Philipp Hungerländer

Semidefinite Optimization Approaches to Applications in Facility Layout and Logistics

DISSERTATION

zur Erlangung des akademischen Grades
Doktor der Sozial- und Wirtschaftswissenschaften

Alpen-Adria-Universität Klagenfurt
Fakultät für Wirtschaftswissenschaften

1. Begutachter: Prof. Miguel F. Anjos, PhD
Canada Research Chair in Discrete
Nonlinear Optimization in Engineering
GERAD & École Polytechnique de Montréal
2. Begutachter: o.Univ.-Prof. Dr. Reinhard Neck
Institut für Volkwirtschaftslehre
Alpen-Adria-Universität Klagenfurt

Juli 2014

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich, dass ich die vorliegende wissenschaftliche Arbeit selbstständig angefertigt und die mit ihr unmittelbar verbundenen Tätigkeiten selbst erbracht habe. Ich erkläre weiters, dass ich keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle aus gedruckten, ungedruckten oder dem Internet im Wortlaut oder im wesentlichen Inhalt übernommenen Formulierungen und Konzepte sind gemäß den Regeln für wissenschaftliche Arbeiten zitiert und durch Fußnoten bzw. durch andere genaue Quellenangaben gekennzeichnet.

Die während des Arbeitsvorganges gewährte Unterstützung einschließlich signifikanter Betreuungshinweise ist vollständig angegeben.

Die wissenschaftliche Arbeit ist noch keiner anderen Prüfungsbehörde vorgelegt worden. Diese Arbeit wurde in gedruckter und elektronischer Form abgegeben. Ich bestätige, dass der Inhalt der digitalen Version vollständig mit dem der gedruckten Version übereinstimmt.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

(Unterschrift) (Ort, Datum)

Abstract

Semidefinite Programming (SDP), Facility Layout and Logistics have been very active research areas over the last decades. SDP is the extension of linear programming to linear optimization over the cone of symmetric positive semidefinite matrices. Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function and logistics is the management of the flow of goods between the point of origin and the point of consumption in order to meet some requirements, for example, of customers or corporations.

The main contributions of this thesis are the comparison of existing and the design of new exact approaches based on linear, quadratic and semidefinite relaxations for several facility layout and logistic problems. Facility layout problems discussed in this thesis are the Single-Row Facility Layout Problem, the k-Parallel Row Ordering Problem and the Multi-Row Facility Layout Problem. These layout problems are of special interest for optimizing flexible manufacturing systems. But they can also be successfully applied to areas like sever design and typewriter keyboard design. Additionally these layout problems arise in a large number of applications in other diverse fields as logistics and scheduling.

Up to now there existed quite diverse exact approaches to the various problem types mentioned above. We aim to highlight their connections and to present a unifying approach by showing that the proposed semidefinite model based on products of ordering variables can be successfully applied to many different layout problems. We show the efficiency of our algorithm by providing extensive computational results for a large variety of problem classes, solving many instances that have been considered in the literature for years to optimality for the first time. In particular we demonstrate that our semidefinite approach is the strongest exact method to date for most row layout problems including the Single-Row (Equidistant) Facility Layout Problem, the Multi-Row Equidistant Facility Layout Problem and several variants of the general Multi-Row Facility Layout Problem.

Due to the generality of our SDP method, we can also use it to simultaneously optimize over multiple machine cells exhibiting different layout types. We formally define the corresponding Combined Cell Layout Problem and indicate its theoretical and practical advantages.

Additionally three new layout problems are introduced, namely the Directed Circular Facility Layout Problem, the Checkpoint Ordering Problem and the weighted Linear Ordering Problem. We indicate the relevance and applications of these new problems and suggest both heuristic and exact methods for solving them.

Furthermore we concern ourselves with two variants of the famous Traveling Salesman Problem. The Target Visitation Problem considers, additionally to the distances travelled, preferences for visiting the different targets. We propose a semidefinite formulation and demonstrate the efficiency of our approach on a variety of benchmark instances. We also conduct a polyhedral study of the corresponding polytope, improving a semidefinite relaxation proposed by Newman. While the Target Visitation Problem has applications in environmental assessment, combat search and rescue and disaster relief, we suggest another variant of the Traveling Salesman Problem with applications in beam melting. We examine the length and structure of the optimal traveling salesman tours considering different types of forbidden neighborhoods on grid graphs.

From an algorithmic point of view the core of the approach used most frequently in this thesis is to approximately solve tight SDP relaxations for the different layout and logistic problems mentioned above by using a dynamic version of the bundle method in conjunction with interior point methods. The resulting fractional solutions constitute lower bounds for the problems. By the use of the hyperplane rounding algorithm of Goemans-Williamson together with an appropriate repair strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe a feasible

layout of the departments or feasible tours respectively. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum.

In summary we extend the application area of semidefinite and combinatorial optimization in facility layout and logistics through the design of new optimization problems and the development of efficient algorithmic frameworks based on semidefinite (and linear) programming.

Acknowledgements

I am grateful to a number of people who have supported me in the development of this thesis and it is my pleasure to highlight them here.

I want to thank my supervisor Prof. Miguel Anjos for many fruitful discussions regarding layout problems and polynomial programming, for inviting me to Montreal, for his open-hearted hospitality, for supporting me at several occasions during the last years and for the large amount of time he devoted to my concerns. His ideas and advice substantiated my thesis.

I also want to thank Prof. Reinhard Neck: Eight years ago he was the person who led me into active research and since then always supported my plans and ideas. Even earlier, when I was a bachelor student, he strongly influenced my wish to become a researcher through is fascinating, challenging and at the same time entertaining lectures. I want to thank him for his enthusiasm concerning our common research and his understanding for all my different requests through all these years.

My thanks go to all members of the Mathematics and Economics Departments at the Alpen-Adria-Universität Klagenfurt for providing me with excellent working conditions. Especially I want to thank Benjamin Hackl who helped me a lot with the drawings in the thesis.

I want to thank my friends Davide, Markus and Michael for sharing a part of their lives with me and for being there whenever I need help. This of course also holds true for my girlfriend Arina, who was very patient, understanding and supportive over the last years, when I spent many evenings and nights working on this thesis - thank you :*

Above all, my thanks go to my parents and my sister, for supporting me in all conceivable ways from when I was young, up to now.

Contents

Part I

Introduction, Motivation and Background Information

Chapter 1

Introduction

The thesis is structured as follows: Part I is a general introduction and motivation of the thesis' topic and summarizes the necessary preliminaries. In chapter 2 we motivate the thesis's topic, give an overview of the research contained and point out relations between the different publications. In order to make the thesis self-contained, we discuss the basic theoretical properties and several important applications of semidefinite programs in Chapter 3. In Chapter 4 we sketch the most important methods for solving semidefinite programs, namely interior-point methods and bundle methods. We discuss in some more detail a dynamic version of the bundle method that we use extensively in Parts II and III to approximately solve computationally challenging semidefinite programs. Finally in Part IV we give a conclusion and point out several research questions and plans.

The main contributions of this thesis are

- the comparison of existing and the design of new exact approaches based on linear, quadratic or semidefinite programming for several well-known applications in facility layout and logistic,
- the computation of optimal solutions for many layout instances, which have been considered in the literature for years, for the first time,
- the design of new facility layout problems and appropriate algorithmic approaches thereto,
- the extension of the area of application of combinatorial optimization problems in facility layout and logistics.

We will elaborate on these points in the following chapter. Before let us briefly point out the publications and research projects incorporated in the thesis. Part II on facility layout planing and design consists of the following original papers that either already appeared in an international journal or where submitted since summer 2014:

- "*Semidefinite Relaxations of Ordering Problems*" [168]
- "*A Computational Study and Survey of Methods for the Single-Row Facility Layout Problem*" [169]
- "*Single-Row Equidistant Facility Layout as a Special Case of Single-Row Facility Layout*" [163]
- "*A Semidefinite Optimization Approach to the Directed Circular Facility Layout Problem*" [157]
- "*A New Modelling Approach for Cyclic Layouts and its Practical Advantages*" [158]
- "*A Semidefinite Optimization Approach for the Parallel Ordering Problem*" [159]
- "*A Semidefinite Optimization Approach to Space-Free Multi-Row Facility Layout*" [164]

- "*A Semidefinite Optimization Approach to Multi-Row Facility Layout*" [167]
- "*Solution Approaches for Equidistant Multi-Row Facility Layout Problems*" [16, 17]
- "*The Checkpoint Ordering Problem*" [160]

Part III on logistics consists of the following original papers that either already appeared in an international journal or where submitted since summer 2014::

- "'*A Semidefinite Optimization Approach to the Target Visitation Problem*" [161]
- "'*New Semidefinite Programming Relaxations for the Linear Ordering and the Traveling Salesman Problem*" [162]
- "*An Exact Approach for the Combined Cell Layout Problem*" [165, 166]

More details on these papers and an outline of their relations can be found in Section 2.5. Note that the thesis does not contain any original journal papers (due to copyright issues), but rather the corresponding preprints that can also be found on "Optimization Online" and on my personal homepage `http://philipphungerlaender.jimdo.com/`.

Furthermore Chapter 18 contains yet unpublished results on

- the weighted Linear Ordering Problem (Section 18.1),
- structural relations between single- and multi-row layouts (Section 18.2, joint work with Anja and Frank Fischer) and
- the Traveling Salesman Problem with forbidden neighbourhoods (Section 18.3, joint work with Anja Fischer).

Chapter 2

Motivation: Areas of Application

2.1 Introduction

In this thesis we use quantitative methods to analyze and improve the decisions made in two fields of Business Economics, namely layout planing (or more general production) and logistics. The improved and often optimal solutions generated by the various algorithms suggested help enterprises in distinct fields to save their resources and thus to enhance the quality of their decisions made.

For me personally it was a great pleasure to do research in production and logistics as it allowed me to combine my two studies graduated, namely Business Economics and Mathematics. On the one hand I worked on various practical applications in the areas production and logistics that I already came across several years ago, when I majored in production, logistic and environmental management. On the other hand I had to apply advanced mathematical methods to obtain appropriate models and later on (near-)optimal solutions for the practical problems at hand.

Accordingly the original papers contained in this thesis were published in (or recently submitted to) high-quality journals from the areas applied mathematics, optimization and quantitative production and logistic research. Furthermore I gave several talks at international conferences with a focus in business and mathematics respectively (for details see the CV attached).

In the following sections we aim to give a general introduction to the thesis's topics and to highlight their relations.

2.2 Facility Layout as Sub-Discipline of Production Research

Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function. This is a well-known operations research problem that arises in different areas of applications. For example, in manufacturing systems, the placement of machines that form a production line inside a plant is a layout problem in which one wishes to minimize the total material flow cost. Another example arises in the design of Very Large Scale Integration (VLSI) circuits in electrical engineering. The objective of VLSI floorplanning is to arrange a set of rectangular modules on a rectangular chip area so that performance is optimized; this is a particular version of facility layout. In general, the objective function may reflect transportation costs, the construction cost of a material-handling system, the costs of laying communication wiring, or simply adjacency preferences among departments.

The variety of applications means that facility layout encompasses a broad class of optimization problems. The survey paper [222] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of departments within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the departments sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem, such as the layout of machines along a production line. The thesis will focus on several problem types from this third area, namely the Single-Row Facility Layout Problem (`SRFLP`), the Directed Circular Facility Layout Problem (`DCFLP`), the k-Parallel Row Ordering Problem (`kPROP`), the Space-Free Multi-Row Facility Layout Problem (`SF-MRFLP`) and the Multi-Row Facility Layout Problem (`MRFLP`). These layout problems are e.g. of special interest for optimizing flexible manufacturing systems (FMSs).

FMSs are automated production systems, typically consisting of numerically controlled machines and material handling devices under computer control, which are designed to produce a variety of parts. In FMSs the layout of the machines has a significant impact on the materials handling cost and time, on throughput, and on productivity of the facility. A poor layout may also negate some of the flexibilities of an FMS [133]. The type of material-handling devices used such as handling robots, automated guided vehicles (AGVs), and gantry robots typically determines machine layout in an FMS [232]. In practice, the most frequently encountered layout types are the single-row layout (Figure 2.1), the double-row and multi-row layouts (Figure 2.2) and the circular layout (Figure 2.3).

Figure 2.1: In a.) an AGV transports parts between the machines moving in both directions along a straight line. In b.) a material-handling industrial robot carries parts between the machines.

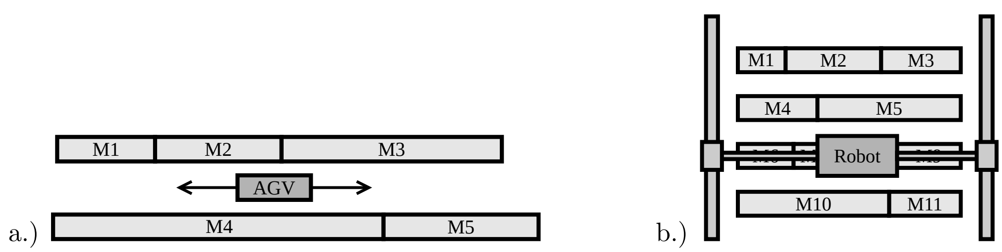


Figure 2.2: In a.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. In b.) a gantry robot is used when the space is limited.

2.3 Relations Between the Layout Problems Studied

Next we want to provide an overview and comparison of the layout problems that will be discussed in detail in the thesis. For each layout problem we give both an intuitive and a mathematical formulation, we point out areas of application and we refer to the most efficient exact and heuristic algorithms. Additionally we try to clarify the workings and differences of the relevant layout problems with the help of a toy example.

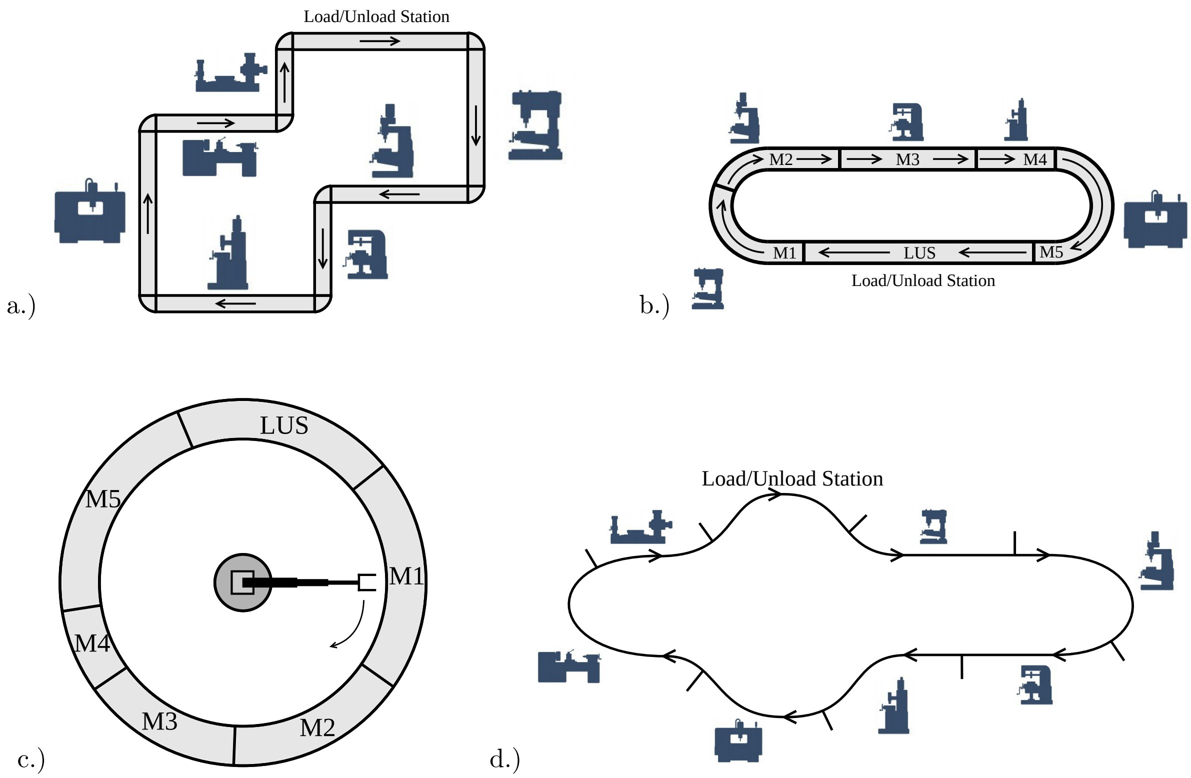


Figure 2.3: In a.) and b.) a conveyor moves in a closed-loop rail in one direction transporting parts among the machines. In c.) a material-handling industrial robot rotates unidirectionally and in d.) single loop AGVs transport parts between the machines.

Furthermore we introduce a new optimization problem that allows for an arbitrary combination and simultaneous optimization of all the other layout problems discussed. At the end of the section we display various relationships between our layout problems and related combinatorial optimization problems.

Many facility layout problems have a strong combinatorial nature. Combinatorial optimization uses heuristic, approximation and exact algorithms to find (near-)optimal solutions for many problems of practical interest whose feasible solutions are given by a finite set. The majority of combinatorial optimization problems is NP-hard. As such, numerous heuristic and metaheuristic approaches have been proposed for the various categories of facility layout problems, see e.g. [148]. However, few methods exist that provide global optimal solutions, or at least a measure of nearness to global optimality, for large instances of layout problems. It is one of the main goals of this thesis to extend and partly improve the available exact approaches.

The easiest known layout type is single-row layout. It arises as the problem of ordering stations on a production line where the material flow is handled by an AGV in both directions on a straight-line path [150] (see Figure 2.1). An instance of the (SRFLP) consists of n one-dimensional departments or machines, with given positive lengths $\ell_1, \ldots, \ell_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{i<j \in [n]} c_{ij} z_{ij}^{\pi},$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_{ij}^{π} is the center-to-center distance between machines i and j with respect to a particular permutation $\pi \in \Pi_n$.

Several practical applications of the (SRFLP) have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291], the arrangement of machines in flexible manufacturing systems [150], the arrangement of books on a shelf and the assignment of disk cylinders to files [252].

The (SRFLP) is one of the few layout problems for which strong global lower bounds and even optimal solutions can be computed for instances of reasonable size. The global optimization approaches for the (SRFLP) are based on relaxations of integer linear programming and semidefinite programming (SDP) formulations. The strongest ones are an LP-based cutting plane algorithm using betweenness variables [8] and an SDP approach using products of ordering variables [169]. For more details on global optimization approaches for the (SRFLP) we refer to the survey article by Anjos and Liers [19].

Another extensively discussed combinatorial optimization problem in the layout literature is the Single-Row Equidistant Facility Layout Problem (SREFLP), sometimes called the one-dimensional machine location problem [269] or the linear machine-cell location problem [308]. In general the (SREFLP) is formulated as follows. Given n machines and flows $f_{ij},\ i,j \in [n],\ i \neq j$ between machines i and j, the aim is to find a one-to-one assignment of the machines to n locations equally spaced along a straight line so as to minimize the sum of the products of distances between the machines with the respective flows. The (SREFLP) arises in many applications in manufacturing and logistics management, including sheet-metal fabrication [15], printed circuit board and disk drive assembly [65] and the optimal design of a flowline in a manufacturing system [308]. Furthermore Bhasker and Sahni [38] applied the (SREFLP) to minimize the total wire length needed when arranging circuit components on a straight line.

Minimum Linear Arrangement(LA) belongs to the graph layout problems and asks for a permutation of the nodes of the underlying graph that optimizes some function of pairwise node distances. (LA) was originally proposed by Harper [131, 132] to develop error-correcting codes with minimal average absolute errors and was since then applied to VLSI design [300], single machine job scheduling [2, 258] and computational biology [180, 226]. It is also used for the layout of entity relationship models [59] and data flow diagrams [109]. For further details on graph layout problems we refer to the survey paper of Díaz et al. [89].

The (DCFLP) seeks to arrange one-dimensional machines with given lengths on a circular material handling system so as to minimize the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction (see Figure 2.3). The material handling system is assumed to move the parts unidirectionally around the circuit following the sequence specified in its process plan. Each machine is capable of picking up and processing the parts from the material handling system [216]. The most commonly used operational strategy for such systems is that all parts enter and exit the system at the loading and unloading stations that do not perform any operation on these parts. Circular material handling systems are mostly preferred because of their relative low initial investment costs, high material handling flexibility and their ability of being easily accommodated to future introduction of new parts and process changes [3, 190].

An instance of the (DCFLP) consists of n one-dimensional machines with given positive lengths $\ell_1, \dots, \ell_n$ and pairwise flows $f_{ij},\ i,j \in [n], i \neq j$. The machines are arranged next to each other on a circle. The objective is to find a permutation π of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines (in clockwise direction) is minimized

$$\min_{\pi \in \Pi_n} \sum_{i,j \in [n], i \neq j} f_{ij} z_{ij}^{\pi},$$

where Π_n is the set of all feasible layouts and z_{ij}^{π} gives the distance between the centroids of machines i and j in the circular layout π (in clockwise direction).

To the best of our knowledge this thesis and the associated papers [157, 158] deal with the (DCFLP) for the first time. The (DCFLP) is a very interesting problem as it is a generalization of several layout problems that have been extensively discussed in the literature. To begin with the (DCFLP) is a generalization of the Directed Circular Arrangement Problem (DCAP) that allows the machines to have arbitrary instead of the same lengths. The (DCAP) was first considered by Liberatore [209] who showed that the problem is NP-hard (hence also the (DCFLP) is NP-hard) and gave an $O(\log n)$-approximation factor algorithm. Later on Naor and Schwartz [231] improved on this result by proposing an $O(\log n \log \log n)$-approximation algorithm. We refer to [31, 209, 231] for several very nice applications of the (DCAP) in the areas of sever design and ring networks. In [157, 158] we show that the (DCFLP) can be modelled as a Linear Ordering Problem (LOP) and we exploit this relationship to tailor efficient heuristic and exact algorithms for the (DCFLP).

Ordering problems are a special class of combinatorial optimization problems, where weights are assigned to each ordering of n objects and the aim is to find an ordering of maximum weight. Even for the simplest case of a linear cost function, ordering problems are known to be NP-hard, i.e. it is extremely unlikely that there exists an efficient (polynomial-time) algorithm for solving ordering problems to optimality.

Ordering problems arise in a large number of applications in such diverse fields as economics, business studies, social choice theory, sociology, archaeology, mathematical psychology, very-large-scale integration and flexible manufacturing systems design, scheduling, graph drawing and computational biology. For further information on the (LOP) we refer to Martí and Reinelt [219]. Semidefinite approaches to ordering problems are discussed in detail in Hungerländer [156].

Furthermore the (DCFLP) is related to the NP-hard [189] Unidirectional Cyclic Layout Problem (UCFLP). The (UCFLP) also considers a circular material handling system and the objective is to find an assignment of n machines to n predetermined candidate locations such that the total handling cost is minimized. The (UCFLP) has two well-known special cases that are at the same time special cases of the (DCFLP): In the balanced unidirectional cyclic layout problem (BUCFLP) the material flow is conserved at each machine, i.e. the total inflow is equal to total outflow at each machine. Another special form of the (UCFLP) is the equidistant unidirectional cyclic layout problem (EUCFLP), where the locations around the unidirectional cyclic material handling system are assumed to be equally distant to each other. Clearly the (EUCFLP) is equivalent to the (DCAP). Additionally Bozer and Rim [45] have shown that the (EUCFLP) and the (BUCFLP) are equivalent. For a recent discussion of several exact and heuristic algorithms for the (UCFLP) and its special cases we refer to [5, 242, 244].

In comparison with the (UCFLP), the (DCFLP) considers machine lengths instead of distances of the locations and hence is an adaption of the Single-Row Facility Layout Problem to circular layouts. Considering machine lengths instead of the location distances (i.e. location lengths) is clearly the preferable modelling approach in many practical applications where the lengths of the machines are the relevant input parameters. Over and above solving the (UCFLP) with heuristic and exact methods is very hard because it is a special (QAP) [45] and the (QAP) is known to be notoriously difficult [211, 311].

An instance of the (COP) consists of n one-dimensional departments, with given positive lengths $\ell_1, \ldots, \ell_n$ and connectivities $c_1, \ldots, c_n$, and a checkpoint on a fixed position, e.g. left-aligned or at the center position. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{i \in [n]} c_i z_i^{\pi}, \tag{2.1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_i^{π} is the distance between the center of department i and the checkpoint with respect to a particular permutation $\pi \in \Pi_n$.

For the (COP) similar applications as for the (SRFLP) are conceivable , e.g. the rooms on a corridor could be arranged such that the weighted sum of their distances with the office of the head is minimized or planes could be assigned to gates such that the weighted sum of their distances from the entrance of the airport terminal is minimized. When comparing the (SRFLP) with the (COP), we observe that the problems are quite similar. One difference is that an (SRFLP) instance has $\binom{n}{2}$ connectivities while an (COP) instance has only n connectivities. On first sight the (SRFLP) seems more difficult than the (COP) (at least this was the first impression of the author). We will show in this thesis that the opposite is true.

The (kPROP) is an extension of the (SRFLP) and at the same time of the (COP) that considers arrangements of the departments along more than one row. An instance of the (kPROP) consists of n one-dimensional departments with given positive lengths $\ell_1, \ldots, \ell_n$, pairwise connectivities c_{ij} and an assignment a of each department to one of the k rows $\mathcal{R} := \{1, \ldots, k\}$. The objective is to find permutations $\pi^1 \in \Pi^1, \ \ldots, \ \pi^k \in \Pi^k$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\pi^1 \in \Pi^1, \ldots, \pi^k \in \Pi^k} \sum_{i<j \in [n]} c_{ij} z_{ij}^{\pi^{a(i)}, \pi^{a(j)}},$$

where $\Pi = \{\Pi^1, \ldots, \Pi^k\}$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{a(i)}, \pi^{a(j)}}$ denotes the horizontal distance between the centroids of departments i and j in the layout $\{\pi^1, \ldots, \pi^k\} \in \Pi$. If the (kPROP) is restricted to two rows we simply call it (PROP). Applications of the (kPROP) are the arrangement of departments along two or more parallel straight lines on a floor plan, the construction of multi-floor buildings and the layout of machines in FMSs.

The (kPROP) can be further extended to the (SF-MRFLP) by additionally optimizing over all possible row assignments. Hence an instance of the (SF-MRFLP) consists of n one-dimensional departments with given positive lengths $\ell_1, \ldots, \ell_n$, pairwise connectivities c_{ij} and a function $r : [n] \to \mathcal{R}$ that assigns each department to one of the k rows. The objective is to find permutations $\pi^1 \in \Pi^1, \ \ldots, \ \pi^k \in \Pi^k$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\pi^1 \times \ldots \times \pi^k \in \Pi^1 \times \ldots \times \Pi^k} \sum_{i<j \in [n]} c_{ij} z_{ij}^{\pi^{r(i)}, \pi^{r(j)}},$$

where $\Pi = \Pi^1 \times \ldots \times \Pi^k$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}$ denotes the horizontal distance between the centroids of departments i and j in the layout $\{\pi^1 \times \ldots \times \pi^k\} \in \Pi$. If we restrict the (SF-MRFLP) to two rows we obtain the Space-Free Double-Row Facility Layout Problem (SF-DRFLP) as a special case. A specific example of the application of the (SF-DRFLP) is in spine layout design. Spine layouts, introduced by Tompkins [295], require departments to be located along both sides of specified corridors along which all the traffic between departments takes place. Although in general some spacing is allowed, layouts with no spacing are much preferable since spacing often translates into higher construction costs for the facility. Algorithms for spine layout design have been proposed, see e.g. [197]. The best method known to date for the (SF-DRFLP) is an algorithm based on a mixed-integer LP (MILP) formulation that was recently proposed in [10].

The Double-Row Facility Layout Problem (DRFLP) is a natural extension of the (SRFLP) in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel (see Figure 2.2). This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the (DRFLP) is especially relevant for real-world applications. The (DRFLP) can be further generalized to the (MRFLP), where the departments are arranged along k parallel rows. Hence the (MRFLP) is generalization of the (SF-MRFLP) in which the rows may not have a common left origin and space is allowed between departments.

The (MRFLP) has many applications such as computer backboard wiring [288], campus planning [90], scheduling [115], typewriter keyboard design [253], hospital layout [95], the layout of machines in an automated manufacturing system [151], balancing hydraulic turbine runners [198], numerical analysis [46], optimal digital signal processors memorry layout generation [302]. Somewhat surprisingly, the development of exact algorithms for the (MRFLP) and the (DRFLP) has received only limited attention in the literature.

In the 1980s Heragu and Kusiak [150, 151] proposed a non-linear programming model and obtained locally optimal solutions to the (SRFLP) and the (DRFLP). Recently Chung and Tanchoco [72] (see also [310]) focused exclusively on the (DRFLP) and proposed a MILP formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Amaral [11] proposed an improved MILP formulation that allowed him to solve instances with up to 12 departments to optimality.

Next let us further clarify the workings and differences of the (SRFLP), the (DCFLP), the (kPROP), the (SF-MRFLP), the (COP) and the (MRFLP) with the help of a toy example: We consider 4 machines with lengths $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4$. Additionally we have given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$ for the row layout problems and pairwise flows $f_{12} = f_{14} = f_{43} = 1,\ f_{13} = f_{42} = 2$ for the (DCFLP). For the (COP) we assign department 1 to row 2 and all other departments to row 1 and hence disregard the connectivity $c_{24} = 2$. Figure 2.4 illustrates the optimal layouts and the according costs for the six different combinatorial optimization problems.

In a cellular manufacturing system, the parts that are similar in their processing requirements are grouped into part families, and the machines needed to process the parts are grouped into machine cells. Ideally, cells should be designed in such a way that the part families are fully processed in a single machine cell so that the machine cells are mutually independent with no inter-cell movement. In a real-world situation however it may be impractical and/or uneconomical to require mutually independent cells. The consequence is that some parts will require processing in more than one cell, see Figure 2.5.

Cell layout usually takes place after the machine cells are determined, see e.g. [71]. While much research has been done on the cell formation problem, the layout of machines within the cells has received less attention. Our concern is finding the optimal layout of each cell in the presence of parts that require processing in more than one cell. The machines of each cell can be arranged in one or multiple rows or on a circle. We denote this problem as Combined Cell Layout Problem (CCLP). Hence in general the (CCLP) allows to combine all the layout problems discussed above. To the best of our knowledge, this thesis and the associated paper [165] deal with the (CCLP) for the first time. Note that the (CCLP) can be considered both an application to facility layout (finding the optimal design for a manufacturing environment) and logistics (moving parts around in a manufacturing environment). This illustrates the close relation and multiple intersections of the two fields. Eventually we decided to include the corresponding paper in the logistic part of the thesis.

Finally let us further highlight various relationships between our layout problems and related combinatorial optimization problems. Clearly the (SREFLP) is a special case of the quadratic assignment problem (QAP) formulated by Koopmans and Beckmann [186]. But the (SREFLP) is also a special case of the (SRFLP) where all departments have the same length and the pairwise connectivities $c_{ij},\ i, j \in [n],\ i < j$ are given as the sum of the flows f_{ij} and f_{ji}. Additionally (LA), which is NP-hard [112] (even if the underlying graph is bipartite [111]), is a special case of the (SREFLP) where all connectivities are equal. Hence the (SREFLP) and the (SRFLP) are also NP-hard and weighted minimum Linear Arrangement (wLA) and the (SREFLP) are equivalent problems. The (SRFLP) and the (COP) are special cases of the (kPROP) which is again special cases of the (SF-MRFLP). The (SRFLP) is also a special case of the weighted Betweenness Problem (wBWP) which is a special case of the Quadratic Ordering Problem (QOP). Furthermore the (SF-MRFLP) is a special case of the (MRFLP) which is a special (CCLP). Also the (DCFLP) and hence the (LOP) are special (CCLPs). Finally the (CCLP) can be modelled as a (QOP). To gain a better overview we display the between the different facility layout problems in Figure 2.6.

2.4 New Variants of the TSP with Applications in Logistics

The Traveling Salesman Problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible tour that visits each city exactly once and returns to the origin city? The (TSP) is of considerable practical importance, especially in the evident transportation and logistics areas. Additionally the NP-hard (TSP) is doubtless the most famous of all (combinatorial)

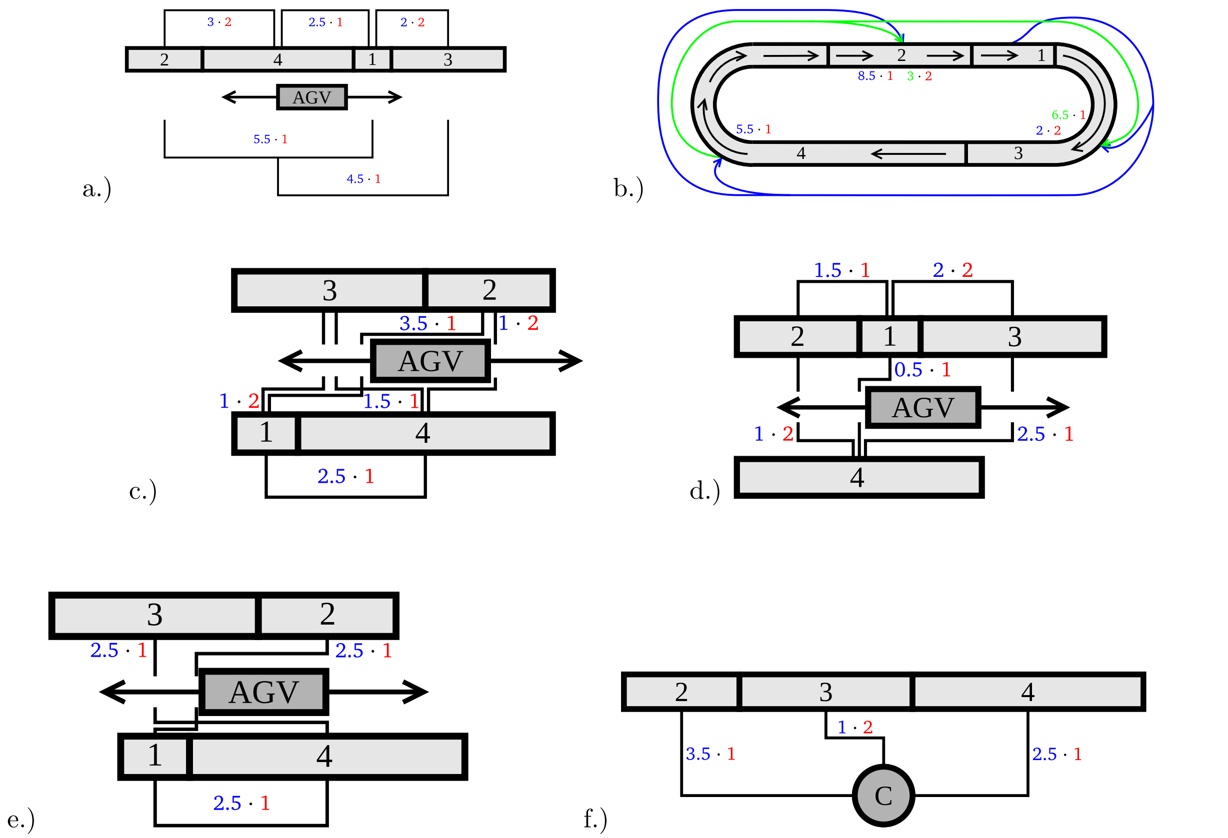


Figure 2.4: We have given the following data: $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4,\ c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2,\ f_{12} = f_{14} = f_{43} = 1,\ f_{13} = f_{42} = 2$. In a.) we display the optimal layout for the (SRFLP) with associated costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we depict the optimal layout for the (DCFLP) with according costs of $2 \cdot 2 + 3 \cdot 2 + 5.5 \cdot 1 + 8.5 \cdot 1 + 6.5 \cdot 1 = 30.5$. In c.) we show the optimal layout for the (PROP) with machines 3 and 2 assign to row 1 and machines 1 and 4 assign to row 2. The associated costs are $3.5 \cdot 1 + 1 \cdot 2 + 1 \cdot 2 + 1.5 \cdot 1 + 2.5 \cdot 1 = 11.5$. In d.) we display the optimal layout for the (SF-DRFLP). The according costs are $1.5 \cdot 1 + 2 \cdot 2 + 0.5 \cdot 1 + 1 \cdot 2 + 2.5 \cdot 1 = 10.5$. Finally we depict the optimal layout for the (DRFLP) in e.). The associated costs are $2.5 \cdot 1 + 2.5 \cdot 1 + 2.5 \cdot 1 = 7.5$.

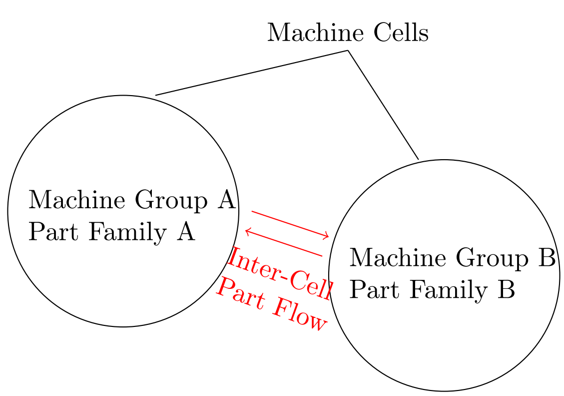


Figure 2.5: Inter-cell part flow occurs in real-world situations.

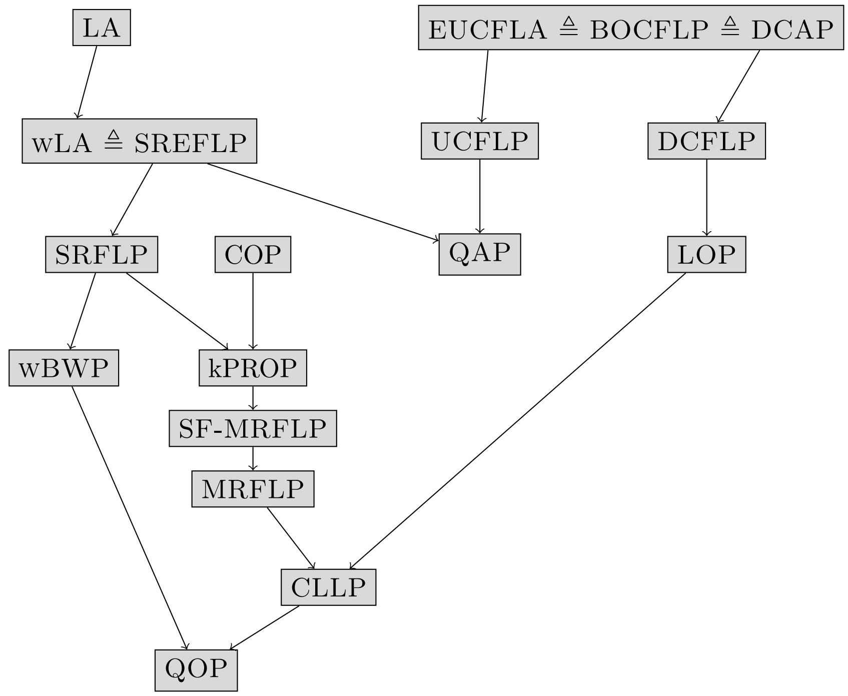


Figure 2.6: Overview of the relationships between several layout and combinatorial optimization problems. "$a \to b$" means that a is a special case of b.

optimization problems with high importance in both operations research and theoretical computer science. We refer to the books [75, 127, 260] and the references therein for extensive material on the (TSP), its variants and various applications, details on many heuristic and exact methods and relevant theoretical results.

Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved completely[1] and even problems with millions of cities can be approximated within a small fraction of 1%.

The (TSP) has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, e.g. in DNA sequencing. In many further applications the (TSP) with additional constraints, such as limited resources or time windows, is of relevance. Another class of important problems in the fields of transportation, distribution and logistics are the Vehicle Routing Problem [79] and its variants that can also be interpreted as extensions of the standard (TSP).

In Part III of the thesis we discuss the (TSP), the (LOP) and a combination of them denoted as Target Visitation Problem (TVP) in more detail. The (TVP) asks for a permutation $(p_1, p_2, \ldots, p_n)$ of n targets with given pairwise weights $w_{ij},\ i,j \in [n], i \neq j$, and pairwise distances $d_{ij},\ i,j \in [n], i \neq j$, maximizing the objective function

$$\sum_{\substack{i,j \in [n] \\ i<j}} w_{p_i,p_j} - \left(\sum_{i=1}^{n-1} d_{p_i,p_{i+1}} + d_{p_n,,p_1} \right).$$

Hence the objective function of the (TVP) considers two often competing factors, namely the overall distance traveled and preferences of visiting some "cities", "targets" or "points of interest" before others.

[1]The Branch-and-Cut algorithm by Applegate et al. [23] holds the current record, solving an instance with 85,900 cities.

The formulation of the (TVP) was inspired by the use of single unmanned aerial vehicles (UAVs) that have been used increasingly over the last decades. Applications of the (TVP) include environmental assessment, combat search and rescue and disaster relief [125].

Despite the manifold and relevant applications of the (TVP), no exact algorithms are available to date for solving it. There exist two heuristics for the (TVP): a very simple one proposed in [125] and a genetic algorithm by Arulselvan et al. [27] that was tested on (TVP) instances with up to 16 targets. Also note that Hildenbrandt et al. [152] are currently working on the first polyhedral study of the (TVP) polytope. Based on their findings they are also developing an exact ILP approach for the (TVP) that has a strong potential to solve large-scale (TVP) instances to optimality. We propose an semidefinite formulation of (TVP) that can be tackled quite efficiently by an exact SDP method for instances with up to 50 targets.

Let us also clarify the workings of the (TVP) with the help of a toy example. We consider 5 targets, where we set target 1 to be the base and hence the first in the ordering. We are given the following (LOP) weights W for the remaining 4 targets and the (TSP) distances D between all five targets:

$$W = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 3 & 5 & 5 & 3 \\ 3 & 0 & 3 & 5 & 5 \\ 5 & 3 & 0 & 3 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 3 & 5 & 5 & 3 & 0 \end{bmatrix} \tag{2.2}$$

Figure 2.7 illustrates the optimal (TSP) and (TVP) tours (the tours are displayed by grey edges) and the optimal (LOP) solution together with their corresponding (LOP) benefits (red edges and numbers) and (TSP) costs (grey numbers). Hence considering travel distances and target preferences simultaneously leads to optimal tours that can be quite different from the optimal (TSP) and (LOP) solutions.

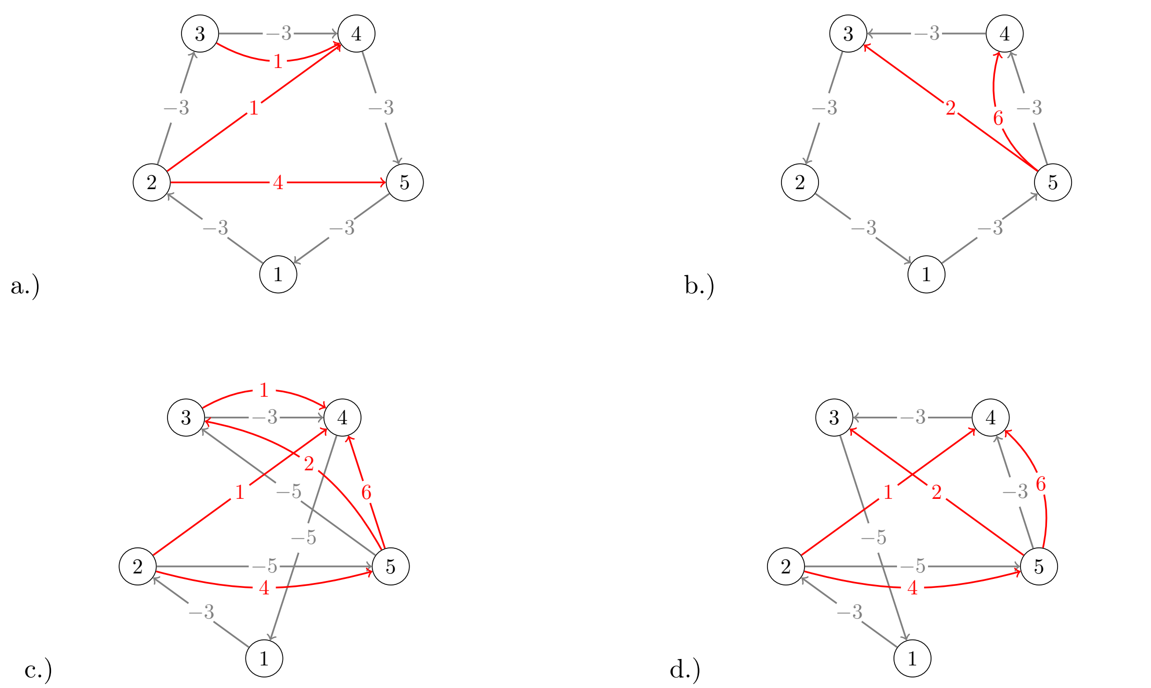


Figure 2.7: We are given 5 targets, where target 1 is the base. In a.) and b.) we display the optimal (TSP) tours with objective value −15 and corresponding (TVP) objective values −9 and −7 respectively. In c.) we depict the optimal (LOP) solution with objective value 14 and associated (TVP) objective value −7. Finally in d.) we display the optimal (TVP) tour with corresponding objective value −6.

in Section 18.3 we study another new variant of the Traveling Salesman Problem with applications in beam melting.

2.5 Detailed Overview of the Publications Building the Thesis

Finally let us give an overview of the original papers and technical reports that build the fundament of the thesis:

- "*Semidefinite Relaxations of Ordering Problems*" [168] (Chapter 5): This published paper presents a systematic investigation of semidefinite optimization based relaxations for the `(QOP)`, extending and improving existing approaches. As all the layout problems discussed above can be modelled as `(QOPs)`, this paper builds the theoretical foundation of the thesis.
- "*A Computational Study and Survey of Methods for the Single-Row Facility Layout Problem*" [169] (Chapter 6): This published paper compares the different modelling approaches for the `(SRFLP)` and applies the SDP approach for general `(QOPs)` to the `(SRFLP)`. In particular, we report optimal solutions for several `(SRFLP)` instances from the literature with up to 42 departments that remained unsolved so far. Secondly we significantly reduce the best known gaps and running times for large instances with up to 110 departments.
- "*Single-Row Equidistant Facility Layout as a Special Case of Single-Row Facility Layout*" [163] (Chapter 7): In this published paper we aim to consolidate two branches of the layout literature. We show that the `(SREFLP)` is not only a special case of the `(QAP)` but also a special case of the `(SRFLP)`. This new connection is relevant as the strongest exact methods for the `(SRFLP)` outperform the best approaches specialized to the `(SREFLP)`. We describe and compare the exact approaches for the `(SRFLP)`, the `(SREFLP)` and `(LA)`. In a computational study we showcase that the strongest exact approach for the `(SRFLP)` clearly outperforms the strongest exact approach tailored to the `(SREFLP)` on medium and large benchmark instances from the literature. Finally we also compare the heuristics for the `(SREFLP)` and the `(SRFLP)`.
- "*A Semidefinite Optimization Approach to the Directed Circular Facility Layout Problem*" [157] (Chapter 8): In this published paper we introduce the `(DCFLP)` and show that the `(DCFLP)` is closely related to the `(SRFLP)`. Hence we adapt the leading exact algorithm for the `(SRFLP)` by suggesting an appropriate modelling approach for the `(DCFLP)` and demonstrate that this algorithmic approach yields promising computational results on a variety of benchmark instances.
- "*A New Modelling Approach for Cyclic Layouts and its Practical Advantages*" [158] (Chapter 9): In this work we show that the `(DCFLP)` allows for a wide range of applications and that it contains several other layout problems that have been discussed extensively in literature as special cases. We model the `(DCFLP)` as a `(LOP)` and solve it using several well studied, efficient heuristics like tabu search or memetic algorithms. Furthermore we argue that the `(DCFLP)` can be solved by both heuristic and exact methods even more efficiently than the `(SRFLP)` that is known to be the easiest layout problem. Finally we validate our findings in a computational study.
- "*A Semidefinite Optimization Approach for the Parallel Ordering Problem*" [159] (Chapter 10): In this paper we present an exact algorithm for the `(kPROP)` that extends the SDP approach for the `(SRFLP)` by modelling inter-row distances as products of ordering variables. Our algorithm is competitive with the strongest method for the `(kPROP)` known so far, namely a MILP formulation that was proposed in [12] very recently. The MILP approach allows to solve instances with up to 23 departments to optimality within a few hours of computing time while our SDP approach yields strong lower bounds for instances of the same size within a few minutes and is able to produce reasonable lower bounds for instances with up to 110 departments.

- "*A Semidefinite Optimization Approach to Space-Free Multi-Row Facility Layout*" [164] (Chapter 11): The contribution of this work is the further extension of the SDP approach from the `(kPROP)` to the `(SF-MRFLP)`. The `(SF-MRFLP)` is a relevant problem in several contexts such as in spine layout design. We develop an SDP approach for the `(SF-MRFLP)` that proves to be competitive with a mixed-integer LP (MILP) formulation for the `(SF-DRFLP)` that was recently proposed in [10]. Computational results show that our SDP approach provides high-quality global bounds in reasonable time for `(SF-DRFLP)` instances with up to 15 departments and for `(SF-MRFLP)` instances with up to 5 rows and 11 departments.

- "*A Semidefinite Optimization Approach to Multi-Row Facility Layout*" [167] (Chapter 12): In this published paper we propose a strategy to extend the algorithmic approach for the `(SF-MRFLP)` to the general `(MRFLP)` and variants thereof. We show that we can model the `(MRFLP)` as a discrete, combinatorial optimization problem by introducing spaces of sufficient number and size. This result allows us to apply our SDP approach for the `(SF-MRFLP)` also to the `(MRFLP)`. We showcase in a computational study that our method yields promising results for several variants of the `(MRFLP)`.

- "*Solution Approaches for Equidistant Multi-Row Facility Layout Problems*" [16, 17] (Chapter 13): In this work we specialize the results from the above paper to the case, where all departments have the same length. We show that for this special case of the problem the number of spaces needed to preserve at least one optimal solution is quite small compared to the general `(MRFLP)`. We exploit this finding for tailoring exact ILP and SDP approaches that outperform other methods proposed for the Equidistant Double-Row Facility Layout Problem. Furthermore comparing our ILP and SDP approaches on a large variety of benchmark instances yields the following finding: The ILP approaches are preferable on small and medium sparse instances, whereas the SDP gives better results on all other types of instances.

- "*The Checkpoint Ordering Problem*" [160] (Chapter 14): In this paper we suggest a new combinatorial optimization problem: Find an ordering of n departments with given lengths such that the total weighted sum of their distances to a given checkpoint is minimized. The Checkpoint Ordering Problem `(COP)` is both of theoretical and practical interest. It has several applications and is conceptually related to the `(SRFLP)`, the `(MRFLP)` and the `(LOP)`. We study the complexity of the `(COP)` and its special cases. The general version of the `(COP)` is NP-hard and can be modeled as a `(QOP)`. We propose an exact semidefinite approach that is highly competitive for several `(QOPs)` to tackle the `(COP)`. Computational experiments indicate that the `(COP)` is very hard to solve in practice. Supported by this finding we additionally suggest a new approach to obtain global bounds for the `(MRFLP)`. We show the practical applicability and benefits of these bounds by providing computational experience on a variety of well-known benchmark instances.

- "*A Semidefinite Optimization Approach to the Target Visitation Problem*" [161] (Chapter 15): In this published paper we propose an exact algorithm for the `(TVP)`. First we show that the `(TVP)` is a special Quadratic Position Problem `(QPP)`. Building on this finding we propose an exact semidefinite optimization approach to tackle the `(TVP)` and finally demonstrate its efficiency on a variety of benchmark instances with up to 50 targets.

- "*New Semidefinite Programming Relaxations for the Linear Ordering and the Traveling Salesman Problem*" [162] (Chapter 16): In 2004 Newman [237] suggested a semidefinite programming relaxation for the Linear Ordering Problem `(LOP)` that is related to the semidefinite program used in the Goemans-Williamson algorithm to approximate the Max Cut problem [121]. Newman [237] showed that her relaxation seems better suited for designing polynomial-time approximation algorithms for the `(LOP)` than the widely-studied standard polyhedral linear relaxations. In this paper we improve the relaxation proposed by Newman [237] and conduct a polyhedral study of the corresponding polytope. Furthermore we relate the relaxation to other linear and semidefinite relaxations for the (LOP) and for the Traveling Salesman Problem and elaborate on its connection to the Max Cut problem.

- "*An Exact Approach for the Combined Cell Layout Problem*" [165, 166] (Chapter 17): In this published paper we propose an exact solution method based on SDP for simultaneously optimizing the layout of two or more cells in a cellular manufacturing system in the presence of parts that require processing in more than one cell. To the best of our knowledge, this is the first exact method proposed for this problem. We consider single-row and directed circular cell layouts but the method can in principle be extended to other layout types. Preliminary computational results suggest that optimal solutions can be obtained for instances with 2 cells and up to 60 machines. An extension of the proceedings paper contained in this thesis has recently been prepared and submitted.

Furthermore Chapter 18 contains yet unpublished material on:

- "*The Weighted Linear Ordering Problem*" (Section 18.1): We define the weighted Linear Ordering Problem that considers individual node weights additional to pairwise weights. We show that the weighted Linear Ordering Problem generalizes row layout problems by allowing for asymmetric cost structures. Additionally we argue that in many applications the optimal ordering obtained is a worthwhile alternative to the optimal solution of the Linear Ordering Problem.

- "*Structural Relations Between Single- and Multi-Row Layouts*" (Section 18.2, joint work with Anja and Frank Fischer): We prove that for equidistant row layout instances the objective value of the optimal solution with m rows is less or equal to the objective value of the optimal single-row layout divided by m. Furthermore we show that for non-equidistant row layout instances no such relation between the objective values of single- and multi-row layouts exists.

- "*The Traveling Salesman Problem with Forbidden Neighbourhoods*" (Section 18.3, joint work with Anja Fischer): The Traveling Salesman Problem with forbidden neighbourhoods asks for a tour of minimal length in which points traversed successively have a given minimal distance from each other. This new variant of the (TSP) is motivated by an application in beam melting, where the points to be fused are arranged in a regular way and can be represented as grid graphs. We examine the length and structure of the optimal tours for different types of forbidden neighborhoods that are the most interesting ones with respect to the application mentioned above.

Chapter 3

Preliminaries: Semidefinite Programming

3.1 Introduction

Semidefinite programming (SDP) is an extension of linear programming (LP), with vector variables replaced by matrix variables and nonnegativity elementwise replaced by positive semidefiniteness. Thus a (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} p^* &:= \inf_X \ \{ \ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \\ \mathcal{P} &:= \left\{ \ X \ \middle| \ \langle A_i, X \rangle = b_i, \ i \in \{1, \ldots, m\}, \ X \in \mathcal{S}_n^+ \ \right\}, \end{aligned} \tag{P}$$

where the data matrices A_i, $i \in \{1, \ldots, m\}$, C and the variable matrix X are in $\mathcal{S}_n$, the space of symmetric $n \times n$ matrices. The essential difference between LP and SDP is that the nonnegative orthant $\mathbb{R}^n_+$ is replaced by the convex, self-dual cone of positive semidefinite matrices $\mathcal{S}_n^+$, which has a nonlinear boundary. Thus (P) is a nonlinear convex programming problem. If the matrix X is restricted to be diagonal, (P) reduces to a linear program. In the following two sections we give a short overview of the basic theoretical properties and main application areas of SDP. We refer the reader to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

3.2 Basic Theoretical Properties

SDP is a relatively new area of optimization as most papers on SDP were written since 1990. The roots of SDP can be traced back to the sixties, when Bellman and Fan [34] derived the first duality theorem. They recognized that much of the duality theory for LP can be extended to SDP by using slightly stronger assumptions. In the following we briefly summarize the basic theoretical properties of semidefinite programs, a comprehensive discussion of this topic can be found for instance in Nesterov and Nemirovskii [235] or De Klerk [83]. Let the Lagrangian dual problem of (D) be given by

$$\begin{aligned} d^* &:= \sup_{y,S} \ \left\{ \ b^\top y \ : \ (y, S) \in \mathcal{D} \right\}, \\ \mathcal{D} &:= \left\{ \ (y, S) \ \middle| \ \sum_{i=1}^m y_i A_i + S = C, \ S \in \mathcal{S}_n^+, \ y \in \mathbb{R}^m \ \right\}. \end{aligned} \tag{D}$$

Then weak duality $\langle C, X\rangle \geq b^\top y$ holds for all $X \in \mathcal{P}$ and $(y, S) \in \mathcal{D}$ due to the Minimax inequality

$$\inf_{x\in S_1} \sup_{y\in S_2} f(x,y) \geq \sup_{y\in S_2} \inf_{x\in S_1} f(x,y), \tag{3.1}$$

that is valid for any function $f : S_1 \times S_2 \to \mathbb{R}$ where S_1 and S_2 are arbitrary sets. Contrary to linear programming, strong duality ($p^* = d^*$) does not hold for SDP in general (see Vandenberghe and Boyd [299] for a standard counterexample). To guarantee strong duality, we additionally have to ask for strict feasibility of `(P)` (or `(D)`), i.e. we have to ask for the existence of a $X \in \mathcal{P} \cap \mathcal{S}_n^{++}$ ($S \in \mathcal{D} \cap \mathcal{S}_n^{++}$).

Theorem 3.1 *Assume that $d^* < \infty$ (resp. $p^* > -\infty$). Further assume that* `(D)` *(resp.* `(P)`*) is strictly feasible. Then $p^* = d^*$ and this value is attained for* `(P)` *(resp.* `(D)`*).*

A proof of this theorem can be found for instance in Duffin [94], Rockafellar [264], Nesterov and Nemirovskii [235] or De Klerk [83] An example where the primal optimal solution is not attained, is e.g. provided by Helmberg [139]. If strong duality holds, it is easy to deduce the following necessary and sufficient optimality conditions

$$X \in \mathcal{P},\ (y, S) \in \mathcal{D},\ XS = 0, \tag{3.2}$$

where we again want to point out a difference compared to linear programming. For SDP strict complementarity , i.e. the existence of an optimal solution (X^*, S^*, y) such that $X^* + S^* \in \mathcal{S}_n^{++}$, does not hold in general (see e.g. the counterexample given by Alizadeh et al. [4]). Except linear programming, SDP has some further interesting special cases. To deduce them, we use the well-known Schur complement theorem [43, Appendix A.5.5].

Theorem 3.2 *Let $M = \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}$ with $A \in \mathcal{S}_n^{++}$ and $C \in \mathcal{S}_n$ be given. Then the Schur complement of A in M is given by $C - B^\top A^{-1}B$ and it holds $M \in \mathcal{S}_n^+$, iff $C - B^\top A^{-1}B \in \mathcal{S}_n^+$.*

Proof. M can be transformed to a block diagonal matrix, using the following similarity transformation

$$\begin{pmatrix} I & 0 \\ -A^{-1}B & I \end{pmatrix} \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix} \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & C - B^\top A^{-1}B \end{pmatrix}.$$

Since a block diagonal matrix is positive semidefinite, iff its diagonal blocks are positive semidefinite, the proof is complete. □

Applying the Schur complement theorem we can rewrite the quadratic constraint

$$(Ax + b)^\top (Ax + b) - (c^\top x + d) \leq 0,\ x \in \mathbb{R}^n,$$

and the second order cone constraint

$$\left\{ (t, x) \;\middle|\; t \geq \sqrt{\sum_{i=1}^n x_i^2} \right\},$$

as semidefinite constraints

$$\begin{pmatrix} I & Ax + b \\ (Ax + b)^\top & c^\top x + d \end{pmatrix} \succcurlyeq 0,\ x \in \mathbb{R}^n,$$

$$\begin{pmatrix} tI & x \\ x^\top & t \end{pmatrix} \succcurlyeq 0.$$

Thus several optimization problems with quadratic or cone constraints, including the well-known convex quadratic programming (QP) problem , are special cases of SDP. Another interesting special case is the nonlinear problem

$$\min_x \left\{ \frac{(c^\top x)^2}{d^\top x} \;\middle|\; Ax \geq b \right\},$$

where we additionally assume that $d^\top x > 0$ if $Ax \geq b$. Also for this problem we can state an equivalent SDP problem

$$\min_{t,x} \left\{ t \;\middle|\; \begin{pmatrix} t & c^\top x & 0 \\ c^\top x & d^\top x & 0 \\ 0 & 0 & \mathrm{Diag}(Ax-b) \end{pmatrix} \succcurlyeq 0 \right\}.$$

Also the classical problem of finding the largest eigenvalue $\lambda_{\max}(A)$ of a symmetric matrix A can be formulated as an SDP problem

$$\min_t \{t \mid tI - A \succcurlyeq 0,\ t \in \mathbb{R}\}.$$

A list of further eigenvalue or matrix norm minimization problems that can be stated as SDP's is provided by Vandenberghe and Boyd [299].

3.3 Applications

There exist important applications of SDP in approximation theory (e.g. non-convex quadratic optimization [234] and nonnegative polynomials [249])), system and control theory [30, 44], and mechanical and electrical engineering (VLSI transistor sizing and pattern recognition [299] and structural design [36]). But in this thesis we are most interested in applications of SDP in combinatorial optimization. An instance of a combinatorial optimization problem is given by a pair (L, f), where L is a countable set of all feasible solutions and f is a function $f : L \to \mathbb{R}$ that assigns an objective value to each element of L. Now the aim is to find an element $i \in L$ with minimal ($f(i) \leq f(u), \forall u \in L$) or maximal ($f(i) \geq f(u), \forall u \in L$) objective value. Thus ordering problems fall into the area of combinatorial optimization. In the following we give a short review of two other important applications of SDP in combinatorial optimization.

The probably most celebrated application is the Lovász θ-function [213] , that maps an undirected graph $G = (V, E)$ to $\mathbb{R}^+$ and is given as the optimal value of the following SDP problem

$$\theta(\overline{G}) := \max_X \left\{ \langle ee^\top, X\rangle : x_{ij} = 0, (i,j) \notin E, \mathbf{Tr}(X) = 1, X \in \mathcal{S}_n^+ \right\}, \tag{3.3}$$

where $e \in \mathbb{R}^{|V|}$ denotes the vector of all-ones, and $\overline{G}$ the complement graph of G, i.e. $\overline{G} = (V, K \setminus E)$, where K consists of all 2-element subsets of V. The θ-function fulfills the following relation known as "sandwich theorem"

$$\omega(G) \leq \theta(\overline{G}) \leq \chi(G),$$

where $\omega(G)$ denotes the clique number of G, and $\chi(G)$ the chromatic number. The sandwich theorem gives a polynomial-time approximation to both $\omega(G)$ and $\chi(G)$ that cannot be off by more than a factor $|V|$. In-approximability results by Håstad [135] and Feige and Kilian [98] show that neither $\omega(G)$ nor $\chi(G)$ can be approximated within a factor $|V|^{1-\varepsilon}$ for any $\varepsilon > 0$. Thus the sandwich theorem yields a very strong approximation guarantee.

Another famous application of SDP to combinatorial optimization is the NP-hard (see Karp [178])

Max-Cut Problem (MC) . Let $G = (V, E)$ be an undirected graph with edge weights $w_{ij} \geq 0$ $(i \neq j)$. Then (MC) consists in finding a partition (S, T) of V with $T = V \setminus S$ such that the weight of the edges in the S-T-cut $\sum_{i \in S, j \in T} w_{ij}$ is maximized. To deduce an SDP relaxation of (MC), we rewrite it as a Boolean quadratic optimization problem by introducing the bivalent variables

$$y_i = \begin{cases} 1 & \text{if vertex } i \in S, \\ -1 & \text{if vertex } i \in T, \end{cases} \qquad \forall i \in V.$$

Thus for a given edge $(i, j) \in E$ we have

$$y_i y_j = \begin{cases} -1 & (i,j) \text{ lies in the } S\text{-}T\text{-cut}, \\ 1 & \text{otherwise.} \end{cases}$$

The weight of the maximum cut is therefore given by

$$\max_{y \in \{-1,1\}^{|V|}} \left\{ \frac{1}{2} \sum_{i<j} w_{ij}(1 - y_i y_j) \right\} = \max_{y \in \{-1,1\}^{|V|}} \frac{1}{4} y^\top L y, \tag{MC}$$

where $L = -W + \mathrm{Diag}(We)$ and W is the matrix with zero diagonal and the (nonnegative) edge weights as off-diagonal entries. Now we use the matrix $Y := yy^\top$ to rewrite (MC)

$$\max \left\{ \frac{1}{4} \langle L, Y \rangle : \mathrm{diag}(Y) = e, Y \succcurlyeq 0, \mathrm{rank}(Y) = 1 \right\}. \tag{MC}$$

Dropping the rank one condition on Y yields the SDP relaxation

$$\max \left\{ \frac{1}{4} \langle L, Y \rangle : \mathrm{diag}(Y) = e, Y \succcurlyeq 0 \right\}. \tag{MC$_1$}$$

In their celebrated paper [121] Goemans and Williamson devised a randomized rounding scheme that uses (MC_1) to generate cuts in the graph. They can prove that one of these cuts gives a 0.878...-polynomial-time-approximation of (MC). Håstad [134] two years later showed that it is NP-complete to approximate (MC) within a factor $\frac{16}{17}$.

Rendl et al. [262] approximately solve (MC_1), strengthened by triangle inequalities with the help of a dynamic version of the bundle method (for details see Section 4.3) and use the obtained upper bounds in a Branch-and-Bound setting for finding exact solutions of (MC). Their approach nearly always outperforms all other approaches for (MC) and works particular well for dense graphs, where linear programming-based methods fail. In this thesis we apply a similar algorithmic approach to tackle (quadratic) ordering problems. As the quadratic ordering polytope is a face of the cut polytope, our method solves (MC) as a special case (if we leave out some constraint classes).

For more details on the θ-function and the Max-Cut Problem and for further applications of SDP to combinatorial optimization see the survey articles [120, 201, 261] and the book of De Klerk [83, Part II].

Chapter 4

Preliminaries: On Solving Semidefinite Programs

4.1 Introduction

In the previous chapter we have mentioned several areas of application for SDP. This of course motivates the research for efficient methods to solve SDP. Bearing the connections between LP and SDP in mind, it is not surprising that interior-point methods (IPMs) have been successfully extended from LP to SDP. They are for sure the most common and the most elegant way for efficiently solving SDPs. As for LP, there exist different variants of (IPMs) (e.g. primal and dual logarithmic barrier methods, affine-scaling methods, potential reduction methods) that have different strengths dependent on the structure of the SDP (for a survey on (IPMs) see e.g. the books of Wright [305] and De Klerk [83]). (IPMs) have polynomial worst-case iteration bounds for the computation of ϵ-optimal solutions, i.e. feasible (X, S) with duality gap $\langle S, X \rangle \leq \epsilon$ for a given tolerance $\epsilon > 0$ (for a more precise statement of the complexity results for (IPMs) see the review by Ramana and Pardalos [256]). Although the theoretical analysis of (IPMs) for LP and SDP is quite similar, there exist major differences concerning implementation and practical performance. In particular exploiting sparsity of the data matrices becomes very difficult for (IPMs) applied to SDP and there is still a lot of current research on this topic (see e.g. the survey articles of Fujisawa et al. [108] and of Nemirovski and Todd [233] or the recent research papers [14, 208]). Thus, in general, state-of-the-art (IPMs) are limited to SDPs involving matrices of dimension n=1000 and having a few thousand constraints.

Semidefinite relaxations that give good approximations for combinatorial optimization problems typically have a very large number of constraints m (e.g. $\Theta(n^2)$ over even $\Theta(n^3)$) and therefore motivated the research on new methods for solving SDP. The Boundary Point Method [217, 255], using quadratic regularization of SDP problems, was successfully applied to compute the θ-function (for details see Section 3.3) for large graphs. Also several first-order methods that use only gradient information have been developed lately. Burer and Monteiro [49] use their projected gradient algorithm for solving a nonconvex, nonlinear programming reformulation of the basic semidefinite relaxation ($\mathtt{MC}_1$) for Max-Cut. Davi, Jarre and Rendl [81, 82, 172] developed a hybrid approach that first uses a first-order method (APD-Method) to generate an approximate solution and then switches to a Krylov subspace algorithm (QMR method) to improve this approximation. They successfully apply their approach for computing the θ-function and the doubly nonnegative relaxation of the Max-Stable-Set-Problem for large graphs.

Finally the bundle method can be used for solving SDPs if the matrix dimension n is not too far beyond 1000. The number of constraints m can be significantly larger (even $\Theta(n^3)$). The bundle method is used for nonsmooth optimization and was introduced in the 1970's by Lemaréchal [202, 203]. Helmberg and Oustry [143] survey its applications to eigenvalue optimization and related problems. Helmberg and Rendl [140, 144] use the spectral bundle method to tackle several combinatorial optimization problems (Max-Cut, θ-function, bisection, frequency assignment problems) and provide a detailed comparison of their approach to other methods available. Fischer et al. [103] describe a dynamic version of the bundle method, where they maintain a basic set of constraints explicitly. They provide strong SDP-based bounds for dense instances of the Max-Cut and Equipartition Problem, which cannot be achieved with any of the other methods mentioned above.

In the following two sections we recall the basic properties and algorithmic machinery of primal-dual path-following (IPMs) and the dynamic version of the bundle method. We will use the dynamic version of the bundle method (that applies a primal-dual path-following method for function evaluation) for the practical solution of semidefinite relaxations of facility layouts and logistic problems in Parts II and III.

4.2 Interior-Point Methods

Nesterov and Nemirovski [235] provided the theoretical background for solving SDPs with (IPMs) by studying linear optimization problems over closed convex cones. They showed that theses problems can be solved in polynomial time by sequential minimization techniques, where the conic constraint is discarded and a suitable, self-concordant barrier term is added to the objective. Self-concordant barriers go to infinity as the boundary of the cone is approached and can be minimized efficiently by Newton's method, as they are smooth convex functions with Lipschitz continuous second derivatives. A computable self-concordant barrier for the cone of semidefinite matrices is given by $f_{\text{bar}}(X) = -\log\ \det(X)$. Practical experience indicates that primal-dual path-following methods are best suited for our purposes (the optimization of a linear function over the elliptope). These methods minimize the duality gap $\langle C, X\rangle - b^\top y = \langle X, S\rangle$ and use a combined primal-dual barrier function $-\log\ \det(XS)$. We assume that strong duality holds and perturb the necessary and sufficient optimality conditions (3.2) to get the following system of equations

$$X \in \mathcal{P},\ (y, S) \in \mathcal{D},\ XS = \mu I, \tag{4.1}$$

where $\mu \in \mathbb{R}^+$. Clearly, $X \succ 0$ and $S \succ 0$ must be satisfied for solutions of (4.1), as $XS = \mu I$ forces X and S to be nonsingular. In fact, (4.1) has a unique solution (X_μ, S_μ, y_μ), iff (D) and (P) are both strictly feasible. Furthermore (X_μ, S_μ, y_μ) form an analytic curve (the central path), parametrized by μ. This can be shown by straightforward application of the implicit function theorem (for proofs of the two basic results mentioned above see e.g. [83, Chapter 3]).

Primal-dual path-following methods use (4.1) to obtain search directions $(\Delta X, \Delta S, \Delta y)$ that approximately satisfy the partly nonlinear, overdetermined $(m + 1 + n^2 + \binom{n+1}{2}$ equations, $m + 2\binom{n+1}{2}$ variables) system

$$\begin{aligned} X + \Delta X \succcurlyeq 0,\ S + \Delta S \succcurlyeq 0,\\ \langle A_i, \Delta X\rangle = 0, \quad i = 1, \ldots, m,\\ \sum_{i=1}^{m} \Delta y_i A_i + \Delta S = 0,\\ (X + \Delta X)(S + \Delta S) = \mu I,\\ \Delta X = \Delta X^\top. \end{aligned} \tag{4.2}$$

The probably most straightforward way to determine approximate solutions of system (4.2) is linearizing its nonlinear equation and then determining a least squares solution (with the Gauss-Newton method), for details see Kruk et al. [192] and De Klerk et al. [85]. The most popular approach (proposed by Zhang [309]) to get approximate solutions of system (4.2) is to drop $\Delta X = \Delta X^\top$ and replace the nonlinear equation by

$$H_P(\Delta X S + X \Delta S) = \mu I - H_P(XS), \tag{4.3}$$

where H_P is defined by

$$H_P(M) := \frac{1}{2}\left[PMP^{-1} + P^{-\top} M^\top P^\top\right],$$

for any matrix M, where P is an arbitrary, nonsingular matrix. This gives now a square linear system that has a unique solution for several choices of P (for details see [281]). Furthermore, if this system has a solution, then ΔX is symmetric. The most common choices for the scaling matrix P, that of course ensure existence and uniqueness of each of the resulting search directions, are the following

- $P = S^{\frac{1}{2}}$ examined by Monteiro [228], Helmberg et al. [146] and Kojima et al. [185],
- $P = X^{-\frac{1}{2}}$ examined by Monteiro [228] and Kojima et al. [185],
- $P = \left[X^{\frac{1}{2}}\left(X^{\frac{1}{2}} S X^{\frac{1}{2}}\right)^{-\frac{1}{2}} X^{\frac{1}{2}}\right]^{\frac{1}{2}}$ examined by Nesterov and Todd [236].

We refer to Todd [294] for an investigation of the theoretical properties of about 20 different search-directions (including the ones mentioned above) used in primal-dual interior-point methods. There exist many variants of primal-dual path-following methods that essentially differ in how μ is update (reduced), and how system (4.2) is symmetrized and solved. A very popular and practically efficient variant that is used in popular SDP software, like Sedumi by Sturm [289], SDPT3 by Toh et al. [51, 297] and CSDP by Borchers [40, 212], is the predictor-corrector method possessing superlinear convergence properties [254]. A practical comparison of many SDP-solvers on several different data is provided by Mittelmann [227] on his website.

4.3 A Dynamic Version of the Bundle Method

Most combinatorial optimization problems can be formulated as linear or quadratic or semidefinite programs in binary variables. Tractable relaxations are obtained by replacing the integrality conditions with bounds on the variables. Then a polyhedral approach is used to get tight relaxations. If the number of cutting planes in the partial description of the convex hull of integer solutions gets "large"[1], this poses a serious challenge even to state of the art software. In theses cases it can be helpful to work with the Lagrangian dual to handle the cutting planes only indirectly. As the Lagrangian dual functional is nonsmooth, a method for nonsmooth optimization has to be applied to it. These iterative algorithms use function and subgradient evaluations of the dual functional to determine a sequence of trial points. When solving semidefinite relaxations of combinatorial optimization problems, we want to maintain some constraints (e.g. semidefiniteness) explicitly, resulting in nontrivial function evaluations. Thus we are interested in algorithms like the bundle method that work well even with a low (less than one hundred) number of function evaluations.

In the following we describe a dynamic version of the bundle method (for further details see [103]). We are going to use exactly this approach to approximately solve the semidefinite relaxations of all facility layout and logistic problems in Parts II and III.

[1] What is "large" of course depends on the number of variables n and the type and structure of the optimization problem.

Let us consider an SDP of the form

$$\begin{aligned} p^* &:= \max_X \{ \langle C, X\rangle \ : \ X \in \mathcal{S} \cap \mathcal{T} \}, \\ \mathcal{S} &:= \{ \ X \mid \langle A_i, X\rangle = a_i \ (i = 1, \ldots, k), \ X \in \mathcal{S}_n^+ \ \}, \\ \mathcal{T} &:= \{ \ X \mid \langle B_i, X\rangle \le b_i \ (i = 1, \ldots, l) \ \}. \end{aligned} \tag{4.4}$$

We assume that maintaining only set $\mathcal{S}$ results in an SDP that is still manageable by `(IPMs)`. But the inclusion of $\mathcal{T}$ would make the SDP computationally far too expensive.[2] Thus we suggest to maintain $X \in \mathcal{S}$ explicitly and put $X \in \mathcal{T}$ into the cost function by taking the partial Lagrangian dual

$$L(X, y) = \langle C, X\rangle + \sum_{i=1}^{l} y_i(b_i - \langle B_i, X\rangle).$$

Now we can rewrite the original problem (4.4) as

$$p^* = \max_{X \in \mathcal{S}} \min_{y \ge 0} L(X, y).$$

Assuming the usual strict feasibility conditions and applying the Minimax inequality (3.1) yields

$$p^* = \min_{y \ge 0} f(y),$$

where $f(y) = \max_{X \in \mathcal{S}} L(X, y)$. Thus to compute p^*, we can minimize f. f is the pointwise maximum of linear functions and therefore continuous and convex but not differentiable at points where the maximum is not unique.

We use the bundle method tailored to our problem (see [154] for a comprehensive survey) to minimize the nonsmooth function f over $y \ge 0$. The bundle method iteratively evalutes f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to solving an SDP over the set $\mathcal{S}$ which we have assumed to be manageable by `(IPMs)`. If we have $f(y^*) = L(X^*, y^*)$, then the maximum of L is attained at $X^* \in \mathcal{S}$ for some given y^*. Setting

$$g_i^* := b_i - \langle B_i, X^*\rangle, \ i = 1, \ldots, l,$$

the inequality

$$f(y) \ge L(X^*, y) = \langle C, X^*\rangle + \sum_{i=1}^{l} y_i(b_i - \langle B_i, X^*\rangle) = f(y^*) + \sum_{i=1}^{l} g_i^*(y_i - y_i^*), \ \forall y \in \mathbb{R}^l, \tag{4.5}$$

defines g^* to be a subgradient of f at y^*.

Now suppose we have evaluated f at $k \ge 1$ feasible points $y_1 \ge 0, \ldots, y_k \ge 0$ with respective maximizers $X_1, \ldots, X_k$ and subgradients $g_1, \ldots, g_k$. We denote the current iterate by $\overline{y} \in \{y_1, \ldots, y_k\}$ and set $f_i := f(y_i)$. The subgradient inequality (4.5) implies

$$f(y) \ge \max_{i \in \{1, \ldots, k\}} \left\{ f_i + g_i^\top (y - y_i) \right\} =: f_m(y), \ \forall y \in \mathbb{R}^l.$$

The minorant $f_m(y)$ is equal to $f(y)$ for $y \in \{y_1, \ldots, y_k\}$ and thus can be used to approximate f in the neighbourhood of the current iterate $\overline{y}$. To simplify the presentation we rewrite the minorant

$$f_m(y) = \max_{\lambda \in \Delta^k} \lambda^\top (H + G^\top y),$$

[2] Exactly this situation occurs most of the semidefinite relaxations considered in this thesis.

where $H = (h_1, \ldots, h_k)^\top$, $h_i = f_i - g_i^\top y_i$, $G = (g_1, \ldots, g_k)$ and Δ^k denotes the k-dimensional standard simplex. Additionally we add a regularization term to f_m

$$f_r(y) := f_m(y) + \frac{1}{2t}\|y - \overline{y}\|^2,$$

where $t \in \mathbb{R}^+$ controls how close we stay to $\overline{y}$. Introducing Lagrange multipliers η for the sign constraints on y and applying again the Minimax inequality yields the following dual problem

$$\min_{y \geq 0} f_r(y) = \max_{\lambda \in \Delta^k, \eta \geq 0} \min_y \lambda^\top (H + G^\top y) + \frac{1}{2t}\|y - \overline{y}\|^2 - y^\top \eta.$$

The inner minimization is a strictly convex unconstrained quadratic optimization problem in y, hence we can replace it by asking that the first-order optimality conditions

$$y = \overline{y} + t(\eta - G\lambda),$$

hold. Using this relation yields the following outer maximization problem

$$\max_{\lambda \in \Delta^k, \eta \geq 0} \lambda^\top (H + G^\top \overline{y}) - \frac{t}{2}\|\eta - G\lambda\|^2 - \overline{y}^\top \eta.$$

We solve this convex quadratic optimization problem in λ and η approximately by keeping alternately one set of the variables constant. Keeping η constant results in a convex quadratic problem over the standard simplex Δ^k and keeping λ constant allows to solve η coordinatewise, see e.g. [147]. We iterate this process several times and then use the estimates $\overline{\lambda}$ and $\overline{\eta}$ to get a new feasible dual point

$$y_{k+1} = \overline{y} + t(\overline{\eta} - G\overline{\lambda}).$$

Minimizing $f_r(y)$ additionally yields a new primal matrix $X_{k+1} := \sum_{i=1}^k \lambda_i X_i$. Finally we evaluate f at y_{k+1} and use some standard criteria to decide whether y_{k+1} becomes the new trial point. Under appropriate stopping conditions (X_k, y_k) converge towards an optimal primal-dual solution pair of the original problem (4.4), see e.g. [105, 204, 274].

To further improve efficiency, we only dualize those constraints in every iteration, which are likely to be active at the optimum. So we face a situation well known from active set methods. If we would know the constraints active at the optimum, we would not care about the other constraints any more. But this information is not explicitly available to us. So we use primal feasibility and dual optimality information to identify important constraints. In every iteration we add constraints (strongly) violated for the actual primal iterate X_k and remove constraints associated to components of the dual multiplier y_k that are close to zero. Thus f changes in the course of the algorithm. A convergence analysis of this dynamic version of the bundle method can be found in [35].

Part II

Facility Layout

Chapter 5

Semidefinite Relaxations of Ordering Problems

Authors: Philipp Hungerländer and Franz Rendl

Abstract: Ordering problems assign weights to each ordering and ask to find an ordering of maximum weight. We consider problems where the cost function is either linear or quadratic. In the first case, there is a given profit if the element u is before v in the ordering. In the second case, the profit depends on whether u is before v and r is before s.

The linear ordering problem is well studied, with exact solution methods based on polyhedral relaxations. The quadratic ordering problem does not seem to have attracted similar attention. We present a systematic investigation of semidefinite optimization based relaxations for the quadratic ordering problem, extending and improving existing approaches. We show the efficiency of our relaxations by providing computational experience on a variety of problem classes.

Keywords: Ordering Problems; Linear Arrangement; Facility Layout; Crossing Minimization; Semidefinite Optimization; Combinatorial Optimization

5.1 Introduction

Ordering problems associate to each ordering (or permutation) of the set $N := \{1, \dots, n\}$ of n objects a profit and the goal is to find an ordering of maximum profit. In the simplest case of the *linear ordering problem (LOP)*, this profit is determined by those pairs $(u, v) \in N \times N$, where u comes before v in the ordering. Thus LOP can be defined as follows. Given an $n \times n$ matrix $D = (d_{ij})$ of integers, find a simultaneous permutation ϕ of the rows and columns of D such that

$$\sum_{i<j} d_{\phi(i),\phi(j)}$$

is maximized. Equivalently, we can interpret d_{ij} as weights of a complete directed graph G with vertex set N. Then LOP consists of finding a complete acyclic subgraph of G of maximum total edge weight.

$$z_* := \max \left\{ \sum\nolimits_{i<j} d_{\phi(i),\phi(j)} : \quad \phi \in \Pi \right\} \tag{5.1}$$

The permutation ϕ gives the ordering of the vertices N of G and the cost function consists of the sum of all edge weights d_{uv} where u comes before v in this ordering. The set of permutations is denoted by Π.

LOP is well known to be NP-complete [112] and arises in a large number of applications in such diverse fields as economy, sociology, graph drawing [173], archaeology, scheduling [39] and also assessment of corruption perception [1]. Two well known examples of LOP are the determination of ancestry relationships [119] and the triangulation of input-output matrices of an economy [60, 205], where the optimal ordering gives some information about the stability of the economic system.

The simplest formulation of LOP uses linear programming in binary variables and will be briefly recalled in section 5.2. It is the basis for exact methods which use the continuous linear relaxation in an enumerative scheme. The resulting bounds are not always of sufficient quality, making this approach impractical for larger instances. In section 5.3 we consider semidefinite relaxations and investigate in some detail how the linear description of the problem can be 'lifted' into the semidefinite model to yield tight approximations. Even though some basic semidefinite models have been proposed in the literature, see for instance [20, 48, 237], we will provide in section 5.3 a systematic investigation on various ways to derive constraints for the semidefinite model. The semidefinite relaxations are the natural setting for the *quadratic ordering problem (QOP)*, where the profit depends on whether u comes before v and r is before s. In section 5.4 we describe some nontrivial applications of QOP, notably the 'multilevel crossing minimization problem' and the 'betweenness problem'.

The last part of the paper contains computational results on a variety of different problem types. We use a complete description of the linear ordering polytope in small dimensions to show that the semidefinite relaxation $(\texttt{SDP}_4)$ below is exact for any instance of size $n \leq 6$ and identifies all but one class of facets of the polytope for $n = 7$. This provides an indication of the potential strength of this approach also for larger instances. We also compare the new model with linear-programming based, and much cheaper bounds on LOP instances from the LOLIB library.

Finally, we provide some improved bounding results for linear arrangement problems and report new optimal solutions for bipartite crossing minimization and single-row layout problems.

5.2 Linear Ordering as a Linear Program in 0-1 variables

The linear ordering problem has a natural formulation as a linear program in 0-1 variables. An instance of the problem is defined by the $n \times n$ matrix $D = (d_{ij})$. We introduce binary variables x_{ij} with $x_{ij} = 1$ if i comes before j and $x_{ij} = 0$ otherwise. Then it is not hard to show that the following constraints describe linear orderings of the set N:

$$x_{ij} + x_{ji} = 1,\ \forall\, i \neq j, \tag{5.2}$$

$$x_{ij} + x_{jk} + x_{ki} \in \{1, 2\},\ \forall\, i, j, k, \tag{5.3}$$

$$x_{ij} \in \{0, 1\},\ \forall\, i \neq j. \tag{5.4}$$

The first condition models the fact that either i is before j or j is before i. The second condition rules out the existence of directed 3-cycles and is sufficient to insure that there is no directed cycle. Hence the feasible solutions of these constraints describe complete acyclic digraphs. Maximizing $\sum_{i \neq j} d_{ij} x_{ij}$ over the constraints (5.2)–(5.4) therefore solves LOP. The equations (5.2) are used to eliminate x_{ji} for $j > i$. This leads to the following formulation of LOP as linear program in binary variables, see [124],

$$z_* = \max\ \left\{ \sum\nolimits_{i<j} (d_{ij} - d_{ji}) x_{ij} + d_{ji} :\ \ x_{ij} \in \{0, 1\},\ \ 0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1,\ \forall\ i < j < k \right\}.$$

The linear relaxation is obtained by leaving out the integrality conditions on the variables. This results in a linear program with $\binom{n}{2}$ variables and $2\binom{n}{3}$ three-cycle inequalities. It poses a serious challenge to standard LP solvers, once $n \approx 200$. State-of-the-art exact algorithms can solve large instances from specific instance classes with up to 150 nodes, while they fail on other much smaller instances with only 50 nodes. For a detailed overview over benchmark instances for LOP solved and not yet solved see [220, Appendix].

The computation time of exact algorithms increases rapidly with problem size. Currently available exact algorithms include a Branch-and-Bound algorithm that uses a linear programming based lower bound by Kaas [175], a Branch-and-Cut algorithm proposed by Grötschel, Jünger and Reinelt [124] and a combined interior-point cutting-plane algorithm by Mitchell and Borchers [225] who explore polyhedral relaxations of the problem and also provide computational results using Branch-and-Cut.

There exist many heuristics and methaheuristics for LOP and some of them are quite good in finding the optimal solution for large instances in reasonable time. For a recent survey and comparison see [220]. Of course these heuristics do not provide an optimality certificate of the solutions found.

In this paper we are mostly interested in the lower bound computation by analyzing matrix liftings of the ordering problem. For this purpose it is convenient to reformulate the problem in variables taking the values -1 and 1. The variable transformation

$$y_{ij} = 2x_{ij} - 1 \tag{5.5}$$

leads to the equivalent problem

$$z_* = \max \left\{ \sum\nolimits_{i<j} (d_{ij} - d_{ji}) \frac{y_{ij}+1}{2} + d_{ji} : \ \ y_{ij} \in \{-1,1\}, \ \ |y_{ij} + y_{jk} - y_{ik}| = 1, \ \forall\, i<j<k \right\}. \tag{5.6}$$

In [138] it is shown that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties are preserved. Omitting the integrality condition $y_{ij} \in \{-1,1\}$ gives the linear relaxation

$$z_{LP} := \max \sum_{i<j} (d_{ij} - d_{ji}) \frac{y_{ij}+1}{2} + d_{ji}, \tag{5.7a}$$

$$\text{subject to } \ -1 \le y_{ij} + y_{jk} - y_{ik} \le 1, \ \forall\, i<j<k, \tag{5.7b}$$

$$-1 \le y_{ij} \le 1, \ \forall\, i<j. \tag{5.7c}$$

The upper bound z_{LP} may lead to gaps between z_* and z_{LP} which are too large for efficient pruning in Branch-and-Bound enumeration. We refer to the column LP-gap in the Tables 5.3 and 5.4 below. Thus it would be desirable to have some tighter approximation available. In the next section we take a closer look at relaxations which are based on semidefinite optimization.

5.3 Semidefinite relaxations

The matrix lifting approach takes a vector y and considers the matrix $Y = yy^T$. We are interested in linear and quadratic orderings and consider the polytope

$$\mathcal{P}_{LQO} := \operatorname{conv}\left\{ \begin{pmatrix}1\\ y\end{pmatrix}\begin{pmatrix}1\\ y\end{pmatrix}^{\top} : \ y \in \{-1,1\}, \ y \text{ satisfies (5.7b)} \right\}.$$

The nonconvex equation $Y - yy^T = 0$ is relaxed to the constraint

$$Y - yy^T \succeq 0,$$

which is convex due to the Schur-complement lemma

$$\begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \succeq 0 \Leftrightarrow Y - yy^T \succeq 0.$$

Moreover, the main diagonal entries of Y correspond to y_{ij}^2, hence $\operatorname{diag}(Y) = e$, the vector of all ones. We therefore conclude that any $Y \in \mathcal{P}_{LQO}$ satisfies

$$Y - yy^T \succeq 0, \quad \operatorname{diag}(Y) = e. \tag{5.8}$$

To simplify notation let us introduce

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{5.9}$$

where $\dim(Z) = \binom{n}{2} + 1 = p$ and $Z = (z_{ij})$. In this case $Y - yy^T \succeq 0 \Leftrightarrow Z \succeq 0$. Hence, the following basic set $\mathcal{B}$ contains $\mathcal{P}_{LQO}$.

$$\mathcal{B} := \{ \ Z : \ \operatorname{diag}(Z) = e, Z \succeq 0 \ \} \tag{5.10}$$

In order to express constraints on y in terms of Y, they have to be reformulated as quadratic conditions in y. A natural way to do this for $|y_{ij} + y_{jk} - y_{ik}| = 1$ consists in squaring both sides, leading to

$$y_{ij}^2 + y_{jk}^2 + y_{ik}^2 + 2(y_{ij,jk} - y_{ij,ik} - y_{ik,jk}) = 1, \ \forall \ i < j < k. \tag{5.11}$$

Since $y_{ij}^2 = 1$, this simplifies to

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \ \forall \ i < j < k. \tag{5.12}$$

In [48] it is shown that these equations formulated in the $\{0, 1\}$ model describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. We now formulate LOP as a semidefinite optimization problem in bivalent variables.

Proposition 5.1 *The problem*

$$\max \{ \ d^T y : \ \ Z \text{ satisfies } (5.9), \ \ Z \in \mathcal{B}, \ \ y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \ \forall \ i < j < k, \ \ y_{ij} \in \{-1, 1\} \ \}$$

is equivalent to LOP.

Proof. Since $y_{ij}^2 = 1$ we have $\operatorname{diag}(Y - yy^T) = 0$, which together with $Y - yy^T \succeq 0$ shows that in fact $Y = yy^T$. The 3-cycle equations (5.12) ensure that $|y_{ij} + y_{jk} - y_{ik}| = 1$ holds. The objective value differs from z_* only in an additive constant. □

We are now dropping the integrality condition on y and obtain the following basic semidefinite relaxation of LOP. The objective function assigns currently only costs to y, but not to Y. From now on we do not make any assumptions on the cost function and simply consider a generic cost function $\langle C, Z \rangle$, where C is a symmetric matrix of order p, and Z is given by (5.9). Thus we get the following basic semidefinite relaxation

$$\max \{ \ \langle C, Z \rangle : \ \ Z \text{ partitioned as in (5.9) satisfies (5.12)}, \ \ Z \in \mathcal{B} \ \}. \tag{$\texttt{SDP}_1$}$$

There are some obvious ways to tighten ($\texttt{SDP}_1$). First of all we observe that Y, and therefore Z in (5.9), is actually a matrix with $\{-1, 1\}$ entries in the original LOP formulation. Hence it satisfies the triangle inequalities, defining the metric polytope $\mathcal{M}$

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e, \ \forall \ i < j < k \right\}. \tag{5.13}$$

We note that the metric polytope is defined through $4\binom{p}{3} \approx \frac{1}{12}n^6$ facets. They are used as triangle inequalities of the max-cut polytope in [214, 279, 285]. The basic relaxation ($\texttt{SDP}_1$) can therefore be improved by asking in addition that $Z \in \mathcal{M}$ yielding $(\texttt{SDP}_2)$.

Another generic improvement was suggested by Lovász and Schrijver in [214]. Applied to our problem, this approach suggests to multiply the 3-cycle inequalities (5.7b)

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0,$$

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. This results in the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \ \forall\, i < j < k,\ l < m, \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \ \forall\, i < j < k,\ l < m. \end{aligned} \tag{5.14}$$

We define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies (5.14)} \ \}, \tag{5.15}$$

consisting of $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ constraints. The basic relaxation $(\texttt{SDP}_1)$ can therefore also be improved by asking in addition that $Z \in \mathcal{LS}$ yielding $(\texttt{SDP}_3)$. In summary, we get the following tractable relaxation of $\mathcal{P}_{LQO}$, part of which (without the matrix cuts (5.15)) has been investigated in [48] for bipartite crossing minimization problems and in [20] for single-row layout problems.

$$z_{SDP} := \max\ \{\ \langle C, Z\rangle :\ \ Z \text{ partitioned as in (5.9) satisfies (5.12)},\ \ Z \in \mathcal{B},\ \ Z \in \mathcal{M},\ \ Z \in \mathcal{LS}\ \} \quad (\texttt{SDP}_4)$$

We close this section with a few simple observations. First, it is not hard to verify that any Z feasible for $(\texttt{SDP}_1)$ has in its first column a vector y which satisfies the 3-cycle inequalities (5.7b). This follows from the semidefiniteness of the following submatrix of Z

$$\begin{pmatrix} 1 & y_{ij} & y_{ik} & y_{jk} \\ y_{ij} & 1 & y_{ij,ik} & y_{ij,jk} \\ y_{ik} & y_{ik,ij} & 1 & y_{ik,jk} \\ y_{jk} & y_{jk,ij} & y_{jk,ik} & 1 \end{pmatrix}.$$

As a consequence, the basic semidefinite relaxation $(\texttt{SDP}_1)$ is at least as strong as the linear relaxation (5.7).

There are further methods to tighten the relaxation. Instead of the Lovász-Schrijver lifting procedure, we could for instance multiply different pairs of 3-cycle inequalities. For further details on this we refer to the forthcoming dissertation [156].

Remark 1 *The original formulation of the ordering problem was done in dimension $2\binom{n}{2}$, as we introduced variables y_{ij} for $i \neq j$. The equations $y_{ij} + y_{ji} = 0$ were then used to eliminate half the variables, leading to a new model in dimension $\binom{n}{2}$. Would we get a stronger semidefinite relaxation by working with matrices of order $2\binom{n}{2}$ instead of $\binom{n}{2}$? It is not difficult to show that this is not the case.*

Let m linear equality constraints $Ay = c$ be given. If there exists some invertible $m \times m$ matrix B, we can partition the linear system in the following way $Ay = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = c$ and then solve for v, $v = B^{-1}(c - Cu)$. Therefore $\begin{bmatrix} 1 \\ u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \\ B^{-1}c & -B^{-1}C \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix} = D \begin{bmatrix} 1 \\ u \end{bmatrix}$, defining the full column rank matrix D. From this it is clear that $\begin{bmatrix} 1 \\ u \\ v \end{bmatrix} \begin{bmatrix} 1 \\ u \\ v \end{bmatrix}^T = D \begin{bmatrix} 1 \\ u \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix}^T D^T$. Therefore $Z := \begin{bmatrix} 1 & u^\top & v^\top \\ u & U & W^\top \\ v & W & V \end{bmatrix} = D \begin{bmatrix} 1 & u^\top \\ u & U \end{bmatrix} D^\top$, thus $\begin{bmatrix} 1 & u^\top \\ u & U \end{bmatrix} \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. Hence we do not weaken the relaxation by first moving into *the subspace, given by the equations, and then lifting the problem to matrix space.*

5.4 Areas of application

In this section we point out some areas of application of the semidefinite relaxations analyzed above.

We have mentioned already that the cost function of the semidefinite relaxation can model profits that depend on products of variables y_{ij} and therefore on the relative position of two pairs of elements in the ordering.

Going from linear to quadratic objective functions usually makes an optimization problem much harder. For example the binary maximization of a linear function over the hypercube, which is trivial, becomes the *maximum cut (MC)* problem [285] and thus NP-hard. In our case LOP is already NP-hard, nonetheless the practical hardness of the *quadratic ordering problem (QOP)* is significantly higher and classical approaches used for LOP are hopeless for QOP. For the semidefinite approach the linear and quadratic variants of the problem are essentially equally hard to solve. We will demonstrate this for *multi-level crossing minimization (MLCM)* and the *weighted betweenness problem (WB)*, which are two special cases of QOP.

5.4.1 Multi-level crossing minimization

Let us define a proper level graph [136] as a graph $G(V, E)$, with vertex set $V = V_1 \cup V_2 \cup \ldots \cup V_q,\ V_i \cap V_j = \emptyset,\ i \neq j$, and edge set $E = E_1 \cup E_2 \cup \ldots \cup E_{q-1},\ E_i \subseteq V_i \times V_{i+1}$. Now MLCM consists of drawing a proper level graph such that the number of edge crossings is minimized when the edges are drawn as straight lines connecting the endnodes.

Next we explain how to count the crossings based on quadratic ordering. We need to consider all distinct pairs $s, t \in V_i$ and all distinct pairs $u, v \in V_{i+1}$ for all i.

Let us give an illustrating example by looking at the concrete cost structure of a small bipartite instance with $V_1 = \{1, 2, 3\}$, $V_2 = \{4, 5, 6, 7\}$ and $E_1 = \{(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 7), (3, 4), (3, 6)\}$. The matrix C_1 is indexed rowwise by the pairs $(1, 2), (1, 3), (2, 3)$. The columns are indexed by all pairs of V_2, i.e. $(4, 5), (4, 6), (4, 7), (5, 6), (5, 7), (6, 7)$. Let us set

$$C = \begin{pmatrix} 0 & -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & -1 \end{pmatrix}.$$

The entry $C_{(1,2),(4,5)} = 0$, because independent of the order of $\{1, 2\}$ and $\{4, 5\}$, there will be exactly one crossing in the subgraph induced by these vertices. Thus, the 0 entries in this matrix reflect all subgraphs, where the number of crossings is independent of the ordering of the vertices. Let us now look at $C_{(1,2),(4,6)} = -1$. The order $(1, 2)$ and $(4, 6)$ yields one crossing in this subgraph. It could be avoided if exactly one of the two orderings is reversed, which amounts to asking that $y_{12}y_{46} = -1$. Maximization will try to make the term $y_{12}y_{46}C_{(1,2),(4,6)}$ equal to one, avoiding the crossing. In a similar way we see that the subgraph induced by the vertices $1, 2, 4, 7$ has no crossing precisely if $y_{12}y_{47} = 1$. Thus we set $C_{(1,2),(4,7)} = 1$.

In the general case we will have crossing matrices C_i for each layer $i = 1, \ldots, q - 1$. We can therefore model MLCM as a quadratic ordering problem with cost matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & C_1 & \dots & 0 & 0 \\ 0 & C_1^\top & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 & C_{q-1} \\ 0 & 0 & 0 & \dots & C_{q-1}^\top & 0 \end{bmatrix}$$

where the $C_i,\ i \in \{1, \dots, q-1\}$ have dimension $\binom{|V_i|}{2} \times \binom{|V_{i+1}|}{2}$ and are determined by the edge set E_i as described above.

5.4.2 Weighted betweenness problem

An input to WB consists of n objects, a set $\mathcal{B}$ of betweenness conditions and a set $\overline{\mathcal{B}}$ of non-betweenness conditions ($\mathcal{B} \cap \overline{\mathcal{B}} = \emptyset$). The elements of $\mathcal{B}$ and $\overline{\mathcal{B}}$ are triples (i, j, k) with associated costs w_{ijk} for not placing respectively placing object j between objects i and k. Now the task in WB is to find a linear ordering of the objects such that the sum of costs is minimized.

To represent the given cost structure in our model, we use the fact that the statement j is between i and k is equivalent to the statement i is in front of j and j is in front of k or k is in front of j and j is in front of i. In terms of our ordering variables, we have that j is between i and k exactly if $y_{ij}y_{jk} = 1$. Thus the cost matrix C which is indexed by all ordered pairs, has a contribution to $C_{(i,j),(j,k)}$ proportional to w_{ijk}.

The cost matrix C for WB is also quite sparse as for n objects we have $O(n^4)$ matrix entries but only $O(n^3)$ triples.

5.5 First computational experiments

($\mathtt{SDP}_4$) is formulated in the space of symmetric matrices of order $p = \binom{n}{2} + 1$ and has p equality constraints $\text{diag}(Z) = e$ together with $\binom{n}{3}$ 3-cycle equations. Additionally there are $O(n^5)$ inequalities from the Lovasz-Schrijver lifting together with $O(n^6)$ triangle inequalities. A direct solution using standard interior-point based methods is therefore only feasible for very small values of n, such as $n \leq 8$.

We use the MATLAB toolbox SEDUMI for semidefinite optimization to solve this relaxation for small n. As a first experiment we consider the full description of the linear ordering polytope in small dimensions, and try to recover the correct right hand side of the facets for $n \in \{6, 7\}$.

We compare the linear programming (LP) relaxation (5.7) to the semidefinite relaxations ($\mathtt{SDP}_1$)–($\mathtt{SDP}_4$) from Section 5.3.

In Table 5.1 we examine all nontrivial facets of the linear ordering polytopes for 6 and 7 nodes. The facets are collected under `http://comopt.ifi.uni-heidelberg.de/software/SMAPO/lop/lop.html`. We also use the same labeling, see column 1. As usual, n denotes the dimension of LOP, and *opt* gives the optimal solution. All relaxations are solved to optimality using the standard settings of SEDUMI [289]. We also include the combinatorial instances Paley 11 and Paley 19, which are notoriously difficult for linear relaxations.

From Table 5.1 we conclude that the triangle inequalities and the matrix cuts are incomparable, as there are instances where ($\mathtt{SDP}_2$) is tighter than ($\mathtt{SDP}_3$) and vice versa. The basic relaxation ($\mathtt{SDP}_1$) improves upon the pure linear model (LP), but does not give the correct facets for $n = 6$. Adding either (5.13) or (5.15) gives the correct facets for $n = 6$. Furthermore the full model ($\mathtt{SDP}_4$) identifies all except one of the facets correctly for $n = 7$.

facet	n	opt	(LP)	$(\mathtt{SDP}_1)$	$(\mathtt{SDP}_2)$	$(\mathtt{SDP}_3)$	$(\mathtt{SDP}_4)$
FC3	6	7	7.5	7.35	7	7	7
FC4, 5	6	8	8.5	8.35	8	8	8
FC3	7	7	7.5	7.35	7	7	7
FC4, 20	7	8	8.5	8.35	8	8	8
FC5	7	9	9.5	9.37	9	9.09	9
FC6	7	9	9.5	9.37	9	9	9
FC10, 25	7	9	9.5	9.37	9.06	9	9
FC21	7	9	9.5	9.37	9	9.01	9
FC7, 9, 22, 24	7	10	10.5	10.37	10.11	10	10
FC8, 13, 23	7	10	10.5	10.37	10.19	10	10
FC11	7	10	10.5	10.37	10	10	10
FC12	7	10	10.5	10.37	10	10.03	10
FC14	7	10	10.5	10.35	10.35	10.24	10.22
FC15, 16	7	11	11.5	11.37	11.22	11	11
FC26	7	11	11.5	11.37	11.23	11	11
FC17, 27	7	13	13.5	13.40	13	13	13
FC18	7	14	14.5	14.40	14.17	14.04	14
FC19	7	14	14.5	14.40	14.10	14.01	14
Paley	11	35	36.67	36.03	36.03	35.92	35.92
Paley	19	107	114	110.70	110.70	110.50	110.50

Table 5.1: Marginal improvement of various semidefinite relaxations as compared to the linear relaxation on facets of the linear ordering polytope for $n = 6$ and $n = 7$

As a first conclusion we observe that the semidefinite approach provides a substantial improvement over the polyhedral approach in the approximation of the linear ordering polytope of small dimensions.

It should also be clear that we need to use algorithmic alternatives to solve instances of reasonable size ($n \geq 30$). In the following section we describe a practical implementation to get approximate solutions of $(\mathtt{SDP}_4)$ also for larger instances, $n \approx 100$.

5.6 A practical implementation for $(\mathtt{SDP}_4)$

Looking at the constraint classes and their sizes in $(\mathtt{SDP}_4)$, it should be clear that maintaining explicitely $O(n^3)$ or more constraints is not an attractive option. We therefore take up the approach suggested in [103] and adapt it to our problem. We first note that maintaining the constraints $Z \succeq 0$, $\operatorname{diag}(Z) = e$ explicitely leads to the following running times in seconds on an Intel Xeon 5160 processor with 3 GHz. We use a standard interior-point method to solve this relaxation, see for instance [146], and summarize the results in Table 5.2.

n	p	time
30	436	3
50	1226	40
70	2416	500
100	4951	3000

Table 5.2: Average computation times (in seconds) using interior-point methods to optimize over (5.10), where Z is of order p.

To get approximate solutions to $(\mathtt{SDP}_4)$, we only maintain $Z \in \mathcal{B}$ explicitely, and deal with all other constraints through Lagrangian duality. For notational convenience, let us formally denote the 3-cycle equations (5.12) by

$$e - A(Z) = 0.$$

The remaining $O(n^6)$ inequalities $\mathcal{M} \cap \mathcal{LS}$ are collected in $e - D(Z) \geq 0$. We consider the partial Lagrangian dual defined through the Lagrangian

$$\mathcal{L}(Z, \lambda, \mu) := \langle C, Z \rangle + \lambda^\top (e - \mathcal{A}(Z)) + \mu^\top (e - \mathcal{D}(Z)).$$

The dual function is thus given by

$$f(\lambda, \mu) := \max_{Z \in \mathcal{B}} \mathcal{L}(Z, \lambda, \mu) = e^\top \lambda + e^\top \mu + \max_{Z \in \mathcal{B}} \langle C - \mathcal{A}^\top(\lambda) - \mathcal{D}^\top(\mu), Z \rangle.$$

It is not hard to verify that $(\texttt{SDP}_4)$ has strictly feasible points, so strong duality holds, and we get

$$z_{SDP} = \min_{\mu \geq 0, \lambda} f(\lambda, \mu) = f(\lambda^*, \mu^*) \leq f(\lambda, \mu), \ \forall \lambda, \mu \ \geq 0.$$

The function f is well-known to be convex but non-smooth. For a given feasible point (λ, μ) the evaluation of $f(\lambda, \mu)$ amounts to solving a problem over (5.10), for timings, see Table 5.2. The primal optimum Z of this semidefinite program also yields an element of the subdifferential of f. Our goal is to use this subgradient information to generate a feasible point (λ, μ) with value close to z_{SDP} with a limited number of function evaluations. To achieve this goal, we use the bundle method, see [154], which is taylored to minimize nonsmooth convex functions. It generates a sequence of iterates (λ_k, μ_k) which converge to an optimal solution. In each iteration, we evaluate f and compute an element of the subdifferential at the iterate. Since this method has a somewhat weak asymptotic convergence behaviour, we limit the number of function evaluations to control the overall computational effort. The computational results in the following sections are obtained with a limited number of such function evaluations, and leave some room for further incremental improvement.

Since inactive inequalities have no influence on f (the corresponding Lagrange multiplier is zero), we concentrate on identifying those inequalities, which are likely to be active at the optimum. Thus the actual set of constraints dualized will change during the iterations of the bundle method. We follow the approach discussed in [103] to add and drop constraints on the fly, and refer the reader interested in further details to this paper.

The high level picture is that we get an acceptable upper bound to z_{SDP} with a few hundred function evaluations of f.

5.7 Computational results for large-scale applications

In this section we give computational results for large-scale instances of LOP, MLCM and WB. We provide improved upper bounds for some hard LOP instances that substantially reduce the duality gap. Even more promising computational results can be given for MLCM and WB which are active areas of research with many interesting applications. MLCM is applied to the representation of the facets of polytopes [63] and inside the Sugiyama framework [290] that is used for schedules, UML diagrams, and flow charts. WB has applications in computational biology [68] and additionally the *single-row facility layout problem (SRFLP)* and the (weighted) *minimum linear arrangement problem (MinLA)* are special cases. Therefore in Subsections 5.7.2, 5.7.3 and 5.7.4 we just point out some selected improvements achieved and refer to separate papers [63, 169] for an in-depth (polyhedral) analysis of the specific problem structures and extensive computational comparisons with other methods to point out their potential and limitations.

5.7.1 Large LOP instances

From a purely theoretical point of view, it is clear that ($\mathtt{SDP}_4$) provides the strongest relaxation. The computational approach described in the previous section typically stops after a preset number of function evaluations and therefore provides only a suboptimal solution. In our preliminary experiments on larger instances, we noticed that it is important to first get 'nearly' feasible with respect to the 3-cycle equations defining ($\mathtt{SDP}_1$). Once this is achieved we start adding the triangle inequalities and the matrix cuts. Since their number is quite large, we analyzed experimentally their effect and noticed that the inclusion of only the most violated triangle inequalities resulted in the quickest improvement of the bound with only a limited number of function evaluations. We therefore concentrate only on getting good approximations to ($\mathtt{SDP}_2$).

In Table 5.3 we summarize upper bounds for some LOP instances for which the optimal solution is not yet known (for details see [220, Tables 10,12,14]). These LOP instances can be downloaded from `http://heur.uv.es/optsicom/LOLIB`. The table identifies the instance by its name and size n. We then provide the best known (integer) solution in the column labeled *bks*. The linear programming bound (5.7) is given in column z_{LP}, The semidefinite bound ($\mathtt{SDP}_2$) is (approximately) determined with 250 function evaluations. Finally, we also provide the relative gap between the best known feasible solution and the bounds in the columns *LP-gap* and *SDP-gap*. The gap (in percent) is computed as $gap = 100\frac{bound-bks}{bks}$.

In Table 5.4 we summarize upper bounds for large-scale instances for which again the optimal solution is not yet known. Here we used ($\mathtt{SDP}_1$) and allowed 25 function evaluations.

For all instances in Tables 5.3 and 5.4 we are able to reduce the gap between lower and upper bound substantially. This reduction of the gaps will also lead to considerably smaller branching trees in a Branch-and-Bound approach. To illustrate this let us mention that applying a Branch-and-Bound algorithm using the lp bounds to the paley graphs 31 and 43 results in bounds still beyond 300 respectively 600 after days of branching.

graph	n	bks	z_{LP}	LP-gap	$\mathtt{SDP}_2^{250}$	SDP-gap
pal31	31	285	310	8.77	297	4.21
pal43	43	543	602	10.87	569	4.79
pal55	55	1045	1084	3.73	1049	0.38
p50-05	50	42907	44196	3.00	43177	0.63
p50-06	50	42325	43765	3.40	42673	0.82
p50-07	50	42640	43977	3.14	42897	0.60
p50-08	50	42666	44655	4.66	43241	1.35
p50-09	50	43711	45183	3.37	43954	0.56
p50-10	50	43575	45346	4.06	44097	1.20
p50-11	50	43527	45132	3.69	43932	0.93
p50-12	50	42808	44671	4.35	43341	1.25
p50-13	50	43169	44872	3.94	43608	1.02
p50-14	50	44519	46272	3.94	44907	0.87
p50-15	50	44866	46479	3.60	45253	0.86

Table 5.3: Bounds for medium size LOP instances

graph	n	bks	z_{LP}	LP-gap	$\mathtt{SDP}_1^{25}$	SDP-gap
N-t1d100.01	100	106852	114468	7.13	110314	3.24
N-t1d100.02	100	105947	114077	7.67	110321	4.13
N-t1d100.03	100	109819	117843	7.31	113926	3.74

Table 5.4: Bounds for large LOP instances

5.7.2 Bipartite crossing minimization

Bipartite crossing minimization (BCM) is a special case of MLCM where the number of levels is set to two. An exact algorithm for this problem has been introduced by Jünger and Mutzel [173] which only performs well on small, sparse instances ($n \leq 12$) [48].

For our experiments, we used the random instances from [48]. These are generated with the Stanford GraphBase generator [184] which is hardware independent. Results are reported for graphs having $n = 14, 16, 18$ vertices on each layer. For each n, we consider graphs with densities $d = 10, 20, \ldots, 90$, (in percent), i.e. with $\left\lfloor \frac{dn^2}{100} \right\rfloor$ edges. For each pair (n, d), we report the average over 10 random instances.

For this type of problem we use the strongest relaxation $(\texttt{SDP}_4)$ as the number of violated matrix cuts (5.15) stays manageable for all instances considered. We also refer to Table 5.10 below.

Once we stop the bound computation, we use the primal solution $\tilde{Z} \in \mathcal{B}$ for hyperplane rounding, as suggested in [121]. Doing this we obtain a $\{-1, 1\}$ vector $\tilde{y}$, which need not be feasible with respect to the cycle equations in (5.6). An infeasible $\tilde{y}$ can be made feasible in an obvious way by flipping the signs of some of its entries, resulting in a feasible $\tilde{y}$. The most expensive step for this rounding process consists in finding the factorization $\tilde{D}\tilde{D}^T = \tilde{Z}$ of $\tilde{Z}$, to carry out the rounding procedure with $\tilde{D}$. Thus, once $\tilde{D}$ is available, we can easily afford to repeat the rounding procedure, and take the overall best solution, In our case we take the best solution out of a 1000 trials.

It turned out that this heuristic found an optimal solution for all BSM instances under consideration. The computation times were in the order of seconds, hence negligible.

Standard heuristics and also some metaheuristics perform quite poorly for MLCM instances of our size [173], [218], but it would be worth comparing metaheuristics like GRASP and tabu search [218] with our heuristic.

In Table 5.5 we compare our approach with the two best methods from [48]. These are the CPLEX MIP-solver, applied to the standard linearization of the objective function in combination with the standard integer programming formulation of the linear ordering problem and a Branch-and-Bound approach using SDP that is similar to our approach.

The results show that the SDP approach of [48] allows for substantial improvements, independent of the somewhat slower machine used in [48]. The main reasons for this improvement is on one hand our careful tuning of the bounding routine, and our rounding procedure, which allowed us to prove optimality at the root node, while [48] had to go through a few steps of branching before being able to prove optimality.

For a generalization of the SDP approach to multiple layers, detailed polyhedral studies of the crossing polytope and an extensive computational comparison with a state-of-the-art ILP approach, we refer to the recent paper by Chimani et al. [63].

5.7.3 Minimum linear arrangement problem

The *minimum linear arrangement problem (MinLA)* is a special case of WB, but can also be defined independently as follows. Given an undirected graph $G(V, E)$ find a permutation $\phi : V \to \{1, \ldots, n\}$ minimizing $\sum_{i,j \in E} |\phi(i) - \phi(j)|$.

$$\phi_1(G) := \min_{\phi \in \Pi} \sum_{i,j \in E} |\phi(i) - \phi(j)|.$$

To see that MinLA can be modeled as (quadratic) ordering problem we note that $\phi_1(G)$ can also be expressed as

$$\phi_1(G) = \min \sum_{(i,j) \in E} \sum_k x_{ik} x_{kj},$$

where x_{ij} satisfies (5.2)–(5.4). The term $\sum_k x_{ik} x_{kj}$ 'counts', how many nodes k lie between i and j in the ordering ϕ defined by x, as $x_{ik} x_{kj} = 1$ precisely if k lies between i and j.

n	d	CPLEX [48]	SDP [48]	(SDP$_4$)
		time	time	time
14	20	40.7	61.9	10.2
14	40	-	97.9	25.3
14	60	-	95.9	20.2
14	80	-	101.9	19.7
16	10	2.1	119.1	2.4
16	20	2701.6	200.9	28.8
16	30	-	432.9	53.5
16	40	-	1432.0	306.3
16	50	-	1181.2	110.2
16	60	-	1186.8	89.0
16	70	-	916.9	79.1
16	80	-	444.9	57.3
16	90	-	224.1	35.6
18	10	5.5	343.2	8.9
18	30	-	1233.7	170.4
18	50	-	-	211.4
18	70	-	2624.98	314.6
18	90	-	601.30	78.6

Table 5.5: Bipartite crossing minimization: comparison with [48]

For MinLA we work again with the strongest relaxation (SDP$_4$) as the number of violated matrix cuts (5.15) stays quite small for all instances considered. Again, we refer to Table 5.10 below.

To get a first idea of the tightness of the SDP approach we apply it to get bounds for $\phi_1(Q_n)$, where Q_n denotes the n-dimensional Cartesian cube. The value

$$\phi_1(Q_n) = 2^{n-1}(2^n - 1)$$

was determined by Harper [131]. For further comparison we also give the lower combinatorial bounds from [145] and [174] respectively. The results are summarized in Table 5.6. The relaxation (SDP$_4$) correctly identifies $\phi_1(Q_n)$ for all $n \leq 4$.

n	Reference [174]	Reference [145]	(SDP$_4$)	Optimum $\phi(Q_n)$
2	5	6	6	6
3	21	24	28	28
4	85	99	120	120
5	341	392	493	496
6	1365	1542	2002	2016

Table 5.6: Bounds for $\phi_1(Q_n)$ on the hypercube Q_n

For MinLA there exists another very recent algorithm of Caprara et al. [54], realized by Schwarz [275], that is preferable to the SDP approach for small graphs and large, sparse graphs.

In Table 5.7 we give some instances from the Boeing Sparse Matrix Collection [93], where we could prove optimality of the upper bounds UB for the first time together with [54] and also improve the best known lower bound for two instances. As we get our lower bounds from the root node relaxation, we are very optimistic to solve these instances when using our bounds in a Branch-and-Bound approach. Table 5.7 starts with the instance name, the number of nodes n, the density d of the instance and the upper bound (UB) obtained by a multi-start local search routine. All cited algorithms were run on a 2× Xeon CPU with 2.5 GHz. The LP-based approach from [55] also yields strong results for large, sparse graphs.

A missing entry indicates that the respective instance was not considered in [55].

Instance				Reference [275]		Reference [55]		$(\mathtt{SDP}_4)$	
name	n	d	UB	LB	Time	LB	Time	LB	Time
can_24	24	24.6	210	210	4.7	203	2.8	210	66.9
fidap005	27	35.8	414	414	4.1	412	4.2	414	124.4
pores_1	30	23.6	383	383	29.9			383	286.5
ibm32	32	18.1	485	485	1241.3			485	306.1
fidapm05	42	27.7	1003	1003	1516.9	998	805.2	1003	6200.1
bcsstk01	48	15.6	1132	1132	40852.8	972	3848.1	1130	10744.5
impcol_b	59	16.4	2076	2000	limit			2074	51082.6
dwt_59	59	6.0	289	289	39.4	258	55.4	289	37925.3
gd95c	62	7.6	506	506	109.7	443	68.3	506	36647.7
can_73	73	5.7	1100	962	limit	971	2016.8	1088	limit

Table 5.7: Comparison of several exact approaches to linear arrangement (time limit is 24h)

5.7.4 The single-row facility layout problem

An instance of the *single-row facility layout problem (SRFLP)* consists of n one-dimensional facilities, with given positive lengths $l_1, \ldots, l_n$, and pairwise weights c_{ij}. Now the task in SRFLP is to find a linear ordering of the departments such that the total weighted sum of the center-to-center distances between all pairs of facilities is minimized. SRFLP is again a special case of WB, where the weights of triples (i, j, k) are set to $w_{ijk} := c_{ik} l_j$ and $\sum_{i<j} \frac{c_{ij}(l_i+l_j)}{2}$ is added as a constant to the objective function.

In Table 5.8 we compare the strongest relaxation $(\mathtt{SDP}_4)$ with another SDP approach from [20]. In [20] they use $(\mathtt{SDP}_1)$ as basic relaxation and then add the 300 to 400 most-violated triangle inequalities (5.13) to the model in every iteration and re-optimize until no more triangle inequalities are violated. Using the bundle method instead of cutting planes and additionally adding the matrix cuts (5.15) yields significantly improved computation times especially for larger instances.

Additionally we compare with the recent Branch-and-Cut algorithm based on a polyhedral study of the distance polytope of SRFLP by Amaral and Letchford [13] and we also give the running times of the approach to SRFLP by Amaral [8] that uses betweenness variables in an LP-based cutting plane algorithm.

The computations in [20] were carried out on a 2.0GHz Dual Opteron, in [13] they used a 2.5 GHz Pentium Dual Core PC and in [8] an Intel Core Duo 1.73 GHz PC was employed. Let us point out that our machine is slower than all the other ones.[1]

The problems are collected from different sources and include well-known benchmark instances [150, 283], instances with clearance requirement [150] and random-generated instances [20].[2] For the upper bound computation we used the heuristic described in Subsection 5.7.2 where again the computation times were in the order of seconds for all instances. These problems are solved to optimality by all methods, so we do not include the optimal value in the table.

Amaral also introduced new instances with 33 and 35 departments, solved them to optimality and pointed out that he could not solve larger instances with his approach as the linear programs involved became too large and too difficult. We can also solve Amaral's new instances [8] to optimality with our approach and even achieve better computation times. For the detailed polyhedral and computational comparison of the above mentioned approaches we refer to [169]. There we also successfully apply $(\mathtt{SDP}_4)$ to the even larger SRFLP instances with up to 100 departments from [21, 22]. In Table 5.9 we state the optimal values and solution times for five new instances with 40 departments, a density of 50 % and random weights between 1 and 10.

[1] For exact numbers of the speed differences see `http://www.cpubenchmark.net/`.
[2] These instances can be downloaded from `http://flplib.uwaterloo.ca/`.

Instance	n	CPU time [20]	CPU time [13]	CPU time [8]	CPU time (SDP_4)
Lit-4	20	26:54	2:22	30	54
Lit-5	30	15:50:57	28:07:49	27:35	9:07
Lit-Cl-5	12	33	4	1	8
Lit-Cl-6	15	5:53	10	3	20
Lit-Cl-7	20	41:32	5:12	40	1:16
Lit-Cl-8	30	51:06:53	17:49:43	1:12:13	14:17
Nugent25-01	25	3:44:38	7:19:44	3:46	2:48
Nugent25-02	25	4:50:27	38:35	9:59	5:46
Nugent25-03	25	5:48:21	1:25:41	4:49	4:11
Nugent25-04	25	4:04:51	39:34	10:19	5:33
Nugent25-05	25	8:22:22	1:18:10	3:47	3:31
Nugent30-01	30	7:41:06	34:00:51	25:41	4:42
Nugent30-02	30	10:41:53	3:56:53	22:43	6:08
Nugent30-03	30	19:32:01	13:08:12	23:14	10:12
Nugent30-04	30	31:03:11	58:20	2:19:22	11:44
Nugent30-05	30	19:54:07	13:03:51	1:05:36	18:30

Table 5.8: Comparison of the most competitive approaches to SRFLP. Times are given in sec, in min:sec or in h:min:sec

Instance	n	Optimal cost	CPU time (SDP_4) (h:min:sec)
N40_1	40	107348.5	1:01:36
N40_2	40	97693	0:52:52
N40_3	40	78589.5	1:21:40
N40_4	40	76669	1:15:58
N40_5	40	103009	2:20:09

Table 5.9: Solution values and times for new large SRFLP instances with 40 departments

5.7.5 Model validation for QOPs

Finally we show that including the matrix cuts (5.15) in the SDP model yields essential improvements for all types of quadratic ordering instances.

First in Table 5.10 we compare the number of inequalities of the different constraint types that are considered by the bundle method using the strongest relaxation (SDP_4) at the last function evaluation. We examine two LOP instances from Table 5.3, a random BCM instance with 16 nodes on each layer and a density of 40 % from Table 5.5 and a MinLA instance from Table 5.6. For the LOP instances the number of matrix cuts (5.15) is very large and therefore constricts the overall performance of the algorithm. In contrast the number of matrix cuts stays quite small for the two quadratic ordering instances.

graph	problem type	p	# (5.12)	# (5.15)	# (5.13)
pal 31	LOP	466	5803	746559	14339
N-p50-05	LOP	1226	4874	399720	6800
bcm.16.40	BCM	241	1262	2524	64903
cube5	MinLA	497	5730	192	36768

Table 5.10: Number of constraints considered by the bundle method

Despite of their small number the matrix cuts help a lot to tighten the relaxation of quadratic ordering instances as can be seen from Table 5.11. In the last line of the table we give in brackets the number of function evaluations (*fe*) needed to prove optimality for the BCM instance.

graph	# fe	opt	$(\mathtt{SDP}_2)$	$(\mathtt{SDP}_4)$
cube5	300	496	490.28	491.47
	400		490.91	492.11
	500		491.21	492.32
	600		491.32	492.41
bcm.16.40	300	1397	1395.18	1395.42
	400		1395.62	1395.88
			1396.01 (595)	1396.01 (455)

Table 5.11: Gain of relaxation tightness through matrix cuts

5.8 Conclusions

In this paper, we have presented a systematic investigation and comparison of SDP based relaxations to ordering problems. We demonstrated that semidefinite relaxations provide substantially tighter bounds than linear programming relaxations for the linear ordering problem. As semidefinite relaxations are the natural setting for quadratic ordering problems, we also applied them to crossing minimization, single-row layout and linear arrangement problems. For all these different fields of application with their rich, mainly independent bodies of literature, we could produce new superior bounding results compared with the multiple, diverse approaches developed so far. This generality distinguishes the semidefinite approach. We only have to adapt the cost function to compute all kinds of quadratic ordering problems and also combinations of them. Another mentionable feature of our approach is that the computation times mainly depend on the number of nodes in the underlying graphs and are not so much influenced by their density structures, which can be a great advantage compared to the other methods for special types of quadratic ordering problems that all depend on exploiting sparsity.

Therefore it seems to be worthwhile to think about ways to further improve the presented approach. There are two (combinable) directions to enhance the presented SDP based relaxations of ordering problems. On the one hand, we could include further constraint classes to further tighten the relaxation and on the other hand, we could incorporate the SDP based bounds in a Branch-and-Bound framework.

Acknowledgement: We thank Gerhard Reinelt and Marcus Oswald from Universität Heidelberg for many fruitful discussions and practical hints regarding the polyhedral approach to LOP. We also acknowledge constructive comments by two anonymous referees that led to the current version of the paper.

Chapter 6

A Computational Study for the Single-Row Facility Layout Problem

Authors: Philipp Hungerländer and Franz Rendl

Abstract: The single-row facility layout problem (SRFLP) is an NP-hard combinatorial optimization problem that is concerned with the arrangement of n departments of given lengths on a line so as to minimize the weighted sum of the distances between department pairs. (SRFLP) is the one-dimensional version of the facility layout problem that seeks to arrange rectangular departments so as to minimize the overall interaction cost. This paper compares the different modelling approaches for (SRFLP) and applies a recent SDP approach for general quadratic ordering problems from Hungerländer and Rendl to (SRFLP). In particular, we report optimal solutions for several (SRFLP) instances from the literature with up to 42 departments that remained unsolved so far. Secondly we significantly reduce the best known gaps and running times for large instances with up to 110 departments.

Keywords: Single-Row Facility Layout; Space Allocation; Semidefinite Optimization; Combinatorial Optimization

6.1 Introduction

An instance of the single-row facility layout problem (SRFLP) consists of n one-dimensional departments, with given positive lengths $l_1, \dots, l_n$, and pairwise connectivities c_{ij}. Now the task in (SRFLP) is to find a permutation π of the departments such that the total weighted sum of the center-to-center distances between all pairs of departments is minimized

$$\min_{\pi \in \Pi} \sum_{i,j \in \mathcal{N}, i<j} c_{ij} z_{ij}^{\pi}, \tag{6.1}$$

where $\mathcal{N} := \{1, \dots, n\}$, Π denotes the set of all layouts and z_{ij}^{π} is the center-to-center distance between departments i and j with respect to π.

Several practical applications of (SRFLP) have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291], the arrangement of machines in flexible manufacturing systems [150], the arrangement of books on a shelf and the assignment of disk cylinders to files [252].

On the one hand (SRFLP) (also known as one-dimensonal space allocation problem) is a special case of the weighted betweenness problem which is again a special case of the quadratic ordering problem. On the other hand the NP-hard [112] minimum linear arrangement problem is a special case of (SRFLP) where all departments have the same length and the connectivities are equal to 0 or 1. Hence (SRFLP) is also

NP-hard.

Accordingly several heuristic algorithms have been suggested to tackle instances of interesting size of (SRFLP), e.g. [80, 123, 128, 149, 151, 193, 266, 267]. However, these heuristic approaches do not provide any optimality certificate, like an estimate of the distance from optimality, for the solution found.

Several exact approaches to (SRFLP) have also been proposed. Simmons [283] first studied (SRFLP) and suggested a branch-and-bound algorithm. Later on Simmons [284] pointed out the possibility of extending the dynamic programming algorithm of Karp and Held [181] to (SRFLP). This was later on implemented by Picard and Queyranne [252]. A nonlinear model was presented by Heragu and Kusiak [151]. Linear mixed integer programs using distance variables were proposed by Love and Wong [215] and Amaral [6]. Amaral [7] achieved a more efficient linear mixed integer program by linearizing a quadratic model based on ordering variables. However all these models suffer from weak lower bounds and hence have high computation times and memory requirements. But just recently Amaral and Letchford [13] achieved significant progress in that direction through the first polyhedral study of the distance polytope for (SRFLP) and showed that their approach is quite effective for instances with challenging size ($n \geq 30$). Amaral [8] suggested an LP-based cutting plane algorithm using betweenness variables that proved to be highly competitive and solved instances with up to 35 departments to optimality. Recently Sanjeevi and Kianfar [268] studied the polyhedral structure of Amaral's betweenness model in more detail and identified several classes of facet defining inequalities.

To obtain tight lower bounds for (SRFLP) without using branch-and-bound, semidefinite programming (SDP) approaches are the best known methods to date. SDP is the extension of linear programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} &\inf_X \{ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \\ &\mathcal{P} := \{ X \mid \langle A_i, X \rangle = b_i, \ i \in \{1, \dots, m\}, \ X \succcurlyeq 0 \}, \end{aligned} \tag{SDP}$$

where the data matrices A_i, $i \in \{1, \dots, m\}$ and C are symmetric. We refer the reader to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

Anjos et al. [22] proposed the first SDP relaxation for (SRFLP) yielding bounds for instances with up to 80 departments. Anjos and Vanelli [20] further tightened the SDP relaxation using triangle inequalities as cutting planes and gave optimal solutions for instances with up to 30 departments that remained unsolved since 1988. Anjos and Yen [21] suggested an alternative SDP relaxation and achieved optimality gaps no greater than 5 % for large instances with up to 100 departments. Recently Hungerländer and Rendl [156, 168] proposed a general approach for quadratic ordering problems, where they further improved on the tightness of the above SDP relaxations. They used a suitable combination of optimization methods to deal with the stronger but more expensive relaxations and applied their method among others to some selected medium (SRFLP) instances. Thereby they solved instances with up to 40 departments to optimality.

The main contributions of this paper are the following: First we describe and compare the most successful modelling approaches to (SRFLP), pointing out their common connections to the maximum cut [32, 130, 285] and the quadratic ordering problem [47, 48]. For further details on this subject see also the recent survey of (SRFLP) by Anjos and Liers [19].

Secondly we apply the approach from [168] for the first time to a broad selection of small, medium and large instances and compare it computationally to the leading algorithms for the different instance sizes. Thereby we demonstrate that this approach clearly dominates all other methods, permitting significant progress for medium as well as large instances. We can give optimal solutions for several medium instances from the literature with up to 42 departments that remained unsolved so far and reduce all the best known gaps for large scale instances by a factor varying from 2 to 110.

Finally we relate the two SDP heuristics from [22] and [168] concerning their computational costs and practical performance.

The paper is structured as follows. In Section 6.2, we put the most competitive algorithms for (SRFLP) into perspective and compare them from a theoretical point of view. In Section 6.3, we conduct an extensive computational study for the SDP approach of Hungerländer and Rendl [168], achieving significant progress for medium and large instances. Finally some conclusions and current research are summarized in Section 6.4.

6.2 The Most Successful Modelling Approaches to (SRFLP)

The most intuitive modelling approach to (SRFLP) using $\binom{n}{2}$ distance variables $z^{\pi}_{ij}, i, j \in \mathcal{N}$ suffers from weak lower bounds of the corresponding LP relaxation and thus large branch-and-bound trees, high computation times and memory requirements. Recently Amaral and Letchford [13] achieved significant progress in that direction by identifying several classes of valid inequalities and using them as cutting planes. Amaral [8] improved the LP relaxation by modelling (SRFLP) via $\binom{n}{3}$ binary betweenness variables. Anjos et. al [22] proposed to model (SRFLP) as a binary quadratic program using $\binom{n}{2}$ ordering variables. They deduced a semidefinite relaxation yielding tighter bounds but being more expensive to compute than the relaxation of Amaral [8]. Later on further SDP approaches have been suggested to improve on the relaxation strength and/or reduce the computational effort involved [20, 21, 168]. In the following subsections we recall the approaches mentioned above and highlight their relations.

6.2.1 Distance-Based LP Formulation of Amaral and Letchford [13]

The polytope containing the feasible distance variables z_{ij} for n departments with lengths $l \in \mathcal{Z}^n$ is called distance polytope and defined as

$$\mathcal{P}^n_{Dis} := \operatorname{conv}\left\{ z \in \mathbb{R}^{\binom{n}{2}} : \exists \pi \in \Pi \ : \ z_{ij} = z^{\pi}_{ij}, \ i, j \in \mathcal{N}, i < j \right\}.$$

Amaral and Letchford [13] show that the equation

$$\sum_{i,j \in \mathcal{N}, i<j} l_i l_j z_{ij} = \frac{1}{6}\left[\left(\sum_{i \in \mathcal{N}} l_i \right)^3 - \sum_{i \in \mathcal{N}} l_i^3 \right],$$

defines the smallest linear subspace that contains $\mathcal{P}^n_{Dis}$. They prove that clique inequalities, strengthened pure negative type inequalities and special types of hypermetric inequalities induce facets of $\mathcal{P}^n_{Dis}$. They further show the validity of rounded psd inequalities and star inequalities for $\mathcal{P}^n_{Dis}$ and use them together with the facet inducing inequalities as cutting planes in a Branch-and-Cut approach.

6.2.2 Betweenness-Based LP Formulation of Amaral [8]

Amaral [8] introduced binary variables $\zeta_{ijk} (i, j, k \in \mathcal{N}, i < j, i \neq k \neq j)$

$$\zeta_{ijk} = \begin{cases} 1, & \text{if department } k \text{ lies between departments } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

Amaral [8] collected these betweenness variables in a vector ζ and defined the betweenness polytope

$$\mathcal{P}^n_{Btw} := \operatorname{conv}\left\{ \zeta \ : \ \zeta \text{ represents an ordering of the elements of } \mathcal{N} \right\}.$$

In order to formulate `(SRFLP)` via ζ an appropriate objective function is needed. For that purpose Amaral [8] used the relation

$$z_{ij}^{\pi} = \frac{1}{2}(l_i + l_j) + \sum_{\substack{k \in \mathcal{N}, \\ i \neq k \neq j}} l_k \zeta_{ijk}, \quad i,j \in \mathcal{N},\ i < j,$$

to rewrite (6.1) in terms of ζ (for details see [8, Proposition 1 and 2])

$$\min_{\zeta \in \mathcal{P}_{Btw}^n} \sum_{\substack{i,j,k \in \mathcal{N}, \\ i<j,k<j}} (c_{ij} l_k - c_{ik} l_j)\, \zeta_{ijk} + \sum_{\substack{i,j \in \mathcal{N}, \\ i<j}} \left(\frac{c_{ij}}{2}(l_i + l_j) + \sum_{\substack{k \in \mathcal{N}, \\ k>j}} c_{ij} l_k \right). \tag{6.2}$$

If department i comes before department j, department k has to be located mutually exclusive either left of department i, or between departments i and j, or right of department j. Thus the following equations are valid for $\mathcal{P}_{Btw}^n$

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i,j,k \in \mathcal{N}, i < j < k. \tag{6.3}$$

In [268] it is shown that these equations describe the smallest linear subspace that contains $\mathcal{P}_{Btw}^n$. To obtain an LP relaxation of `(SRFLP)`, the integrality conditions on ζ are replaced with 0-1 bounds:

$$0 \leq \zeta_{ijk} \leq 1, \quad i,j,k \in \mathcal{N}, i < j. \tag{6.4}$$

To further strengthen the relaxation, Amaral [8] came up with additional valid inequalities. Let a subset $\{i,j,k,d\} \subset \mathcal{N}$ be given. On the one hand department d can not be located between the departments i and j, i and k and j and k at the same time. On the other hand if department d is between departments i and k then it also lies between departments i and j or j and k. Thus the inequalities

$$\zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \leq 2, \quad i,j,k,d \in \mathcal{N}, i < j < k \tag{6.5}$$

and

$$-\zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \geq 0, \quad \zeta_{ijd} - \zeta_{jkd} + \zeta_{ikd} \geq 0, \quad \zeta_{ijd} + \zeta_{jkd} - \zeta_{ikd} \geq 0, \quad i,j,k,d \in \mathcal{N}, i < j < k, \tag{6.6}$$

are valid for $\mathcal{P}_{Btw}^n$. Sanjeevi and Kianfar [268] showed that (6.6) unlike (6.5) are facet defining for $\mathcal{P}_{Btw}^n$.

Amaral [8] further generalizes (6.6) to a more complicated set of inequalities: Let $\beta \leq n$ be an even integer and let $S \subseteq \mathcal{N}$. For each $d \in S$, and for any partition (S_1, S_2) of $S \setminus \{d\}$ such that $|S_1| = \frac{1}{2}\beta$, the inequality

$$\sum_{p,q \in S_1, p<q} \zeta_{pqd} + \sum_{p,q \in S_2, p<q} \zeta_{pqd} \leq \sum_{p \in S_1, q \in S_2, p<q} \zeta_{pqd} \tag{6.7}$$

is valid [8] and also facet-defining [268] for $\mathcal{P}_{Btw}^n$. Note that (6.6) is a special case of (6.7) with $\beta = 4$.

Minimizing (6.2) over (6.3)–(6.6) gives the basic linear relaxation `(LP)`. To construct stronger relaxations from `(LP)` Amaral [8] proposes to use the inequalities $(6.7)_{\beta=6}$ as cutting planes (for details see Subsection 6.3.1 below).

6.2.3 Matrix-Based Relaxations of Anjos et al. [20, 21, 22]

Another way to get good lower bounds for (SRFLP) is the usage of matrix-based relaxations. They can be deduced from the betweenness-based approach above by introducing bivalent ordering variables $y_{ij}(i,j \in \mathcal{N}, i<j)$

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise,} \end{cases} \tag{6.8}$$

and using them to express the betweenness variables ζ via the transformations

$$\zeta_{ijk} = \frac{1+y_{ik}y_{kj}}{2},\ i<k<j, \qquad \zeta_{ijk} = \frac{1-y_{ki}y_{kj}}{2},\ k<i<j, \qquad \zeta_{ijk} = \frac{1-y_{ik}y_{jk}}{2},\ i<j<k, \tag{6.9}$$

for $i,j,k \in \mathcal{N}$. Using (6.9) we can easily rewrite the objective function (6.2) and equalities (6.3) in terms of ordering variables

$$K - \sum_{\substack{i,j\in\mathcal{N}\\ i<j}} \frac{c_{ij}}{2} \left(\sum_{\substack{k\in\mathcal{N}\\ k<i}} l_k y_{ki} y_{kj} - \sum_{\substack{k\in\mathcal{N}\\ i<k<j}} l_k y_{ik} y_{kj} + \sum_{\substack{k\in\mathcal{N}\\ k>j}} l_k y_{ik} y_{jk} \right), \tag{6.10}$$

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k \in \mathcal{N}, i<j<k, \tag{6.11}$$

where $K := \left(\sum_{\substack{i,j\in\mathcal{N}\\ i<j}} \frac{c_{ij}}{2} \right) \left(\sum_{k\in\mathcal{N}} l_k \right)$. In [48] it is shown that the equations (6.11) formulated in a $\{0,1\}$ model describe the smallest linear subspace that contains the quadratic ordering polytope

$$\mathcal{P}^n_{QO} := \text{conv}\ \{\ yy^\top\ :\ y \in \{-1,1\},\ |y_{ij}+y_{jk}-y_{ik}| = 1\ \}.$$

To obtain matrix-based relaxations we collect the ordering variables in a vector y and consider the matrix $Y = yy^\top$. The main diagonal entries of Y correspond to y_{ij}^2 and hence $\text{diag}(Y) = e$, the vector of all ones. Now we can formulate (SRFLP) as the following optimization problem, first proposed in [22]

$$\min\ \{\ \langle C, Y\rangle + K :\ Y \text{ satisfies (6.11)},\ \ \text{diag}(Y) = e,\ \text{rank}(Y) = 1,\ Y \succcurlyeq 0\ \}, \tag{SRFLP}$$

where the cost matrix C is deduced from (6.10). Dropping the rank constraint yields the basic semidefinite relaxation of (SRFLP)

$$\min\ \{\ \langle C, Y\rangle + K :\ Y \text{ satisfies (6.11)},\ \ \text{diag}(Y) = e,\ Y \succcurlyeq 0\ \}, \tag{SDP$_1$}$$

providing a lower bound on the optimal value of (SRFLP). To be able to tackle larger instances Anjos and Yen [21] proposed to sum up the $O(n^3)$ constraints (6.11) over k yielding the $O(n^2)$ constraints

$$\sum_{\substack{k\in\mathcal{N}\\ i\neq k\neq j}} y_{ij}y_{jk} - \sum_{\substack{k\in\mathcal{N}\\ i\neq k\neq j}} y_{ij}y_{ik} - \sum_{\substack{k\in\mathcal{N}\\ i\neq k\neq j}} y_{ik}y_{jk} = -(n-2), \quad i,j \in \mathcal{N}, i<j. \tag{6.12}$$

They showed that the following optimization problem using (6.12) instead of (6.11)

$$\min\ \{\ \langle C, Y\rangle + K :\ Y \text{ satisfies (6.12)},\ \ \text{diag}(Y) = e,\ \text{rank}(Y) = 1,\ Y \succcurlyeq 0\ \},$$

is again an exact formulation of (SRFLP). Dropping the rank-one constraint yields a weaker but also cheaper semidefinite relaxation than ($\texttt{SDP}_1$)

$$\min \{ \langle C, Y \rangle + K : \ Y \text{ satisfies (6.12)}, \ \ \operatorname{diag}(Y) = e, \ Y \succcurlyeq 0 \}. \qquad (\texttt{SDP}_0)$$

As Y is actually a matrix with $\{-1, 1\}$ entries in the original (SRFLP) formulation, Anjos and Vanelli [20] proposed to further tighten ($\texttt{SDP}_1$) by adding the triangle inequalities, defining the metric polytope $\mathcal{M}$ and known to be facet-defining for the cut polytope, see e.g. [88]

$$\mathcal{M} = \left\{ Y : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{i,j} \\ Y_{j,k} \\ Y_{i,k} \end{pmatrix} \leq e, \ \ 1 \leq i < j < k \leq \binom{n}{2} \right\}. \qquad (6.13)$$

Using the linear transformations (6.9) it is straightforward to show the equivalence of a subset of the triangle inequalities with the betweenness constraints (6.5) and (6.6) from above. Along the same lines inequalities (6.7) can be connected to general clique inequalities. Adding the triangle inequalities to ($\texttt{SDP}_1$), Anjos and Vanelli [20] achieved the following relaxation of (SRFLP)

$$\min \{ \langle C, Y \rangle + K : \ Y \text{ satisfies (6.11)}, \ Y \in \mathcal{M}, \ \operatorname{diag}(Y) = e, \ Y \succcurlyeq 0 \}. \qquad (\texttt{SDP}_2)$$

As solving ($\texttt{SDP}_2$) directly with an interior-point solver like CSDP gets far too expensive, they suggest to use the $\approx \frac{1}{12} n^6$ triangle inequalities as cutting planes in their algorithmic framework (for details see Subsection 6.3.1 below). Let us also mention that so far all SDP approaches to (SRFLP) refrained from using other clique inequalities to further tighten the SDP relaxations because of their large number. We will argue in the conclusions that using well-designed subsets of larger clique inequalities, like e.g. pentagonal inequalities, which can be connected to the betweenness constraints $(6.7)_{\beta=6}$, could be a promising direction to improve current SDP approaches.

6.2.4 Strengthened Matrix-Based Relaxation

Recently Hungerländer and Rendl [168] suggested a further strengthening of ($\texttt{SDP}_2$) and an alternative algorithmic approach to solve such large SDP relaxations. To this end we introduce the matrix

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \qquad (6.14)$$

and relax the equation $Y - yy^\top = 0$ to

$$Y - yy^T \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0,$$

which is convex due to the Schur-complement lemma. Note that $Z \succcurlyeq 0$ is in general a stronger constraint than $Y \succcurlyeq 0$. Additionally we use an approach suggested by Lovász and Schrijver in [214] to further improve on the strength of the relaxation. This yields the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \ \forall \, i,j,k,l,m \in \mathcal{N}, i < j < k, \ l < m, \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \ \forall \, i,j,k,l,m \in \mathcal{N}, i < j < k, \ l < m, \end{aligned} \qquad (6.15)$$

that are generated by multiplying the 3-cycle inequalities valid for the ordering problem

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0,$$

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. These constraints define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{ \ Z : Z \text{ satisfies (6.15)} \ \}, \tag{6.16}$$

consisting of $\approx \frac{1}{3}n^5$ inequalities. In summary, we come up with the following relaxation of (SRFLP)

$$\min \{ \ \langle C, Y \rangle + K : \ Y \text{ satisfies (6.11)}, \ Z \in (\mathcal{M} \cap \mathcal{LS}), \ \operatorname{diag}(Z) = e, \ Z \succcurlyeq 0 \ \}. \tag{SDP$_3$}$$

A similar relaxation (without the LS-cuts (6.15)) was used in [48] for bipartite crossing minimization. In [168] (SDP$_3$) is applied to different special cases of the quadratic ordering problem like the linear ordering problem, the linear arrangement problem, multi-level crossing minimization and of course (SRFLP). It is also demonstrated there that adding the LS-cuts to the relaxation pays off in practice.

To make (SDP$_3$) computationally tractable Hungerländer and Rendl [168] suggest to deal with the triangle inequalities (6.13) and LS-cuts (6.15) through Lagrangian duality (for details see Subsection 6.3.1 below and [168, Section 6]).

6.3 Computational Comparison

In this section we give a computational comparison of all state-of-the-art approaches to (SRFLP) on a broad selection of small, medium and large instances from the literature. Using the approach from [168] we solve several instances to optimality for the first time and improve on the gaps of all currently unsolved instances.

6.3.1 Comparison of Globally Optimal Methods for Small and Medium Instances

In Table 6.1 we computationally compare the four most competitive approaches to (SRFLP) for small and medium instances. These are the integer linear programming (ILP) approaches of Amaral and Letchford [13] and Amaral [8], the SDP approach of Anjos and Vanelli [20] building on relaxation (SDP$_2$) and the SDP approach from [168] building on relaxation (SDP$_3$).

Anjos and Vanelli [20] start with the basic relaxation (SDP$_1$) and then enhance it with violated triangle inequalities (6.13) in every iteration (using the interior-point solver CSDP version 5.0) until no more triangle inequalities are violated.

Amaral and Letchford [13] suggest an ILP Branch-and-Cut algorithm based on the distance variables z_{ij}. They use a cheap initial LP relaxation with only $O(n^2)$ non-zero coefficients and apply exact separation routines for triangle and special strengthened pure negative type inequalities and heuristic ones for clique, rounded psd and star inequalities. They suggest a specialised branching rule to avoid the use of additional binary variables and use a primal heuristic based on multi-dimensional scaling to obtain feasible layouts.

Amaral [8] proposes an ILP cutting plane algorithm based on the betweenness variables ζ_{ijk} that improves on the results in [20] and [13]. For computational usage of the betweenness model Amaral [8] suggests to alternate between solving (LP) and strengthening (LP) (by searching for cutting planes $(6.7)_{\beta=6}$ violated at the optimal solution of the current (LP) and adding them to (LP)). Amaral [8] also introduces new instances with 33 and 35 departments, solves them to optimality and points out that he cannot solve larger instances with his approach as the involved linear programs become too large and too difficult to solve with the currently available LP solvers.

Recently Hungerländer and Rendl [168] proposed an algorithm to provide lower bounds to (SDP$_3$). Their method is building on subgradient optimization techniques, such as the bundle method [103, 154] and deals with the inequality constraints (6.13) and (6.15) through Lagrangian duality. This results in an iterative process where each iteration amounts to solving a semidefinite program of the form

$$\min\{\langle \tilde{C}, Y\rangle \ : \ \operatorname{diag}(Z) = e, \ Y \succcurlyeq 0\},$$

where $\tilde{C}$ changes in each iteration due to the update of the Langrange multipliers. The order of the SDP is $\binom{n}{2} + 1$. Once this matrix order goes beyond 5000, i.e. if $n \approx 100$, the computation times become prohibitive, due to matrix operations (multiplications, Cholesky decomposition) with generically dense matrices. Due to the iterative nature of algorithm the SDP lower bound improves quickly in the beginning and its progress slows down relatively smoothly. A similar algorithmic approach was successfully applied to the maximum cut problem [262]. In [168] Hungerländer and Rendl already demonstrated that their algorithm clearly outperforms the SDP approach suggested in [20] on some selected (SRFLP) instances.

In Table 6.1 we give a full computational comparison of the four most successful exact approaches to (SRFLP) on all available instances from the literature, including well-known benchmark instances [6, 7, 8, 151, 283], instances with clearance requirement [150] and random-generated instances [20]. [1] The table identifies the instance by its name, source and size n and gives the times required by the four approaches to find a layout and prove its optimality.

The computations in [20] were carried out on a 2.0GHz Dual Opteron with 16 GB RAM, Amaral [8] used an Intel Core Duo, 1.73 GHz PC with 1 GB RAM, in [13] a 2.5 GHz Pentium Dual Core PC with 2 GB RAM was employed, whereas for applying the approach from [168] we use an Intel Xeon 5160 processor with 3 GHz and 2 GB RAM.

For small instances with up to 20 departments the ILPs are preferable to the SDP approaches whereas the SDP approach from [168] outperforms the other approaches on the larger instances. The difference between the approaches strongly grows with the problem size. Note that we do not take into account the speed of the machines, as it does not differ too much and thus does not affect the conclusions drawn above. Our machine is the quickest and about 2.5 times faster than the one in [8], which is the slowest. [2]

This motivates us to tackle larger instances with the approach from [168]. We summarize the results for the five instances with 40 departments, a density of 50 % and random lengths and connectivities between 1 and 10 in Table 6.2.[3]

We succeed in providing optimal solutions within reasonable time for all these instances that can hardly be solved to optimality with one of the other three approaches.

6.3.2 Comparison of Gaps Achieved by SDP-Based Approaches on Large Instances

In this subsection we compare the most competitive approaches to (SRFLP) for obtaining tight bounds of large instances. These are the algorithms of Anjos and Yen [21] building on relaxations (SDP_0) and (SDP_1) respectively and again the approach of Hungerländer and Rendl [168] building on relaxation (SDP_3). For solving relaxations (SDP_0) and (SDP_1), Anjos and Yen [21] use the interior-point solver CSDP (version 5.0). In Tables 6.3 and 6.4 we compare the three SDP approaches on instances with $36 - 100$ departments taken from [22] and [21].[4]

In [168] the constraints $Z \succcurlyeq 0$ and $\operatorname{diag}(Z) = e$ are maintained explicitly. The evaluations of an appropriate function over this set constitute the computational bottleneck and are responsible for more than 99% of the overall running time for large instances. To control the computational effort we restrict the number of function evaluations to 500 for instances with up to 64 departments and to 250 for larger instances. This limitation of the number of function evaluations leaves some room for further incremental

[1] Most of the instances can be downloaded from http://flplib.uwaterloo.ca/.
[2] For exact numbers of the speed differences see http://www.cpubenchmark.net/.
[3] These instances and the corresponding optimal orderings are available from http://flplib.uwaterloo.ca/.
[4] Most of the instances can be downloaded from http://flplib.uwaterloo.ca/. Our improved gaps and the corresponding orderings are also available there.

Instance	Source	n	Anjos/Vanelli [20]	Amaral/Letchford [13]	Amaral [8]	Hungerländer/Rendl [168]
S5	[283]	5		0.1	0.1	0.1
S8	[283]	8		0.5	0.1	0.6
S8H	[283]	8	0.2	0.1	0.1	2.3
S9	[283]	9		0.1	0.1	0.7
S9H	[283]	9		2.4	0.1	9.2
S10	[283]	10	3.4	0.4	0.2	0.6
S11	[283]	11	32.6	0.7	0.3	1.3
P15	[6]	15			2.8	19.7
P17	[7]	17			8.4	34.9
P18	[7]	18			13.3	32.5
H_20	[151]	20	26:54	2:22	30.8	54.3
H_30	[151]	30	15:50:57	28:07:49	27:35	9:07
Cl_5	[151]	5	0.1	0.1	0.2	0.1
Cl_6	[151]	6	0.4	0.1	0.1	0.1
Cl_7	[151]	7	1.2	0.3	0.1	0.6
Cl_8	[151]	8	1.8	0.1	0.1	0.4
Cl_12	[151]	12	32.8	4.0	0.6	7.9
Cl_15	[151]	15	5:53	9.6	3.2	19.6
Cl_20	[151]	20	41:32	5:12	40.1	1:16
Cl_30	[151]	30	51:06:53	17:49:43	1:12:19	14:17
N25_01	[20]	25	3:44:38	7:19:44	3:46	2:48
N25_02	[20]	25	4:50:27	38:35	9:59	5:46
N25_03	[20]	25	5:48:21	1:25:41	4:49	4:11
N25_04	[20]	25	4:04:51	39:34	10:19	5:33
N25_05	[20]	25	8:22:22	1:18:10	3:47	3:31
N30_01	[20]	30	7:41:06	34:00:51	25:41	4:42
N30_02	[20]	30	10:41:53	3:56:53	22:43	6:08
N30_03	[20]	30	19:32:01	13:08:12	23:14	10:12
N30_04	[20]	30	31:03:11	58:20	2:19:22	11:44
N30_05	[20]	30	19:54:07	13:03:51	1:05:36	18:30
Am33_01	[8]	33			1:15:57	19:28
Am33_02	[8]	33			2:35:22	48:07
Am33_03	[8]	33			2:22:32	36:33
Am35_01	[8]	35			1:35:04	17:30
Am35_02	[8]	35			5:27:34	41:01
Am35_03	[8]	35			2:17:52	53:14

Table 6.1: Results for (SRFLP) instances with up to 35 departments. The running times are given in sec, in min:sec or in h:min:sec respectively.

Instance	n	Optimal cost	Time SDP Hungerländer/Rendl [168]
N40_1	40	107348.5	1:01:36
N40_2	40	97693	52:52
N40_3	40	78589.5	1:21:40
N40_4	40	76669	1:15:58
N40_5	40	103009	2:20:09

Table 6.2: Results for 5 new (SRFLP) instances with 40 departments. The running times are given in min:sec or in h:min:sec.

improvement.

The SDP relaxations ($\texttt{SDP}_0$), ($\texttt{SDP}_1$), ($\texttt{SDP}_2$) and ($\texttt{SDP}_3$) are closely related to the standard SDP relaxation for the max-cut problem used in the seminal paper of Goemans and Williamson [121] to obtain high quality feasible solutions providing upper bounds. However the hyperplane rounding idea suggested in [121] cannot be applied directly to (SRFLP) to get a good layout because it yields a $\{-1, 1\}$ vector $\tilde{y}$, which need not be feasible with respect to the three cycle equations (6.11). That is why Anjos et al. [22] propose a different procedure to obtain a good feasible layout from the optimal solution of the SDP relaxation whereas Hungerländer and Rendl [168] suggest to apply a repair strategy to the infeasible $\tilde{y}$.

Anjos et al. [22] propose to use the entries $y^*_{ij,kl}$ of the optimal matrix Y^* of the SDP relaxation in the following way to obtain a good feasible layout: Fix a row ij and compute the values

$$\omega_k^{ij} = \frac{1}{2}\left(n + 1 + \sum_{l \in \mathcal{N}, k \neq l} y^*_{ij,kl}\right), \quad k \in \mathcal{N}.$$

These values are motivated by the fact that if Y^* is rank-one, then the values $\omega_k^{ij}, k \in \mathcal{N}$ are all distinct and belong to $\mathcal{N}$ and thus give a permutation of $\mathcal{N}$. In general, $\text{rank}(Y^*) > 1$ and thus a permutation can be obtained by sorting $w_k^{ij}, k \in \mathcal{N}$ in either decreasing or increasing order (since the objective value is the same). The output of the SDP-based heuristic is the best layout found by considering every row ij of Y^* with $i, j \in \mathcal{N}, i < j$.

In [168] it is suggested to take the $\{-1, 1\}$ vector $\tilde{y}$ obtained from hyperplane rounding and make it feasible with respect to the 3-cycle inequalities by flipping the signs of some of its entries appropriately. Computational experiments demonstrated that the repair strategy is not as critical as one might assume [63, 168]. For example we know from multi-level crossing minimization that the heuristic clearly dominates traditional heuristic approaches.

The heuristic of Anjos et. al [22] is much cheaper than the one in [168] as they have to factorize Y^* to carry out the rounding procedure. Nonetheless the computation times of both heuristics are negligible compared to the computational effort for the lower bound computation. We compared both heuristics concerning the quality of the produced layouts on many test instances and found out that the heuristic from [168] is clearly superior. This is also supported by a comparison of the upper bounds achieved by both approaches in Tables 6.3 and 6.4, where the heuristic from [168] improves on the one of Anjos et. al [22] on all instances considered.

When comparing the running times of the three approaches we do not take into account that Anjos and Yen [21] use a machine (2.4GHz Quad Opteron with 16 GB of RAM) that is more than 1.5 times faster and has 8 times the memory of our machine. [5]

In Table 6.3 we compare the three approaches for problems with 36 to 56 departments for which no optimal solution was known before. The table identifies the instance by its name and size n. We then provide the lower bound "lb" and the best layout found "blf" as well as the associated running times for the different approaches. Finally we give the running times that the approach of Hungerländer and Rendl [168] needs to improve on the gaps of the two other approaches "improve gap ($\texttt{SDP}_0$)" and "improve gap ($\texttt{SDP}_1$)".

[5] For details see http://www.cpubenchmark.net/.

Instance	n	SDP Anjos/Yen using (SDP$_0$) [21]			SDP Anjos/Yen using (SDP$_1$) [21]			SDP Hungerländer/Rendl [168] - restricted to 500 function eval.					
		lb	blf	time	lb	blf	time	lb	blf	time	gap in %	improve gap (SDP$_0$)	improve gap (SDP$_1$)
ste36-1	36	9851	10328	7:15	10087.5	10301	14:57	10287	10287	14:50	0	2:23	2:55
ste36-2	36	170759.5	182649	7:12	175387	181910	14:17	181508	181508	25:25	0	2:05	2:39
ste36-3	36	96090	104041.5	7:13	98739	102179.5	13:42	101643.5	101643.5	24:01	0	2:35	3:09
ste36-4	36	91103	96854.5	7:16	94650.5	96080.5	14:23	95805.5	95805.5	16:15	0	2:01	3:49
ste36-5	36	87688	92563.5	7:19	89533	91893.5	14:25	91651.5	91651.5	17:58	0	2:00	3:09
sko42-1	42	24517	25779	20:07	24807	25724	45:21	25521	25525	2:23:09	0.02	5:20	7:34
sko42-2	42	207357	218117.5	20:21	210785	217296.5	45:14	216099.5	216120.5	2:43:34	0.01	5:57	9:38
sko42-3	42	167783.5	174694.5	20:10	169944.5	173854.5	47:32	173245.5	173267.5	2:47:18	0.01	8:01	17:51
sko42-4	42	131536	139630	19:21	133429.5	138829	48:18	137379	137615	2:53:05	0.17	4:55	7:24
sko42-5	42	238669.5	250501.5	20:18	242925.5	249327.5	45:41	248238.5	248238.5	1:08:42	0	6:37	11:13
sko49-1	49	39333.5	41379	59:55	39794.5	41308	2:48:57	40895	41012	4:36:21	0.29	33:33	57:43
sko49-2	49	403024.5	418370	1:03:30	407741.5	418288	2:50:32	416142	416178	8:27:34	0.01	19:11	31:43
sko49-3	49	313923.5	326004	1:02:13	317628.0	325747	2:51:45	324464	324512	8:03:03	0.02	19:15	26:20
sko49-4	49	229809.5	238380.5	1:05:34	232368	237894.5	2:50:40	236718.5	236755	9:15:14	0.02	21:59	32:01
sko49-5	49	645406.5	673303	1:04:13	652638	671508	2:51:45	666130	666143	9:30:22	0.002	35:04	35:04
sko56-1	56	61789.5	64454	3:05:19	62496.5	64396	8:40:40	63971	64027	12:36:33	0.09	41:55	1:03:12
sko56-2	56	480473.5	499700	3:09:35	486426.5	498836	9:07:10	496482	496561	15:59:27	0.02	41:05	1:11:58
sko56-3	56	164609.5	171963	3:08:16	166441.5	171860	8:57:50	169644	171032	16:22:56	0.82	1:00:39	1:53:27
sko56-4	56	302572.5	325803	2:55:51	306550.5	315175	9:00:52	312656	313497	15:17:25	0.27	52:24	1:26:44
sko56-5	56	575501.5	595593.5	2:56:20	582117.5	594477.5	8:57:53	591915.5	592335.5	17:46:46	0.07	1:08:30	1:34:20

Table 6.3: Results for well-known (SRFLP) instances with 36–56 departments. n gives the number of departments, “lb” denotes the lower bound, “blf” gives the objective value of the best layout found and “improve gap (SDP$_0$)” and “improve gap (SDP$_1$)” denote the running times that our algorithm based on relaxation (SDP$_4$) needs to improve on the gaps of the other two approaches. The running times are given in min:sec or in h:min:sec respectively.

The results show that the SDP approaches of Anjos and Yen [21] allow for substantial improvement. On the one hand the approach from [168] reduces the difference between best layout and lower bound for all instances by factors that are, except once, > 10 (both lower and upper bounds are improved for all instances). On the other hand it reaches the gaps achieved by the other two approaches considerably faster. Further it is worthwhile to note that all instances with 36 departments and even one instance with 42 departments can be solved to optimality for the first time. We also experimented with further raising the number of function evaluations. In doing so we obtained another slight improvement of the lower bounds but we couldn't solve additional instances to optimality. Hence we think that the chosen number of function evaluations yields a reasonable trade-off between bound-quality and running time.

In Table 6.4 we compare the cheaper approach from [21] using relaxation $(\mathtt{SDP}_0)$ (the other one gets too expensive for these instances) to the approach from [168] for problems with 60 to 100 departments.

The results show that the SDP approach of Anjos and Yen [21] again allows for some improvement. On the one hand the SDP approach from [168] reduces the difference between best layout and lower bound for all instances by factors going from clearly above 10 to 2 as the instance sizes grow (again both lower and upper bounds are improved for all instances). On the other hand the gaps achieved by the approach of Anjos and Yen [21] are reached in average in about half the time by the approach from [168].

In Table 6.5 we compare the approach of Hungerländer and Rendl [168] with the ILP Branch-and-Cut algorithm of Amaral and Letchford [13], already described in the previous subsection, on three random generated instances with 110 departments from [13]. For their computations Amaral and Letchford [13] set a time limit of 2.5 days and reported results only for the root node because of the challenging size of the instances.

When comparing the running times of the two approaches we do not take into account that Amaral and Letchford [13] use a machine (Intel Core 2 Duo 3.33GHz computer with 3 GB RAM) that is slightly faster than ours. [5]

Again the lower bounds are improved for all instances by the approach from [168]. The gaps achieved by the approach of Amaral and Letchford [13] are reached in about two thirds of the time in average.

While our approach suffers from high computation times for these large instances, its memory requirements are negligible. They were below 30 % of the available RAM for all instances.

Let us finally compare the SDP-based heuristic from [168] with the recent tabu search based heuristic of Samarghandi and Eshghi [267] and the recent permutation-based genetic algorithm of Datta et al. [80] on the 20 "AKV"-instances [22]. On five instances all three heuristics yield the same upper bound, on 5 instances the heuristics from [267] and [80] yield the same best value, on 5 instances the algorithm of Datta et al. [80] generates the best feasible layouts and on 5 instances the approach from [168] produces the best upper bounds. In general the SDP-based heuristic seems to be preferable when $n \leq 70$ and computation time is not a critical factor as its performance depends on the quality of the lower bounds from the SDP relaxation. The "sko"-instances [21] were not considered in [267] and [80], hence for these instances the lower and upper bounds presented in Tables 6.3 and 6.4 are the best known ones to date.

6.4 Conclusions and Current Research

This paper improves on the practical results for (SRFLP). The SDP approach of Hungerländer and Rendl [168] provides optimal solutions for several instances with up 42 departments for the first time. Additionally it significantly reduces the duality gap and running times for large instances with up to 110 departments. These achievements are the consequence of the interaction of the following three advancements:

- the usage of a stronger SDP relaxation,
- the appropriate algorithmic approach to this relaxation,
- a stronger upper bound heuristic.

Instance	n	SDP Anjos/Yen using $(\texttt{SDP}_0)$ [21]				SDP Hungerländer/Rendl [168] - restricted to 250 function eval.				
		lb	blf	time	gap in %	lb	blf	time	gap in %	improve gap $(\texttt{SDP}_0)$
AKV-60-01	60	1473338.5	1478464.0	5:39:13	0.35	1477134	1477834	12:38:16	0.05	4:42:33
AKV-60-02	60	829956.5	844695.0	5:08:10	1.78	841472	841776	11:08:16	0.04	2:01:14
AKV-60-03	60	641723	650533.5	4:50:48	1.38	647031.5	648337.5	9:51:06	0.20	2:58:00
AKV-60-04	60	389733	400669.0	4:55:19	2.81	397951	398406	10:49:59	0.11	1:57:18
AKV-60-05	60	316284.5	319103.0	5:05:28	0.89	318792	318805	12:39:37	0.004	2:54:16
sko64-1	64	93388	97842	8:16:08	4.77	96569	97194	13:08:05	0.65	2:15:21
sko64-2	64	619258	636602.5	8:36:06	2.80	633420.5	634332.5	14:28:38	0.14	2:44:31
sko64-3	64	402165.5	418083.5	8:47:21	3.96	412820.5	414384.5	14:04:55	0.38	4:54:09
sko64-4	64	285762.5	300469	8:38:01	5.15	295145	298155	13:55:45	1.02	2:48:40
sko64-5	64	488035	505185.5	8:47:49	3.51	501059.5	502063.5	13:53:04	0.20	2:30:01
AKV-70-01	70	1513741.5	1533075	24:25:30	1.28	1526359	1528560	26:41:34	0.14	10:36:44
AKV-70-02	70	1424673.5	1444720	24:20:39	1.41	1439122	1441028	26:11:27	0.13	7:56:27
AKV-70-03	70	1503311.5	1526830.5	23:11:47	1.56	1517803.5	1518993.5	26:15:14	0.08	6:59:29
AKV-70-04	70	951725	972389	22:56:51	2.17	967316	969150	27:28:48	0.19	6:22:30
AKV-70-05	70	4207969.5	4218730.5	23:42:47	0.26	4213774.5	4218002.5	28:16:05	0.10	9:38:10
sko72-1	72	135280.5	140209	20:26:35	3.64	138885	139231	29:33:19	0.25	5:22:09
sko72-2	72	690377	716873	19:58:29	3.84	707643	715611	29:40:41	0.11	11:06:29
sko72-3	72	1026164	1063314.5	22:19:25	3.62	1048930.5	1061762.5	32:38:47	0.12	11:57:10
sko72-4	72	898586.5	924542.5	20:20:37	2.89	916229.5	924019.5	33:58:28	0.85	8:26:20
sko72-5	72	415320.5	432062.5	20:21:15	4.03	426224.5	430288.5	31:39:43	0.95	6:23:57
AKV-75-01	75	2377176	2394812.5	40:15:12	0.74	2387590.5	2393600.5	37:57:53	0.25	22:19:37
AKV-75-02	75	4294138	4322967	42:23:20	0.67	4309185	4322492	39:28:38	0.31	21:08:44
AKV-75-03	75	1230123.5	1255634	38:27:39	2.07	1243136	1249251	38:21:06	0.49	11:48:54
AKV-75-04	75	3911919	3950444.5	41:27:49	0.99	3936460.5	3941845.5	38:42:58	0.14	17:57:02
AKV-75-05	75	1763890.5	1797676	43:09:58	1.92	1786154	1791469	41:10:37	0.30	10:43:21
AKV-80-01	80	2045170.5	2073453.5	49:07:29	1.38	2063346.5	2070391.5	58:24:49	0.34	21:03:27
AKV-80-02	80	1903788	1923506	48:31:48	1.04	1918945	1921202	58:47:15	0.12	18:42:50
AKV-80-03	80	3237288.5	3256577	49:22:31	0.60	3245254	3251413	58:17:19	0.19	26:04:02
AKV-80-04	80	3730569	3747950	52:16:43	0.47	3739657	3747829	58:50:47	0.22	35:17:04
AKV-80-05	80	1555271.5	1594228	47:03:04	2.51	1585491	1590847	58:30:30	0.34	13:12:47
sko81-1	81	197416.5	207229	47:42:37	4.97	203424	207063	52:44:10	1.79	18:28:22
sko81-2	81	507726	527239.5	49:02:44	3.84	518711.5	526157.5	59:58:08	1.44	22:45:43
sko81-3	81	942850.5	979816	47:45:13	3.92	962886	979281	58:17:40	1.70	17:27:37
sko81-4	81	1971210.5	2042462	46:48:01	3.62	2019058	2035569	57:21:49	0.82	17:33:03
sko81-5	81	1267977	1311605	50:42:29	3.44	1293905	1311166	58:59:28	1.33	22:49:57
sko100-1	100	367048.5	380981	214:49:05	3.80	375999	380562	191:47:21	1.21	108:20:47
sko100-2	100	2024668	2089757.5	240:13:08	3.21	2056997.5	2084924.5	201:46:52	1.36	116:16:55
sko100-3	100	15750362	16251391.5	236:03:51	3.18	15987840.5	16216076.5	212:38:54	1.43	109:48:22
sko100-4	100	3148661	3266569	255:53:11	3.74	3200643	3263493	204:14:39	1.96	133:18:35
sko100-5	100	1002763.5	1040987.5	219:33:25	3.81	1021584.5	1040929.5	201:29:27	1.89	111:11:49

Table 6.4: Results for well-known $(\texttt{SRFLP})$ instances with 60–100 departments. n gives the number of departments, "lb" denotes the lower bound, "blf" gives the objective value of the best layout found and "improve gap $(\texttt{SDP}_0)$" denotes the running times that our algorithm based on relaxation $(\texttt{SDP}_4)$ needs to improve on the gaps of the approach by Anjos and Yen. The running times are given in h:min:sec.

Instance	ILP Amaral/Letchford [13]				SDP Hungerländer/Rendl [168] - restricted to 250 function eval.				
	lb	blf	time	gap in %	lb	blf	time	gap in %	improve gap [13]
AL110-1	139045719	144331884.5	60:00:00	3.80	143108697	144301616	393:46:29	0.83	42:08:16
AL110-2	82676930	86065390	60:00:00	4.10	84953191	86060345	376:00:02	1.30	45:41:16
AL110-3	2155583	2234803.5	60:00:00	3.68	2211923.5	2234854.5	370:36:41	1.04	42:24:01

Table 6.5: Results for new random generated $(\texttt{SRFLP})$ instances with 110 departments. n gives the number of departments, "lb" denotes the lower bound, "blf" gives the objective value of the best layout found and "improve gap $(\texttt{SDP}_0)$" denotes the running times that our algorithm based on relaxation $(\texttt{SDP}_4)$ needs to improve on the gaps of the approach by Anjos and Yen. The running times are given in h:min:sec.

There are three (combinable) directions to further improve current SDP approaches. Firstly we could include well-designed subsets of order $\leq O(n^6)$ of larger clique inequalities, like e.g. the $\approx \frac{1}{240}n^{10}$ pentagonal inequalities, in the presented relaxations. Secondly we could incorporate the achieved bounds in a branch-and-bound framework and thirdly we could try to speed up the computations over the elliptope that constitute the computational bottleneck of our algorithm, e.g. by using first-order methods instead of interior-point methods.

Acknowledgements: We thank three anonymous referees for their constructive comments and suggestions for improvement leading to the present version of the paper.

Chapter 7

Single-Row Equidistant Layout as a Special Case of Single-Row Layout

Authors: Philipp Hungerländer

Abstract: In this paper we discuss two particular layout problems, namely the Single-Row Equidistant Facility Layout Problem (SREFLP) and the Single-Row Facility Layout Problem (SRFLP). Our aim is to consolidate the two respective branches in the layout literature. We show that the (SREFLP) is not only a special case of the Quadratic Assignment Problem but also a special case of the (SRFLP). This new connection is relevant as the strongest exact methods for the (SRFLP) outperform the best approaches specialized to the (SREFLP). We describe and compare the exact approaches for the (SRFLP), the (SREFLP) and Linear Arrangement that is again a special case of the (SREFLP). In a computational study we showcase that the strongest exact approach for the (SRFLP) clearly outperforms the strongest exact approach tailored to the (SREFLP) on medium and large benchmark instances from the literature.

Keywords: Single-Row Facility Layout; Space Allocation; Machine Location; Quadratic Ordering Problem; Quadratic Assignment Problem; Semidefinite Optimization

7.1 Introduction

Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function that may reflect transportation costs and construction costs. This is a well-known operations research problem that arises in different areas of applications. For example, in manufacturing systems, the placement of machines that form a production line inside a plant so that performance is optimized is a layout problem. Another example arises in the design of Very Large Scale Integration (VLSI) circuits in electrical engineering where one aims to arrange a set of rectangular modules on a rectangular chip area.

The variety of applications means that facility layout encompasses a broad class of optimization problems. The survey paper of Meller and Gau [222] divides facility layout research into three categories. The first category is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of departments with unequal area requirements within a facility so as to minimize the total material handling costs. The second category deals with extensions of the basic problem that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem, such as the layout of machines or departments along one row (e.g. a production line) or cellular layout design. In the following we will discuss two particular problems from this third area and aim to consolidate the two respective branches in the layout literature.

This Single-Row Facility Layout Problem (SRFLP), sometimes called the one-dimensional space allocation problem [252], consists of finding the optimal arrangement of rectangular machines or departments next to each other along one row so as to minimize the total weighted sum of the center-to-center distances between all pairs of departments. Hence an instance of the (SRFLP) consists of n one-dimensional departments, with given positive lengths $l_1, \ldots, l_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{i<j \in [n]} c_{ij} z_{ij}^{\pi}, \tag{7.1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_{ij}^{π} is the center-to-center distance between departments i and j with respect to a particular permutation $\pi \in \Pi_n$.

Several practical applications of the (SRFLP) have been identified in the literature. It arises for example as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle (AGV) travelling in both directions on a straight-line path [150]. Further applications are the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291] and the arrangement of books on a shelf and the assignment of disk cylinders to files [252].

The (SRFLP) has interesting connections to other combinatorial optimization problems. It is a special case of the Weighted Betweenness Problem which is again a special case of the Quadratic Ordering Problem. The (SRFLP) is also related to the NP-hard [189] Unidirectional Cyclic Layout Problem that aims to find an assignment of n machines to n predetermined candidate locations on a circular material handling system such that the total handling cost is minimized (for further details and references see e.g. Altınel and Öncan [5]).

Another extensively discussed combinatorial optimization problem in the layout literature is the Single-Row Equidistant Facility Layout Problem (SREFLP), sometimes called the one-dimensional machine location problem [269] or the linear machine-cell location problem [308]. The (SREFLP) has been investigated quite thoroughly in the last three decades, as it models one of the major challenges encountered in designing a large manufacturing system. The (SREFLP) is normally formulated as follows. Given n machines and flows $f_{ij},\ i,j \in [n],\ i \neq j$ between machines i and j, the aim is to find a one-to-one assignment of the machines to n locations equally spaced along a straight line so as to minimize the sum of the products of distances between the machines with the respective flows.

The (SREFLP) arises in many applications in manufacturing and logistics management, including sheet-metal fabrication [15], printed circuit board and disk drive assembly [65] and the optimal design of a flowline in a manufacturing system [308]. Furthermore Bhasker and Sahni [38] applied the (SREFLP) to minimize the total wire length needed when arranging circuit components on a straight line.

The (SREFLP) is a special case of the quadratic assignment problem (QAP) formulated by Koopmans and Beckmann [186] and hence any algorithm for the latter can also be applied to the former. However, studying special cases of a more general problem may lead to the discovery of more efficient algorithms tailored to solve these special cases. For the (QAP) the algorithms of Christofides and Benavent [70] for the Tree (QAP) and Drezner [91] for the grey pattern (QAP) are such examples.

But the (SREFLP) is also a special case of the (SRFLP) where all departments have the same length and the pairwise connectivities $c_{ij},\ i,j \in [n],\ i < j$ are given as the sum of the flows f_{ij} and f_{ji}. While exact methods and heuristics especially designed for the (SREFLP) clearly outperform general methods for the (QAP), this is not the case for approaches to the (SRFLP) (for details see the computational study in Section 7.4).

Furthermore minimum Linear Arrangement (LA), which is NP-hard [112] (even if the underlying graph is bipartite [111]), is a special case of the (SREFLP) where all connectivities are equal. Hence the (SREFLP) and the (SRFLP) are also NP-hard and weighted (LA) and the (SREFLP) are equivalent problems.

(LA) belongs to the graph layout problems that ask for a permutation of the nodes of the underlying graph that optimizes some function of pairwise node distances. (LA) was originally proposed by Harper [131, 132] to develop error-correcting codes with minimal average absolute errors and was since then applied to VLSI design [300], single machine job scheduling [2, 258] and computational biology [180, 226]. It is also used for the layout of entity relationship models [59] and data flow diagrams [109]. There exist approximation algorithms for (LA) with performance guarantee $O(\log n)$ [41, 257] and $O(\sqrt{\log n}\log\log n)$ [58, 99]. For further details on graph layout problems we refer to the survey paper of Díaz et al. [89].

In the following we will discuss the relations between (LA), the (SREFLP) and the (SRFLP) that have been disregarded in the literature so far in more detail. The main contributions of this paper are:

1. We describe and compare the most successful modelling and algorithmic approaches to (LA), the (SREFLP) and the (SRFLP).
2. We apply the strongest exact approach for the (SRFLP) to all (SREFLP) benchmark instances and favorably compare it to the leading exact algorithm for the (SREFLP).
3. We relate the heuristics for the (SREFLP) and the (SRFLP).

The paper is structured as follows. In Section 7.2 we put the most competitive exact approaches and heuristics for (LA), the (SREFLP) and the (SRFLP) into perspective and compare them from a theoretical point of view. In Section 7.3 we present a semidefinite optimization approach for Quadratic Ordering Problems that we apply to all (SREFLP) benchmark instances from the literature in Section 7.4. Finally, conclusions and future research directions are given in Section 7.5.

7.2 Survey of Exact Approaches and Heuristics for Single-Row Facility Layout

Due to their strong combinatorial nature and the involved NP-hardness, numerous heuristic and meta-heuristic approaches have been proposed for (LA) [174, 221, 250, 251, 265], the (SREFLP) [239, 269, 271, 272, 301, 307, 308] and the (SRFLP) [80, 123, 128, 149, 151, 193, 266, 267].

For most facility layout problems there exist few methods that provide global optimal solutions, or at least a measure of nearness to global optimality, for large instances. The main exceptions are the problems in consideration.

Exact methods for (LA), the (SREFLP) and the (SRFLP) can be divided with respect to the variable and relaxation type that they are using. The problems are formulated either in $\binom{n}{2}$ integer distance variables modelling the distances between all pairs of departments or $\binom{n}{2}$ binary position variables modelling the positions of the departments. It is also possible that $\binom{n}{3}$ betweenness variables or $\binom{n}{2}$ ordering variables modeling the relative order between all triples or pairs of departments are used. Furthermore the enumerative scheme (in general a Branch & Bound approach) is based either on cheap linear relaxations or on more expensive but also stronger semidefinite relaxations.

Semidefinite Programming (SDP) is the extension of Linear Programming (LP) from the cone of non-negative real vectors to the cone of symmetric positive semidefinite matrices. (SDP) includes (LP) as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} &\inf_X \{ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \\ &\mathcal{P} := \{ X \mid \langle A_i, X \rangle = b_i, \ i \in \{1, \dots, m\}, \ X \succcurlyeq 0 \}, \end{aligned} \tag{SDP}$$

where the data matrices $A_i,\ i \in \{1,\dots,m\}$ and C are symmetric. We refer the reader to the handbooks of Wolkowicz et al. [303] and Anjos and Lasserre [18] for a thorough coverage of the theory, algorithms and software in this field, as well as a discussion of many application areas where (SDP) has had a major impact.

The theoretically fastest known exact algorithm for (LA) is based on dynamic programming [187]. Back in 2009 this approach that is restricted to instances of size $n \leq 30$ was the strongest exact method for (LA). Hence finding global optimal solutions or at least tight global bounds for large (LA) instances is already very challenging. Recently one (SDP)-based method and two (LP)-based approaches were developed that are applicable to large instances with $n \geq 40$. The algorithm of Caprara et al. [54] that is using betweenness variables and was realized by Schwarz [275], is the most competitive exact method for small graphs and large, sparse graphs with $n \leq 200$. For even larger graphs the algorithm proposed by Caprara et al. [55] that is using position variables is the method of choice as it can provide reasonable bounds for sparse graphs with up to $n \approx 1000$. The semidefinite approach of Hungerländer and Rendl [168] yields competitive results for most sparse instances with $n \leq 100$ and is the method of choice for dense instances (edge density $\geq 30\%$) with $n \geq 30$. For a detailed survey and comparison of exact methods for (LA) see Hungerländer [156].

In contrast to (LA), the benchmark instances for the (SREFLP) and also the (SRFLP) are in general very dense. Hence it is difficult to efficiently generalize methods based on exploiting sparsity from (LA) to these problems. For the (SREFLP) Kouvelis et al. [191] used a dynamic programming algorithm to solve instances with $n \leq 20$ and Palubeckis [246] proposed an (LP)-based Branch & Bound approach based on distance variables in conjunction with a tabu search heuristic to obtain optimal solutions for instances with $n \leq 35$ and global bounds for instances with $n \leq 60$.

For the (SRFLP) several exact approaches have been proposed. Simmons [283] first studied the (SRFLP) and suggested a Branch & Bound algorithm. Later on Simmons [284] pointed out the possibility of extending the dynamic programming algorithm of Karp and Held [181] to the (SRFLP). This was later on implemented by Picard and Queyranne [252]. A nonlinear model was presented by Heragu and Kusiak [151]. (LP)-based approaches using distance variables were proposed by Love and Wong [215] and Amaral [6]. Amaral [7] achieved a more efficient (LP)-based method by linearizing a quadratic model formulated in ordering variables. However all these models suffer from weak lower bounds and hence have high computation times and memory requirements. But just recently Amaral and Letchford [13] achieved significant progress in that direction through the first polyhedral study of the distance polytope for the (SRFLP) and showed that their approach is quite effective for instances with challenging size ($n \geq 30$). Amaral [8] suggested an (LP)-based cutting plane algorithm using betweenness variables that proved to be highly competitive and solved instances with up to 35 departments to optimality.

Anjos et al. [22] proposed the first semidefinite relaxation for the (SRFLP) yielding bounds for instances with up to 80 departments. Anjos and Vannelli [20] further tightened this relaxation using triangle inequalities as cutting planes and obtained optimal solutions for instances with up to 30 departments that remained unsolved since 1988. Anjos and Yen [21] suggested an alternative semidefinite relaxation and achieved optimality gaps no greater than 5 % for large instances with up to 100 departments. Recently Hungerländer and Rendl [168] proposed a general (SDP)-based approach for quadratic ordering problems, where they further improved on the tightness of the existing relaxations. They used a suitable combination of optimization methods to deal with their stronger but more expensive relaxations and applied their method among others to all known benchmark instances for the (SRFLP) [169]. They provided global optimal solutions for instances with up to 42 departments, and obtained tighter bounds than the Anjos-Yen relaxation for instances with up to 100 departments. Hence their algorithm achieved the best practical performance to date among all exact approaches to the (SRFLP). For a detailed survey and comparison of exact methods for the (SRFLP) see Hungerländer [156]. We will recap the (SDP)-based algorithm of Hungerländer and Rendl in the following section and apply it to all known benchmark instances for the (SREFLP) in Section 7.4.

7.3 A Semidefinite Optimization Approach for Quadratic Ordering Problems

In this section we will elaborate on the (SDP)-based approach by Hungerländer and Rendl [168] that has not only been successfully applied to (LA) and the (SRFLP) but also to several further Quadratic Ordering Problems [61, 62, 63, 156]. First we will deduce a quadratic programming formulation in ordering variables of the (SRFLP). We will rewrite this intractable formulation in matrix notation and use standard techniques for relaxing it. Hence we will obtain a basic semidefinite relaxation that can be tightened by adding several classes of valid constraints. Finally we will describe a suitable combination of optimization methods that can be used to obtain strong lower bounds and feasible layouts from the semidefinite relaxation.

7.3.1 A Matrix-Based Formulation for Single-Row Facility Layout

The center-to-center distances between departments can be encoded using betweenness variables that again can be expressed as products of ordering variables $y_{ij} (i, j \in [n], i < j)$

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j, \\ -1, & \text{otherwise.} \end{cases} \tag{7.2}$$

Any feasible ordering of the departments has to fulfill the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1, \quad i, j, k \in [n], \; i < j < k. \tag{7.3}$$

It is well-known that the 3-cycle inequalities together with integrality conditions on the ordering variables suffice to describe feasible orderings, see e.g. [296, 306]. Now we can rewrite the objective function (7.1) in terms of ordering variables

$$K - \sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \left(\sum_{\substack{k \in [n] \\ k<i}} l_k y_{ki} y_{kj} - \sum_{\substack{k \in [n] \\ i<k<j}} l_k y_{ik} y_{kj} + \sum_{\substack{k \in [n] \\ k>j}} l_k y_{ik} y_{jk} \right), \tag{7.4}$$

where

$$K := \left(\sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \right) \left(\sum_{k \in [n]} l_k \right). \tag{7.5}$$

We collect the ordering variables in a vector y and reformulate the (SRFLP) as a quadratic program in ordering variables.

Theorem 7.1 *Minimizing* (7.4) *over* $y \in \{-1, 1\}^{\binom{n}{2}}$ *fulfilling* (7.3) *solves the* (SRFLP).

Finally we rewrite (7.4) in matrix notation as follows:

$$\min \{ \; \langle C_Y, Y \rangle + K : \; y \in \{-1, 1\}^{\binom{n}{2}} \text{ satisfies } (7.3) \; \}, \tag{SRFLP}$$

where $Y := yy^\top$ and the cost matrix C_Y is deduced from (7.4).

In the following subsection we use matrix-based relaxations to get tight lower bounds for the (SRFLP). Similar relaxations have already been successfully applied to combinatorial optimization problems arising in the area of graph drawing [48, 61, 62, 63].

7.3.2 Semidefinite Relaxations

We apply standard techniques to construct (SDP) relaxations over the linear-quadratic ordering polytope

$$\mathcal{P}_{LQO} := \operatorname{conv}\left\{ \begin{pmatrix}1\\ y\end{pmatrix}\begin{pmatrix}1\\ y\end{pmatrix}^{\top} : \ y \in \{-1,1\}^{\binom{n}{2}}, \ y \text{ satisfies (7.3)} \right\}.$$

First we relax the nonconvex equation $Y - yy^{\top} = 0$ to the positive semidefinite constraint

$$Y - yy^{\top} \succcurlyeq 0.$$

Moreover, the main diagonal entries of Y correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(Y) = e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(y,Y) := \begin{pmatrix} 1 & y^{\top} \\ y & Y \end{pmatrix}, \tag{7.6}$$

where $\dim(Z) = \binom{n}{2} + 1 =: \Delta$. The Schur complement lemma [43, Appendix A.5.5] implies $Y - yy^{\top} \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. We therefore conclude that $\mathcal{P}_{LQO}$ is contained in the elliptope

$$\mathcal{E} := \{ \ Z : \ \operatorname{diag}(Z) = e, \ Z \succcurlyeq 0 \ \}.$$

In order to express constraints on y in terms of Y, they have to be reformulated as quadratic conditions in y. A natural way to do this for the 3-cycle inequalities $|y_{ij} + y_{jk} - y_{ik}| = 1$ consists in squaring both sides. Additionally using $y_{ij}^2 = 1$, we obtain

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i,j,k \in [n], \ i < j < k. \tag{7.7}$$

Buchheim et al. [48] show that these 3-cycle equations formulated in the $\{0,1\}$ model[1] describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. The 3-cycle inequalities are implicitly ensured by the 3-cycle equations together with $Z \succcurlyeq 0$ [156, Proposition 4.2].

Next we can formulate the (SRFLP) as a semidefinite optimization problem in binary variables [see 156, for a proof].

Theorem 7.2 *The problem*

$$\min \left\{ \ K + \langle C_Z, Z \rangle : Z \ \textit{satisfies (7.7)}, \ Z \in \mathcal{E}, \ y \in \{-1,1\}^{\binom{n}{2}} \ \right\}$$

where Z is given by (7.6), K is defined in (7.5) and the cost matrix C_Z is given by

$$C_Z := \begin{pmatrix} 0 & 0 \\ 0 & C_Y \end{pmatrix},$$

is equivalent to the (SRFLP)*.*

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of the (SRFLP):

$$\min \ \{K + \langle C_Z, Z \rangle : Z \text{ satisfies (7.7)}, \ Z \in \mathcal{E} \ \}. \tag{SDP$_2$}$$

There are several ways to tighten the above relaxation. We will concentrate on two of them that have

[1] Helmberg [138] shows that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties are preserved.

been successfully applied for the (SRFLP).

First we notice that Z is generated as the outer product of the vector $\begin{pmatrix} 1 & y \end{pmatrix}^\top$ that has merely $\{-1, 1\}$ entries in the non-relaxed SDP formulation. Hence any feasible solution of the (SRFLP) also belongs to the metric polytope $\mathcal{M}$ that is defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets.

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e, \ \ 1 \le i < j < k \le \Delta \right\}. \tag{7.8}$$

A second class of strengthening constraints for Quadratic Ordering Problems was proposed by Lovász and Schrijver [214]. They suggest to multiply the 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \ge 0, \qquad 1 + y_{ij} + y_{jk} - y_{ik} \ge 0. \tag{7.9}$$

by the nonnegative expressions

$$1 - y_{lo} \ge 0, \qquad 1 + y_{lo} \ge 0, \qquad l, o \in [n], \ l < o. \tag{7.10}$$

This results in the following $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ inequalities:

$$\begin{aligned} -1 - y_{lo} \le y_{ij} + y_{jk} - y_{ik} + y_{ij,lo} + y_{jk,lo} - y_{ik,lo} \le 1 + y_{lo}, \\ -1 + y_{lo} \le y_{ij} + y_{jk} - y_{ik} - y_{ij,lo} - y_{jk,lo} + y_{ik,lo} \le 1 - y_{lo}, \end{aligned} \tag{7.11}$$

for $i, j, k, l, o \in [n], \ i < j < k, \ l < o$. We define the corresponding polytope $\mathcal{LS}$:

$$\mathcal{LS} := \{ \ Z : Z \text{ satisfies (7.11)} \ \}. \tag{7.12}$$

In summary we get the following tractable semidefinite relaxation of $\mathcal{P}_{LQO}$:

$$\min \{ K + \langle C_Z, Z \rangle : Z \text{ satisfies (7.7)}, \ Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \ \}. \tag{SDP$_4$}$$

All variables in Z with cost coefficient greater than zero appear in a 3-cycle equality and thus are tightly constrained in the relaxation. This fact explains why the various linear and semidefinite relaxations for the (SRFLP) that are based on betweenness or ordering variables produce very tight bounds in the root node relaxation even for large instances. Notice that (SDP$_4$) can be easily applied to general Quadratic Ordering Problems by adapting its objective function. For the experiments in Section 7.4 we will apply (SDP$_4$) to the (SREFLP) to allow for a direct comparisons with the well-studied (SRFLP).

7.3.3 Computing Lower and Upper Bounds

The core of our semidefinite approach is to solve our semidefinite relaxation (SDP$_4$) by using the bundle method in conjunction with interior point methods. The resulting fractional solutions constitute lower bounds. By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e. integer solutions that describe a feasible layout of the departments. Hence, in the end we have some feasible solution, together with a proof how far this solution could possibly be from the true optimum. We will discuss these two steps in more detail in the following.

While theoretically tractable, it is clear that (SDP$_4$) has an impractically large number of constraints. Indeed, even including only $O(n^3)$ constraints is not realistic for instances of size $n \ge 20$. For this reason, we adopt an approach originally suggested by Fischer et al. [103] and since then applied to the Max-Cut Problem [262] and several ordering problems [156]. Initially, we only explicitly ensure that Z lies in the elliptope $\mathcal{E}$. This can be achieved efficiently with standard interior-point methods, see e.g. Helmberg et al. [146]. All other constraints are handled through Lagrangian duality in the objective function f. By using the bundle method [103] that iteratively conducts function evaluations of f and makes improvement steps we obtain an approximate minimizer of f that is guaranteed to yield a lower bound to the optimal solution of $(\texttt{SDP}_4)$. Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations to control the overall computational effort.

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans and Williamson [121] to the approximate solution of $(\texttt{SDP}_4)$. We take the resulting vector and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities (7.3). Computational experiments demonstrated that the repair strategy is not as critical as one might assume [63, 169]. For example for the (SRFLP) the (SDP)-based rounding heuristic performs comparably to the strongest heuristics [80, 267]. Further notice that our heuristic clearly outperforms another (SDP)-based heuristic proposed by Anjos et al. [22].

7.4 Computational Comparison

We report the results for different computational experiments with our semidefinite relaxation $(\texttt{SDP}_4)$. All benchmark instances used can be downloaded together with the best layouts found from `http://anjos.mgi.polymtl.ca/flplib`. The (SDP) computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. In Table 7.1 we compare our approach for Quadratic Ordering Problems described in Section 7.3 with the strongest approach specialized to the (SREFLP) designed by Palubeckis [246] on benchmark instances with up to 35 departments taken from several sources [240, 241, 270, 308]. Palubeckis coded his (LP) -based Branch & Bound approach in the C programming language and ran it on a Pentium M 1733 MHz notebook that is about four times slower than our machine.[2] Notice that we do not take into account the speed of the machines in Table 7.1, as it does not affect the conclusions drawn.

For small instances with up to 20 departments the specialized (LP)-based Branch & Bound algorithm is preferable to the (SDP)-based approach whereas the (SDP)-based approach clearly outperforms the (LP) -based algorithm on the larger instances with $n \geq 25$. The difference between the two approaches strongly grows with the problem size.

The instances "N-25" and "N-30" from Table 7.1 were also implicitly considered in the (SRFLP) literature where they are denoted as "N25_01" and "N30_01" and each connectivity $c_{ij},\ i,j \in [n],\ i < j$ is exactly the half of $f_{ij} + f_{ji}$ which results in an optimal objective value of half the size. The instances "N25_02" – "N25_5" and "N30_02" – "N30_5" were obtained by leaving the connectivities and assigning random integers ≤ 30 for the lengths of the departments [20]. In Table 7.2 we computationally compare the four most competitive approaches to the (SRFLP) for the instances discussed above. Again we do not take into account the speed of the machines, as it does not differ too much and thus does not affect the conclusions drawn. Our machine is the quickest and about 2.5 times faster than the one of Amaral [8], which is the slowest. [3]

Comparing the results in Tables 7.1 and 7.2 we find that the approach of Palubeckis [246] is also outperformed by the (LP)-based cutting plane algorithm of Amaral [8] on "N-25" and "N-30". Furthermore we can deduce that an instance does not become a lot harder for the approaches to the (SRFLP) if we allow arbitrary department lengths instead of isochronous ones.

Next we compare our approach with the one of Palubeckis on large benchmark instances with up to 60 departments taken from Yu and Sarker [308]. We restrict the bundle method to 500 function evaluations. This limitation of the number of function evaluations sacrifices some possible incremental improvement of the bounds. We summarize the results in Table 7.3.

[2] For exact numbers of the speed differences see `http://www.cpubenchmark.net/`.
[3] For exact numbers of the speed differences see `http://www.cpubenchmark.net/`.

Instance	Source	n	Optimum	Palubeckis [246]	Hungerländer and Rendl [168]
O-5	[241]	5	150	1	1
O-6		6	292	1	1
O-7		7	472	1	1
O-8		8	784	1	1
O-9		9	1032	1	1
O-10		10	1402	1	1
O-15		15	5134	2	4
O-20		20	12924	37	37
S-12	[270]	12	4431	1	4
S-13		13	5897	1	4
S-14		14	7316	1	15
S-15		15	8942	2	19
S-16		16	11019	3	27
S-17		17	13172	5	50
S-18		18	15699	8	48
S-19		19	18700	22	1:43
S-20		20	21825	55	2:09
S-21		21	24891	1:41	3:20
S-22		22	28607	3:48	3:57
S-23		23	33046	8:12	3:15
S-24		24	37498	13:41	3:31
S-25		25	42349	36:21	7:07
Y-6	[308]	6	1372	1	1
Y-7		7	1801	1	1
Y-8		8	2302	1	1
Y-9		9	2808	1	1
Y-10		10	3508	1	2
Y-11		11	4022	1	4
Y-12		12	4793	1	10
Y-13		13	5471	1	9
Y-14		14	6445	1	34
Y-15		15	7359	2	26
Y-20		20	12185	23	1:14
Y-25		25	20357	22:38	5:14
Y-30		30	27673	16:17:07	17:46
Y-35		35	38194	459:08:51	25:50
N-12	[240]	12	1000	1	1
N-14		14	1866		4
N-15		15	2186	2	17
N-16a		16	3050		8
N-16b		16	2400		6
N-17		17	3388		13
N-18		18	3986		16
N-20		20	5642	41	1:11
N-21		21	5084		1:16
N-22		22	6184		1:20
N-24		24	8270		2:12
N-25		25	9236	24:03	2:48
N-30		30	16494	12:34:18	4:42

Table 7.1: Results for (SREFLP) instances with up to 35 departments. We compare the specialised (LP) -based Branch & Bound approach by Palubeckis [246] and the general (SDP)-based approach by Hungerländer and Rendl [168]. The running times are given in sec, in min:sec or in h:min:sec respectively. A missing entry indicates that the instance was not considered by the respective approach.

Instance	n	Optimum	Anjos and Vannelli [20]	Amaral and Letchford [13]	Amaral [8]	Hungerländer and Rendl [168]
N25_01	25	4618	3:44:38	7:19:44	3:46	2:48
N25_02	25	37116.5	4:50:27	38:35	9:59	5:46
N25_03	25	24301	5:48:21	1:25:41	4:49	4:11
N25_04	25	48291.5	4:04:51	39:34	10:19	5:33
N25_05	25	15623	8:22:22	1:18:10	3:47	3:31
N30_01	30	8247	7:41:06	34:00:51	25:41	4:42
N30_02	30	21582.5	10:41:53	3:56:53	22:43	6:08
N30_03	30	45449	19:32:01	13:08:12	23:14	10:12
N30_04	30	56873.5	31:03:11	58:20	2:19:22	11:44
N30_05	30	115268	19:54:07	13:03:51	1:05:36	18:30

Table 7.2: Results for (SRFLP) instances with up to 30 departments. The running times are given in sec, in min:sec or in h:min:sec respectively.

Instance	n	(LP)-based B & B approach by Palubeckis [246]			(SDP)-based approach by Hungerländer and Rendl [168]				
		Best lower bound	Best layout	Gap (%)	Best lower bound	Best layout	Gap (%)	Time	Improve B & B lower bound
Y-40	40	43891	47561	8.36	47561	47561	0	2:14:47	1:27
Y-45	45	58551	62890	7.41	62849	62904	0.09	3:44:43	4:09
Y-50	50	76520	83127	8.63	83086	83127	0.05	6:03:27	5:32
Y-60	60	102828	112055	8.97	111884	112126	0.22	19:57:35	15:06

Table 7.3: Results for well-known (SREFLP) instances with 40–60 departments. "B & B" is a shortcut for "Branch & Bound". "Improve B & B lower bound" denotes the running times that the (SDP)-based approach needs to improve on the lower bound of the (LP)-based approach. The bundle method is restricted to 500 function evaluations and the running times are given in min:sec or in h:min:sec respectively.

The results show that the specialized (LP)-based approach by Palubeckis [246] allows for substantial improvement. While the tabu search heuristic generates very strong layouts, the lower bounds are quite weak and can be improved very quickly by our (SDP)-based approach.

The standard (SRFLP) benchmark set also contains some large instances with isochronous department lengths. For the sake of completeness we summarize the best known bounds for these instances lifted from Hungerländer and Rendl [169] in Tables 7.4 and 7.5. It would be interesting to compare the tabu search heuristic of Palubeckis [246] on these instances with the strongest heuristics for the (SRFLP):

1. the tabu search based heuristic of Samarghandi and Eshghi [267],
2. the permutation-based genetic algorithm of Datta et al. [80],
3. and our (SDP)-based rounding heuristic.

Further notice that the gaps obtained are quite similar to the ones in Table 7.3 and also stay in the same order of magnitude if we allow arbitrary integer department lengths.

7.5 Conclusions And Future Research

This paper dealt with two particular layout problems, namely the (SREFLP) and the (SRFLP), and aimed to consolidate the two respective branches in the layout literature. We showed that the (SREFLP) is not only a special case of the (QAP) but also a special case of the (SRFLP). This new connection is relevant as the strongest exact methods for the (SRFLP) outperform the best approaches specialized to the (SREFLP). We described and compared the exact approaches for the (SRFLP), the (SREFLP) and (LA) that is again a special case of the (SREFLP) and showed that the (SDP)-based approach of Hungerländer and Rendl [168] for the (SRFLP) outperforms the strongest exact approach tailored to the (SREFLP) from Palubeckis [246] on medium and large (SREFLP) benchmark instances.

Instance	n	Hungerländer and Rendl [169]			
		Best lower bound	Best layout	Gap (%)	Time
ste36-1	36	10287	10287	0	14:50
sko42-1	42	25521	25525	0.02	2:23:09
sko49-1	49	40895	41012	0.29	4:36:21
sko56-1	56	63971	64027	0.09	12:36:33

Table 7.4: Results for well-known (SREFLP) instances with 36–56 departments that have isochronous lengths. The bundle method is restricted to 500 function evaluations and the running times are given in min:sec or in h:min:sec respectively.

Instance	n	Hungerländer and Rendl [169]			
		Best lower bound	Best layout	Gap (%)	Time
sko64-1	64	96569	97194	0.65	13:08:05
sko72-1	72	138885	139231	0.25	29:33:19
sko81-1	81	203424	207063	1.79	52:44:10
sko100-1	100	375999	380562	1.21	191:47:21

Table 7.5: Results for well-known (SREFLP) instances with 64–100 departments that have isochronous lengths. The bundle method is restricted to 250 function evaluations and the running times are given in h:min:sec.

The strength of the discussed (SDP)-based approach is its general applicability. It can not only be successfully applied to the (SRFLP) and its special cases but to any Quadratic Ordering Problem, including for instance applications in the area of graph drawing.

Therefore it seems to be worthwhile to think about ways to further improve the (SDP)-based approach. There are three (combinable) ways to do so. First we could include additional constraint classes to further tighten the underlying semidefinite relaxation. Secondly we could incorporate the bounds obtained in a Branch & Bound framework and thirdly we could try to speed up the computations over the elliptope by using first-order instead of interior-point methods.

Furthermore one could try to identify other layout problems with quadratic ordering structure. A first attempt in this direction has been made by Hungerländer and Anjos [164] for the Multi-row Facility Layout Problem. Another promising application seems to be the Equidistant Unidirectional Cyclic Layout Problem.

Chapter 8

An SDP Approach to the Directed Circular Facility Layout Problem

Authors: Philipp Hungerländer

Abstract: We propose a new combinatorial optimization problem, the so-called Directed Circular Facility Layout Problem (DCFLP). The (DCFLP) seems to be quite interesting as it contains several other problems that have been discussed extensively in literature as special cases. We show that the (DCFLP) is closely related to the Single-Row Facility Layout Problem (SRFLP) and hence we adapt the leading algorithmic method for the (SRFLP) by suggesting an appropriate modelling approach for the (DCFLP). Finally we show that this algorithmic approach yields promising computational results on a variety of benchmark instances.

Keywords: Facility planning and design; Flexible manufacturing systems; Cyclic layout; Semidefinite Programming; Global Optimization; Combinatorial Optimization

8.1 Introduction

Facility layout is concerned with the optimal location of machines inside a plant according to a given objective function that may reflect transportation costs, the construction cost of a material-handling system, the costs of laying communication wiring, or simply adjacency preferences among machines. Facility layout is a well-known operations research problem and arises in different areas of applications.

Flexible manufacturing systems (FMSs) are automated production systems, typically consisting of numerically controlled machines and material handling devices under computer control, which are designed to produce a variety of parts. A fundamental problem in the design of a FMS is that of determining the layout of machines [207]. The layout of the machines has a significant impact on the materials handling cost and time, on throughput, and on productivity of the facility. A poor layout may also negate some of the flexibilities of an FMS [133].

The survey paper by [222] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of machines within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the machines sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously.

The third category is concerned with specially structured instances of the problem, such as the layout of machines along a production line. This paper is concerned with finding global upper and lower bounds for one such type of structured instances, namely the Directed Circular Facility Layout Problem (DCFLP).

The (DCFLP) seeks to arrange the machines on a circular material handling system so as to minimize the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction. The material handling system is assumed to move the parts unidirectionally around the circuit following the sequence specified in its process plan. Each machine is capable of picking up and processing the parts from the material handling system [216]. The most commonly used operational strategy for such systems is that all parts enter and exit the system at the loading and unloading stations that do not perform any operation on these parts. Circular material handling systems are mostly preferred because of their relative low initial investment costs, high material handling flexibility and their ability of being easily accommodated to future introduction of new parts and process changes [3, 190].

To the best of our knowledge this is the first paper dealing with the (DCFLP). The (DCFLP) might be a quite interesting problem as it is a generalization of several layout problems that have been extensively discussed in the literature. To begin with the (DCFLP) is a generalization of the Directed Circular Arrangement Problem (DCAP) allowing the machines to have arbitrary instead of the same lengths. The (DCAP) was first considered by [209] who showed that the problem is NP-hard (hence also (DCFLP) is NP-hard) and gave an $\tilde{O}(\sqrt{n})$-approximation factor algorithm. Later on [231] improved on this result by proposing an $O(\log n \log \log n)$-approximation algorithm. We refer to [31], [209] and [231] for several very nice applications of the (DCAP) in the areas of sever design and ring networks.

Furthermore the (DCFLP) is related to the Unidirectional Cyclic Layout Problem (UCFLP). The (UCFLP) also considers a circular material handling system and the objective is to find an assignment of n machines to n predetermined candidate locations such that the total handling cost is minimized. Compared to the (UCFLP), the (DCFLP) considers machine lengths instead of distances of the locations and hence is an adaption of the Single-Row Facility Layout Problem (for details see below) to circular layouts. The NP-hard [189] (UCFLP) has two well-known special cases that are at the same time special cases of the (DCFLP): In the balanced unidirectional cyclic layout problem (BUCFLP) the material flow is conserved at each machine, i.e. the total inflow is equal to total outflow at each machine. Another special form of the (UCFLP) is the equidistant unidirectional cyclic layout problem (EUCFLP), where the locations around the unidirectional cyclic material handling system are assumed to be equally distant to each others. Clearly the (EUCFLP) is equivalent to the (DCAP). Additionally [45] have shown that the (EUCFLP) and the (BUCFLP) are equivalent. For more details on (UCFLP) (especially for a recent discussion of several exact and heuristic algorithms) and its special cases see e.g. [5, 242, 244] and the references therein.

There exist only few methods that provide global optimal solutions for large instances of layout problems. One exception is the case of the Single-Row Facility Layout Problem (SRFLP). that consists of finding the optimal location of rectangular machines next to each other along one row so as to minimize the total weighted sum of the center-to-center distances between all pairs of machines. The (SRFLP) arises e.g. as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle travelling in both directions on a straight-line path [150].

On the one hand the (SRFLP) is a special case of the weighted betweenness problem which is again a special case of the quadratic ordering problem. On the other hand the NP-hard [112] Weighted Minimum Linear Arrangement (LA) problem is a special case of the (SRFLP) where all machines have the same length. Hence (SRFLP) is also NP-hard. For information and references on exact methods, heuristic algorithms and polynomially-solvable special cases of (LA), we refer the reader to [89]. For more details on the (SRFLP) we refer to the survey article by [19].

There exist global optimization approaches for the (SRFLP) that are based on relaxations of integer linear programming and semidefinite programming (SDP) formulations. In this paper we will adapt a global optimization approaches based on SDP that has been applied very successfully to the (SRFLP) [169]. SDP is the extension of linear programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. We refer the reader to the handbooks of [303] and [18] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

The main contributions of this paper are the following. We introduce a new combinatorial optimization problem, denoted by `(DCFLP)` that contains several, extensively and recently discussed special cases like the `(DCAP)`, the `(BUCFLP)` and the `(EUCFLP)`. We show that the `(DCFLP)` is closely related to the `(SRFLP)` and hence we adapt the leading algorithmic method for the `(SRFLP)` by suggesting an appropriate modelling approach for the `(DCFLP)`. Finally we show that this algorithmic approach yields promising computational results on a variety of benchmark instances.

The paper is structured as follows. In Section 8.2 we deduce a mathematical formulation based on bivalent ordering variables for the `(DCFLP)`. This formulation builds the basis for the matrix-based relaxations and the according algorithmic approach proposed in Section 8.3. Finally computational experience with our SDP approach is reported in Section 8.4.

8.2 Mathematical Formulation

An instance of the Directed Circular Facility Layout Problem `(DCFLP)` consists of n one-dimensional machines with given positive lengths $\ell_1, \dots, \ell_n$ and pairwise flows $f_{ij},\ i,j \in [n], i \neq j$ with $[n] := \{1, 2, \dots, n\}$. The machines are arranged next to each other on a circle. The objective is to find a permutation π of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines (in clockwise direction) is minimized:

$$\min_{\pi \in \Pi_n} \sum_{i,j \in [n], i \neq j} f_{ij} z_{ij}, \tag{8.1}$$

where Π is the set of all feasible layouts and z_{ij} gives the distance between the centroids of machines i and j in the circular layout π (in clockwise direction).

To model the `(DCFLP)` we introduce the binary ordering variables $y_{ij},\ i,j \in [n],\ i < j$:

$$y_{ij} = \begin{cases} 1, & \text{if machine } i \text{ lies before machine } j \\ -1, & \text{otherwise.} \end{cases}$$

Any feasible ordering of the machines on the circle has to fulfill the 3-cycle inequalities:

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1, \quad i,j,k \in [n],\ i < j < k. \tag{8.2}$$

It is well-known that the 3-cycle inequalities together with integrality conditions on the variables suffice to describe feasible orderings, see e.g. [296] or [306]. Next we introduce the distance variables $d_{ij},\ i,j \in [n],\ i < j$ that give the difference of the sums of the lengths of the machines in front of machine i and j respectively

$$\begin{aligned} d_{ij} = {} & \left(\frac{\ell_i}{2} + \sum_{k \in [n],\ k<i} \ell_k \frac{1 + y_{ki}}{2} + \sum_{k \in [n],\ k>i} \ell_k \frac{1 - y_{ik}}{2} \right) - \\ & \left(\frac{\ell_j}{2} + \sum_{k \in [n],\ k<j} \ell_k \frac{1 + y_{kj}}{2} + \sum_{k \in [n],\ k>j} \ell_k \frac{1 - y_{jk}}{2} \right), \\ & \qquad i,j \in [n],\ i < j, \end{aligned}$$

The d_{ij} are linear expressions in the ordering variables. To destroy symmetry and reduce the dimension of the problem, we fix machine 1 to be first in the ordering. Hence we set $y_{1j} = 1,\ j \in [n],\ j \neq 1$. This results (after some additional simplifications) in the following adaption of the distance variables

$$\begin{aligned} d_{1j} &= \frac{1}{2}\left(\sum_{k\in[n],\ 1<k<j} \ell_k y_{kj} + \sum_{k\in[n],\ k>j} \ell_k y_{jk} - L\right), \\ &\qquad\qquad\qquad\qquad j \in [n],\ j \neq 1, \\ d_{ij} &= \frac{1}{2}\left(\sum_{k\in[n],\ 1<k<i} \ell_k y_{ki} - \sum_{k\in[n],\ k>i} \ell_k y_{ik} - \right. \\ &\qquad \left. \sum_{k\in[n],\ 1<k<j} \ell_k y_{kj} + \sum_{k\in[n],\ k>j} \ell_k y_{jk}\right), \\ &\qquad\qquad\qquad\qquad i,j \in [n],\ 1 < i < j. \end{aligned} \tag{8.3}$$

Next we express the distances between machines i and j on the circle, denoted by $z_{ij},\ i,j \in [n],\ i \neq j$, via the distance variables

$$\begin{aligned} z_{1j} &= -d_{1j}, \quad z_{j1} = L + d_{1j}, \quad j \in [n],\ 1 < j, \\ z_{ij} &= -d_{ij} + \frac{1-y_{ij}}{2}L,\ z_{ji} = d_{ij} + \frac{1+y_{ij}}{2}L, \\ &\qquad\qquad i,j \in [n],\ 1 < i < j, \end{aligned} \tag{8.4}$$

where L denotes the sum of the lengths of all machines

$$L = \sum_{k\in[n]} \ell_k.$$

Now we can rewrite the objective function (8.1) with the help of (8.4) as a linear function in $\binom{n-1}{2}$ ordering variables. This yields (after some simplifications)

$$\min_{y\in\{-1,1\}^{\binom{n-1}{2}}} f(y) \tag{8.5}$$

where

$$\begin{aligned} f(y) := \frac{L}{2} \sum_{i,j\in[n],1<i<j} (f_{ij} + f_{ji}) + L \sum_{j\in[n],1<j} f_{j1} + \\ \sum_{i,j\in[n],1<i<j} (f_{ji} - f_{ij}) \left(d_{ij} + \frac{Ly_{ij}}{2}\right) + \sum_{j\in[n],1<j} (f_{j1} - f_{1j}) d_{1j}, \end{aligned}$$

and y is a vector collecting all the ordering variables.

In summary we have deduced a second formulation of the `(DCFLP)`.

Theorem 8.1 *Solving* (8.5) *subject to* (8.2) *and* (8.3) *is equivalent to the* `(DCFLP)`.

Proof. The inequalities (8.2) together with the integrality conditions on y suffice to induce a feasible layout on the circle. Equation (8.3) connects the ordering with the distance variables and finally the definition of the objective function ensures that the distances between machines are computed correctly. □ □

In the next section we propose matrix-based relaxations (for further details see [156]) for getting tight lower bounds to the `(DCFLP)`.

8.3 Semidefinite Relaxations

The matrix lifting approach takes the vector y and considers the matrix $Y = yy^\top$. Our object of interest is the linear-quadratic ordering polytope

$$\mathcal{P}_{LQO} := \operatorname{conv}\left\{ \begin{pmatrix}1\\y\end{pmatrix}\begin{pmatrix}1\\y\end{pmatrix}^\top : y \in \{-1,1\}, y \text{ satisfies (8.2)} \right\}.$$

We apply standard techniques to construct SDP relaxations. First we relax the nonconvex equation $Y - yy^\top = 0$ to the positive semidefinite constraint

$$Y - yy^\top \succcurlyeq 0.$$

Moreover, the main diagonal entries of Y correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(Y) = e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^\top \\ y & Y \end{pmatrix},$$

where $\dim(Z) = \binom{n}{2} + 1 =: \Delta$. The Schur complement lemma [43, Appendix A.5.5] implies $Y - yy^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. We therefore conclude that $\mathcal{P}_{LQO}$ is contained in the elliptope

$$\mathcal{E} := \{\ Z : \ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}.$$

In order to express constraints on y in terms of Y, they have to be reformulated as quadratic conditions in y. A natural way to do this for the 3-cycle inequalites $|y_{ij} + y_{jk} - y_{ik}| = 1$ consists in squaring both sides. Now applying $y_{ij}^2 = 1$ to the resulting equations gives

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i,j,k \in [n],\ i < j < k. \tag{8.6}$$

In [48] it is shown that these 3-cycle equations formulated in the $\{0,1\}$ model[1] describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. The 3-cycle inequalities are implicitly ensured by the 3-cycle equations together with $Z \succcurlyeq 0$ [156, Proposition 4.2].

Next we can formulate the (DCFLP) as a semidefinite optimization problem in binary variables.

Theorem 8.2 *The following optimization problem is equivalent to the* (DCFLP)*:*

$$\min\ \{\ K + \langle C_Z, Z\rangle : Z \text{ satisfies } (8.6),\ Z \in \mathcal{E},\ y \in \{-1,1\}\ \}$$

where $K := \frac{L}{2}\sum_{i,j\in[n],1<i<j}(f_{ij} + f_{ji}) + L\sum_{j\in[n],1<j} f_{j1} - \frac{L}{2}\sum_{j\in[n],1<j}(f_{j1} - f_{1j})$ *, the cost matrix* C_Z *is given by*

$$C_Z := \begin{pmatrix} 0 & f_y^\top \\ f_y & 0 \end{pmatrix},$$

and the cost vector f_y *is deduced by equating the coefficients of the following equation:*

$$4 f_y^\top y = f(y) - K$$

Proof. Since $y_i^2 = 1,\ i \in \{1, \ldots, \Delta - 1\}$ we have $\operatorname{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. The 3-cycle equations (8.6) ensure that $|y_{ij} + y_{jk} - y_{ik}| = 1$ holds.

[1] In [138] it is shown that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties are preserved.

Finally the objective value reflects the total cost of the layout encoded by y due to the definition of the cost matrix C_Z and the constant K. □ □

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of the (DCFLP):

$$\min \{K + \langle C_Z, Z\rangle : Z \text{ satisfies } (8.6),\ Z \in \mathcal{E}\ \}. \tag{SDP$_2$}$$

There are several ways to tighten (SDP$_2$). We will concentrate on two of them that have been successfully applied to the (SRFLP).

First we notice that Z is generated as the outer product of the vector $\begin{pmatrix}1 & y\end{pmatrix}$ that holds merely $\{-1, 1\}$ entries in the non-relaxed SDP formulation. Hence any feasible solution of the (DCFLP) also belongs to the metric polytope $\mathcal{M}$ that is defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets.

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e, 1 \le i < j < k \le \Delta \right\}.$$

A second class of strengthening constraints for our problem was proposed by [214]. They suggest to multiply the 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \ge 0, \qquad 1 + y_{ij} + y_{jk} - y_{ik} \ge 0. \tag{8.7}$$

by the nonnegative expressions

$$1 - y_{lo} \ge 0, \qquad 1 + y_{lo} \ge 0, \qquad l, o \in [n],\ l < o. \tag{8.8}$$

This results in the following $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ inequalities:

$$\begin{gathered} -1 - y_{lo} \le y_{ij} + y_{jk} - y_{ik} + y_{ij,lo} + y_{jk,lo} - y_{ik,lo} \le 1 + y_{lo}, \\ -1 + y_{lo} \le y_{ij} + y_{jk} - y_{ik} - y_{ij,lo} - y_{jk,lo} + y_{ik,lo} \le 1 - y_{lo}, \\ i, j, k, l, o \in [n],\ i < j < k,\ l < o. \end{gathered} \tag{8.9}$$

Hence we define the corresponding polytope $\mathcal{LS}$:

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (8.9)\ \}. \tag{8.10}$$

In summary we get the following tractable semidefinite relaxation of the (DCFLP):

$$\begin{aligned} \min \{&K + \langle C_Z, Z\rangle : Z \text{ satisfies} \\ &(8.6), Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS})\}. \end{aligned} \tag{SDP$_4$}$$

The core of our SDP approach is to solve our SDP relaxation (SDP$_4$), using a dynamic bundle method in conjunction with interior point methods (for details see [103]). The resulting fractional solutions constitute lower bounds for the (DCFLP). By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe a feasible layout of the machines on the circle. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum. We will discuss these two steps and further algorithmic aspects in more detail in a forthcoming paper.

8.4 Computational Experience

We report the results for the (DCFLP) using the semidefinite relaxation (SDP_4). To generate reasonable data for our experiments we use well-known benchmark instances for (LA) and the (SRFLP) and adapt them to the (DCAP) and the (DCFLP) as follows: We transfer the number of machines and their lengths. For adapting the flows we propose two variants. Either we just set $f_{ij} := c_{ij}$ and $f_{ji} := 0$ or we decide randomly (with the same probability for both cases) if $f_{ij} := c_{ij}, f_{ji} := 0$ or $f_{ij} := 0, f_{ji} := c_{ij}$. The first variant we denote as "one-way", the second one as "random".

We restrict ourselves to cases where either f_{ij} or f_{ji} is zero. If both f_{ij} and f_{ji} are greater than zero then we could remodel the problem by setting $f_{ij}^n := f_{ij} - \min(f_{ij}, f_{ji})$, $f_{ji}^n := f_{ji} - \min(f_{ij}, f_{ji})$ and adding the constant $\min(f_{ij}, f_{ji}) \cdot L$ to the objective function. Notice that for such problems the additional constant in the objective function just reduces the relative gap.

We restrict the number of function evaluations of the bundle method to control the overall computational effort. This limitation sacrifices some possible incremental improvement of the bounds. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 24 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. All the instances and the best layouts found can be downloaded from `http://anjos.mgi.polymtl.ca/flplib`. Additionally we compare the results with the best known ones for the (SRFLP) that were achieved using the same SDP relaxation, the same algorithmic approach and the same machine (for details see [169]). We summarize the computational results for small, medium and large instances in Tables 8.1, 8.2 and 8.3 respectively. Due to space restrictions we only include detailed computational results for one instance of each instance type (there exist up to five instances of each instance type).

For all instances with up to 36 machines our SDP approach is able to find the optimal layout. The "one-way" variant of the (DCFLP) proves to be easier than the (SRFLP) as we are able to solve all considered instances with up to 80 machines to optimality. The "random" variant of the (DCFLP) seems to be practically harder than the (SRFLP), but still we are able to obtain reasonable bounds below 26 % even for the largest instances. Although all the circular layouts are always at least as expensive as the according row layouts for all proposed benchmark instances, this is not true in general (a counter-example with 5 machines will be included in a forthcoming paper).

We also ran our algorithm on various well-known (LA) benchmark instances from the literature. There we could solve nearly all instances (independent of the density of the underlying graph) to optimality for both variants "one-way" and "random" for graphs with up to 100 vertices.[2]

Finally we created (BUCFLP) instances following the lines of [242] and [293]. We generated 10 instances for low, medium and high variations in the part flow and for 20 to 70 machines. We could find the optimal solution for all instances with ≤ 30 machines for all variation types within half an hour but did not succeed to close the gap for most instances with more than 40 machines independent of the variation type. Hence the Integer Linear Programming (ILP) approach by [242] that is specialized to the (BUCFLP) is preferable, as their method yields the optimal solution for all instances with up to 50 machines, again independent of the variation type. But if we want to find reasonable bounds for even larger instances with 60 machines or more, then our SDP method is of use as it provides tighter bounds than the competing ILP approach. E.g. for 60 machines we obtain gapes ranging from 6 to 22 % within 10 hours of computing time.[2]

[2]Detailed computational results are omitted due to space restrictions.

Instance	Source	n	(SRFLP)		(DCFLP) one-way		(DCFLP) random	
			Optimal cost	Time	Optimal cost	Time	Optimal cost	Time
H_20	[151]	20	15549	54.3	21122	21.2	22057	42.9
H_30	[151]	30	44965	9:07	62853	2:32	57648	6:30
Cl_20	[151]	20	119.710	1:16	161.290	45.8	158.820	25.4
Cl_30	[151]	30	334.870	14:17	447.490	2:53	434.470	7:14
N25_01	[20]	25	4618	2:48	6221	3:56	5921	3:54
N30_01	[20]	30	8247	4:42	11067	5:00	10216	2:55
Am33_01	[8]	33	60704.5	19:28	84034.5	7:28	79955.5	10:32
Am35_01	[8]	35	69439.5	17:30	95812.5	46:14	90732.5	11:05
ste36-1	[21]	36	10287	14:50	13476	11:12	19548	7:41

Table 8.1: Results for (SRFLP) and (DCFLP) instances with up to 36 machines. The optimal solutions are found for all instances and layout types within 1 hour. The running times are given in sec or in min:sec respectively.

Instance	Source	n	(SRFLP)				(DCFLP) one-way				(DCFLP) random			
			Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time
N40_1	[169]	40	107348.5		0	1:01:36	54285.5		0	1:43:12	148925.5	156684.5	5.21	3:44:15
sko42-1	[21]	42	25521	25525	0.02	2:23:09	34112		0	41:34	31683	36890	16.44	4:34:48
sko49-1	[21]	49	40895	41012	0.29	4:36:21	50697		0	35:47	50010	52844	5.67	8:25:36
sko56-1	[21]	56	63971	64027	0.09	12:36:33	81074		0	2:31:07	79609	89069	11.88	15:49:54

Table 8.2: Results for (SRFLP) and (DCFLP) instances with between 40 and 56 machines. The bundle method is restricted to 500 function evaluations. The running times are given in min:sec or in h:min:sec respectively.

Instance	Source	n	(SRFLP)				(DCFLP) one-way				(DCFLP) random			
			Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time
AKV-60-01	[21]	60	1477134	1477834	0.05	12:38:16	2042653		0	4:20:38	2479392	3032824	22.32	10:22:20
sko64-1	[21]	64	96569	97194	0.65	13:08:05	122479		0	6:33:07	118187	138447	17.14	16:27:22
AKV-70-01	[21]	70	1526359	1528560	0.14	26:41:34	2185508		0	14:38:14	2670853	3320917	24.34	23:42:27
sko72-1	[21]	72	138885	139231	0.25	29:33:19	168547		0	15:00:02	171274	212845	24.27	37:14:22
AKV-75-01	[21]	75	2387590.5	2393600.5	0.25	37:57:53	3457598.5		0	17:48:21	4030236.5	5046105.5	25.21	40:21:01
AKV-80-01	[21]	80	2063346.5	2070391.5	0.34	58:24:49	2837168.5		0	26:41:11	3544971.5	4346950.5	22.62	52:32:09

Table 8.3: Results for (SRFLP) and (DCFLP) instances with between 60 and 80 machines. The bundle method is restricted to 250 function evaluations. The running times are given in h:min:sec.

Chapter 9

A New Modelling Approach for Cyclic Layouts and its Practical Advantages

Authors: Philipp Hungerländer

Abstract: We propose a new facility layout problem, the Directed Circular Facility Layout Problem (DCFLP). The (DCFLP) allows for a wide range of applications and it contains several other layout problems that have been discussed extensively in literature as special cases. We model the (DCFLP) as a Linear Ordering Problem and solve it using several well studied, efficient heuristics like tabu search or memetic algorithms. Furthermore we argue that the (DCFLP) can be solved by both heuristic and exact methods even more efficiently than the single-row facility layout problem that is known to be the easiest layout problem. Finally we validate our findings in a computational study.

Keywords: Facility planning and design; Flexible manufacturing systems; Cyclic layout; Tabu Search; Linear Ordering Problem.

9.1 Introduction

Facility layout is concerned with the optimal location of machines inside a plant according to a given objective function that may reflect transportation costs, the construction cost of a material-handling system, the costs of laying communication wiring, or simply adjacency preferences among machines. Facility layout is a well-known operations research problem and arises in different areas of application. For a recent study see e.g. [28].

The survey paper by Meller and Gau [222] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of machines within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the machine sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem, such as the layout of machines along a production line. This article is concerned with finding strong feasible layouts for one such type of structured instances, namely the Directed Circular Facility Layout Problem (DCFLP).

Flexible manufacturing systems (FMSs) are automated production systems, typically consisting of numerically controlled machines and material handling devices under computer control, which are designed to produce a variety of parts. A fundamental problem in the design of a FMS is that of determining the layout of machines [207]. The layout of the machines has a significant impact on the materials handling cost and time, on throughput, and on productivity of the facility. A poor layout may also negate some of the flexibilities of an FMS [133].

The type of material-handling device used such as handling robots, automated guided vehicles (AGVs), and gantry robots typically determines machine layout in an FMS [232]. In practice, the most frequently encountered layout types are the single-row layout (Figure 9.1), the double-row and multi-row layouts (Figure 9.2) and the circular layout (Figure 9.3).

Figure 9.1: In a.) an AGV transports parts between the machines moving in both directions in a straight line. In b.) a material-handling industrial robot carries parts between the machines.

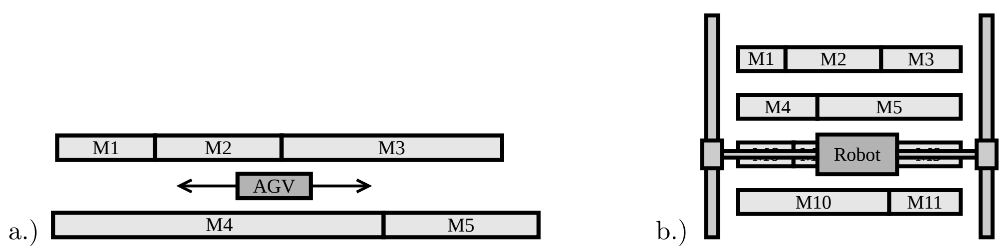


Figure 9.2: In a.) again an AGV transports parts between the machines. In b.) a gantry robot is used when the space is limited.

Hence the easiest layout type is single-row layout. It arises as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle travelling in both directions on a straight-line path [150]. An instance of the Single-Row Facility Layout Problem `(SRFLP)` consists of n one-dimensional machines, with given positive lengths $l_1, \ldots, l_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{i<j \in [n]} c_{ij} z_{ij}^{\pi},$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_{ij}^{π} is the center-to-center distance between machines i and j with respect to a particular permutation $\pi \in \Pi_n$.

Several practical applications of the `(SRFLP)` have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291], the arrangement of machines in flexible manufacturing systems [150], the arrangement of books on a shelf and the assignment of disk cylinders to files [252]. Accordingly several heuristic algorithms have been suggested to tackle instances of interesting size of the `(SRFLP)`, the best ones to date are [80, 188, 267].

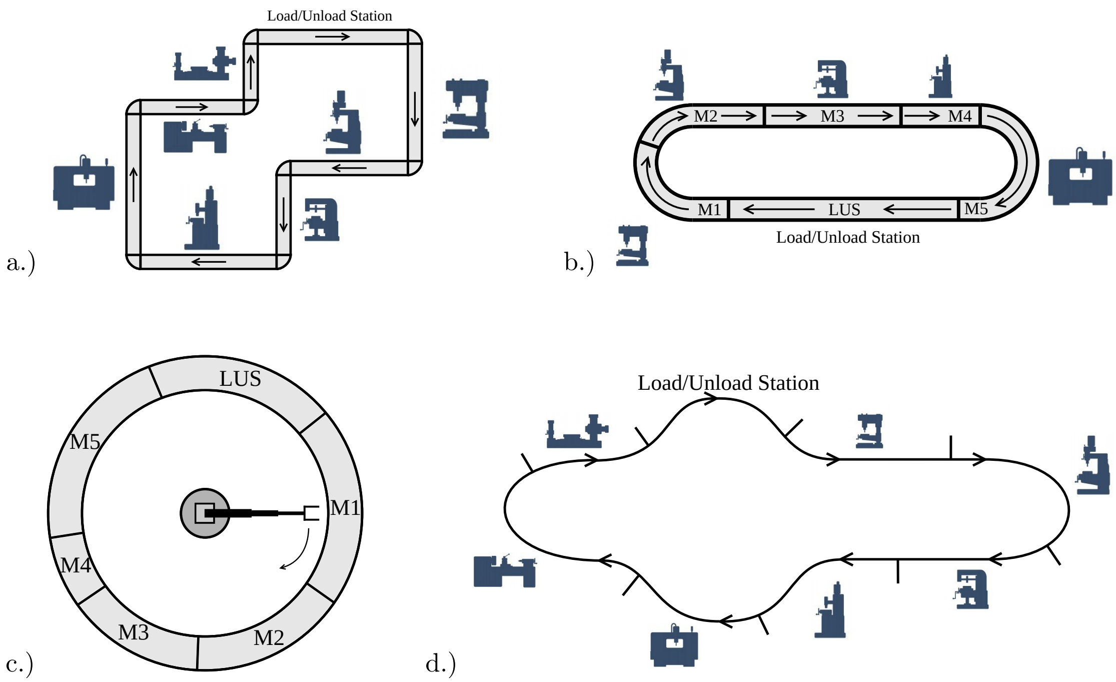


Figure 9.3: In a.) and b.) a conveyor moves in a closed-loop rail in one direction transporting parts among the machines. In c.) a material-handling industrial robot rotates unidirectionally and in d.) single loop AGVs transport parts between the machines.

From an algorithmic and modelling point of view the (SRFLP) is the easiest layout problem known to date. Thus it is one of the few layout problems for which strong global lower bounds and even optimal solutions can be computed for instances of reasonable size. This in turn allows a well-based assessment of the quality of different heuristics. The global optimization approaches for the (SRFLP) are based on relaxations of integer linear programming and semidefinite programming (SDP) formulations. The strongest ones are an LP-based cutting plane algorithm using betweenness variables [8] and an SDP approach using products of ordering variables [169].

The (SRFLP) has several connections to other combinatorial optimization problems. It is a special case of the weighted betweenness problem which is again a special case of the quadratic ordering problem. On the other hand the NP-hard [112] Weighted Minimum Linear Arrangement (LA) problem and the equidistant (SRFLP) [163] are special cases of the (SRFLP) where all machines have the same length. Hence the (SRFLP) is also NP-hard. For information and references on exact methods, heuristic algorithms and polynomially-solvable special cases of (LA), we refer the reader to [89] and [275]. For more details on the (SRFLP) we refer to the survey article by Anjos and Liers [19].

The Directed Circular Facility Layout Problem (DCFLP) seeks to arrange the machines on a circular material handling system so as to minimize the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction. The material handling system is assumed to move the parts unidirectionally around the circuit following the sequence specified in its process plan. Each machine is capable of picking up and processing the parts from the material handling system [216]. The most commonly used operational strategy for such systems is that all parts enter and exit the system at the loading and unloading stations that do not perform any operation on these parts. Circular material handling systems are mostly preferred because of their relative low initial investment costs, high material handling flexibility and their ability of being easily accommodated to future introduction of new parts and process changes [3, 190].

An instance of the Directed Circular Facility Layout Problem (DCFLP) consists of n one-dimensional machines with given positive lengths $\ell_1, \dots, \ell_n$ and pairwise flows $f_{ij},\ i,j \in [n], i \neq j$. The machines are arranged next to each other on a circle (see again Figure 9.3). The objective is to find a permutation π of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines (in clockwise direction) is minimized

$$\min_{\pi \in \Pi_n} \sum_{i,j \in [n], i \neq j} f_{ij} z_{ij}, \tag{9.1}$$

where Π is the set of all feasible layouts and z_{ij} gives the distance between the centroids of machines i and j in the circular layout π (in clockwise direction).

Let us further clarify the differences between the (SRFLP) and the (DCFLP) with the help of a toy example: We consider 4 machines with lengths $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4$. Additionally we have given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$ for the (SRFLP) and pairwise flows $f_{12} = f_{14} = f_{43} = 1,\ f_{13} = f_{42} = 2$ for the (DCFLP). Figure 9.4 illustrates the optimal layouts and the according costs for the two combinatorial optimization problems.

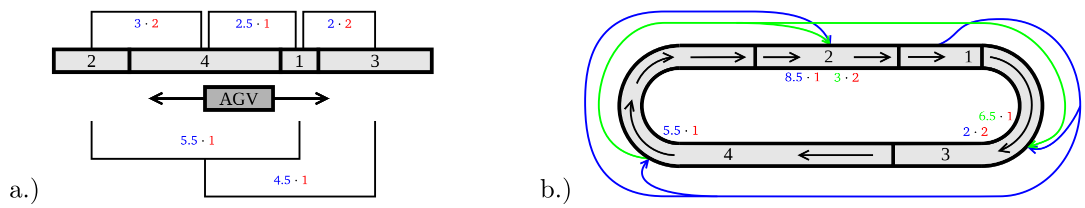


Figure 9.4: We have given the following data: $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4,\ c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2,\ f_{12} = f_{14} = f_{43} = 1,\ f_{13} = f_{42} = 2$. In a.) we display the optimal layout for the (SRFLP) with associated costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we depict the optimal layout for the (DCFLP) with associated costs of $2 \cdot 2 + 3 \cdot 2 + 5.5 \cdot 1 + 8.5 \cdot 1 + 6.5 \cdot 1 = 30.5$.

To the best of our knowledge this is the first journal article dealing with the (DCFLP) (extending the proceedings paper [157], but focusing on heuristic methods rather than exact ones). The (DCFLP) is a very interesting problem as it is a generalization of several layout problems that have been extensively discussed in the literature. To begin with the (DCFLP) is a generalization of the Directed Circular Arrangement Problem (DCAP) that allows the machines to have arbitrary instead of the same lengths. The (DCAP) was first considered by Liberatore [209] who showed that the problem is NP-hard (hence also the (DCFLP) is NP-hard) and gave an $O(\log n)$-approximation factor algorithm. Later on Naor and Schwartz [231] improved on this result by proposing an $O(\log n \log\log n)$-approximation algorithm. We refer to [31, 209] and [231] for several very nice applications of the (DCAP) in the areas of sever design and ring networks.

Furthermore the (DCFLP) is related to the NP-hard [189] Unidirectional Cyclic Layout Problem (UCFLP). The (UCFLP) also considers a circular material handling system and the objective is to find an assignment of n machines to n predetermined candidate locations such that the total handling cost is minimized. The (UCFLP) has two well-known special cases that are at the same time special cases of the (DCFLP): In the balanced unidirectional cyclic layout problem (BUCFLP) the material flow is conserved at each machine, i.e. the total inflow is equal to total outflow at each machine. Another special form of the (UCFLP) is the equidistant unidirectional cyclic layout problem (EUCFLP), where the locations around the unidirectional cyclic material handling system are assumed to be equally distant to each other. Clearly the (EUCFLP) is equivalent to the (DCAP). Additionally Bozer and Rim [45] have shown that the (EUCFLP) and the (BUCFLP) are equivalent. For a recent discussion of several exact and heuristic algorithms for the (UCFLP) and its special cases we refer to [5, 244] and [242].

In comparison with the (UCFLP), the (DCFLP) considers machine lengths instead of distances of the locations and hence is an adaption of the Single-Row Facility Layout Problem to circular layouts. Considering machine lengths instead of the location distances (i.e. location lengths) is clearly the preferable modelling approach in many practical applications where the lengths of the machines are the relevant input parameters. Additionally solving the (UCFLP) with heuristic and exact methods is very hard because it is a (special type of the) quadratic assignment problem [45] and quadratic assignment problems are known to be notoriously difficult [211, 311]. Therefore optimizing cyclic layouts was said to be clearly harder than optimizing row layouts. In this article we aim to reveal that the opposite is true.

We will show that the (DCFLP) can be modelled as a Linear Ordering Problem (LOP). Hence it is, despite its manifold practical applicability and seemingly high complexity, even easier than the (SRFLP) from a modelling and algorithmic point of view, because the easiest models for the (SRFLP) are based on betweenness variables which are linear-quadratic expressions of ordering variables. In its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation ϕ of the rows and columns of W such that

$$\sum_{i<j\in[n]} w_{\phi(i),\phi(j)},$$

is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of nodes i and j either arc (i, j) or arc (j, i), but not both. Then (LOP) consists of finding an acyclic tournament, i.e. a tournament without directed cycles, of G of maximum total edge weight. The (LOP) is equivalent to the acyclic subdigraph problem and the feedback arc set problem. It is well known to be NP-hard [112] and it is even NP-hard to approximate (LOP) within the factor $\frac{65}{66}$ [238].

The (LOP) arises in a large number of applications in such diverse fields as economy, sociology (determination of ancestry relationships [119]), graph drawing (one sided crossing minimization [173]), archaeology, scheduling [39], assessment of corruption perception [1] and ranking in sports tournaments. In 1959 Kemeny [182] posed the first application of the (LOP) (Kemeny's problem) concerning the aggregation of individual orderings to a common one in the best possible way. The probably most established application of the (LOP) is the triangulation of input-output matrices of an economy. [205, 206] was awarded the Nobel Prize in 1973 for his research on input-output analysis.

The main contributions of this article are the following. We introduce the (DCFLP), a new facility layout problem that contains several, extensively and recently discussed layout types, like the (DCAP), the (BUCFLP) and the (EUCFLP), as special cases. We argue that the (DCFLP) allows for a wide range of applications and can be modelled as a (LOP). Hence the (DCFLP) is even easier to solve for heuristic and exact methods than the (SRFLP), the easiest layout problem in the literature. This finding contradicts the current opinion that optimizing cyclic layouts is harder than optimizing row layouts. Finally we support our argumentation by a computational study where we compare the leading heuristics for the (LOP) and the (SRFLP) and assess them using strong lower bounds obtained by an exact method based on semidefinite programming that is applicable to both problem types.

The article is structured as follows. In Section 9.2 we deduce a mathematical formulation for the (DCFLP) based on binary ordering variables. This formulation allows us to tackle the (DCFLP) using heuristic and exact algorithms proposed for the well-studied and surveyed (LOP). In Section 9.3 we recall the heuristics proposed for the (LOP) and argue that the tabu search algorithm by Glover and Laguna [118] is best suited for user in practice to tackle the (DCFLP). In Section 9.4 we give a short description of an exact algorithm based on semidefinite programming that has been successfully applied to both the (LOP) and the (SRFLP). This feature allows for a good and fair assessment and comparison of heuristic methods for both the (DCFLP) and the (SRFLP) in Section 9.5. Section 9.6 concludes the article.

9.2 Mathematical Formulation

To model the (DCFLP) we introduce the $O(n^2)$ binary ordering variables $x_{ij},\ i < j \in [n]$:

$$x_{ij} = \begin{cases} 1, & \text{if machine } i \text{ is located before machine } j, \\ 0, & \text{otherwise.} \end{cases}$$

Any feasible ordering of the machines on the circle has to fulfill the 3-cycle inequalities:

$$0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1, \quad i < j < k \in [n]. \tag{9.2}$$

It is well-known that the 3-cycle inequalities together with integrality conditions on the variables suffice to describe feasible orderings, see e.g. [296] or [306]. Next we introduce the distance variables $d_{ij},\ i < j \in [n]$, that give the difference of the sums of the lengths of the machines in front of machine i and machine j respectively:

$$\begin{aligned} d_{ij} = &\left(\frac{\ell_i}{2} + \sum_{k \in [n],\ k<i} \ell_k x_{ki} + \sum_{k \in [n],\ k>i} \ell_k (1 - x_{ik}) \right) \\ &- \left(\frac{\ell_j}{2} + \sum_{k \in [n],\ k<j} \ell_k x_{kj} + \sum_{k \in [n],\ k>j} \ell_k (1 - x_{jk}) \right),\ i,j \in [n],\ i < j. \end{aligned}$$

The d_{ij} are linear expressions in the ordering variables x_{ij}. To destroy symmetry and reduce the dimension of the problem, we fix machine 1 to be first in the ordering. Hence we set $x_{1j} = 1,\ j \in [n],\ j \neq 1$. This results (after some additional simplifications) in the following adaption of the distance variables

$$\begin{aligned} d_{1j} &= -\frac{\ell_1}{2} - \frac{\ell_j}{2} - \sum_{k \in [n],\ 1<k<j} \ell_k x_{kj} - \sum_{k \in [n],\ k>j} \ell_k (1 - x_{jk}),\ j \in [n],\ j \neq 1, \\ d_{ij} &= \left(\frac{\ell_i}{2} + \sum_{k \in [n],\ 1<k<i} \ell_k x_{ki} + \sum_{k \in [n],\ k>i} \ell_k (1 - x_{ik}) \right) \\ &\quad - \left(\frac{\ell_j}{2} + \sum_{k \in [n],\ 1<k<j} \ell_k x_{kj} + \sum_{k \in [n],\ k>j} \ell_k (1 - x_{jk}) \right),\ 1 < i < j \in [n]. \end{aligned} \tag{9.3}$$

Next we express the distances between machines i and j on the circle, denoted by $z_{ij},\ i,j \in [n],\ i \neq j$, via the distance variables

$$\begin{aligned} z_{1j} &= -d_{1j}, \quad z_{j1} = L + d_{1j}, \quad j \in [n],\ 1 < j, \\ z_{ij} &= -d_{ij} + (1 - x_{ij})L, \quad z_{ji} = d_{ij} + x_{ij}L, \quad 1 < i < j \in [n], \end{aligned} \tag{9.4}$$

where $L = \sum_{k \in [n]} \ell_k$ denotes the sum of the lengths of all machines. Now we can rewrite the objective function (9.1) with the help of (9.4) as a linear function in $\binom{n-1}{2}$ ordering variables:

$$\min_{x \in \{0,1\}^{\binom{n-1}{2}}} f(x) \tag{9.5}$$

where

$$f(x) := L \sum_{i,j\in[n],1<i<j} f_{ij} + \sum_{i,j\in[n],1<i<j} (f_{ji} - f_{ij})(-d_{ij} + Lx_{ij}) + \sum_{j\in[n],1<j} [(f_{j1} - f_{1j})d_{1j} + f_{j1}L],$$

and x is a vector collecting all the ordering variables.

In summary we have deduced the following formulation of the (DCFLP) based on ordering variables.

Theorem 9.1 *The problem* (9.5) *subject to* (9.2) *and* (9.3) *is equivalent to the* (DCFLP).

Proof. The inequalities (9.2) together with the integrality conditions on x suffice to induce a feasible layout on the circle. Equation (9.3) connects the ordering with the distance variables and finally the definition of the objective function ensures that the distances between machines are computed correctly. □

Let us close this section by pointing out the connection of the (LOP) to the (SRFLP) that is modelled most conveniently using the $O(n^3)$ binary betweenness variables $\zeta_{ijk},\ i,j,k \in [n], i < j, i \neq k \neq j$:

$$\zeta_{ijk} = \begin{cases} 1, & \text{if machine } k \text{ lies between machines } i \text{ and } j. \\ 0, & \text{otherwise.} \end{cases}$$

The betweenness variables can be written as linear-quadratic expressions of ordering variables

$$\zeta_{ijk} = 2x_{ik}x_{kj} - x_{ik} - x_{kj} + 1,\ i < k < j \in [n], \quad \zeta_{ijk} = x_{ki} + x_{kj} - 2x_{ki}x_{kj},\ k < i < j \in [n],$$
$$\zeta_{ijk} = x_{ik} + x_{jk} - 2x_{ik}x_{jk},\ i < j < k \in [n].$$

In the following section we will exploit the fact that the (DCFLP) can be modelled as a (LOP) and propose several methods to compute strong feasible layouts for the (DCFLP).

9.3 Computing Feasible Layouts

There exist many and several very strong heuristics for the (LOP). The earliest heuristic for the LOP is a construction heuristic and was introduced in 1958 by Chenery and Watanabe [60]. Further construction methods are due to Becker and Aujac [29]. In general construction methods perform weak in practice, hence it is reasonable to look for improvement possibilities after having constructed some ordering. Local improvement methods that are based on some sort of local search can be expected to obtain optimum or near-optimum solutions for easy problems of medium size, but even more important they are a very powerful concept for the design of meta-heuristics. The most important improvement methods for the (LOP) are local search, improvements based on exchanges of elements, Kernighan-Lin improvements [183] and the heuristic of Chanas and Kobylanski [57].

Insert moves are used as the primary mechanism to move from one solution to another in all relevant heuristics for both the (LOP) and the (SRFLP). But the costs for these insert moves differ significantly for these two problems. For the (LOP) an insertion can be done in $O(n)$ time because only the costs in the row and column of the shifted element have to be considered in the matrix W. For the (SRFLP) an insert move needs $O(n^2)$ time because in principle all entries of the cost matrix C have to be considered. This difference suggests that heuristics for the (DCFLP) obtain in general stronger solutions with less computational effort than heuristics for the (SRFLP).

Metaheuristics [117] are a combination of simple heuristics with some scheme of randomization and additional features which can be interpreted as learning mechanism and systematic exploration of search spaces. All known metaheuristics for the (LOP) have been tested in a large variety of benchmark instances with up to 500 elements in [219] and the best ones are (in descending order of their rank value obtained by a non-parametric Friedman test) are Genetic and Memetic Algorithms [106, 224, 273], Tabu Search [118, 195], Variable Neighborhood Search [110, 194], Scatter Search [42, 52, 116] and the Greedy Randomized Adaptive Search Procedure [52, 100]. The differences among these best methods, e.g. with respect to the rank value or the average percentage deviations from the best solutions found or the score statistics [263], are very small. It seems that (among these methods) the implementation details, specially the incremental computation of the move value, are more important then the specific choice of the method [220]. This statement is in line with our reasoning above that heuristics for the (LOP) obtain in general stronger feasible layouts than heuristics for the (SRFLP).

There are no black-boxes or codes of metaheuristics for the (LOP) available on the Internet. As the differences among the best methods are very small, we decided to implement the tabu search method suggested by Glover and Laguna [118] because all relevant implementation parameters are given in this paper and the fine-tuning process is less involved compared to the memetic algorithm in [273]. Glover and Laguna [118] complemented the basic tabu search procedure with a long-term intensification based on the path relinking methodology, and with a long-term diversification based on the REVERSE operation proposed by Chanas and Kobylanski [57]. Both long-term strategies incorporate frequency information recorded during the application of the short-term phase. Due to clear description of the method in [118], we could do the implementation within a few days and facilely reproduce the results of their computational experiments. This easy reproducibility is an important feature for practitioners looking for an efficient layout in their production plant. In the next section we propose an exact algorithm based on semidefinite programming for obtaining tight lower bounds to the (DCFLP).

9.4 Computing Lower Bounds

Currently available exact algorithms for the (LOP) include a Branch-and-Bound algorithm that uses a linear programming based lower bound by Kaas [175], a Branch-and-Cut algorithm proposed by Grötschel et al. [124] and a combined interior-point cutting-plane algorithm by Mitchell and Borchers [225] who explore polyhedral relaxations of the problem and also provide computational results using Branch-and-Cut. The current state-of-the-art Branch-and-Cut algorithm was developed by the working group of Prof. Reinelt in Heidelberg and is based on sophisticated cut generation procedures (for details see [220]). It can solve large instances from specific instance classes with up to 150 objects, while it fails on other much smaller instances with only 50 objects ("RandB" problems). We propose an exact algorithm based on semidefinite programming (SDP) that has proved to be competitive with the state-of-the-art Branch-and-Cut algorithm for the (LOP) (for a detailed analysis of the strengths and weaknesses of both approaches see [168]) and is the method of choice for the (SRFLP) [169] (and other ordering problems [62, 63]). Hence we are able to obtain strong lower bounds for both the (LOP) and the (SRFLP) using the same algorithm. This allows us to provide a fair assessment and comparison of heuristic methods for both the (DCFLP) and the (SRFLP).

Let us give a short description of our SDP approach. SDP is the extension of linear programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} &\inf_X \{ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \\ &\mathcal{P} := \{ \ X \mid \langle A_i, X \rangle = b_i, \ i \in \{1, \dots, m\}, \ X \succcurlyeq 0 \ \}, \end{aligned} \tag{SDP}$$

where the data matrices A_i, $i \in \{1, \dots, m\}$ and C are symmetric. We refer the reader to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

The core of our approach is to solve an SDP relaxation of the (DCFLP), using the bundle method [103] in conjunction with interior point methods [146]. The resulting fractional solutions constitute lower bounds for the (DCFLP). By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe a feasible layout of the machines on the circle. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum.[1]

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans-Williamson [121] to the solution of our SDP relaxation.We take the resulting vector $\overline{y}$ and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities (9.6). Computational experiments demonstrated that the repair strategy is not as critical as one might assume [63]. For example for the (SRFLP) the SDP-based rounding heuristic performs comparably to the strongest heuristics [80, 188, 267].

Let us give a more detailed description of the implementation of our heuristic. We consider a vector y', that encodes a feasible layout of the departments on the circle. The algorithm stops after 1000 executions of step 2. (Notice that before its 501st execution of step 2, we perform step 1 again. As step 1 is quite expensive, we refrain from executing it too often.)

1. Let Y'' be the current primal fractional solution of the semidefinite relaxation for the (DCFLP) obtained by the bundle method or an interior-point solver. Compute the convex combination $R := \lambda(y'y'^{\top}) + (1-\lambda)Y''$, using some random $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^{\top}$ of R.

2. Apply Goemans-Williamson hyperplane rounding to D and obtain a $-1/+1$ vector $\overline{y}$.

3. Compute the induced objective value $z(\overline{y})$. If $z(\overline{y}) \geq z(y')$: goto step 2.

4. If $\overline{y}$ satisfies all 3-cycle inequalities: set $y' := \overline{y}$ and goto 2. Else: modify $\overline{y}$ by changing the signs of one of three variables in all violated inequalities and goto step 3.

The final y' is the heuristic solution. If the duality gap is not closed after the heuristic, we return to the SDP optimization algorithm and then retry the heuristic (retaining the last vector y').

9.5 Computational Experience

To generate reasonable data for our experiments we use well-known benchmark instances for the (SRFLP) and (LA) and adapt them to the (DCAP) and the (DCFLP) as follows: We transfer the number of machines and their lengths. For adapting the flows we propose two variants. Either we just set $f_{ij} := c_{ij}$ and $f_{ji} := 0$ or we decide randomly (with the same probability for both cases) if $f_{ij} := c_{ij}, f_{ji} := 0$ or $f_{ij} := 0, f_{ji} := c_{ij}$. The first variant we denote as "one-way", the second one as "random".

We restrict ourselves to cases where either f_{ij} or f_{ji} is zero. If both f_{ij} and f_{ji} are greater than zero then we could remodel the problem by setting $f^n_{ij} := f_{ij} - \min(f_{ij}, f_{ji})$, $f^n_{ji} := f_{ji} - \min(f_{ij}, f_{ji})$ and adding the constant $\min(f_{ij}, f_{ji}) \cdot L$ to the objective function. Notice that for such problems the additional constant in the objective function just reduces the relative gap.

We report the results for the (DCFLP) using the tabu search procedure by Glover and Laguna [118] for computing feasible layouts and the exact algorithm based on SDP described in the previous section for obtaining tight lower bounds. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 24 GB RAM, running Debian 5.0 in 64-bit mode. The SDP approach was implemented in Matlab 7.7 and the TS heuristic was written in C. We run the tabu search heuristic on all considered instances for 0.1 seconds. All the instances and the best layouts found can be downloaded from http://anjos.mgi.polymtl.ca/flplib. Additionally the according ordering problems in standard format together with the best orderings found are available at the library of sample instances for the linear ordering problem LOLIB http://www.optsicom.es/lolib/.

[1] Details on the lower bound computation for the (DCFLP) can be found in the appendix.

We compare the results for the (DCFLP) with the best known ones for the (SRFLP) that were obtained by using

- the same SDP relaxation, the same algorithmic approach and the same machine concerning the lower bound computation (for details see [169]).
- 4 different heuristics for generating strong feasible layouts:
  - a permutation-based genetic algorithm [80],
  - two tabu search based heuristics [188, 267],
  - and the SDP rounding heuristic described in the previous section.

The SDP approach is able to solve all 40 instances with up to 36 machines to optimality independent of the layout type considered. Additionally the tabu search heuristic is always able find the optimal layout for both (DCFLP) variants. In Table 9.1 we summarize the computational results for the medium and large instances proposed in [21, 22]. We state detailed numbers only for one out of 5 instances of each instance type but we would like to point out that the results are very homogeneous.[2]

Instance	n	(SRFLP)			(DCFLP) one-way			(DCFLP) random		
		Best lower bound	Best layout	Gap (%)	Best lower bound	Best layout	Gap (%)	Best lower bound	Best layout	Gap (%)
N40_1	40	107348.5		0	154285.5		0	148925.5	151300.5	1.59
sko42-1	42	25521	25525	0.02	34112		0	31683	31993	0.98
sko49-1	49	40895	41012	0.29	50697		0	50010	51031	2.04
sko56-1	56	63971	64027	0.09	81074		0	79609	82388	3.49
AKV-60-01	60	1477134	1477834	0.05	2042653		0	2479392	2608666	5.21
sko64-1	64	96569	96915	0.36	122479		0	118187	123706	4.67
AKV-70-01	70	1526359	1528537	0.14	2185508		0	2670853	2797161	4.73
sko72-1	72	138885	139174	0.21	168547		0	171274	179425	4.76
AKV-75-01	75	2387590.5	2393456.5	0.25	3457598.5		0	4030236.5	4216271.5	4.62
AKV-80-01	80	2063346.5	2069097.5	0.28	2837168.5		0	3544971.5	3723924.5	5.05

Table 9.1: Costs of the best feasible layouts and lowe(SRFLP) and (DCFLP) instances with between 60 and 80 machines. For the (SRFLP) the best feasible layouts were obtained by the SDP rounding strategy or one of the heuristics described in Section 9.5 [80, 188, 267]. For the (DCFLP) the best feasible layouts were obtained by the tabu search procedure suggested in [118] that clearly outperformed the SDP rounding heuristic.

The "one-way" variant of the (DCFLP) proves to be easier for both the heuristic and the exact algorithm than the (SRFLP). On the one hand our exact algorithm is able to solve all instances considered with up to 80 machines to optimality and on the other hand the tabu search heuristic always finds the optimal layout whereas for the (SRFLP) different heuristics provide the best known layouts for different instances (for details see [188]). The "random" variant of the (DCFLP) is practically harder than the (SRFLP) for our exact method , but still we are able to obtain reasonable bounds below 7 % even for the largest instances. The lower bounds for the "random" instances were obtained by the tabu search heuristic that clearly outperformed the SDP rounding heuristic.

Notice that there is no obvious or direct relation between the objective values of the optimal layouts of the (SRFLP) and the (DCFLP). In our experiments the cost of the optimal (DCFLP) layouts is always larger or the same the cost of the optimal (SRFLP) layouts. But we can give a very simple instance with three machines ($l_1 = l_2 = l_3 = 1,\ c_{12} = c_{13} = c_{23} = f_{12} = f_{13} = f_{23} = 1$) where the optimal (DCFLP) layout is cheaper (objective value 3) than the corresponding optimal (SRFLP) layout (objective value 4).

We also applied the SDP method and the tabu search heuristic to 16 well-known (LA) benchmark instances with up to 100 machines proposed in [53, 89, 277]. On these instances our exact SDP algorithm performs quite good:

[2]Detailed results for all instances considered can be found in the appendix.

- For two out of the 16 (LA) instances, where the duality gap is not closed, the SDP approach yields better feasible layouts and stronger lower bounds than the best heuristics and ILP approaches specialized to (LA).
- We find the optimal layouts for all "one-way" instances, whereas the tabu search heuristic does not find 4 out of 16.
- We cannot close the gap for 3 out of 16 "random" instances, but our SDP round heuristic produces better layouts than the tabu search method that cannot find 6 out of 16 optimal layouts.

In summary our computational study supports the claim that the (DCFLP) is easier to solve than the (SRFLP). Especially the tabu search heuristic for the (LOP) is by several orders of magnitude faster than comparable heuristics for the (SRFLP) and additionally provides layouts of higher quality most of the time.

9.6 Outlook and Conclusions

We have proposed the Directed Circular Facility Layout Problem (DCFLP) that represents a new modelling approach for cyclic layouts. The (DCFLP) allows for a wide range of applications and can be solved very efficiently, e.g. via tabu search or memetic algorithms, because it is a special Linear Ordering Problem (LOP). Furthermore we provided theoretical arguments and a computational study supporting our claim that the (DCFLP) can be solved by heuristic and exact methods even more efficiently than the Single-Row Facility Layout Problem (SRFLP) that is known as the easiest layout problem.

As a future research topic we suggest to combine and extend the very efficient heuristics for the (DCFLP) and the (SRFLP) to the Combined Cell Layout Problem (CCLP). The (CCLP) is concerned with the minimization of the material- handling costs in a cellular manufacturing system with two or more cells in the presence of parts that require processing in more than one cell. The machines of each cell can be arranged in a row or on a circle. Hence the (CCLP) allows to model more complex layout types but still builds on well-studied and efficiently solvable basic problems. There even exist lower bounds for reasonably sized instances of the (CCLP) to assess heuristic approaches [165]. Another worthwhile direction of research is to enhance the heuristics for the (LOP) and thus for the (DCFLP) with dynamic and stochastic aspects as well as the ability to consider multiple objectives.

APPENDIX

Semidefinite Models and Relaxations for the (DCFLP)

For applying an exact approach based on semidefinite programming (SDP) it is helpful to model the (DCFLP) by the binary variables $y_{ij},\ i,j \in [n],\ i<j$:

$$y_{ij} = \begin{cases} 1, & \text{if machine } i \text{ is located before machine } j, \\ -1, & \text{otherwise.} \end{cases}$$

For these variables we can rewrite Theorem 9.1 as follows

Corollary 9.2 *The problem*

$$\begin{aligned} \min_{y \in \{-1,1\}^{\binom{n-1}{2}}} \ & \frac{L}{2} \sum_{i,j \in [n], 1<i<j} (f_{ij} + f_{ji}) + L \sum_{j \in [n], 1<j} f_{j1} \\ & + \sum_{i,j \in [n], 1<i<j} (f_{ji} - f_{ij}) \left(-D_{ij} + \frac{L y_{ij}}{2} \right) + \sum_{j \in [n], 1<j} (f_{j1} - f_{1j}) D_{1j}, \end{aligned}$$

subject to

$$D_{1j} = \frac{1}{2} \left(\sum_{k \in [n],\ 1<k<j} \ell_k y_{kj} + \sum_{k \in [n],\ k>j} \ell_k y_{jk} - L \right),\ j \in [n],\ j \neq 1,$$

$$D_{ij} = \frac{1}{2} \left(\sum_{\substack{k \in [n], \\ 1<k<i}} \ell_k y_{ki} - \sum_{\substack{k \in [n], \\ k>i}} \ell_k y_{ik} - \sum_{\substack{k \in [n], \\ 1<k<j}} \ell_k y_{kj} + \sum_{\substack{k \in [n], \\ k>j}} \ell_k y_{jk} \right),\ 1<i<j \in [n],$$

and the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1, \quad i<j<k \in [n], \tag{9.6}$$

is equivalent to the (DCFLP)*.*

Proof. The distance variables D_{1j} and D_{ij} can be obtained analogously to the distance variables d_{1j} and d_{ij} in Section 9.2 and ensure that the distances between machines are computed correctly. The 3-cycle inequalities (9.6) together with the integrality conditions on y suffice to induce a feasible layout on the circle. □

The matrix lifting approach takes now the vector y and considers the matrix $Y = yy^\top$. Our object of interest is the linear-quadratic ordering polytope

$$\mathcal{P}_{LQO} := \text{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top : y \in \{-1,1\}, y \text{ satisfies (9.6)} \right\}.$$

We apply standard techniques to construct SDP relaxations. First we relax the nonconvex equation $Y - yy^\top = 0$ to the positive semidefinite constraint

$$Y - yy^\top \succcurlyeq 0.$$

Moreover, the main diagonal entries of Y correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(Y)=e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(y,Y) := \begin{pmatrix} 1 & y^\top \\ y & Y \end{pmatrix},$$

where $\dim(Z) = \binom{n}{2}+1 =: \Delta$. The Schur complement lemma [43, Appendix A.5.5] implies $Y - yy^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. We therefore conclude that $\mathcal{P}_{LQO}$ is contained in the elliptope

$$\mathcal{E} := \{\ Z:\ \operatorname{diag}(Z)=e,\ Z \succcurlyeq 0\ \}.$$

In order to express constraints on y in terms of Y, they have to be reformulated as quadratic conditions in y. A natural way to do this for the 3-cycle inequalites $|y_{ij}+y_{jk}-y_{ik}|=1$ consists in squaring both sides. Now applying $y_{ij}^2=1$ to the resulting equations gives

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i,j,k \in [n],\ i<j<k. \tag{9.7}$$

In [48] it is shown that these 3-cycle equations formulated in the $\{0,1\}$ model[3] describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. The 3-cycle inequalities are implicitly ensured by the 3-cycle equations together with $Z \succcurlyeq 0$ [156, Proposition 4.2].

Next we can formulate the `(DCFLP)` as a semidefinite optimization problem in binary variables.

Theorem 9.3 *The following optimization problem is equivalent to the* `(DCFLP)`*:*

$$\min\ \{\ K + \langle C_Z, Z\rangle : Z \textit{ satisfies } (9.7),\ Z \in \mathcal{E},\ y \in \{-1,1\}\ \}$$

where $K := \frac{L}{2}\sum_{i,j\in[n],1<i<j}(f_{ij}+f_{ji}) + L\sum_{j\in[n],1<j} f_{j1} - \frac{L}{2}\sum_{j\in[n],1<j}(f_{j1}-f_{1j})$ *, the cost matrix* C_Z *is given by*

$$C_Z := \begin{pmatrix} 0 & f_y^\top \\ f_y & 0 \end{pmatrix},$$

and the cost vector f_y *is deduced by equating the coefficients of the following equation:*

$$4 f_y^\top y = f(y) - K$$

Proof. Since $y_i^2 = 1,\ i \in \{1,\dots,\Delta-1\}$ we have $\operatorname{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. The 3-cycle equations (9.7) ensure that $|y_{ij}+y_{jk}-y_{ik}|=1$ holds. Finally the objective value reflects the total cost of the layout encoded by y due to the definition of the cost matrix C_Z and the constant K. □

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of the `(DCFLP)`:

$$\min\ \{K + \langle C_Z, Z\rangle : Z \text{ satisfies } (9.7),\ Z \in \mathcal{E}\ \}. \tag{SDP$_2$}$$

There are several ways to tighten (SDP$_2$). We will concentrate on two of them that have been successfully applied to the `(SRFLP)`.

First we notice that Z is generated as the outer product of the vector $\begin{pmatrix}1 & y\end{pmatrix}$ that holds merely $\{-1,1\}$ entries in the non-relaxed SDP formulation. Hence any feasible solution of the `(DCFLP)` also belongs to

[3] In [138] it is shown that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties are preserved.

the metric polytope $\mathcal{M}$ that is defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets.

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \leq e, 1 \leq i < j < k \leq \Delta \right\}.$$

A second class of strengthening constraints for our problem was proposed by [214]. They suggest to multiply the 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \qquad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0. \tag{9.8}$$

by the nonnegative expressions

$$1 - y_{lo} \geq 0, \qquad 1 + y_{lo} \geq 0, \qquad l, o \in [n],\ l < o. \tag{9.9}$$

This results in the following $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ inequalities:

$$\begin{aligned} -1 - y_{lo} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lo} + y_{jk,lo} - y_{ik,lo} \leq 1 + y_{lo}, \qquad & i,j,k,l,o \in [n], \\ -1 + y_{lo} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lo} - y_{jk,lo} + y_{ik,lo} \leq 1 - y_{lo}, \qquad & i < j < k,\ l < o. \end{aligned} \tag{9.10}$$

Hence we define the corresponding polytope $\mathcal{LS}$:

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (9.10)\ \}. \tag{9.11}$$

In summary we get the following tractable semidefinite relaxation of the (DCFLP):

$$\min \left\{ K + \langle C_Z, Z \rangle : Z \text{ satisfies } (9.7), Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \right\}. \tag{SDP$_4$}$$

Most of the small facets that are usually used for separation in linear programming based Branch-and-Cut approaches for the (LOP) are already implicitly included in (SDP_4) [156, Proposition 4.2]. Hence it is not surprising that (SDP_4) yields essentially stronger bounds, of course at higher expenses, than linear programming relaxations in practice. These stronger bounds lead to considerably smaller branching trees in a Branch-and-Bound approach which often overcompensates the more expensive bound computation.

To illustrate this let us take a look at two notoriously difficult (LOP) instances, namely the paley graphs with 31 or 43 nodes. Their best known solutions are 285 and 543 respectively. Applying the state-of-the-art ILP Branch-and-Cut approach [219] to these instances results in bounds beyond 300 respectively 600 after days of branching while the proposed SDP approach yields the upper bounds 297 respectively 569 within 10 minutes. Of course there are also many other (LOP) problem classes with up to 250 objects where the linear programming bounds are already quite strong and thus the ILP Branch-and-Cut approach yields the optimal solution.

While theoretically tractable, it is clear that (SDP_4) has an impractically large number of constraints. Indeed, even solving a semidefinite relaxations containing only $O(n^3)$ constraints is not realistic for instances of size $n \geq 20$. For this reason, we adopt an approach originally suggested in [103]. Initially, we only explicitly ensure that Z lies in the elliptope $\mathcal{E}$. This can be achieved efficiently with standard interior-point methods, see e.g. [146]. All other constraints are handled through Lagrangian duality. Thus the objective function f becomes non-smooth and the evaluation of f for a given feasible point amounts to solving a problem over $\mathcal{E}$. In our experiments, we use a primal-dual interior-point method that also provides a subgradient of f to conduct the function evaluations. Using these ingredients, we get an approximate minimizer of f using subgradient optimization techniques such as the bundle method [103].

Detailed Computational Results

Let us point out that we can always round the lower bound $\langle C_Z, Z\rangle$ up to the next integer because 0.5 can only occur in the constant term K. We restrict the number of function evaluations of the bundle method to control the overall computational effort. This limitation sacrifices some possible incremental improvement of the bounds.

For small and medium (SRFLP) instances we restrict the bundle method to a maximum of 500 function evaluations and summarize the results in Tables 9.2 and 9.3. For our experiments we use the well-known benchmark instances from [6, 7, 8, 21, 151, 283], randomly generated instances from [20, 169] and instances with clearance requirements from [150].

In Table 9.4 we summarize the results for large instances with 60 to 80 machines taken from [21] and [22]. This time we restrict the bundle method to a maximum of 250 function evaluations to reduce the very high running times. For the (LA) instances we restrict the bundle method to a maximum of 500 function evaluations. and summarize the results of our experiments in Table 9.5.

Instance	Source	n	(SRFLP)		(DCFLP) one-way		(DCFLP) random	
			Optimal cost	Time	Optimal cost	Time	Optimal cost	Time
S8	[283]	8	801	0.6	1068	0.3	985	0.3
S8H	[283]	8	2324.5	2.3	2456.5	0.3	2392.5	0.3
S9	[283]	9	2469.5	0.7	3056.5	0.3	2963.5	0.5
S9H	[283]	9	4695.5	9.2	5120.5	0.4	5030.5	0.5
S10	[283]	10	2781.5	0.6	3791.5	0.4	3574.5	0.9
S11	[283]	11	6933.5	1.3	9355.5	1.0	8189.5	0.8
P15	[6]	15	6305	19.7	8284	2.4	8041	3.5
P17	[7]	17	9254	34.9	12717	5.8	13105	21.1
P18	[7]	18	10650.5	32.5	14450.5	6.8	14726.5	24.1
H_20	[151]	20	15549	54.3	21122	21.2	22057	42.9
H_30	[151]	30	44965	9:07	62853	2:32	57648	6:30
Cl_5	[151]	5	1.100	0.1	1.490	0.4	1.270	0.1
Cl_6	[151]	6	1.990	0.1	2.480	0.3	2.440	0.2
Cl_7	[151]	7	4.730	0.6	4.890	0.3	5.530	0.3
Cl_8	[151]	8	6.295	0.4	8.015	0.4	7.835	0.3
Cl_12	[151]	12	23.365	7.9	30.225	1.6	28.055	1.8
Cl_15	[151]	15	44.600	19.6	61.480	3.0	61.310	3.2
Cl_20	[151]	20	119.710	1:16	161.290	45.8	158.820	25.4
Cl_30	[151]	30	334.870	14:17	447.490	2:53	434.470	7:14
N25_01	[20]	25	4618	2:48	6221	3:56	5921	3:54
N25_02	[20]	25	37116.5	5:46	53933.5	1:00	53162.5	1:36
N25_03	[20]	25	24301	4:11	34784	3:16	31698	1:23
N25_04	[20]	25	48291.5	5:33	70468.5	1:19	69385.5	2:54
N25_05	[20]	25	15623	3:31	22256	45.5	21540	1:05
N30_01	[20]	30	8247	4:42	11067	5:00	10216	2:55
N30_02	[20]	30	21582.5	6:08	30890.5	22:52	28194.5	6:16
N30_03	[20]	30	45449	10:12	64275	6:21	62721	7:23
N30_04	[20]	30	56873.5	11:44	84051.5	24:04	83023.5	6:51
N30_05	[20]	30	115268	18:30	164981	5:15	164662	12:18
Am33_01	[8]	33	60704.5	19:28	84034.5	7:28	79955.5	10:32
Am33_02	[8]	33	67684	48:07	94504	2:39	101700	12:12
Am33_03	[8]	33	69942.5	36:33	98414.5	3:08	105357.5	12:58
Am35_01	[8]	35	69439.5	17:30	95812.5	46:14	90732.5	11:05
Am35_02	[8]	35	61712	41:01	86175	15:34	184637.5	20:33
Am35_03	[8]	35	69002.5	53:14	96865.5	13:14	90215.5	14:49
ste36-1	[21]	36	10287	14:50	13476	11:12	19548	7:41
ste36-2	[21]	36	181508	25:25	237692	33:29	336125	28:05
ste36-3	[21]	36	101643.5	24:01	138237.5	13:38	200207.5	16:01
ste36-4	[21]	36	95805.5	16:15	131592.5	9:15	195123.5	18:06
ste36-5	[21]	36	91651.5	17:58	133715.5	7:42	172998.5	11:11

Table 9.2: Results for (SRFLP) and (DCFLP) instances with up to 36 machines. The optimal solutions are found for all instances and layout types within 1 hour. The running times of the SDP approach are given in sec or in min:sec respectively.

Instance	Source	n	(SRFLP)				(DCFLP) one-way				(DCFLP) random			
			Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time
N40_1	[169]	40	107348.5		0	1:01:36	154285.5		0	1:43:12	148925.5	151300.5	1.59	3:44:15
N40_2	[169]	40	97693		0	52:52	135486		0	16:15	144880	146971	1.44	3:55:13
N40_3	[169]	40	78589.5		0	1:21:40	118601.5		0	1:22:06	120570.5	120754.5	0.15	3:38:52
N40_4	[169]	40	76669		0	1:15:58	115049		0	5:40:28	118906	120042	0.96	4:07:58
N40_5	[169]	40	103009		0	2:20:09	153266		0	1:11:50	145625	145744	0.08	3:39:24
sko42-1	[21]	42	25521	25525	0.02	2:23:09	34112		0	41:34	31683	31993	0.98	4:34:48
sko42-2	[21]	42	216099.5	216120.5	0.01	2:43:34	308129.5		0	35:45	307766.5	311251.5	1.13	4:25:24
sko42-3	[21]	42	173245.5	173267.5	0.01	2:47:18	252477.5		0	23:43	248828.5	253216.5	1.76	4:52:47
sko42-4	[21]	42	137379	137615	0.17	2:53:05	194866		0	41:00	192648	194569	1.00	4:31:40
sko42-5	[21]	42	248238.5		0	1:08:42	356611.5		0	39:36	353175.5	357471.5	1.22	4:36:23
sko49-1	[21]	49	40895	41012	0.29	4:36:21	50697		0	35:47	50010	51031	2.04	8:25:36
sko49-2	[21]	49	416142	416178	0.01	8:27:34	573272		0	1:01:02	574750	588591	2.41	8:29:54
sko49-3	[21]	49	324464	324512	0.02	8:03:03	446309		0	50:33	444335	457742	3.02	8:50:43
sko49-4	[21]	49	236718.5	236755	0.02	9:15:14	330308.5		0	1:09:11	322226.5	329119.5	2.14	8:30:50
sko49-5	[21]	49	666130	666143	0.002	9:30:22	896555.5		0	1:34:58	881914.5	904923.5	2.61	8:32:15
sko56-1	[21]	56	63971	64027	0.09	12:36:33	81074		0	2:31:07	79609	82388	3.49	15:49:54
sko56-2	[21]	56	496482	496561	0.02	15:59:27	680388.5		0	2:48:35	679368.5	706263.5	3.96	15:37:06
sko56-3	[21]	56	169644	171032	0.82	16:22:56	223781		0	2:54:10	232759	239005	2.68	15:35:38
sko56-4	[21]	56	312656	313497	0.27	15:17:25	436350		0	4:38:53	434260	448307	3.23	15:14:04
sko56-5	[21]	56	591915.5	592335.5	0.07	17:46:46	833410		0	3:39:03	855870	887712	3.72	16:08:06

Table 9.3: Results for (SRFLP) and (DCFLP) instances with between 40 and 56 machines. The bundle method is restricted to 500 function evaluations. The running times are given in min:sec or in h:min:sec respectively. For the (SRFLP) the best feasible layouts were obtained by the SDP rounding strategy or one of the heuristics described in Section 9.5 [80, 188, 267]. For the (DCFLP) the best feasible layouts were obtained by the tabu search procedure suggest in [118].

Instance	n	(SRFLP)				(DCFLP) one-way				(DCFLP) random			
		Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time	Best lower bound	Best layout	Gap (%)	Time
AKV-60-01	60	1477134	1477834	0.05	12:38:16	2042653		0	4:20:38	2479392	2608666	5.21	10:22:20
AKV-60-02	60	841472	841776	0.04	11:08:16	1224508		0	5:27:32	1345517	1406480	4.53	11:07:06
AKV-60-03	60	647031.5	648337.5	0.20	9:51:06	927362		0	4:20:59	1104305	1148412	3.99	10:34:24
AKV-60-04	60	397951	398406	0.11	10:49:59	610184		0	8:38:01	663133	680905	2.68	9:48:47
AKV-60-05	60	318792	318805	0.004	12:39:37	451681		0	4:11:27	576075	591985	2.76	10:15:09
sko64-1	64	96569	96915	0.36	13:08:05	122479		0	6:33:07	118187	123706	4.67	16:27:22
sko64-2	64	633420.5	634332.5	0.14	14:28:38	919477		0	10:11:12	913297	951454	4.18	15:31:21
sko64-3	64	412820.5	414327.5	0.38	14:04:55	553264.5		0	6:50:30	587192.5	610927.5	4.04	15:48:09
sko64-4	64	295145	297332.0	0.74	13:55:45	394514		0	6:06:59	389265	407336	4.64	15:33:45
sko64-5	64	501059.5	501922.5	0.17	13:53:04	679565		0	9:27:46	686896	717466	4.45	15:31:38
AKV-70-01	70	1526359	1528537	0.14	26:41:34	2185508		0	14:38:14	2670853	2797161	4.73	23:42:27
AKV-70-02	70	1439122	1441028	0.13	26:11:27	2082572		0	14:00:41	2533911	2674120	5.53	30:03:37
AKV-70-03	70	1517803.5	1518993.5	0.08	26:15:14	2207952.5		0	19:30:29	2395187.5	2532646.5	5.74	36:09:58
AKV-70-04	70	967316	968796	0.19	27:28:48	1420836.5		0	27:38:48	1557019.5	1634485.5	4.98	33:47:29
AKV-70-05	70	4213774.5	4218002.5	0.10	28:16:05	6023331.5		0	16:49:54	7252641.5	7667993.5	5.73	31:38:56
sko72-1	72	138885	139174	0.21	29:33:19	168547		0	15:00:02	171274	179425	4.76	37:14:22
sko72-2	72	707643	715611	0.62	29:40:41	918687		0	14:10:09	980256	1038406	5.93	30:37:27
sko72-3	72	1048930.5	1054110.5	0.49	32:38:47	1470058.5		0	20:12:00	1617738.5	1697744.5	4.95	37:31:45
sko72-4	72	916229.5	920086.5	0.42	33:58:28	1255362.5		0	26:08:17	1368468.5	1443741.5	5.50	33:42:30
sko72-5	72	426224.5	428248.5	0.47	31:39:43	542872.5		0	11:52:34	563290.5	595530.5	5.72	31:08:31
AKV-75-01	75	2387590.5	2393456.5	0.25	37:57:53	3457598.5		0	17:48:21	4030236.5	4216271.5	4.62	40:21:01
AKV-75-02	75	4309185	4321190	0.28	39:28:38	6170887		0	19:46:45	7285028	7664408	5.21	41:04:15
AKV-75-03	75	1243136	1248423	0.43	38:21:06	1895325		0	48:06:32	2182004	2316993	6.18	42:01:47
AKV-75-04	75	3936460.5	3941816.5	0.14	38:42:58	5496300		0	19:40:09	6670688	6946526	4.14	39:17:19
AKV-75-05	75	1786154	1791408	0.30	41:10:37	2480432		0	16:42:28	2844624	2990000	5.11	39:50:32
AKV-80-01	80	2063346.5	2069097.5	0.28	58:24:49	2837168.5		0	26:41:11	3544971.5	3723924.5	5.05	52:32:09
AKV-80-02	80	1918945	1921136	0.11	58:47:15	2785739		0	25:24:34	3198286	3370233	5.38	61:54:05
AKV-80-03	80	3245254	3251413	0.19	58:17:19	5113068		0	31:46:18	5963020	6240366	4.65	57:18:22
AKV-80-04	80	3739657	3746515	0.18	58:50:47	5695886		0	37:15:57	6488517	6856745	5.68	58:24:15
AKV-80-05	80	1585491	1588885	0.21	58:30:30	2247171.5		0	32:03:30	2461083.5	2610297.5	6.06	51:36:19

Table 9.4: Results for (SRFLP) and (DCFLP) instances with between 60 and 80 machines. The bundle method is restricted to 250 function evaluations. The running times are given in h:min:sec. For the (SRFLP) the best feasible layouts were obtained by the SDP rounding strategy or one of the heuristics described in Section 9.5 [80, 188, 267]. For the (DCFLP) the best feasible layouts were obtained by the tabu search procedure suggest in [118].

Instance	n	Density	(LA)			(DCAP) one-way			(DCAP) random		
			Best lower bound	Best layout	time (sec)	Best lower bound	Best layout	time (sec)	Best lower bound	Best layout	time (sec)
can_24	24	0.246	210		1:07	311		26.5	379		43.2
fidap005	27	0.358	414		2:04	702		48.1	981		1:10
pores_1	30	0.236	383		4:47	383		1:34	679		2:15
ibm32	32	0.181	485		5:06	638		24:34	648		10:54
bcspwr01	39	0.062	106		17:16	131		16:07	129		10:35
fidapm05	42	0.277	1003		1:43:20	1989		14:31	2745		22:07
bcspwr02	49	0.050	161		1:27:18	218		58:23	190		46:13
will57	57	0.079	335		5:06:47	571		3:08:23	1120		2:42:52
dwt_59	59	0.060	289		10:32:05	327		3:01:50	811		3:34:27
impcol_b	59	0.164	2074 / 2076		14:11:23	2659		3:50:44	4193 / 4346		21:00:32
can_62	62	0.041	210		8:21:30	271		5.55:29	310		4:59:59
gd95c	62	0.076	506		10:10:48	796		2:58:02	1370		6:49:52
dwt_66	66	0.059	192		1:01:00	346		9:24:23	1025		6:41:40
dwt_72	72	0.029	167		21:28:54	175		20:50:14	234 / 236		59:05:46
can_73	73	0.057	1090 / 1100		35:01:49	1304		41:55:45	1572 / 1579		57:04:44
tub100	100	0.029	246		59:48:10	246		222:20:27	934		226:34:24

Table 9.5: Results for (LA) and (DCAP) instances with up to 100 machines. The bundle method is restricted to 500 function evaluations. The running times are given in min:sec or in h:min:sec respectively. All the best feasible layouts were obtained by the SDP rounding heuristic for both (LA) and (DCAP). 22 out of the 32 best feasible (DCAP) layouts were also found by the tabu search procedure suggest in [118].

Chapter 10

An SDP Approach to the Parallel Row Ordering Problem

Authors: Philipp Hungerländer

Abstract: The k-Parallel Row Ordering Problem (kPROP) is an extension of the Single-Row Facility Layout Problem (SRFLP) that considers arrangements of the departments along more than one row. We propose an exact algorithm for the (kPROP) that extends the semidefinite programming approach for the (SRFLP) by modelling inter-row distances as products of ordering variables. For $k = 2$ rows, our algorithm is competitive with a mixed integer programming (MIP) formulation that was proposed by Amaral [12] very recently. The MIP approach allows to solve instances with up to 23 departments to optimality within a few days of computing time while our semidefinite programming approach yields tight global bounds for instances of the same size within a few minutes on a similar machine. Additionally our algorithm is able to produce reasonable global bounds for instances with up to 100 departments . We show that our approach is also applicable for $k \geq 3$ rows and even yields better computational results for a larger number of rows.

Keywords: Facilities planning and design; Flexible manufacturing systems; Semidefinite Programming; Global Optimization

10.1 Introduction

Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function. This is a well-known operations research problem that arises in different areas of applications. For example, in manufacturing systems, the placement of machines that form a production line inside a plant is a layout problem in which one wishes to minimize the total material flow cost. Another example arises in the design of Very Large Scale Integration (VLSI) circuits in electrical engineering. The objective of VLSI floorplanning is to arrange a set of rectangular modules on a rectangular chip area so that performance is optimized; this is a particular version of facility layout. In general, the objective function may reflect transportation costs, the construction cost of a material-handling system, or simply adjacency preferences among departments.

The variety of applications means that facility layout encompasses a broad class of optimization problems. The survey paper [222] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of departments within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the departments sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem. This paper will focus on a problem from this third area, namely the k-Parallel Row Ordering Problem `(kPROP)`. In this introduction we will highlight the relations of the `(kPROP)` to other problems from this third category like the Single-Row Facility Layout Problem `(SRFLP)`, the Space-Free Multi-Row Facility Layout Problem `(SF-MRFLP)` and the Multi-Row Facility Layout Problem `(MRFLP)`. These layout problems are e.g. of special interest for optimizing flexible manufacturing systems (FMSs).

FMSs are automated production systems, typically consisting of numerically controlled machines and material handling devices under computer control, which are designed to produce a variety of parts. In FMSs the layout of the machines has a significant impact on the materials handling cost and time, on throughput, and on productivity of the facility. A poor layout may also adulterate some of the flexibilities of an FMS [133]. The type of material-handling devices used such as handling robots, automated guided vehicles (AGVs), and gantry robots typically determines machine layout in an FMS [232]. In practice, two of the most frequently encountered layout types are the single-row layout (Figure 10.1) and multi-row layouts (Figure 10.2).

Figure 10.1: In a.) an AGV transports parts between the machines moving in both directions along a straight line. In b.) a material-handling industrial robot carries parts between the machines.

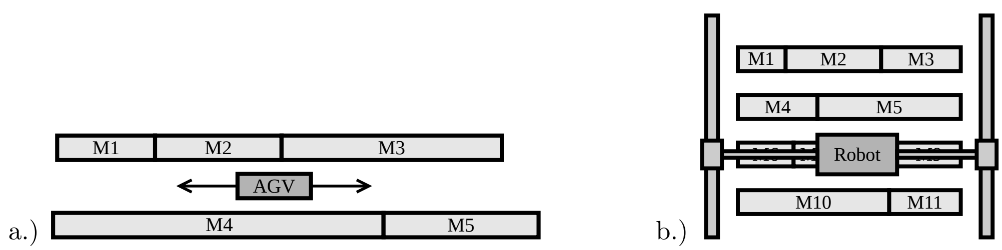


Figure 10.2: In a.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. In b.) a gantry robot is used when the space is limited.

The Single-Row Facility Layout Problem `(SRFLP)` The easiest known layout type is single-row layout. It arises as the problem of ordering stations on a production line where the material flow is handled by an AGV in both directions on a straight-line path [150]. An instance of the `(SRFLP)` consists of n one-dimensional machines, with given positive lengths $\ell_1, \dots, \ell_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi}, \tag{10.1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_{ij}^{π} is the center-to-center distance between machines i and j with respect to a particular permutation $\pi \in \Pi_n$.

Several practical applications of the (SRFLP) have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291], the arrangement of machines in flexible manufacturing systems [150], the arrangement of books on a shelf and the assignment of disk cylinders to files [252]. Accordingly several heuristic algorithms have been suggested to tackle instances of interesting size of the (SRFLP), the best ones to date are [80, 188, 267].

The (SRFLP) is NP-hard, even if all department lengths are equal and the connectivities are binary [113] . The (SRFLP) is one of the few layout problems for which strong global lower bounds and even optimal solutions can be computed for instances of reasonable size. The global optimization approaches for the (SRFLP) are based on relaxations of integer linear programming (ILP) and semidefinite programming (SDP) formulations. The strongest ILP approach is an LP-based cutting plane algorithm using betweenness variables [8] that can solve instances with up to 35 departments within a few hours. The strongest SDP approach to date using products of ordering variables [169] is even stronger and can solve instances with up to 42 departments within a few hours. More details on global optimization approaches for the (SRFLP) can be found below in Section 10.2.

The k-Parallel Row Ordering Problem (kPROP) The (kPROP) is an extension of the (SRFLP) that considers arrangements of the departments along more than one row. An instance of the (kPROP) consists of n one-dimensional departments with given positive lengths $\ell_1, \ldots, \ell_n$, pairwise connectivities c_{ij} and an assignment r of each department to one of the k rows $\mathcal{R} := \{1, \ldots, k\}$. The objective is to find permutations $\pi^1 \in \Pi^1, \ldots, \pi^k \in \Pi^k$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\pi^1 \in \Pi^1, \ldots, \pi^k \in \Pi^k} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}, \tag{10.2}$$

where $\Pi = \{\Pi^1, \ldots, \Pi^k\}$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}$ denotes the distance[1] between the centroids of departments i and j in the layout $\{\pi^1, \ldots, \pi^k\} \in \Pi$. If the (kPROP) is restricted to two rows we simply call it (PROP). Applications of the (kPROP) are the arrangement of departments along two or more parallel straight lines on a floor plan, the construction of multi-floor buildings and the layout of machines in FMSs. The (kPROP) was very recently introduced by Amaral [12] that proposed a mixed integer programming (MIP) formulation. From a computational point of view his MIP approach allows to solve instances with up to 23 departments to optimality within a few days.

Further Variants of Multi-Row Layouts The (kPROP) can be further extended to the Space-Free Multi-Row Facility Layout Problem (SF-MRFLP) by additionally optimizing over all possible row assignments. Hence an instance of the (SF-MRFLP) consists of n one-dimensional departments with given positive lengths $\ell_1, \ldots, \ell_n$, pairwise connectivities c_{ij} and a function $r : [n] \to \mathcal{R}$ that assigns each department to one of the k rows. The objective is to find permutations $\pi^1 \in \Pi^1, \ldots, \pi^k \in \Pi^k$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\pi^1 \times \ldots \times \pi^k \in \Pi^1 \times \ldots \times \Pi^k} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi^{r(i)}, \pi^{r(j)}},$$

[1] We will discuss two different ways for defining the distance between pairs of departments in Subsection 10.3.1.

where $\Pi = \Pi^1 \times \ldots \times \Pi^k$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{r(i)},\pi^{r(j)}}$ denotes the distance between the centroids of departments i and j in the layout $\{\pi^1 \times \ldots \times \pi^k\} \in \Pi$. If we restrict the (SF-MRFLP) to two rows we obtain the Space-Free Double-Row Facility Layout Problem (SF-DRFLP) as a special case. A specific example of the application of the (SF-DRFLP) is in spine layout design. Spine layouts, introduced by Tompkins [295], require departments to be located along both sides of specified corridors along which all the traffic between departments takes place. Although in general some spacing is allowed, layouts with no spacing are much preferable since spacing often translates into higher construction costs for the facility. Algorithms for spine layout design have been proposed, see e.g. [197]. The best methods known to date for the (SF-DRFLP) are an algorithm based on a MIP formulation proposed by [10] and an SDP approach suggest by Hungerländer and Anjos [164] that is also applicable to the (SF-MRFLP). The MIP formulation allows to solve instances with up to 13 departments to optimality within a few hours of computing time. Amaral [10] also proposed two heuristics (based on 2-opt and 3-opt) and showed that these heuristics can handle larger instances with up to 30 departments. Hungerländer and Anjos [164] extend the SDP approach from this paper and provide high-quality global bounds in reasonable time for (SF-DRFLP) instances with up to 15 departments and for (SF-MRFLP) instances with up to 5 rows and 11 departments.

The Double-Row Facility Layout Problem (DRFLP) is a natural extension of the (SRFLP) in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel (see Figure 10.2). This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the (DRFLP) is especially relevant for real-world applications. The (DRFLP) can be further generalized to the (MRFLP), where the departments are arranged along k parallel rows. Hence the (MRFLP) is a generalization of the (SF-MRFLP) in which the rows may not have a common left origin and space is allowed between departments.

The (MRFLP) has many applications such as computer backboard writing [288], campus planning [90], scheduling [115], typewriter keyboard design [253], hospital layout [95], the layout of machines in an automated manufacturing system [151], balancing hydraulic turbine runners [198], numerical analysis [46], optimal digital signal processors memory layout generation [302]. Different extensions of the (MRFLP) like considering a clearance between any two adjacent machines given as a fuzzy set [114] or the design of a FMS in one or multiple rows [101] have been proposed and tackled with genetic algorithms. Somewhat surprisingly, the development of exact algorithms for the (DRFLP) and the (MRFLP) has received only limited attention in the literature. In the 1980s Heragu and Kusiak [150] proposed a non-linear programming model and obtained locally optimal solutions to the (SRFLP) and the (DRFLP). Recently Chung and Tanchoco [72] (see also [310]) focused exclusively on the (DRFLP) and proposed a MIP formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Amaral [11] proposed an improved MIP formulation that allowed him to solve instances with up to 12 departments to optimality.

A toy example for illustrating and comparing different layout types Next let us further clarify the workings and differences of the (SRFLP), the (kPROP), the (SF-MRFLP) and the (MRFLP) with the help of a toy example: We consider 4 machines with lengths $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4$. Additionally we are given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$. Figure 10.3 illustrates the optimal layouts and the corresponding costs for the different problems.

Outline The main contributions of this article are the following. We propose the first SDP approach for the (PROP) that is at the same time the first (exact) approach to the (kPROP) for $k \geq 3$. We show the connections and differences of the formulations and relaxations for the (SRFLP) and the (kPROP) and argue that in general the (kPROP) is essentially harder to solve than the (SRFLP). In a computational study we demonstrate that for the (PROP) our algorithm is competitive with a MIP formulation that was proposed

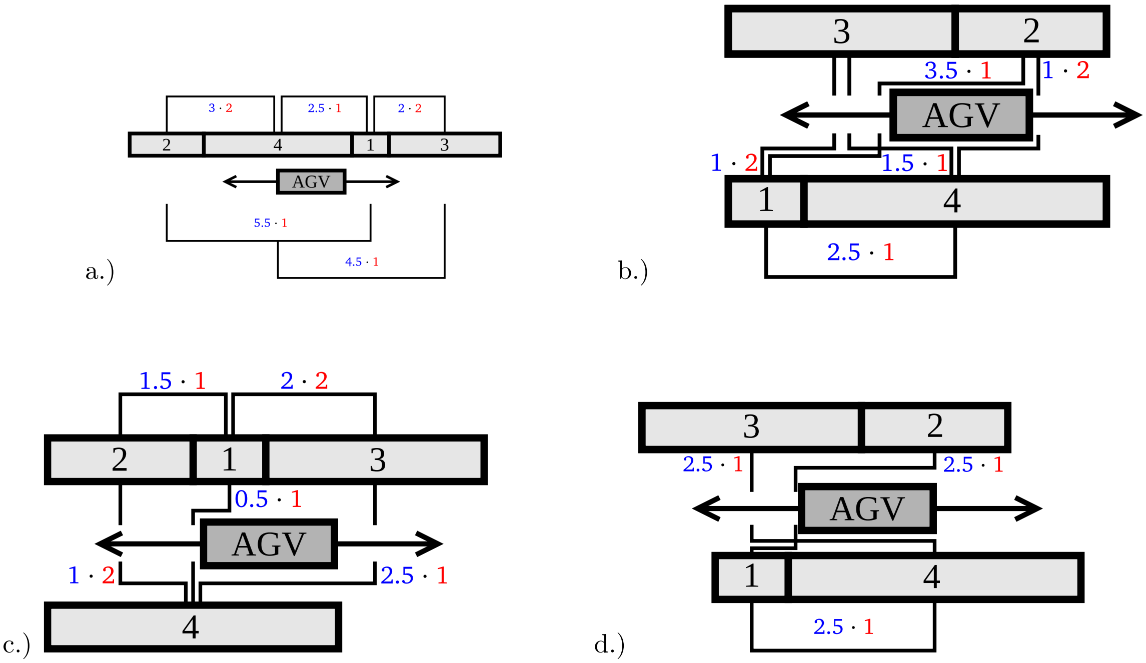


Figure 10.3: We are given the following data: $\ell_1 = 1$, $\ell_2 = 2$, $\ell_3 = 3$, $\ell_4 = 4$, $c_{12} = c_{14} = c_{34} = 1$, $c_{13} = c_{24} = 2$. In a.) we display the optimal layout for the (SRFLP) with associated costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we depict the optimal layout for the (PROP) with machines 3 and 2 assign to row 1 and machines 1 and 4 assign to row 2. The corresponding costs are $3.5 \cdot 1 + 1 \cdot 2 + 1 \cdot 2 + 1.5 \cdot 1 + 2.5 \cdot 1 = 11.5$. In c.) we show the optimal layout for the (SF-DRFLP) with associated costs of $1.5 \cdot 1 + 2 \cdot 2 + 0.5 \cdot 1 + 1 \cdot 2 + 2.5 \cdot 1 = 10.5$. Finally we display the optimal layout for the (DRFLP) in d.). The corresponding costs are $2.5 \cdot 1 + 2.5 \cdot 1 + 2.5 \cdot 1 = 7.5$.

by Amaral [12] very recently. The MIP approach allows to solve instances with up to 23 departments to optimality within a few days of computing time while our SDP approach yields strong global bounds for instances of the same size within a few minutes on a similar machine. Additionally our approach is able to produce reasonable global bounds for instances with up to 100 departments. Finally we demonstrate that our approach is also applicable for (kPROP) instances with $k \geq 3$ rows: We propose two different methods for calculating the distances of departments located in non-adjacent rows and show that for one of the two variants the computational results even improve for a larger number of rows.

The article is structured as follows. In Section 10.2 we recall different formulations and relaxations for the (SRFLP). In Section 10.3 we discuss the possibilities to extend the different (SRFLP) formulations for the (kPROP). In particular we argue that a reasonably tight SDP relaxation for the (kPROP) consists of the relaxation for the (SRFLP) plus two further constraint classes. In Section 10.4 we explain how the proposed SDP relaxations can be solved efficiently and we suggest a heuristic that generates feasible layouts from the solutions of the SDP relaxations. Computational results demonstrating the strength and potential of our SDP approach for the (kPROP) are presented in Section 10.5. Section 10.6 concludes the paper.

10.2 Formulations and Relaxations for the (SRFLP)

There exist several different LP and SDP formulations for the (SRFLP) based on betweenness, distance or ordering variables. MIPs using distance variables were proposed by Love and Wong [215] and Amaral [6]. Both models suffer from weak lower bounds and hence have high computation times and memory requirements. Recently Amaral and Letchford [13] achieved significant progress in that direction through the first polyhedral study of the distance polytope for the (SRFLP) and showed that their approach can solve instances with up to 30 departments within a few hours of computing time. In the following we will describe two formulations for the (SRFLP) that relate to the most competitive exact approaches for the (SRFLP).

10.2.1 Zero-One Linear Programming via Betweenness Variables

To model the (SRFLP) as a preferably easy zero-one LP, let us introduce the betweenness variables $\zeta_{ijk},\ i,j,k \in [n],\ i<j,\ i \neq k \neq j$,

$$\zeta_{ijk} = \begin{cases} 1, & \text{if department } k \text{ lies between departments } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

We collect these betweenness variables in a vector ζ and rewrite (10.1) in terms of ζ (for details see [8, Proposition 1 and 2]):

$$\min_{\zeta \in \mathcal{P}^n_{Btw}} \sum_{\substack{i,j,k \in [n],\\ i<j,k<j}} (c_{ij}\ell_k - c_{ik}\ell_j)\,\zeta_{ijk} + \sum_{\substack{i,j \in [n]\\ i<j}} \left(\frac{c_{ij}}{2}(\ell_i + \ell_j) + \sum_{\substack{k \in [n],\\ k>j}} c_{ij}\ell_k \right). \tag{10.3}$$

where P^n_{Btw} denotes the betweenness polytope

$$\mathcal{P}^n_{Btw} := \text{conv}\,\{\zeta \ :\ \zeta \text{ represents an ordering of the elements of } [n]\}.$$

If department i comes before department j, department k has to be located mutually exclusive either left of department i, or between departments i and j, or right of department j. Thus the following equations are valid for the betweenness polytope $\mathcal{P}^n_{Btw}$

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i<j<k \in [n]. \tag{10.4}$$

In [268] it is shown that these equations describe the smallest linear subspace that contains $\mathcal{P}^n_{Btw}$. To obtain a tight LP relaxation several additional classes of valid inequalities can be deduced. We refer to Amaral [8] for a description of an exact algorithm based on the above formulation that is able to solve instances with up to 35 departments to optimality within a few hours.

10.2.2 Semidefinite Programming via Ordering Variables

Another way to get tight global bounds for (SRFLP) is the usage of SDP relaxations. SDP is the extension of LP to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} &\inf_X \{\ \langle C, X\rangle \ :\ X \in \mathcal{P}\}, \\ &\mathcal{P} := \{\ X \mid \langle A_i, X\rangle = b_i,\ i \in \{1,\dots,m\},\ X \succcurlyeq 0\ \}, \end{aligned} \tag{SDP}$$

where the data matrices $A_i,\ i \in \{1,\dots,m\}$ and C are symmetric. For further information on SDP we refer to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

We can deduce an SDP formulation for the (SRFLP) from the betweenness-based approach above by introducing bivalent ordering variables $y_{ij},\ i,j \in [n],\ i < j$,

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise,} \end{cases} \tag{10.5}$$

and using them to express the betweenness variables ζ via the transformations

$$\zeta_{ijk} = \frac{1 + y_{ik}y_{kj}}{2},\ i < k < j, \qquad \zeta_{ijk} = \frac{1 - y_{ki}y_{kj}}{2},\ k < i < j, \qquad \zeta_{ijk} = \frac{1 - y_{ik}y_{jk}}{2},\ i < j < k, \tag{10.6}$$

for $i, j, k \in [n]$. Using (10.6) we can easily rewrite the objective function (10.3) and equalities (10.4) in terms of ordering variables

$$K - \sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \left(\sum_{\substack{k \in [n] \\ k<i}} \ell_k y_{ki} y_{kj} - \sum_{\substack{k \in [n] \\ i<k<j}} \ell_k y_{ik} y_{kj} + \sum_{\substack{k \in [n] \\ k>j}} \ell_k y_{ik} y_{jk} \right), \tag{10.7}$$

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i < j < k \in [n], \tag{10.8}$$

with

$$K = \left(\sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \right) \left(\sum_{k \in [n]} \ell_k \right). \tag{10.9}$$

In [48] it is shown that the equations (10.8) formulated in a $\{0,1\}$ model describe the smallest linear subspace that contains the quadratic ordering polytope

$$\mathcal{P}_{QO}^n := \operatorname{conv} \{ \ yy^\top \ : \ y \in \{-1,1\},\ |y_{ij} + y_{jk} - y_{ik}| = 1 \ \}.$$

To obtain matrix-based relaxations we collect the ordering variables in a vector y and consider the matrix $Y = yy^\top$. The main diagonal entries of Y correspond to y_{ij}^2 and hence $\operatorname{diag}(Y) = e$, the vector of all ones. Now we can formulate the (SRFLP) as a semidefinite program, first proposed in [22]

$$\min \{ \ \langle C, Y \rangle + K : \ Y \text{ satisfies (10.8)},\ \ \operatorname{diag}(Y) = e,\ \operatorname{rank}(Y) = 1,\ Y \succcurlyeq 0 \ \}, \tag{SRFLP}$$

where the cost matrix C is deduced from (10.7). Dropping the rank constraint yields the basic semidefinite relaxation of the (SRFLP)

$$\min \{ \ \langle C, Y \rangle + K : \ Y \text{ satisfies (10.8)},\ \ \operatorname{diag}(Y) = e,\ Y \succcurlyeq 0 \ \}, \tag{$\texttt{SDP}_{\texttt{trivial}}$}$$

providing a lower bound on the optimal value of the (SRFLP).

As Y is actually a matrix with $\{-1,1\}$ entries in the original (SRFLP) formulation, Anjos and Vanelli [20] proposed to further tighten ($\texttt{SDP}_{\texttt{trivial}}$) by adding the triangle inequalities, defining the metric polytope $\mathcal{M}$ and known to be facet-defining for the cut polytope, see e.g. [88]

$$\mathcal{M} = \left\{ Y : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{i,j} \\ Y_{j,k} \\ Y_{i,k} \end{pmatrix} \le e,\ i < j < k \in \binom{n}{2} \right\}. \tag{10.10}$$

Adding the triangle inequalities to (SDP$_{\texttt{trivial}}$), we obtain the following relaxation of the (SRFLP)

$$\min \{ \langle C, Y \rangle + K : \; Y \text{ satisfies (10.8)}, \; Y \in \mathcal{M}, \; \text{diag}(Y) = e, \; Y \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{basic}})$$

As solving (SDP$_{\texttt{basic}}$) directly with an interior-point solver like CSDP gets far too expensive, Anjos and Vannelli [20] suggest to use the $\approx \frac{1}{12}n^6$ triangle inequalities as cutting planes in their algorithmic framework.

Recently Hungerländer and Rendl [169] suggested a further strengthening of (SDP$_{\texttt{basic}}$) and an alternative algorithmic approach to solve such large SDP relaxations. To this end we introduce the matrix

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \qquad (10.11)$$

and relax the equation $Y - yy^\top = 0$ to

$$Y - yy^T \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0,$$

which is convex due to the Schur-complement lemma. Note that $Z \succcurlyeq 0$ is in general a stronger constraint than $Y \succcurlyeq 0$. Additionally we use an approach suggested by Lovász and Schrijver in [214] to further improve on the strength of the semidefinite relaxation. This yields the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \qquad & i < j < k \in [n], \; l < m \in [n] \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \qquad & i < j < k \in [n], \; l < m \in [n] \end{aligned} \qquad (10.12)$$

that are generated by multiplying the 3-cycle inequalities valid for the ordering problem

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0,$$

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. These constraints define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{ Z : Z \text{ satisfies (10.12)} \}, \qquad (10.13)$$

consisting of $\approx \frac{1}{3}n^5$ constraints. In summary, we come up with the following relaxation of the (SRFLP)

$$\min \{ \langle C, Y \rangle + K : \; Y \text{ satisfies (10.8)}, \; Z \in (\mathcal{M} \cap \mathcal{LS}), \; \text{diag}(Z) = e, \; Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{standard}})$$

To make (SDP$_{\texttt{standard}}$) computationally tractable Hungerländer and Rendl [169] suggest to deal with the triangle inequalities (10.10) and LS-cuts (10.12) through Lagrangian duality (for details see Subsection 10.4.1 below). Similar relaxations have been applied recently to different types of quadratic ordering problem like the linear ordering problem, the linear arrangement problem and multi-level crossing minimization [48, 64, 168]. For more details on global optimization approaches for the (SRFLP) we refer to the survey article by Anjos and Liers [19].

10.3 Formulations and Relaxations for the (kPROP)

Recently Amaral [12] extended his own approach for the (SRFLP) based on distance variables [6] to the (PROP)[2] and argued that a (PROP) with n departments may be solved faster than a (SRFLP) with n departments. We want to complement this statement: This is true if we model both the (SRFLP) and the (PROP) with distance variables (for convincing theoretical arguments and computational comparisons see

[2]Possible next steps to improve on this approach are tightening the lower bounds by adding cutting planes connected to the distance polytope in the style of [13] and the generalization of the approach to the (kPROP).

[12]), but one should also bear in mind that the two strongest approaches for the (SRFLP) are other ones. As we will see below

- it is not possible to extend the approach of [8] to the (PROP) because the distances of departments from the two different rows cannot be modelled as linear terms in betweenness variables and

- solving an SDP relaxation for the (SRFLP) and the (kPROP) with n departments results in about the same computational effort but the lower bounds from the (SRFLP) relaxations are tighter than the ones from the (PROP) relaxations.

These arguments fall into place with the fact that the largest (PROP) instances with proven optimal solutions have 23 departments but the largest (SRFLP) instances with proven optimal solutions have 42 departments. Hence the (SRFLP) is practically clearly easier to solve than the (PROP) (at least with the models at hand at the moment).

10.3.1 A Semidefinite Formulation via Ordering Variables

In the following we show how to generalize the SDP approach for the (SRFLP) to the (kPROP). Let us start with expressing the center-to-center distances of pairs of departments i and j, $i, j \in [n]$, $i < j$ from different rows as quadratic terms of ordering variables. To do so we sum up the lengths of the departments left of i and j respectively and then take the difference of the two sums. Finally we multiply the whole term by the ordering variable y_{ij}. This ensures a correct calculation of the distances of departments from different rows through the constraints (10.15):

$$z_{ij}^y = \frac{1}{2}(\ell_i + \ell_j) + \sum_{\substack{k \in [n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ki}y_{kj}}{2} + \sum_{\substack{k \in [n],\ i<k<j,\\ r(k)=r(i)}} \ell_k \frac{1 + y_{ik}y_{kj}}{2} + \sum_{\substack{k \in [n],\ k>j,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ik}y_{jk}}{2}, \ r(i) = r(j), \tag{10.14a}$$

$$z_{ij}^y = y_{ij}\left(d_j - d_i\right), \qquad r(i) \neq r(j), \qquad d_i = \frac{\ell_i}{2} + \sum_{\substack{k \in [n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1 + y_{ki}}{2} + \sum_{\substack{k \in [n],\ k>i,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ik}}{2}. \tag{10.14b}$$

To ensure non-negative distances for all feasible layouts we have to introduce the following, additional constraints:

$$z_{ij}^y \geq 0, \qquad i, j \in [n],\ i < j,\ r(i) \neq r(j). \tag{10.15}$$

In summary we are able to rewrite (10.2) with the help of ordering variables.

Theorem 10.1 *Minimizing* $\sum_{i,j \in [n],\ i<j} c_{ij} z_{ij}^y$ *over* $y \in \{-1, 1\}^{\binom{n}{2}}$, (10.8), (10.14) *and* (10.15) *solves the* (kPROP).

Proof. The equations (10.8) model transitivity for $y \in \{-1, 1\}^{\binom{n}{2}}$ [48] and hence suffice together with the integrality conditions on y and (10.15) to induce all feasible layouts. Thus by definition of the distances z_{ij}^y in (10.14), the objective value $\sum_{i,j \in [n], i<j} c_{ij} z_{ij}^y$ gives the costs of a feasible layout. □

Next we rewrite the objective function in terms of matrices and obtain a matrix-based formulation:

$$\min \left\{ \langle C_d, Z \rangle \ : \ y \in \{-1, 1\}^{\binom{n}{2}},\ y \text{ satisfies (10.8) and (10.15)} \right\}, \tag{kPROP}$$

where the cost matrix C_d is deduced by equating the coefficients of the following equation

$$
\begin{aligned}
2\langle C_d, Z\rangle \stackrel{!}{=} & \sum_{\substack{i,j\in[n],\ i<j\\ r(i)=r(j)}} c_{ij}\left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik}y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj}\right)\\
& + \sum_{\substack{i<j\in[n],\\ r(i)\neq r(j)}} c_{ij}y_{ij}\left(L_{r(i)} - L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik}\right.\\
& \left. - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk}\right) + \sum_{h\in\mathcal{R}}\left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij}\right)\left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i\right)\right],
\end{aligned}
$$

and L_i denotes the sum of the length of the departments on row i

$$L_i = \sum_{\substack{k\in[n],\\ r(k)=i}} \ell_k, \quad i\in\mathcal{R}.$$

Finally we can further rewrite the above matrix-based formulation as an SDP:

Theorem 10.2 *The problem*

$$\min\left\{\ \langle C_d, Z\rangle : Z \text{ satisfies } (10.8) \text{ and } (10.15)\ ,\ Z\in\mathcal{E},\ y\in\{-1,1\}^{\binom{n}{2}}\ \right\}$$

is equivalent to the (kPROP).

Proof. Since $y_{ij}^2 = 1,\ i,j\in[n],\ i<j$ we have $\operatorname{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. By Theorem 10.1, integrality on Y together with (10.8) and (10.15) suffice to induce all feasible layouts of the (kPROP) and the objective function $\langle C_d, Z\rangle$ gives the correct costs for all feasible layouts. □

Looking at the (kPROP) with $k\geq 3$ we can model the distance between two departments i and j located in non-adjacent rows alternatively as the sum of the distances of the centers of the departments i and j to the common left origin:

$$
\begin{aligned}
z_{ij}^y =& \frac{\ell_i+\ell_j}{2} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k\frac{1+y_{ki}}{2} + \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k\frac{1-y_{ik}}{2}\\
& + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k\frac{1+y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k\frac{1-y_{jk}}{2}, \quad |r(i)-r(j)|>1.
\end{aligned}
$$

For an illustration and comparison of the direct distance calculation from above and the “indirect” one suggested now see Figure 10.4.

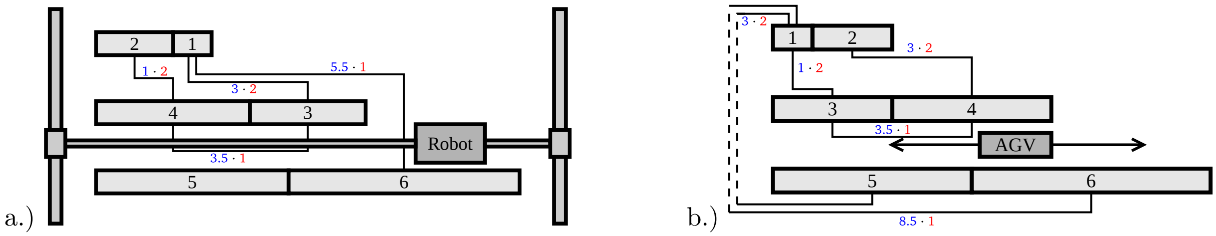


Figure 10.4: Illustration and comparison of the different distance calculation for departments in non-adjacent rows. We are given the following data: $\ell_i = i,\ i \in \{1, \ldots, 6\},\ c_{16} = c_{34} = 1,\ c_{13} = c_{15} = c_{24} = 2.$ Departments 1 and 2 are assigned to row 1, departments 3 and 4 are assigned to row 2 and departments 5 and 6 are assigned to row 3. In a.) a gantry robot is used that can travel "directly" between departments in non-adjacent rows. We display the optimal (PROP) layout for the direct distance calculation with associated costs of $1 \cdot 2 + 3 \cdot 2 + 5.5 \cdot 1 + 3.5 \cdot 1 = 17$. In b.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. If the AGV has to transport parts between machines in non-adjacent rows it has to leave one corridor on the left and enter the other corridor also on the left. We depict the optimal (PROP) layout for the indirect distance calculation. The proper costs are $1 \cdot 2 + 3 \cdot 2 + 3.5 \cdot 1 + 3 \cdot 2 + 8.5 \cdot 1 = 26$.

The alternative cost matrix C_i can be obtained from:

$$
\begin{aligned}
2\langle C_i, Z\rangle \overset{!}{=} & \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)}} c_{ij} \left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik} y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj} \right) \\
& + \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|=1}} c_{ij} y_{ij} \left(L_{r(i)} - L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} \right. \\
& \left. + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right) + \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|>1}} c_{ij} \left(L_{r(i)} + L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} \right. \\
& \left. + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right) + \sum_{h\in\mathcal{R}} \left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij} \right) \left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i \right) \right].
\end{aligned}
$$

Note that we do not have to concern ourselves with horizontal distances because the assignment of the departments to the rows is input data and hence the sum of the horizontal distances is a predetermined constant.

The decision on how to define the distances of departments located in non-adjacent rows is determined by technical conditions in practice. In Subsection 10.5.3 we will computationally compare the two proposed variants with respect to the tightness of the related relaxations. In the following subsection we will deduce a computationally tractable, preferably tight semidefinite relaxation for the (kPROP).

10.3.2 Semidefinite Relaxations

As we are able to model the (kPROP) on the same variables as the (SRFLP) (namely products of ordering variables) we can adopt the strongest SDP relaxation from the previous section:

$$\min \{ \langle C_d, Z \rangle : \ Y \text{ satisfies (10.8) and (10.15)}, \ Z \in (\mathcal{M} \cap \mathcal{LS}), \ \mathrm{diag}(Z) = e, \ Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{standard}})$$

But as the objective function of the (kPROP) is more complex as the one of the (SRFLP) we will try to deduce additional valid inequalities to tighten the relaxation. Therefore we propose triangle inequalities relating the distances between three departments that are clearly valid for both alternative ways to calculate the distances between departments located in non-adjacent rows:

$$z^y_{ij} + z^y_{ik} \geq z^y_{jk}, \qquad z^y_{ij} + z^y_{ik} \geq z^y_{jk}, \qquad z^y_{ik} + z^y_{jk} \geq z^y_{ij}, \qquad i, j, k \in [n], \ i < j < k. \qquad (10.16)$$

By an inductive argument it is very easy to see that the above constraints imply similar constraints for more than three departments. Hence let us define the polytope

$$\mathcal{DV} := \{ Z : Z \text{ satisfies (10.16)} \} \qquad (10.17)$$

containing the $3\binom{n}{3}$ triangle inequalities relating the distances between 3 or more departments. Adding these constraints to ($\texttt{SDP}_{\texttt{standard}}$) yields

$$\min \{ \langle C_d, Z \rangle : \ Y \text{ satisfies (10.8) and (10.15)}, \ Z \in (\mathcal{M} \cap \mathcal{LS} \cap \mathcal{DV}), \ \mathrm{diag}(Z) = e, \ Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{full}})$$

It was demonstrated in [168] that using $\mathcal{M}$ and $\mathcal{LS}$ in the semidefinite relaxation pays off for several ordering problems including the (SRFLP) in practice. But for the (kPROP) we will refrain from using $\mathcal{M} \cap \mathcal{LS}$ due to several reasons that we will be discussed in Section 10.5 in detail. Instead we will work with the following SDP relaxation

$$\min \{ \langle C_d, Z \rangle : \ Y \text{ satisfies (10.8) and (10.15)}, \ Z \in \mathcal{DV}, \ \mathrm{diag}(Z) = e, \ Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{cheap}})$$

10.4 On Solving SDP Relaxations

The core of our SDP approach is to solve our SDP relaxation ($\texttt{SDP}_{\texttt{cheap}}$), using the bundle method in conjunction with interior point methods. The resulting fractional solutions constitute lower bounds for the exact SDP formulation of the (kPROP). By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe feasible layouts of the departments. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum. We will discuss these two steps in more detail in the following.

10.4.1 Computing Lower Bounds

Looking at the constraint classes and their sizes in the relaxation ($\texttt{SDP}_{\texttt{cheap}}$), it is clear that explicitly maintaining $O(n^3)$ or more constraints is not an attractive option. We therefore consider an approach originally suggested in [103], which was applied to the max cut problem [262] and several ordering problems [62, 168], and adapt it for the (kPROP). Initially, we only aim at explicitly ensuring the constraints $\mathrm{diag}(Z) = e$ and $Z \succcurlyeq 0$, which can be achieved with standard interior point methods, see, e.g. [146].

All other constraints are handled through Lagrangian duality in the objective function f. Thus the objective function f becomes non-smooth. The bundle method [103] iteratively evaluates f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to solving an SDP with the constraints $\operatorname{diag}(Z) = e$ and $Z \succcurlyeq 0$ that can be solved efficiently by using again interior point methods. Finally we obtain an approximate minimizer of f that is guaranteed to yield a lower bound to the optimal solution of $(\texttt{SDP}_{\texttt{cheap}})$. Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations that are responsible for more than 95% of the required running time to control the overall computational effort. This limitation of the number of function evaluations leaves some room for further incremental improvement. We next describe how a feasible layout can be obtained from a solution to $(\texttt{SDP}_{\texttt{cheap}})$.

10.4.2 Obtaining Feasible Layouts

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans-Williamson [121] to the fractional solution of the SDP relaxation. We take the resulting vector $\overline{y}$ and flip the signs of some of its entries to make it feasible with respect to (10.15) and the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1. \tag{10.18}$$

that are well-known [296, 306] to ensure feasible orderings of n elements. Computational experiments demonstrated that repair strategies of this type are not as critical as one might assume. For example, in multi-level crossing minimization this SDP rounding heuristic clearly dominates traditional heuristic approaches [64].

Let us give a more detailed description of the implementation of our heuristic. We consider a vector y' that encodes a feasible layout of the departments in all rows. The algorithm stops after 100 executions of step 2. (Note that before the 51st execution of step 2, we perform step 1 again. As step 1 is quite expensive, we refrain from executing it too often.)

1. Let Y'' be the current primal (fractional) solution of $(\texttt{SDP}_{\texttt{full}})$ (or some other semidefinite relaxation) obtained by the bundle method or an interior-point solver. Compute the convex combination $R := \lambda(y'y'^{\top}) + (1-\lambda)Y''$ using a randomly generated $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^{\top}$ of R.

2. Apply Goemans-Williamson hyperplane rounding to D and obtain a $-1/+1$ vector $\overline{w}$ (cf. [262]).

3. Compute the induced objective value $z(\overline{y}) := \begin{pmatrix} 1 \\ \overline{y} \end{pmatrix}^{\top} C_d \begin{pmatrix} 1 \\ \overline{y} \end{pmatrix}$. If $z(\overline{y}) \geq z(y')$: go to step 2.

4. If $\overline{y}$ satisfies (10.15) and (10.18): set $y' := \overline{y}$ and go to 2. Else: modify $\overline{y}$ by first changing the signs of one of three variables in all violated 3-cycle inequalities, afterwards flipping signs to satisfy (10.15) and go to step 3.

The final y' is the heuristic solution. If the duality gap is not closed after the heuristic, we continue approximating $(\texttt{SDP}_{\texttt{cheap}})$ with the help of the bundle method and then retry the heuristic (retaining the last vector y').

10.5 Computational Experiments

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. We use the (PROP) instances from [12] and define new (kPROP) instances by adapting (SRFLP) and (PROP) instances from the literature. For each instance considered, our computational objective is to obtain the best possible solution for a given assignment of the departments to 2 – 5 rows. This demonstrates that our relaxations and methodology are in principle applicable to (kPROP) instances with any given number of rows. All the instances can be downloaded from http://anjos.mgi.polymtl.ca/flplib. Let us finally mention that we can round the lower bound $\langle C, Z\rangle$ to the nearest integer because 0.5 can only occur in the constant term. First we review the best known practical results for the (SRFLP) in Subsection 10.5.1 and then we explain our algorithmic strategies for the (PROP) in detail in Subsection 10.5.2. Finally we expand our computational results to the case of 3 and more rows in Subsection 10.5.3.

10.5.1 Review: (SRFLP) Instances

Let us start with giving the characteristics and the optimal (SRFLP) solutions of the instances considered in our computational study in Table 10.1. All results obtained are provided by the SDP approach [169], which is the strongest to date, applied to ($\mathtt{SDP}_{\mathtt{standard}}$) on our machine. Its main features are summarized in Subsection 10.2.2. Using a direct extension of the (SRFLP) approach to obtain the results of the following two subsections allows a fair comparison of the computational (SRFLP) and (kPROP) results.

Instance	Source	Size	SRFLP			
		(n)	Best lower bound	Best layout	Gap (%)	Time (sec) [169]
S11	[283]	11	6933.5		0	1
P15	[6]	15	6305		0	20
P16_a	[12]	16	14829		0	34
P16_b	[12]	16	11878.5		0	34
P17	[7]	17	9254		0	35
P18	[7]	18	10650.5		0	33
H_20	[151]	20	15549		0	54
P20_a	[12]	20	24180.5		0	1:03
P20_b	[12]	20	25218		0	1:07
P21_a	[12]	21	13432.5		0	1:13
P21_b	[12]	21	22964		0	1:21
P22_a	[12]	22	17064		0	1:00
P22_b	[12]	22	29770		0	1:07
P23_a	[12]	23	19589		0	1:14
P23_b	[12]	23	30257.5		0	1:06
N25_05	[20]	25	15623		0	3:31
H_30	[151]	30	44965		0	14:17
N30_05	[20]	30	115268		0	18:30
Am33_03	[8]	33	69942.5		0	36:33
Am35_03	[8]	35	69002.5		0	53:14
ste36.5	[21]	36	91651.5		0	17:58
N40_5	[169]	40	103009		0	2:20:09
sko42-5	[21]	42	248238.5		0	1:08:42
sko49-5	[21]	49	666130	666143	0.002	9:30:22
sko56-5	[21]	56	591915.5	592335.5	0.07	17:46:46
AKV-60-05	[22]	60	318792	318805	0.004	12:39:37
sko64-5	[21]	64	501059.5	502063.5	0.20	13:53:04
AKV-70-05	[22]	70	4213774.5	4218002.5	0.10	28:16:05
sko72-5	[21]	72	426224.5	430288.5	0.95	31:39:43
AKV-75-05	[22]	75	1786154	1791469	0.30	41:10:37
AKV-80-05	[22]	80	1585491	1590847	0.34	58:30:30
sko81-5	[21]	81	1293905	1311166	1.33	58:59:28
sko100-5	[21]	100	1021584.5	1040929.5	1.89	201:29:27

Table 10.1: Characteristics and optimal (SRFLP) results for instances with between 11 and 100 departments. The bundle method is restricted to 500 function evaluations for $42 \leq n \leq 56$ and 250 function evaluations for $n \geq 60$. The running times are given in sec or min:sec or in h:min:sec respectively.

10.5.2 (PROP) Instances

We started with computing the exact relaxation values for small and medium sized (PROP) instances. Among n departments, suppose that there are t departments with some characteristic in common so that they should be arranged along one row, leaving the remaining departments to be arranged on a parallel row. For reasons of efficiency we used 10 function evaluations of the bundle method applied to ($\mathtt{SDP}_{\mathtt{cheap}}$) to obtain an initial set of constraints to add to the relaxation ($\mathtt{SDP}_{\mathtt{basic}}$). We then solved the resulting relaxation using Sedumi [289]; added all violated inequality constraints from ($\mathtt{SDP}_{\mathtt{cheap}}$); solved again using Sedumi; and repeated this process until no more violations were found. We also tried to solve ($\mathtt{SDP}_{\mathtt{cheap}}$) directly but the running times were at least one order of magnitude slower. Additionally we tried to solve ($\mathtt{SDP}_{\mathtt{full}}$) instead of ($\mathtt{SDP}_{\mathtt{cheap}}$) which resulted in slightly improved lower bounds but tremendously larger running times because many of the $O(n^6)$ triangle inequalities and $O(n^5)$ LS-cuts are active at the optimum. E.g. even for the smallest instance *S11* with only 11 departments and an optimal solution value of 3895.5 we need more than 10 minutes to compute ($\mathtt{SDP}_{\mathtt{full}}$), we have to consider 2745 constraints and the lower bound is 3600.5. For comparison we need only 1 second to compute ($\mathtt{SDP}_{\mathtt{cheap}}$) as we have to consider only 113 constraints and the lower bound obtained is 3563.5. The results obtained are summarized in Table 10.2, where the optimal solutions are provided by [12].

Instance	$t = \lfloor n/2 \rfloor$				$t = \lfloor n/3 \rfloor$			
	Lower bound	Optimal solution	Gap (%)	Time (sec)	Lower bound	Optimal solution	Gap (%)	Time (sec)
S11	3563.5	3895.5	9.3	0.6	5336.5	5404.5	1.3	1.3
P15	3269	3435	5.1	5.7	3651	3754	2.8	5.7
P16_a	7326	7630	4.1	9.8	9006	9813	9.0	10.3
P16_b	6039.5	6239.5	3.3	8.0	8579.5	9091.5	6.0	7.4
P20_a	11707.5	12609.5	7.7	30.5	14702.5	15874.5	8.0	44.1
P20_b	12169	12936	6.3	34.3	17903	19167	7.1	25.2
P21_a	6582.5	7006.5	6.4	100.2	8466.5	9141.5	8.0	78.8
P21_b	11105	11705	5.4	101.1	12838	13887	8.2	89.9
P22_a	8348	8874	6.3	99.7	11347	12238	7.9	104.9
P22_b	14673	15714	7.1	160.7	17590	19183	9.1	179.5
P23_a	9599	10242	6.7	148.4	13416	14294	6.6	145.4
P23_b	14884.5	15802.5	6.2	157.3	19936.5	21116.5	5.9	253.1
Instance	$t = \lfloor n/4 \rfloor$				$t = \lfloor n/5 \rfloor$			
	Lower bound	Optimal solution	Gap (%)	Time (sec)	Lower bound	Optimal solution	Gap (%)	Time (sec)
S11	5840.5	5852.5	0.2	1.1	5840.5	5852.5	0.2	1.1
P15	4445	4537	2.1	2.9	4445	4537	2.1	2.9
P16_a	10999	11409	3.7	6.9	11958	12279	2.7	6.6
P16_b	9392.5	9636.5	2.6	11.1	11031.5	11256.5	2.0	8.0
P20_a	17372.5	18185.5	4.7	65.3	20618.5	21215.5	2.9	37.0
P20_b	21784	22801	4.7	35.1	23148	23902	3.3	46.7
P21_a	11243.5	11765.5	4.7	66.1	11959.5	12385.5	3.6	94.1
P21_b	17862	18564	4.0	62.3	20489	20825	1.6	63.9
P22_a	14730	15385	4.4	148.0	15617	16114	3.2	89.4
P22_b	22859	23534	3.0	99.9	24702	25044	1.4	122.3
P23_a	17085	17812	4.3	139.2	18100	18619	2.9	220.8
P23_b	25427.5	26004.5	2.3	168.6	29395.5	29892.5	1.7	200.0

Table 10.2: (PROP) results for ($\mathtt{SDP}_{\mathtt{cheap}}$) and given row assignments using the bundle method in conjunction with Sedumi. Among n departments, t departments are arranged along one row, leaving the remaining departments to be arranged on a parallel row.

While the computing time clearly grows with the instance size, the gaps seem to be independent of the number of departments considered. However the size of t has an obvious influence on the gaps: The more departments are located in the same row, the smaller the gaps are in average, i.e. (PROP) instances that are "similar" to (SRFLP) instances are easier. This finding perfectly falls into place with our argument above that the (SRFLP) is easier to solve than the (PROP). Contrary to that the MIP approach by Amaral [12] that provides the optimal solution for instances with up to 23 departments works better on balanced instances, i.e. on a machine that is similar to our computer[3] his approach needs in average around 17.5 hours for instances with 23 departments and $t = \lfloor n/2 \rfloor$ and in average around 96 hours for instances with 23 departments and $t = \lfloor n/5 \rfloor$. In the following we will show that our SDP algorithm, in contrast to the MIP approach, can also be used the obtain reasonably tight lower bounds for (PROP) instances with up to 100 departments.

As a first experiment in that direction we generate row assignments of similar row lengths that are the most difficult ones for our approach (for details see Table 10.2). To do so we select the row assignments using the following simple heuristic: We first randomly assign 25% of the departments to each of the two rows; then the remaining 50% of the departments are added one at a time by taking the longest remaining department and adding it to the shorter row. Such balanced row assignments are often of interest in the design of layouts in practice, see e.g. [197]. Our heuristic quickly yields assignments for which the total row lengths are very close; see the second-to-last column of Table 10.4. We summarize the results averaged over 10 row assignments selected by our heuristic in Table 10.3. We used the same algorithmic approach as described above for the lower bound computation. The upper bounds are provided by the heuristic described in Subsection 10.4.2. Additionally we state the average number of inequalities of ($\text{SDP}_{\text{cheap}}$) that we considered when obtaining the optimal solution.

Instance	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average number of inequalities	Average time
P17	4501.5	4722	2.68	10.05	5.82	265.7	41
P18	5153	5503.5	3.85	11.51	8.36	298.6	1:07
H_20	7520	8046	4.97	10.86	7.70	400.7	4:03
N25_05	7385	7986	5.62	11.56	8.79	659.1	23:06
H_30	21028	22848	6.64	13.74	9.63	1057.6	2:12:30
N30_05	53854	58221	5.89	13.46	9.27	1201.3	2:37:19
Am33_03	32847	35904.5	7.59	13.88	9.31	1580.7	5:52:13
Am35_03	32142	35273	8.64	12.89	9.74	1666.3	10:27:58
ste36.5	44786.5	46794.5	1.36	5.54	3.66	1633.6	12:40:15

Table 10.3: (PROP) results for ($\text{SDP}_{\text{cheap}}$) and given row assignments using the bundle method in conjunction with Sedumi. The results are averages over 10 row assignments. For the heuristically selected row assignments the total row lengths are very close. The running times are given in sec or min:sec or in h:min:sec respectively.

We have to call Sedumi 3 times on average to solve ($\text{SDP}_{\text{cheap}}$) exactly. It is interesting to note that in all our experiments, the gap changes only marginally after the first call to Sedumi, i.e. lower bounds of nearly the same quality are already obtained after one third of the computing time. We assume that the slightly larger gaps compared to Table 10.2 are due to non-optimal upper bounds because the performance of our rounding heuristic depends of course on the quality of the fractional starting solutions provided. If these starting solutions are already off a few percent the obtained layouts are good but not optimal in general. Additionally we can observe that the growing size of the SDP matrices and the growing number of inequalities considered when obtaining the optimal solution result in rapidly growing running times for increasing n.

Hence for large instances with $n \geq 40$ departments we apply only the bundle method (without Sedumi) to ($\text{SDP}_{\text{cheap}}$). Also in this case the large number of violated LS-cuts has only little effect on the bound

[3]For exact numbers of the speed differences see http://www.cpubenchmark.net/.

quality but slows down the bundle computation significantly which altogether leads to a computationally better performance of our approach applied to (SDP$_{\texttt{cheap}}$) compared to (SDP$_{\texttt{full}}$). We report results for instances with up to 100 departments, again averaged over 10 balanced row assignments selected by our heuristic. The experiments quickly become very time consuming which is evidenced by the growth of the running times in Table 10.4 below, as well as in Table 10.1 for solving the simpler (SRFLP) relaxation. We restrict the bundle method to 125 function evaluations of the objective function f. This limitation of the number of function evaluations sacrifices some possible incremental improvement of the bounds. Table 10.4 summarizes the results we obtained.

Instance	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average difference of row lengths	Average time (sec)
P17	4435	4737	4.68	10.62	7.29	1.8	25
P18	5080	5462.5	5.09	14.32	9.63	1.0	32
H_20	7402	8149	8.54	12.40	10.03	2.0	48
N25_05	7254	7945	6.37	15.33	10.45	0.4	2:09
H_30	20659.5	22801	9.18	18.70	13.34	2.0	5:13
N30_05	52756.5	58425	7.29	13.55	10.45	1.8	5:10
Am33_03	32058	35958.5	10.45	20.41	15.39	1.6	9:14
Am35_03	31521	34794.5	8.77	18.48	14.83	1.2	12:00
ste36.5	41409.5	47259.5	7.14	19.94	12.91	1.0	13:28
N40_5	46877.5	55220	13.73	21.75	17.53	0.0	24:24
sko42-5	113606	127639.5	11.36	19.43	15.54	1.0	32:40
sko49-5	291004.5	349137	17.46	23.10	20.20	2.0	1:21:44
sko56-5	261686	306133.5	15.91	22.54	19.66	1.0	3:17:29
AKV-60-05	145702	171280	17.56	22.42	19.41	1.0	4:46:03
sko64-5	219646	261257.5	18.95	24.78	21.56	1.0	6:20:28
AKV-70-05	1861211	2196942.5	18.04	21.36	19.62	1.2	12:33:52
sko72-5	185496	222924.5	19.77	24.76	22.14	0.0	14:00:31
AKV-75-05	793712	946626	18.37	23.17	20.61	2.2	17:36:27
AKV-80-05	708327.5	836043	16.99	23.11	19.82	2.2	26:19:44
sko81-5	568646.5	691250	20.94	23.62	22.10	1.0	28:45:03
sko100-5	441133	552389.5	24.37	27.01	25.61	1.0	98:08:37

Table 10.4: (PROP) results for (SDP$_{\texttt{cheap}}$) and given row assignments using the bundle method restricted to 125 function evaluations. The results are averages over 10 row assignments. For the heuristically selected row assignments the total row lengths are very close. “Lower bound” gives the worst lower bound over the 10 instances and “Upper bound” states the best upper bound over the 10 instances. The running times are given in sec or min:sec or in h:min:sec respectively.

Comparing the results in Tables 10.3 and 10.4 shows that the lower bounds of the bundle method quickly get close to the exact (SDP$_{\texttt{cheap}}$) bounds even though the number of function evaluations is capped at 125. Furthermore, while the running times in Table 10.3 grow very quickly with the problem size, the computation times of the bundle method in Table 10.4 are not so strongly affected by the problem size. Hence this approach yields bounds competitive with the exact optimal value of (SDP$_{\texttt{cheap}}$) at only a fraction of the computational cost. For comparison the (SRFLP) results from Table 10.1 were obtained by the same algorithmic approach but using a different SDP relaxation ((SDP$_{\texttt{standard}}$) instead of (SDP$_{\texttt{cheap}}$)) and a higher number of function evaluations. We can observe that the number of function evaluations mainly determines the running times and that the size of the gaps clearly differs for (SRFLP) and (PROP) instances due to the different objective functions.

10.5.3 (kPROP) Instances with $k \geq 3$

Finally we aim to analyze the effect of an increased number of rows on our SDP approach. Hence we adapt the experiments from the previous subsection to (kPROP) instances with $3 \leq k \leq 5$. Let us start with the small and medium size instances from [12]. We use again the bundle method in conjunction with Sedumi applied to (SDP$_{\texttt{cheap}}$) because this approach clearly outperforms the other two variants discussed above (solving (SDP$_{\texttt{cheap}}$) directly or using (SDP$_{\texttt{full}}$)) due to the same reasons. We choose again t departments with some characteristic in common so that they should be arranged along the first row, leaving the remaining departments to be arranged equally on the other rows. We compare the two different objective functions discussed in Subsection 10.3.1, for an illustration see Figure 10.4. The results obtained are summarized in Table 10.5 for 3 rows and in Table 10.6 for 4 and 5 rows. The upper bounds are provided by the heuristic described in Subsection 10.4.2.

Instance	$k=3,\quad t=\lfloor n/3 \rfloor,\quad \langle C_d, Z\rangle$				$k=3,\quad t=\lfloor n/3 \rfloor,\quad \langle C_i, Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	2827.5	3127.5	10.6	1.1	4279.5	4323.5	1.0	1.2
P15	2134	2296	7.6	10.2	3383	3489	3.2	11.2
P16_a	4779	5234	9.5	10.7	7490	7939	6.0	9.4
P16_b	3937.5	4293.5	9.0	6.6	6107.5	6330.5	3.7	13.0
P20_a	7815.5	8608.5	10.2	91.6	12913.5	13348.5	3.4	48.4
P20_b	8282	9446	14.1	47.2	13678	14073	2.9	67.9
P21_a	4457.5	4976.5	11.6	116.6	7060.5	7491.5	6.1	101.1
P21_b	7580	8156	7.6	121.2	11942	12388	3.7	111.1
P22_a	5659	6479	14.5	219.5	9442	9829	4.1	175.6
P22_b	9612	10398	8.2	271.3	15931	16865	5.9	219.4
P23_a	6589	7545	14.5	307.8	10961	11383	3.9	223.2
P23_b	10267.5	12090.5	17.8	215.8	17531.5	18103.5	3.3	378.1
Instance	$k=3,\quad t=\lfloor n/4 \rfloor,\quad \langle C_d, Z\rangle$				$k=3,\quad t=\lfloor n/4 \rfloor,\quad \langle C_i, Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	4507.5	4700.5	4.3	1.3	5793.5	5851.5	1.0	1.7
P15	3893	4043	3.9	6.4	5374	5474	1.9	8.5
P16_a	6163	6639	7.7	10.7	9439	9615	2.0	16.1
P16_b	4872.5	5230.5	7.4	11.0	7459.5	7583.5	1.7	17.5
P20_a	9893	11482.5	16.1	61.0	17911.5	18186.5	1.5	55.8
P20_b	10456	11664	11.6	62.1	15491	16020	3.4	50.5
P21_a	5739.5	6434.5	12.1	135.8	8171.5	8405.5	2.9	126.8
P21_b	8868	10216	15.2	142.4	13364	13575	1.6	123.8
P22_a	8075	9044	12.0	160.1	11029	11317	2.6	196.6
P22_b	11768	13148	11.7	129.6	17572	18219	3.7	224.4
P23_a	9711	10779	11.0	337.7	12952	13434	3.7	263.4
P23_b	15838.5	16696.5	5.4	269.8	21236.5	22001.5	3.6	207.9
Instance	$k=3,\quad t=\lfloor n/5 \rfloor,\quad \langle C_d, Z\rangle$				$k=3,\quad t=\lfloor n/5 \rfloor,\quad \langle C_i, Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	4507.5	4700.5	4.3	1.3	5793.5	5851.5	1.0	1.7
P15	3893	4043	3.9	6.4	5374	5474	1.9	8.5
P16_a	9493	9746	2.7	13.8	12794	12845	0.4	12.7
P16_b	8922.5	9137.5	2.4	13.1	10873.5	10981.5	1.0	13.5
P20_a	5874.5	6762.5	15.1	108.2	11770.5	12082.5	2.7	68.5
P20_b	14693	15505	5.5	65.9	19237	19615	2.0	69.3
P21_a	7893.5	8606.5	9.0	106.4	10351.5	10482.5	1.3	114.8
P21_b	12068	12928	7.1	99.2	15982	16204	1.4	163.5
P22_a	10936	11669	6.7	203.2	13828	14100	2.0	259.2
P22_b	17445	18595	6.6	125.1	23147	23371	1.0	232.0
P23_a	13006	13696	5.3	202.8	16001	16351	2.2	311.4
P23_b	21200.5	22020.5	3.9	227.9	25683.5	26313.5	2.5	362.7

Table 10.5: (kPROP) results with $k = 3$ for (SDP$_3$) and given row assignments using the bundle method in conjunction with Sedumi. Among n departments, t departments are arranged along the first two rows, leaving the remaining departments to be arranged on row 3.

Again the computing time clearly grows with the instance size and the gaps are clearly influenced in the same way as above by t: The more departments are located in one row, the smaller the gaps are in average. The number of rows and the distance calculation for non-adjacent rows have no influence on the running time but a considerable effect on the gaps:

1. The gaps are essentially smaller for the objective function $\langle C_i, Z\rangle$ compared to $\langle C_d, Z\rangle$, i.e. C_d contains more quadratic terms in ordering variables and hence the corresponding SDP relaxation provides less tight lower bounds.

2. For the objective function $\langle C_i, Z\rangle$ the average gaps decrease for an increasing number of rows as the number of quadratic terms in orderings variables in C_i decreases.

3. For objective function $\langle C_d, Z\rangle$ the average gaps increase for an increasing number of rows as the

Instance	$k=4,\quad t=\lfloor n/4\rfloor,\quad \langle C_d,Z\rangle$				$k=4,\quad t=\lfloor n/4\rfloor,\quad \langle C_i,Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	2977.5	3156.5	6.0	1.6	4723.5	4733.5	0.2	1.5
P15	2101	2403	14.4	7.0	3754	3783	0.8	8.8
P16_a	3554	3896	9.6	11.8	7090	7191	1.4	14.0
P16_b	3026.5	3401.5	12.4	9.9	5898.5	6000.5	1.7	12.4
P20_a	6622.5	7756.5	17.1	68.3	13448.5	13524.5	0.6	103.5
P20_b	6131	7092	15.7	90.1	11973	12160	1.6	115.2
P21_a	3583.5	3942.5	10.0	138.4	7592.5	7831.5	3.2	107.4
P21_b	5687	6336	11.4	82.1	11832	11948	1.0	163.5
P22_a	4778	5675	18.8	128.2	10334	10477	1.4	166.1
P22_b	7412	8171	10.2	291.3	15777	15805	0.2	306.7
P23_a	5719	6751	18.1	272.6	12081	12334	2.1	324.1
P23_b	8559.5	10003.5	16.9	309.4	17200.5	17241.5	0.2	303.1
Instance	$k=4,\quad t=\lfloor n/5\rfloor,\quad \langle C_d,Z\rangle$				$k=4,\quad t=\lfloor n/5\rfloor,\quad \langle C_i,Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	2977.5	3156.5	6.0	1.6	4723.5	4733.5	0.2	1.5
P15	2101	2403	14.4	7.0	3754	3783	0.8	8.8
P16_a	5460	6022	10.3	11.3	9514	9630	1.2	13.9
P16_b	4330.5	4717.5	9.0	19.9	7239.5	7299.5	0.8	13.0
P20_a	4724.5	5210.5	10.3	95.9	10475.5	10532.5	0.6	107.4
P20_b	7764	8645	11.4	94.3	14885	15005	0.8	95.0
P21_a	4247.5	4888.5	15.1	96.9	8143.5	8181.5	0.5	186.7
P21_b	7034	7931	12.8	140.9	13664	13682	0.1	155.5
P22_a	6238	6959	11.6	169.6	11100	11136	0.3	209.2
P22_b	9434	10566	12.0	194.7	18186	18285	0.5	204.4
P23_a	7606	8382	10.2	241.9	12963	13054	0.7	434.0
P23_b	13562.5	14352.5	5.8	431.8	20962.5	21016.5	0.3	478.8
Instance	$k=5,\quad t=\lfloor n/5\rfloor,\quad \langle C_d,Z\rangle$				$k=5,\quad t=\lfloor n/5\rfloor,\quad \langle C_i,Z\rangle$			
	Lower bound	Upper bound	Gap (%)	Time (sec)	Lower bound	Upper bound	Gap (%)	Time (sec)
S11	1545.5	1742.5	12.8	1.3	3329.5	3329.5	0	0.7
P15	1500	1566	4.4	7.5	3193	3239	1.4	8.1
P16_a	2821	3181	12.8	9.4	7043	7073	0.4	13.4
P16_b	2396.5	2786.5	16.3	6.1	5817.5	5934.5	2.0	13.0
P20_a	4724.5	5210.5	10.3	95.9	10475.5	10532.5	0.6	107.4
P20_b	4898	5408	10.4	55.8	10996	11038	0.4	97.8
P21_a	2732.5	3038.5	11.2	147.3	6044.5	6076.5	0.5	155.9
P21_b	4534	5054	11.5	115.2	10366	10483	1.1	154.9
P22_a	3749	4403	17.5	213	8432	8438	0.1	220.0
P22_b	5923	6615	11.7	165.1	14180	14224	0.3	190.7
P23_a	4612	5293	14.8	199.3	10082	10104	0.2	308.3
P23_b	7075.5	8396.5	18.7	222.1	15035.5	15057.5	0.2	300.8

Table 10.6: (kPROP) results with $k = 4$ and $k = 5$ for (SDP$_3$) and given row assignments using the bundle method in conjunction with Sedumi. Among n departments, t departments are arranged along the first $k-1$ rows, leaving the remaining departments to be arranged on the last row.

number of quadratic terms in orderings variables in C_d increases.

We have seen in the previous subsection that using only the bundle method (without Sedumi) is the preferable approach for larger (PROP) instances. The same holds true for (kPROP) instances with $k \geq 3$. Hence we solely extend the experiments from Table 10.4 for 3 and 5 rows respectively. Hence we again generate row assignments of similar row lengths. To do so we select the row assignments using the following simple heuristic: We first randomly assign $\frac{50}{k}$% of the departments to each of the k rows; then the remaining 50% of the departments are added one at a time by taking the longest remaining department and adding it to the shortest row. This heuristic quickly yields assignments for which the total row lengths are very close; see the second-to-last column of Table 10.7. We again restrict the bundle method to 125 function evaluations of the objective function f which sacrifices some possible incremental improvement of the bounds. We summarize the results averaged over 10 row assignments selected by our heuristic in Table 10.7. The upper bounds are provided by the heuristic described in Subsection 10.4.2.

The number of rows and the different objective functions influence the computational results for large (kPROP) instances in the same way as above for small and medium (kPROP) instances:

- The number of rows and the distance calculation for non-adjacent rows have no effect on the running time.
- The gaps are essentially smaller for the objective function $\langle C_i, Z\rangle$ compared to $\langle C_d, Z\rangle$.

Instance	$k = 3, \quad \langle C_d, Z\rangle$							$k = 3, \quad \langle C_i, Z\rangle$					
	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average difference of row lengths	Average time	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average time
P17	3083.5	3417	8.75	19.99	14.24	1.3	30	4491.5	4715	2.62	6.73	4.31	26
P18	3513.5	3842.5	9.36	18.24	13.63	1.7	37	5113.5	5299.5	1.72	7.44	4.45	32
H_20	4950	5718	10.48	18.04	13.74	1.5	55	7150.5	7731	1.77	8.13	4.48	51
N25_05	4737	5507	11.11	17.69	15.07	0.7	2:24	7282.5	7912	1.08	8.64	4.07	2:15
H_30	13632	15408	13.03	20.82	17.31	1.5	6:08	21705.5	23265	4.38	7.18	5.66	5:44
N30_05	35034.5	39536	10.04	21.98	16.11	2.2	5:38	54644	59798	2.97	9.43	5.55	5:35
Am33_03	21187.5	24958.5	15.21	22.31	19.57	1.5	9:31	35025	37088.5	3.33	9.77	5.57	9:47
Am35_03	20515.5	23552.5	13.92	21.17	17.69	1.0	12:38	33972	36098.5	3.56	6.96	5.24	13:12
ste36.5	29326	35684.5	16.57	31.71	23.57	1.5	14:19	43378.5	47433.5	2.13	9.53	5.02	14:20
N40_5	29726.5	37388	19.54	28.58	24.83	0.7	25:54	54514.5	59497	4.79	9.61	6.42	26:59
sko42-5	73870	86835.5	16.70	24.68	21.28	0.8	32:48	122741.5	130627.5	3.98	8.80	6.68	35:37
sko49-5	187774.5	231802	23.45	27.51	25.16	2.4	1:26:00	319754	350438	7.72	10.95	9.05	1:27:53
sko56-5	169046	209771.5	23.18	27.90	25.47	0.7	3:10:14	280373	307337.5	6.15	9.62	7.83	3:10:54
AKV-60-05	93118.5	118576	25.25	34.64	28.46	1.1	4:54:59	159002	173876	4.42	9.38	7.38	5:11:26
sko64-5	140187	175694.5	25.12	30.82	27.59	0.7	7:03:37	236929	261468.5	7.87	12.20	9.80	7:37:32
AKV-70-05	1179882	1489955.5	25.27	30.37	27.70	1.4	12:25:00	2015403	2236313.5	5.73	12.63	8.78	13:44:50

Instance	$k = 5, \quad \langle C_d, Z\rangle$							$k = 5, \quad \langle C_i, Z\rangle$					
	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average difference of row lengths	Average time	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average time
P17	1883	2125	9.61	16.73	13.29	1.9	32	4113	4142	0.09	1.15	0.62	23
P18	2095	2465.5	8.52	19.45	13.76	1.9	40	4553	4562.5	0.11	1.50	0.64	29
H_20	3190	3641	12.64	20.45	16.00	1.4	1:01	6648	6668	0.21	1.74	0.80	45
N25_05	2885	3301	12.19	21.90	17.16	0.9	2:40	6700	6760	0.13	1.02	0.63	2:05
H_30	8078	9535	15.86	24.26	20.10	1.3	6:47	19873.5	20169	0.41	1.66	1.07	5:32
N30_05	21021	23970	10.97	26.61	20.48	2.2	6:44	51491.5	52171	0.44	1.32	0.85	5:23
Am33_03	13020.5	15468.5	15.69	27.93	22.84	1.5	11:08	29509	29828.5	0.49	1.17	0.71	9:34
Am35_03	12362.5	14753.5	19.34	27.81	23.34	1.1	14:41	30070	30561.5	0.71	1.83	1.24	14:29
ste36.5	17727	23431.5	19.39	42.88	29.70	1.4	14:44	57881.5	58551.5	0.74	2.72	1.57	15:33
N40_5	18871	23351	21.72	30.23	26.45	0.6	26:29	49102	49553.5	0.54	1.40	1.02	27:32
sko42-5	43394.5	54402.5	23.63	30.09	26.80	1.2	33:57	108072.5	109073.5	0.46	1.46	0.92	35:47
sko49-5	109725.5	142625	26.62	31.72	29.20	2.7	1:24:15	276888.5	279585	0.88	1.85	1.27	1:31:42
sko56-5	98792.5	128605.5	28.73	33.72	30.97	1.0	3:05:20	252663.5	254116.5	0.53	1.19	0.75	3:34:49
AKV-60-05	56122	73434	28.74	38.78	33.12	1.2	5:00:21	139695	140687	0.31	1.45	0.67	5:33:54
sko64-5	81613	106964.5	30.54	37.18	34.34	0.8	7:06:06	212293	214853.5	0.82	1.56	1.27	7:45:49
AKV-70-05	696599	913425.5	29.02	34.13	32.39	1.6	12:18:10	1704224.5	1717412.5	0.39	1.04	0.66	14:56:07

Table 10.7: (kPROP) results with $k = 3$ and $k = 5$ for (SDP_{cheap}) and given row assignments using the bundle method. The results are averages over 10 row assignments. For the heuristically selected row assignments the total row lengths are very close. "Lower bound" gives the worst lower bound over the 10 instances and "Upper bound" states the best upper bound over the 10 instances. The running times are given in sec or min:sec or in h:min:sec respectively.

- For the objective function $\langle C_i, Z\rangle$ the average gaps decrease for an increasing number of rows.
- For the objective function $\langle C_d, Z\rangle$ the average gaps increase for an increasing number of rows.

10.6 Conclusion

In this paper we proposed a new semidefinite programming approach for the k-Parallel Row Ordering Problem that extends the semidefinite programming approach for the Single-Row Facility Layout Problem by modelling inter-row distances as products of ordering variables. Our computational results show that our approach provides high-quality global bounds in reasonable time for instances with up to 100 departments and 5 rows.

The next step in extending our approach are the consideration of further valid inequalities in our SDP relaxations, the incorporation of spacing within the rows in the optimization process and the use of the SDP approach within a suitable enumeration scheme to globally optimize instances of double-row and multi-row layout. For a first step in that direction we refer to Hungerländer and Anjos [164].

Chapter 11

An SDP Approach to Space-Free Multi-Row Facility Layout

Authors: Philipp Hungerländer and Miguel F. Anjos

Abstract: Facility layout is a well-known operations research problem that arises in numerous areas of applications. The multi-row facility layout problem is concerned with placing departments along one or several rows so as to optimize objectives such as material handling and space usage. The particular cases of the single-row and double-row facility layout problems are of special interest in the manufacturing context where materials flow between stations located along linear corridors. Significant progress has been made in recent years on solving single-row problems to global optimality using semidefinite optimization models. The contribution of this paper is the extension of the semidefinite programming approach to the special case of multi-row layout in which all the rows have a common left origin and no empty space is allowed between departments. We call this special case the space-free multi-row facility layout problem. Although this problem may seem overly restrictive, it is a relevant problem in several contexts such as in spine layout design. Our computational results show that the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for space-free double-row instances with up to 15 departments and for space-free multi-row instances with up to 5 rows and 11 departments.

Keywords: Facilities planning and design; Flexible manufacturing systems; Cell layout; Semidefinite Programming; Global Optimization

11.1 Introduction

Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function. This is a well-known operations research problem that arises in different areas of applications. For example, in manufacturing systems, the placement of machines that form a production line inside a plant is a layout problem in which one wishes to minimize the total material flow cost. Another example arises in the design of Very Large Scale Integration (VLSI) circuits in electrical engineering. The objective of VLSI floorplanning is to arrange a set of rectangular modules on a rectangular chip area so that performance is optimized; this is a particular version of facility layout. In general, the objective function may reflect transportation costs, the construction cost of a material-handling system, the costs of laying communication wiring, or simply adjacency preferences among departments. Some facility layout problems are dynamic, meaning that the layout may have to change over time (e.g. due to expected changes in the production process).

The variety of applications means that facility layout encompasses a broad class of optimization problems. The survey paper [222] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of departments within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the departments sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem, such as the layout of machines along a production line. This paper is concerned with finding global upper and lower bounds for one such type of structured instances, namely the multi-row facility layout problem (MRFLP) in which the departments are to be placed so as to form one or more parallel rows.

Most facility layout problems have a strong combinatorial nature and turn out to be NP-hard. As such, numerous heuristic and metaheuristic approaches have been proposed for the various categories of problems, see e.g. [148]. However, few methods exist that provide global optimal solutions, or at least a measure of nearness to global optimality, for large instances of layout problems. One exception is the case of the single-row facility layout problem (SRFLP). This problem, sometimes called the one-dimensional space allocation problem [252], consists of finding the optimal location of rectangular departments next to each other along one row so as to minimize the total weighted sum of the center-to-center distances between all pairs of departments. It arises for example as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle (AGV) travelling in both directions on a straight-line path. The SRFLP has interesting connections to other combinatorial optimization problems such as the maximum-cut problem, the quadratic linear ordering problem, and the linear arrangement problem. We refer the reader to [19] for more details.

Global optimization approaches for the SRFLP are based on relaxations of integer linear programming (ILP) or semidefinite programming (SDP) formulations. Semidefinite programming is the extension of linear programming (LP) from the cone of non-negative real vectors to the cone of symmetric positive semidefinite matrices. SDP includes LP as a special case, namely when all the matrices involved are diagonal. Several excellent solvers for SDP are now available. We refer the reader to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where SDP has had a major impact.

The SRFLP is a special case of the more general MRFLP. Another particular case of the MRFLP is the double-row facility layout problem (DRFLP). The DRFLP is a natural extension of the SRFLP in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel. This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the DRFLP is especially relevant for real-world applications. A specific example of the application of the DRFLP is in spine layout design. Spine layouts, introduced by Tompkins [295], require departments to be located along both sides of specified corridors along which all the traffic between departments takes place. Although in general some spacing is allowed, layouts with no spacing are much preferable since spacing often translates into higher construction costs for the facility.

Somewhat surprisingly, the MRFLP and DRFLP have received only limited attention in the literature. In the 1980s Heragu and Kusiak [150, 151] proposed a non-linear programming model and obtained locally optimal solutions to SRFLPs and DRFLPs. Recently Chung and Tanchoco [72] (see also [310]) focused exclusively on the DRFLP and proposed a mixed-integer LP (MILP) formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Amaral [11] proposed an improved MILP formulation that allowed him to solve instances with up to 12 departments to optimality. Algorithms for spine layout design have been proposed, see e.g. [197].

In this paper, as a first step towards developing an SDP-based global optimization approach to the general MRFLP, we extend the SDP-based methodology for SRFLP originally proposed in [22] to the particular version of the MRFLP in which all the rows have a common left origin and no empty space is allowed between departments. We call it the *space-free multi-row facility layout problem (SF-MRFLP)*. This is an interesting special case, not only because it is relevant for the application of spine layout design, but also because it captures much of the inherent difficulty of layout problems. This difficulty is characterized by the large number of pairwise interactions between departments reflected by a high density of the objective function matrix in the SDP relaxation: generally speaking, more pairwise interactions lead to SDP approaches outperforming (M)ILP. Our computational results show that the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for space-free double-row instances with up to 15 departments and for space-free multi-row instances with up to 5 rows and 11 departments. In particular we show that our method is competitive with a MILP formulation for the SF-MRFLP restricted to 2 rows that was proposed in [10] very recently.

This paper is structured as follows. In Section 11.2 we introduce the SRFLP and discuss the issues to address in extending the SDP models from single-row to multi-row problems. In Section 11.3 we formally state the SF-MRFLP and present new formulations of it. The SDP relaxations are presented in Section 11.4 and a heuristic to obtain feasible layouts from the solutions of the SDP relaxations is proposed in Section 11.5. Computational results demonstrating the strength and potential of our SDP framework are presented in Section 11.6. Finally, conclusions and future research directions are given in Section 11.7.

11.2 From Single-Row to Multi-Row Layout

Our starting point are the most successful models for SRFLP. To introduce these, let $\pi = (\pi_1, \dots, \pi_n)$ denote a permutation of the indices $[n] := \{1, 2, \dots, n\}$ of the departments, so that the leftmost department is π_1, the department to the right of it is π_2, and so on, with π_n being the last department in the arrangement. Given a permutation π and two distinct departments i and j (and assuming that there is no space between the departments), the center-to-center distance between i and j is $\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j$, where ℓ_i is the positive length of department i, and $D_\pi(i,j)$ denotes the sum of the lengths of the departments between i and j in the ordering defined by π. Solving the SRFLP consists of finding a permutation of $[n]$ that minimizes the weighted sum of the distances between all pairs of departments. In other words, the problem is:

$$\min_{\pi \in \Pi_n} \sum_{i<j} c_{ij} \left[\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j \right] \tag{SRFLP}$$

where c_{ij} is the connectivity between departments i and j, and Π_n denotes the set of all permutations of $[n]$. Since the lengths of the departments are constant, it is clear that the crux of the problem is to minimize $\sum_{i<j} c_{ij} D_\pi(i,j)$ over all permutations $\pi \in \Pi_n$.

The key information to express the quantity $D_\pi(i,j)$ can be encoded using *betweenness variables*. These are $\binom{n}{3}$ binary variables $\zeta_{ijk},\ i,j,k \in [n],\ i < j, i \neq k \neq j$ defined by:

$$\zeta_{ijk} = \begin{cases} 1, & \text{if department } k \text{ lies between departments } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

Not all possible combinations of values of the variables ζ_{ijk} correspond to permutations of $[n]$. Specifically, given three departments i, j, and k, exactly one of them must be located between the other two. This fact is expressed using the following equations:

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i,j,k \in [n],\ i < j < k. \tag{11.1}$$

Anjos and Yen [21] show that these equations precisely characterize the combinations of values of the variables ζ_{ijk} that describe permutations of $[n]$.

We collect all the betweenness variables in a vector ζ. Since every permutation $\pi \in \Pi_n$ can be encoded as one such vector ζ, we express the center-to-center distance between departments i and j as

$$D_\pi(i,j) = \sum_{k\in[n],\ i\neq k\neq j} \ell_k \zeta_{ijk}, \quad i,j \in [n],\ i<j,$$

and hence express SRFLP as

$$\sum_{i,j\in[n],\ i<j} \frac{c_{ij}}{2}(\ell_i+\ell_j) + \min_{\zeta\in\mathcal{P}^n_{Btw}} \sum_{i,j\in[n],\ i<j} c_{ij}\left(\sum_{k\in[n],\ i\neq k\neq j} \ell_k\zeta_{ijk}\right),$$

where $\mathcal{P}^n_{Btw}$ is the betweenness polytope:

$$\mathcal{P}^n_{Btw} := \operatorname{conv}\left\{\zeta \ :\ \zeta\in\{0,1\}^{\binom{n}{3}} \text{ and } \zeta_{ijk}+\zeta_{ikj}+\zeta_{jki}=1, \quad i,j,k\in[n],\ i<j<k\right\}.$$

Sanjeevi and Kianfar [268] show that the equations (11.1) describe the smallest linear subspace that contains $\mathcal{P}^n_{Btw}$. Buchheim et al. [48] proved a similar result in the context of quadratic linear ordering problems.

The betweenness polytope is the structure common to most of the recent LP and SDP relaxations for the SRFLP. One visible difference between these two approaches is that SDP approaches define the binary variables in terms of $\{-1,1\}$ instead of $\{0,1\}$. In fact this makes no difference: Helmberg [138] proved that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of binary problems in such a way that the resulting bounds remain the same and structural properties are preserved.

The SDP relaxation proposed in Anjos et al. [22] was used by Anjos and Vannelli [20] to solve SRFLPs with up to 30 departments to global optimality. This was improved on by Amaral [8] who used an LP relaxation of $\mathcal{P}^n_{Btw}$ to solve instances with up to 35 departments. Global lower bounds for very large instances with up to 100 departments were provided by Anjos and Yen [21] using a modified SDP relaxation. More recently, Hungerländer and Rendl [169] provided global optimal solutions for instances with up to 42 departments, and tighter bounds than the Anjos-Yen relaxation for instances with up to 100 departments. The relaxation in [169] achieved the best practical performance to date among all approaches for the SRFLP.

While these approaches work extremely well for the SRFLP, none of them can be applied directly to the MRFLP. This is because there are three modeling issues that arise in the MRFLP but not in the SRFLP:

1. Assigning each department to exactly one row;
2. Expressing the center-to-center distance between departments assigned to different rows;
3. Handling the possibility of empty space between departments.

The fundamental limitation is that the betweenness variables used for the state-of-the-art LP and SDP relaxations are not sufficient to capture these issues.

Models for the DRFLP have been proposed by two groups of authors. Heragu and Kusiak [150] proposed a non-linear programming model that provides locally optimal solutions to SRFLPs and DRFLPs, while Chung and Tanchoco [72] (see also [310]) used a MILP formulation for the DRFLP. The latter approach is only able to provide global optimal solutions for DRFLPs of small sizes. For larger instances, locally optimal solutions were obtained by using the MILP formulation in conjunction with heuristics for assigning the departments to the rows in advance.

This paper proposes an SDP-based model that can provide global upper and lower bounds, and in some cases global optimal solutions, for the SF-MRFLP. The proposed model extends the tight SDP relaxations in [22, 169] and the algorithmic approaches in [169] to the SF-MRFLP. Our SDP relaxations further assume that each department is already assigned to one of the rows but this restriction is overcome by optimizing the relaxations over all row assignments or, for large instances, over a chosen subset of assignments. The issue of allowing empty space between departments will be addressed in future research.

11.3 Formulations for Space-Free Multi-Row Layout

An instance of the SF-MRFLP consists of n one-dimensional departments with given positive lengths $\ell_1, \dots, \ell_n$, pairwise connectivities c_{ij} and a function $r: [n] \rightarrow \mathcal{R}$ that assigns each department to one of the m rows $\mathcal{R} := \{1, \dots, m\}$. The objective is to find permutations $\pi^1 \in \Pi^1, \dots, \pi^m \in \Pi^m$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\Pi^1 \times \ldots \times \Pi^m} \sum_{i,j \in [n], i<j} c_{ij} z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}, \tag{SF-MRFLP}$$

where $\Pi^1 \times \ldots \times \Pi^m$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}$ denotes the horizontal distance between the centroids of departments i and j in the layout $\pi^1 \times \ldots \times \pi^m$.

We define the m-row betweenness polytope

$$\mathcal{P}_{Btw}^{n,m} := \text{conv}\left\{\zeta \;:\; \zeta \text{ represents orderings of the } n \text{ departments on the } m \text{ rows }\right\},$$

and introduce the binary ordering variables $y_{ij},\ i,j \in [n],\ r(i) = r(j),\ i < j$:

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise.} \end{cases} \tag{11.2}$$

We can express the betweenness variables ζ as quadratic terms in ordering variables:

$$\zeta_{ijk} = \frac{1 - y_{ik}y_{jk}}{2}, \qquad \zeta_{ikj} = \frac{1 + y_{ij}y_{jk}}{2}, \qquad \zeta_{jki} = \frac{1 - y_{ij}y_{ik}}{2}, \tag{11.3}$$

for $i,j,k \in [n],\ r(i) = r(j) = r(k),\ i < j < k$, and thus rewrite (11.1) (generalized for m rows) as

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k \in [n],\ r(i) = r(j) = r(k),\ i < j < k. \tag{11.4}$$

It was shown in [64] that (11.4) describes the smallest linear subspace that contains the multi-level quadratic ordering polytope

$$\mathcal{P}_{MQO} := \text{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^{\top} \;:\; y_{ij} \in \{-1, 1\},\ y \text{ satisfies (11.4)} \right\},$$

where y is a vector collecting the ordering variables. We can also use y to express the center-to-center distances of pairs of departments $i,j \in [n],\ i < j$:

$$z_{ij}^{y} = \begin{cases} D_{ij}, & r(i) = r(j) \\ |d_{ij}|, & r(i) \neq r(j) \end{cases} \tag{11.5}$$

where

$$D_{ij} = \frac{1}{2}(\ell_i + \ell_j) + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ki}y_{kj}}{2} + \sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k \frac{1+y_{ik}y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ik}y_{jk}}{2},$$

and

$$d_{ij} = \left[\frac{\ell_i}{2} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1+y_{ki}}{2} + \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ik}}{2}\right] - \left[\frac{\ell_j}{2} + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k \frac{1+y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k \frac{1-y_{jk}}{2}\right].$$

To linearize the absolute value in (11.5), we introduce binary ordering variables $x_{ij},\ i,j \in [n],\ r(i) \neq r(j), i<j$ for departments in different rows

$$x_{ij} = \begin{cases} 1, & \text{if the center of department } i \text{ lies before the center of department } j \\ -1, & \text{otherwise,} \end{cases}$$

and let x be the vector collecting these linear ordering variables. The following constraints have to hold for x:

$$x_{ij}d_{ij} \leq 0, \quad i,j \in [n],\ r(i) \neq r(j),\ i<j. \tag{11.6}$$

We can thus rewrite (11.5) as

$$z_{ij}^{x,y} = \begin{cases} D_{ij}, & r(i) = r(j) \\ -x_{ij}d_{ij}, & r(i) \neq r(j) \end{cases} \tag{11.7}$$

for all $i,j \in [n],\ i<j$. Now we can rewrite the objective function of SF-MRFLP in terms of x and y:

$$\sum_{i,j\in[n],i<j} c_{ij} z_{ij}^{x,y}. \tag{11.8}$$

In summary we have deduced a second formulation of SF-MRFLP.

Theorem 11.1 *Minimizing* (11.8) *over* $x_{ij}, y_{ij} \in \{-1,1\}$, (11.4) *and* (11.6) *solves SF-MRFLP.*

Proof. The equations (11.4) together with the integrality conditions on y suffice to induce all feasible layouts within the rows. The integrality conditions on x together with (11.6) ensure that we incorporate the distances between departments in different rows with the correct sign in the objective function. □

Next we rewrite (11.8) in terms of matrices and obtain a matrix-based fomulation:

$$\min\left\{K + c_x^\top x + \langle C_V, V\rangle + \langle C_Y, Y\rangle \ :\ x,y \in \{-1,1\},\ (x,y) \text{ satisfy (11.4) and (11.6) }\right\}, \tag{SF-MRFLP}$$

where $K := \frac{1}{2}\sum_{h\in\mathcal{R}}\left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij}\right)\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} \ell_i\right)\right]$, $Y := yy^\top$, $V := xy^\top$. The cost vector c_x

and the cost matrices C_Y and C_V are deduced by equating the coefficients of the following equations:

$$2\langle C_Y, Y\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)}} c_{ij}\left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} y_{ik}y_{kj}\ell_k - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} y_{ki}y_{kj}\ell_k - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} y_{ki}y_{kj}\ell_k\right),$$

$$2\langle C_V, V\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)\neq r(j)}} c_{ij}x_{ij}\left(\sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk}\right),$$

$$2c_x^\top x \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)\neq r(j)}} c_{ij}x_{ij}\left(L_{r(i)} - L_{r(j)}\right),$$

where L_i denotes the sum of the length of the departments on row i:

$$L_i = \sum_{k\in[n],\ r(k)=i} \ell_k, \quad i \in \mathcal{R}.$$

Note that we can model the distance between two departments i and j located in non-adjacent rows alternatively as the sum of the distances of the centers of the departments i and j to the common left origin:

$$\begin{aligned} z_{ij}^y =& \frac{\ell_i+\ell_j}{2} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1+y_{ki}}{2} + \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ik}}{2} \\ &+ \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k \frac{1+y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k \frac{1-y_{jk}}{2}, \quad |r(i)-r(j)|>1. \end{aligned}$$

For an illustration and comparison of the direct distance calculation from above and the "indirect" one suggested now see Figure 11.1.

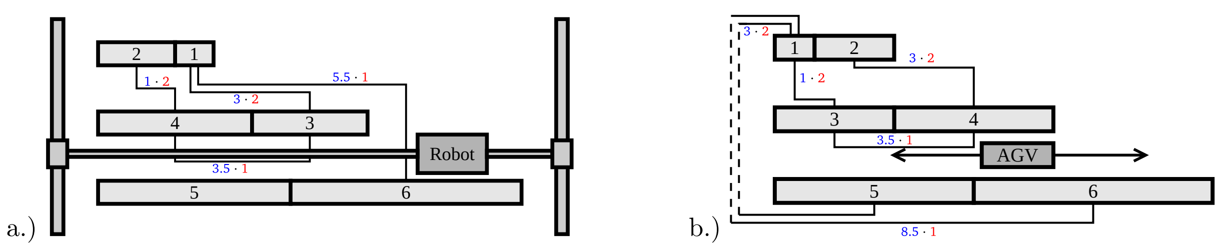


Figure 11.1: Illustration and comparison of the different distance calculation for departments in non-adjacent rows. We are given the following data: $\ell_i = i,\ i \in \{1,\ldots,6\},\ c_{16} = c_{34} = 1,\ c_{13} = c_{15} = c_{24} = 2.$ Departments 1 and 2 are assigned to row 1, departments 3 and 4 are assigned to row 2 and departments 5 and 6 are assigned to row 3. In a.) a gantry robot is used that can travel "directly" between departments in non-adjacent rows. We display the optimal layout for the direct distance calculation with associated costs of $1\cdot 2 + 3\cdot 2 + 5.5\cdot 1 + 3.5\cdot 1 = 17$. In b.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. If the AGV has to transport parts between machines in non-adjacent rows it has to leave one corridor on the left and enter the other corridor also on the left. We depict the optimal layout for the indirect distance calculation. The correspondent costs are $1\cdot 2 + 3\cdot 2 + 3.5\cdot 1 + 3\cdot 2 + 8.5\cdot 1 = 26$.

The alternative cost function $K + c_x^\top x + c_y^\top y + \langle C_V, V\rangle + \langle C_Y, Y\rangle$ can be obtained from:

$$2\langle C_Y, Z\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)}} c_{ij}\left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik}y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj}\right),$$

$$2\langle C_V, Z\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|=1}} c_{ij}x_{ij}\left(\sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk}\right),$$

$$2c_y^\top x \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|>1}} c_{ij}\left(\sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk}\right),$$

$$2c_x^\top x \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|=1}} c_{ij}x_{ij}\left(L_{r(i)} - L_{r(j)}\right),$$

$$K \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|>1}} c_{ij}\left(L_{r(i)} + L_{r(j)}\right) + \sum_{h\in\mathcal{R}}\left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij}\right)\left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i\right)\right].$$

The decision on how to define the distances of departments located in non-adjacent rows is determined by technical conditions in practice. In Subsection 11.6.6 we will computationally compare the two proposed variants with respect to the tightness of the related relaxations. In the following section we use matrix-based relaxations to get tight lower bounds for SF-MRFLP.

11.4 Semidefinite Relaxations

We collect the ordering variables in a vector

$$w := \begin{pmatrix} x \\ y \end{pmatrix},$$

and consider the matrix variable $W = ww^\top$. Our object of interest is the multi-row ordering polytope

$$\mathcal{P}_{MRO} := \operatorname{conv}\left\{ \begin{pmatrix} 1 \\ w \end{pmatrix}\begin{pmatrix} 1 \\ w \end{pmatrix}^\top \ :\ w \in \{-1,1\},\ w \text{ satisfies (11.4) and (11.6)} \right\},$$

We apply standard techniques to construct SDP relaxations. First we relax the nonconvex equation $W - ww^\top = 0$ to the positive semidefinite constraint

$$W - ww^\top \succcurlyeq 0.$$

Moreover, the main diagonal entries of W correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(W) = e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(w, W) := \begin{pmatrix} 1 & w^\top \\ w & W \end{pmatrix}, \tag{11.9}$$

where $\dim(Z) = \binom{n}{2} + 1 =: \Delta$. By the Schur complement theorem, $W - ww^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. We therefore conclude that $\mathcal{P}_{MRO}$ is contained in the elliptope

$$\mathcal{E} := \{\ Z :\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}, \tag{11.10}$$

which is studied in detail by Laurent and Poljak [199, 200].

Next we can formulate SF-MRFLP as a semidefinite optimization problem in binary variables.

Theorem 11.2 *The problem*

$$\min\ \{\ \langle C_Z, Z\rangle : Z \text{ satisfies } (11.4) \text{ and } (11.6),\ Z \in \mathcal{E},\ w \in \{-1, 1\}\ \}$$

where the cost matrix C_Z is given by

$$C_Z := \begin{pmatrix} K & c_x^\top & 0 \\ c_x & 0 & C_V \\ 0 & C_V & C_Y \end{pmatrix},$$

is equivalent to SF-MRFLP.

Proof. Since $w_i^2 = 1,\ i \in \{1, \ldots, \Delta - 1\}$ we have $\operatorname{diag}(W - ww^\top) = 0$, which together with $W - ww^\top \succcurlyeq 0$ shows that in fact $W = ww^\top$ is integral. By Theorem 11.1, integrality on W together with (11.4) and (11.6) suffice to describe SF-MRFLP. □

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of SF-MRFLP:

$$\min\ \{\langle C_Z, Z\rangle : Z \text{ satisfies } (11.4) \text{ and } (11.6),\ Z \in \mathcal{E}\ \}. \tag{$\texttt{SDP}_{\texttt{basic}}$}$$

There are several ways to tighten the above relaxation. This is the topic of the next two subsections.

11.4.1 Tightening the Semidefinite Relaxation by Exploiting Binarity and Ordering Properties

Since Z is generated as the outer product of the vector $\begin{pmatrix} 1 & w \end{pmatrix}^\top$ that has merely $\{-1, 1\}$ entries in the non-relaxed SDP formulation, any feasible solution of SF-MRFLP also belongs to the metric polytope $\mathcal{M}$:

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e,\ \ 1 \le i < j < k \le \Delta \right\}. \tag{11.11}$$

We note that $\mathcal{M}$ is defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets. They are the triangle inequalities of the max-cut polytope, see e.g. [88].

We can impose a transitivity relation on all ordering variables by imposing the 3-cycle inequalities on w:

$$-1 \le w_{ij} + w_{jk} - w_{ik} \le 1, \quad i, j, k \in [n],\ i < j < k.$$

These may rule out some optimal solutions but preserve at least one optimal solution. Squaring them we obtain the 3-cycle equalities

$$w_{ij}w_{jk} - w_{ij}w_{ik} - w_{ik}w_{jk} = -1, \quad i, j, k \in [n],\ i < j < k, \tag{11.12}$$

which are a strengthening of (11.4) and model transitivity for $w \in \{-1, 1\}$ [48]. Additionally the 3-cycle equalities together with $Z \succcurlyeq 0$ ensure the 3-cycle inequalities also for non-integral w [156, Proposition 4.2].

Another generic improvement was proposed by Lovász and Schrijver [214]. Applied to our problem, their approach suggests to multiply the 3-cycle inequalities

$$1 - w_{ij} - w_{jk} + w_{ik} \geq 0, \qquad 1 + w_{ij} + w_{jk} - w_{ik} \geq 0. \tag{11.13}$$

by the nonnegative expressions

$$1 - w_{lo} \geq 0, \qquad 1 + w_{lo} \geq 0, \qquad l, o \in [n],\ l < o. \tag{11.14}$$

This results in the following $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ inequalities:

$$\begin{aligned} -1 - w_{lo} \leq w_{ij} + w_{jk} - w_{ik} + w_{ij,lo} + w_{jk,lo} - w_{ik,lo} \leq 1 + w_{lo}, \\ -1 + w_{lo} \leq w_{ij} + w_{jk} - w_{ik} - w_{ij,lo} - w_{jk,lo} + w_{ik,lo} \leq 1 - w_{lo}, \end{aligned} \tag{11.15}$$

for $i, j, k, l, o \in [n],\ i < j < k,\ l < o$. We define the corresponding polytope $\mathcal{LS}$:

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (11.15)\ \}. \tag{11.16}$$

We can also deduce lower and upper bounds on the sum of the inter-row ordering variables for each of the $\binom{m}{2}$ pairs of rows:

$$\delta_l^{c_1,c_2} \leq \sum_{\substack{i,j \in [n], i<j, \\ r(i)=c_1 \neq c_2 = r(j)}} x_{ij} \leq \delta_u^{c_1,c_2}, \quad c_1, c_2 \in \mathcal{R},\ c_1 < c_2, \tag{11.17}$$

where $\delta_l^{c_1,c_2}$ and $\delta_u^{c_1,c_2}$ are dependent on the given data.[1] We obtain $\delta_l^{c_1,c_2}$ by sorting the departments in row c_1 by decreasing length and the departments in row c_2 by increasing length, then computing the sum of the inter-row ordering variables. Analogously we compute $\delta_u^{c_1,c_2}$ by sorting the departments in row c_1 by increasing length and the departments in row c_2 by decreasing length. If two centers are located exactly below each other we break the symmetry and tighten the bounds by setting the respective variables to $+1$ in the first case and to -1 in the second case. Thus it can happen that $\delta_l^{c_1,c_2} > \delta_u^{c_1,c_2}$; in this case we set $\delta_l^{c_1,c_2} = \delta_u^{c_1,c_2} := \frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2} + \left(\frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2} \bmod 2\right)$ to preserve an optimal solution.

Fact 11.3 *The constraints* (11.17) *can rule out some optimal solutions but preserve at least one optimal solution and thus are valid for tightening the semidefinite relaxation.*

We can rewrite (11.17) as

$$\frac{\delta_l^{c_1,c_2} - \delta_u^{c_1,c_2}}{2} \leq \sum_{\substack{i,j \in [n], i<j, \\ r(i)=c_1 \neq c_2 = r(j)}} x_{ij} - \frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2} \leq \frac{\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2}}{2}, \quad c_1, c_2 \in \mathcal{R},\ c_1 < c_2.$$

[1] Notice that we can assume w.l.o.g. that $i < j$ for $r(i) = c_1 \neq c_2 = r(j)$.

Squaring yields

$$0 \le \left(\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \right)^2 - (\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} + \left(\frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2}\right)^2 \le \left(\frac{\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2}}{2}\right)^2, \quad \text{if } (\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2}) \bmod 4 = 0, \tag{11.18a}$$

$$1 \le \left(\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \right)^2 - (\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} + \left(\frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2}\right)^2 \le \left(\frac{\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2}}{2}\right)^2, \quad \text{if } (\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2}) \bmod 4 = 2. \tag{11.18b}$$

To obtain the lower bound we exploit the fact that the inter-row ordering variables are $\{-1, 1\}$. Hence if $\delta_u^{c_1,c_2} - \delta_l^{c_1,c_2} \le 2$ then (11.18) defines equalities on the sum of products of inter-row variables.

Fact 11.4 *The smallest subspace containing the multi-row polytope is defined by* (11.4)*, but we build our semidefinite relaxation on an even smaller subspace that contains at least one optimal solution. This subspace is defined by* (11.12) *and possibly additional equations from* (11.17) *and* (11.18)*.*

We can also multiply

$$\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2} \ge 0, \qquad \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \ge 0,$$

by (11.13) and (11.14). This results in the following inequalities

$$\begin{aligned} \delta_l^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le w_{lo} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - w_{lo}\delta_l^{c_1,c_2} \le \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2}, \\ \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_u^{c_1,c_2} \le w_{lo}\delta_u^{c_1,c_2} - w_{lo} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij}, \end{aligned} \tag{11.19}$$

for $c_1, c_2 \in \mathcal{R},\ c_1 < c_2,\ l, o \in [n],\ l < o$ and

$$\begin{aligned} \delta_l^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le \overline{w} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \overline{w}\delta_l^{c_1,c_2} \le \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2}, \\ \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_u^{c_1,c_2} \le \overline{w}\delta_u^{c_1,c_2} - \overline{w} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij}, \end{aligned} \tag{11.20}$$

for $\overline{w} := w_{kl} + w_{lo} - w_{ko}$ and $c_1, c_2 \in \mathcal{R},\ c_1 < c_2,\ k, l, o \in [n],\ k < l < o$.

We gather the $O(n^2m^2)$ inequalities based on the bounds on the sum of the inter-row variables and define the polytope $\mathcal{SI}$:

$$\mathcal{SI} := \{\ Z : Z \text{ satisfies } (11.17) - (11.20)\ \}. \tag{11.21}$$

11.4.2 Tightening the Semidefinite Relaxation through the Distance Variables

We now turn our attention to the distance variables. We can tighten constraints (11.6) as follows:

$$x_{ij} d_{ij} \leq d_{ij}, \quad x_{ij} d_{ij} \leq -d_{ij}, \qquad i,j \in [n],\ r(i) \neq r(j),\ i < j, \tag{11.22}$$

which holds true for $x_{ij} \in \{-1, 1\}$. Similar constraints also hold for the intra-row variables y:

$$y_{ij} d_{ij} \leq d_{ij}, \quad y_{ij} d_{ij} \leq -d_{ij}, \qquad i,j \in [n],\ r(i) = r(j),\ i < j. \tag{11.23}$$

We can gather (11.22) and (11.23) together into a single expression in terms of w:

$$w_{ij} d_{ij} \leq d_{ij}, \quad w_{ij} d_{ij} \leq -d_{ij}, \qquad i,j \in [n],\ i < j. \tag{11.24}$$

Fact 11.5 *The constraints* (11.24) *are valid.*

For $i,j \in [n],\ r(i) = r(j),\ i < j$, the expressions $y_{ij} d_{ij}$ and D_{ij} represent different ways to express the distance between two departments within the same row. They are equivalent for integral y but we decided to express the intra-row distances in the objective function (11.8) via D_{ij} as it is common for the SRFLP. Notice that all variables in both $y_{ij} d_{ij}$ and D_{ij} appear in a 3-cycle equality (11.12) and thus are tightly constrained in the relaxation. This fact also explains why the various linear and semidefinite relaxations for SRFLP that include these equations on the betweenness variables produce such tight bounds even for very large instances.

Another class of constraints are the triangle inequalities relating the distances between three departments, where we use again D_{ij} to measure the intra-row distances

$$\begin{aligned} &D_{ij} + D_{ik} \geq D_{jk}, \quad D_{ij} + D_{jk} \geq D_{ik}, \quad D_{jk} + D_{ik} \geq D_{ij}, \\ &\qquad i < j < k \in [n],\ r(i) = r(j) = r(k), \end{aligned} \tag{11.25a}$$

$$\begin{aligned} &-x_{ik} d_{ik} - x_{jk} d_{jk} \geq D_{ij}, \quad -x_{ik} d_{ik} - D_{ij} \geq -x_{jk} d_{jk}, \quad D_{ij} - x_{jk} d_{jk} \geq -x_{ik} d_{ik}, \\ &\qquad i < j < k \in [n],\ r(i) = r(j),\ r(i) \neq r(k), \end{aligned} \tag{11.25b}$$

$$\begin{aligned} &-x_{ij} d_{ij} - x_{ik} d_{ik} \geq D_{jk}, \quad -x_{ij} d_{ij} - D_{jk} \geq -x_{ik} d_{ik} \quad D_{jk} - x_{ik} d_{ik} \geq -x_{ij} d_{ij}, \\ &\qquad i < j < k \in [n],\ r(i) \neq r(j),\ r(j) = r(k), \end{aligned} \tag{11.25c}$$

$$\begin{aligned} &-x_{ij} d_{ij} - x_{ik} d_{ik} \geq -x_{jk} d_{jk}, \quad -x_{ij} d_{ij} - x_{jk} d_{jk} \geq -x_{ik} d_{ik}, \quad -x_{jk} d_{jk} - x_{ik} d_{ik} \geq -x_{ij} d_{ij}, \\ &\qquad i < j < k \in [n],\ r(i) \neq r(j) \neq r(k) \neq r(i). \end{aligned} \tag{11.25d}$$

For the exact SF-MRFLP formulation these constraints implicitly hold.

Fact 11.6 *The constraints* (11.25) *are valid.*

Furthermore these constraints imply the distance constraints for more than three departments.

Theorem 11.7 *The triangle inequalities* (11.25) *imply all the distance constraints involving more than three departments.*

Proof. Consider a general distance constraint for $\gamma > 3$ departments

$$\sum_{h=1}^{\gamma-1} -w_{i_h i_{h+1}} d_{i_h i_{h+1}} \geq -w_{i_1 i_\gamma} d_{i_1 i_\gamma},$$

where

$$-w_{i_h i_{h+1}} d_{i_h i_{h+1}} := \begin{cases} D_{i_h i_{h+1}}, & r(h) = r(h+1), \\ -x_{i_h i_{h+1}} d_{i_h i_{h+1}}, & r(h) \neq r(h+1). \end{cases}$$

We show that (11.25) implies the above inequality. We start out with the left hand side of the inequality and use $-w_{i_1 i_2} d_{i_1 i_2} - w_{i_2 i_3} d_{i_2 i_3} \geq -w_{i_1 i_3} d_{i_1 i_3}$ to obtain

$$-w_{i_1 i_2} d_{i_1 i_2} - w_{i_2 i_3} d_{i_2 i_3} - \sum_{h=3}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}} \geq -w_{i_1 i_3} d_{i_1 i_3} - \sum_{h=3}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}}.$$

Next we use $-w_{i_1 i_3} d_{i_1 i_3} - w_{i_3 i_4} d_{i_3 i_4} \geq -w_{i_1 i_4} d_{i_1 i_4}$ in the same fashion. The process can be repeated resulting in the chain of inequalities

$$-\sum_{h=1}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}} \geq \ldots \geq -w_{i_1, i_{\gamma-1}} d_{i_1, i_{\gamma-1}} - w_{i_{\gamma-1} i_\gamma} d_{i_{\gamma-1} i_\gamma} \geq -w_{i_1 i_\gamma} d_{i_1 i_\gamma}.$$

□

Hence we define the polytope

$$\mathcal{DV} := \{ \, Z : Z \text{ satisfies (11.25)} \, \} \tag{11.26}$$

using the $3\binom{n}{3}$ triangle inequalities relating the distances between 3 or more departments.

11.4.3 Tightest Semidefinite Relaxation and Solution Methodology

Gathering all the results in Section 11.4, we get the following relaxation of $\mathcal{P}_{MRO}$:

$$\min \left\{ \langle C_Z, Z \rangle : Z \text{ satisfies (11.12) and (11.24)}, \; Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS} \cap \mathcal{SI} \cap \mathcal{DV}) \right\}. \tag{$\texttt{SDP}_{\texttt{full}}$}$$

While theoretically tractable, it is clear that ($\texttt{SDP}_{\texttt{full}}$) has an impractically large number of constraints. Indeed, even including only $O(n^3)$ constraints is not realistic for instances of size $n \geq 20$. For this reason, we adopt an approach originally suggested in [103] and since then applied to the max-cut problem [262] and several ordering problems [62, 64, 168]. Initially, we only explicitly ensure that Z lies in the elliptope $\mathcal{E}$. This can be achieved efficiently with standard interior-point methods, see e.g. [146]. All other constraints are handled through Lagrangian duality.

For notational convenience, let us formally denote the equations in ($\texttt{SDP}_{\texttt{full}}$) by $e - \mathcal{A}(Z) = 0$. Similarly we write the inequalities in ($\texttt{SDP}_{\texttt{full}}$) as $g - \mathcal{D}(Z) \geq 0$. Using the Lagrangian

$$\mathcal{L}(Z, \lambda, \mu) := \langle C, Z \rangle + \lambda^\top (e - \mathcal{A}(Z)) + \mu^\top (g - \mathcal{D}(Z)),$$

we obtain the partial Lagrangian dual

$$f(\lambda, \mu) := \min_{Z \in \mathcal{B}} \mathcal{L}(Z, \lambda, \mu) = e^\top \lambda + g^\top \mu + \min_{Z \in \mathcal{B}} \langle C - \mathcal{A}^\top(\lambda) - \mathcal{D}^\top(\mu), Z \rangle.$$

Since ($\texttt{SDP}_{\texttt{full}}$) has strictly feasible points, strong duality holds and we can solve the relaxation through $\max_{\mu \geq 0, \lambda} f(\lambda, \mu)$.

The function f is well-known to be convex but non-smooth. For a given feasible point (λ, μ) the evaluation of $f(\lambda, \mu)$ amounts to optimizing over $\mathcal{E}$. We do this using a primal-dual interior-point method which also provides a primal feasible $Z_{\lambda,\mu}$ yielding a subgradient of f. Using these ingredients, we get an approximate minimizer of f using the bundle method [103]. Thanks to the use of the bundle method, we quickly obtain a good initial set of constraints. On the other hand, since the rate of convergence is slow, we limit the number of function evaluations to control the overall computational effort. These evaluations nevertheless constitute the computational bottleneck for larger instances as there they are responsible for more than 95% of the required running time. Detailed computational results are given in Section 11.6.

We next describe how a feasible layout can be obtained from a solution to any of the SDP relaxations.

11.5 Obtaining Feasible Layouts

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans-Williamson [121] to the solution of the SDP relaxation. We take the resulting vector $\overline{w}$ and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities

$$-1 \leq \overline{y}_{ij} + \overline{y}_{jk} - \overline{y}_{ik} \leq 1 \tag{11.27}$$

within the rows and the inequalities for the inter-row variables (11.6). Computational experiments demonstrated that this repair strategy is not as critical as one might assume. For example, in multi-level crossing minimization this SDP rounding heuristic clearly dominates traditional heuristic approaches [64].

Let us give a more detailed description of the implementation of our heuristic. We consider a vector w' that encodes a feasible layout of the departments in all rows. The algorithm stops after 100 executions of step 2. (Note that before the 51st execution of step 2, we perform step 1 again. As step 1 is quite expensive, we refrain from executing it too often.)

1. Let W'' be the current primal (fractional) solution of $(\texttt{SDP}_{\texttt{full}})$ (or some other semidefinite relaxation) obtained by the bundle method or an interior-point solver. Compute the convex combination $R := \lambda(w'w'^{\top}) + (1-\lambda)W''$ using a randomly generated $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^{\top}$ of R.

2. Apply Goemans-Williamson hyperplane rounding to D and obtain a $-1/+1$ vector $\overline{w}$ (cf. [262]).

3. Compute the induced objective value $z(\overline{w}) := \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}^{\top} C_Z \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}$. If $z(\overline{w}) \geq z(w')$: go to step 2.

4. If $\overline{w}$ satisfies (11.27) and (11.6): set $w' := \overline{w}$ and go to 2. Else: modify $\overline{w}$ by first changing the signs of one of three variables in all violated 3-cycle inequalities, afterwards flipping signs of the inter-row ordering variables to satisfy (11.6) and go to step 3.

The final w' is the heuristic solution. If the duality gap is not closed after the heuristic, we return to the SDP optimization algorithm and then retry the heuristic (retaining the last vector w').

11.6 Computational Experience

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7.

We define MRFLP instances using the data from SRFLP instances in the literature. as well as data randomly generated in the same way as in [169], namely with a density of 50% and with lengths and connectivities varying randomly between 1 and 10.

Table 11.1 gives the characteristics of the SRFLP instances that we considered. These include well-known benchmark instances from [6, 151, 283], randomly generated instances from [20, 169], and instances with clearance requirements from [150]. All the instances can be downloaded from `http://www.gerad.`

ca/files/Sites/Anjos/flplib.html. We use the latter without taking the clearance requirement into account, Hence we could round on 5 as the lengths of the departments are multiples of 10. In general, while for the SRFLP we can round to the nearest integer because 0.5 can only occur in the constant term, for the MRFLP we can round the lower bound only to 0.5 as the constant term is different for distinct row assignments.

Instance	Source	Size (n)	SRFLP	
			Optimal SRFLP solution	Time (sec) [169]
S_8	[283]	8	801	0.6
SH_8	[283]	8	2324.5	2.3
S_9	[283]	9	2469.5	0.7
SH_9	[283]	9	4695.5	9.2
S_10	[283]	10	2781.5	0.6
S_11	[283]	11	6933.5	1.3
H_5	[151]	5	800	0.1
H_6	[151]	6	1480	0.1
H_7	[151]	7	3680	0.6
H_8	[151]	8	4725	0.4
H_12	[151]	12	17945	7.9
H_15	[151]	15	45840	19.6
Rand_5	new	5	147.5	0.1
Rand_6	new	6	420	0.4
Rand_7	new	7	344	0.3
Rand_8	new	8	382	1.3
Rand_9	new	9	1024.5	2.2
Rand_10	new	10	1697	3.1
Rand_11	new	11	1564	2.0
Rand_12	new	12	2088	8.4
Rand_13	new	13	3101.5	7.8
Rand_14	new	14	3653	17.9
Rand_15	new	15	5345.5	19.2
P_15	[6]	15	6305	19.7

Table 11.1: Characteristics of SRFLP instances with between 5 and 15 departments

For each instance considered, our computational objective is to obtain the best possible solution for a placement of the departments in two to five rows, thus showing that our relaxations and methodology are practical for MRFLP instances with any given number of rows. First we explain our algorithmic strategies in detail for the DRFLP in Subsections 11.6.1 – 11.6.5 and then we expand our computational results to the case of 3 and more rows in Subsection 11.6.6.

11.6.1 Global Optimization of Small DRFLP Instances Using ($\mathtt{SDP}_{\mathtt{full}}$)

For small DRFLP instances, the relaxation ($\mathtt{SDP}_{\mathtt{full}}$) can be solved for each of the $2^{n-1}-1$ possible row assignments. From the obtained bounds, we can deduce global upper and lower bounds: these are the minima of all upper and lower bounds respectively. Note that we could also incorporate horizontal distances between the (non-adjacent) rows and straightforwardly (without additional computational effort) calculate a correspondent term considering the horizontal distances between all pairs of departments for each row assignment.

We restricted the running time per instance to 24 hours. The upper bounds were obtained using the SDP rounding heuristic in Section 11.5. The results are summarized in Table 11.2. We point out that the lower and upper bound were often obtained from different row assignments. Looking at the running times and their growth rates, we deduce that this approach is realistic only for instances with fewer than 8 departments within the 24-hour time limit.

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems			Computational statistics		
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time w/ bundle (sec)	Total time w/o bundle (sec)
H_5	420	450	7.14	7.14	1.54	7	144.2	9.6	8.7
Rand_5	52.5	52.5	0	7.65	1.03	10	141.3	7.9	9.0
H_6	703	720	2.42	10.96	3.30	7	477.4	130.3	384.8
Rand_6	189.5	190.5	0.53	6.18	1.70	7	424.8	94.2	317.6
H_7	1639.5	1700	3.69	14.05	2.69	6	1205.2	6177.5	47619.6
Rand_7	166	166	0	13.21	2.20	16	1016.0	2704.0	19448.3

Table 11.2: Computational results for $(\mathtt{SDP}_{\mathtt{full}})$

The last two columns of Table 11.2 illustrate the impact of an important computational strategy. We start with the basic relaxation:

$$\min\left\{\langle C_Z, Z\rangle : Z \text{ satisfies (11.12) and (11.24)},\ Z\in\mathcal{E}\right\}. \qquad (\mathtt{SDP}_{\mathtt{basic}})$$

For the results labelled "w/ bundle" we used 10 function evaluations of the bundle method and 3 constraint updates to obtain an initial set of constraints to add to the relaxation $(\mathtt{SDP}_{\mathtt{basic}})$. We then solved the resulting relaxation using Sedumi [289]; added violated inequality constraints (from all the inequalities in $(\mathtt{SDP}_{\mathtt{full}})$); solved again using Sedumi; and repeated this process until no more violations were found. Alternatively one can skip the search for an initial set of inequalities using the bundle method and proceed straight to using Sedumi starting from the relaxation $(\mathtt{SDP}_{\mathtt{basic}})$. The times for this alternative approach are labelled "w/o bundle". The important observation is that the use of the initial set of inequalities yields a speed-up of one order of magnitude in the running time for the largest instances. The same effect was observed for the linear relaxation of the linear ordering problem for very large instances ($n \geq 150$) [156, Section 10.3].

11.6.2 Analysis of the Practical Impact of the Various Constraint Classes

Because $(\mathtt{SDP}_{\mathtt{full}})$ is too expensive to be solved for $n \geq 8$, we examine the efficiency (impact on computation time) and effectiveness (impact on bound quality) of the various constraint classes. The aim is to find a smaller relaxation that contains the most important constraints with respect to bound quality.

Our starting relaxation is again $(\mathtt{SDP}_{\mathtt{basic}})$. This model reflects the fundamental structure of the original problem in the sense that it would suffice to obtain the optimal solution if we additionally imposed integrality conditions on the ordering variables (see Theorem 11.1).

First we examine the practical effect of the constraint sets $\mathcal{DV}$ and $\mathcal{SI}$. The computational results are summarized in Table 11.3. The results support the conclusion that $\mathcal{DV}$ is both effective and efficient. On the other hand, the impact of $\mathcal{SI}$ is much less.

Adding $\mathcal{DV}$ to $(\mathtt{SDP}_{\mathtt{basic}})$, we obtain a relaxation that is improved but still computationally cheap:

$$\min\left\{\langle C_Z, Z\rangle : Z \text{ satisfies (11.12) and (11.24)},\ Z\in(\mathcal{E}\cap\mathcal{DV})\right\}. \qquad (\mathtt{SDP}_{\mathtt{cheap}})$$

Next we examine the effects of adding $\mathcal{LS}$ and $\mathcal{M}$ to this new relaxation $(\mathtt{SDP}_{\mathtt{cheap}})$. The results are summarized in Table 11.4. We observe that neither $\mathcal{M}$ nor $\mathcal{LS}$ is particularly efficient. We also tested the relaxation $(\mathtt{SDP}_{\mathtt{basic}}) \cap \mathcal{M} \cap \mathcal{LS}$ and found that the overall gaps for these same instances are always over

Instance	$(\texttt{SDP}_{\texttt{basic}})$			$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{DV}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	5.2	72.75	7.1	4.5	10.43	12.9
Rand_5	4.5	98.11	6.3	4.5	0	26.5
H_6	11.5	171.70	10.6	12.3	8.27	19.5
Rand_6	12.4	126.79	10.0	13.8	1.06	20.9
H_7	32.0	188.14	14.5	31.7	4.36	30.8
Rand_7	30.9	114.19	13.2	31.7	0	31.0
Instance	$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{SI}$			$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{DV} \cap \mathcal{SI}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	5.4	71.43	25.9	5.3	10.43	30.3
Rand_5	5.0	90.91	23.1	5.1	0	29.3
H_6	13.9	104.26	39.7	15.7	8.27	43.9
Rand_6	15.4	83.17	35.1	14.9	1.06	37.9
H_7	45.4	139.44	79.7	39.4	4.36	62.1
Rand_7	41.8	97.62	49.9	39.0	0	59.8

Table 11.3: Study of the impact of constraint classes $\mathcal{DV}$ and $\mathcal{SI}$

50%. Furthermore, the running times are much higher than for $(\texttt{SDP}_{\texttt{cheap}})$.

In summary, our computational results in this section strongly suggest that $(\texttt{SDP}_{\texttt{cheap}})$ provides the best tradeoff between computational time and quality of the bounds. Of course, the constraint classes not included in $(\texttt{SDP}_{\texttt{cheap}})$ still help tighten the relaxation but it is more efficient to use them within the improvement strategy proposed in Section 11.6.5.

11.6.3 Optimizing Over All Row Assignments Using $(\texttt{SDP}_{\texttt{cheap}})$

We run again the algorithmic approach of Section 11.6.1 but using the relaxation $(\texttt{SDP}_{\texttt{cheap}})$. The results are reported in Table 11.5.

Comparing with the performance of $(\texttt{SDP}_{\texttt{full}})$ documented in Table 11.2, we see that $(\texttt{SDP}_{\texttt{cheap}})$ runs at least one order of magnitude faster for instances of size $n = 6$ and $n = 7$ with only a mild deterioration of the lower bounds, and hence of the gap.

Using $(\texttt{SDP}_{\texttt{cheap}})$, we are able to compute bounds for instances of sizes up to $n = 14$ within the 24-hour time limit. We observe that the quality of the bounds does not deteriorate as the size increases, and that the running time increases by a factor of 3 for each unit increase in n.

11.6.4 Using Bounds in the Enumeration

It is possible to further reduce the computational effort within the enumeration scheme using previously acquired lower-bound knowledge. This is because the computation of a lower bound can be stopped if its current value is already above the current global lower bound.

The impact of this strategy depends on the order in which we look at the row assignments; those with the weakest lower bounds should be computed first. We propose the following heuristic to obtain a reasonably good ordering:

- Order the row assignments in increasing difference of the sums of the lengths of the departments in each row
- If two or more assignments are tied, further sort them in increasing difference between the sum of connectivities within the rows and the sum of connectivities between the rows. Specifically for the

Instance	$(\texttt{SDP}_{\texttt{cheap}})$			$(\texttt{SDP}_{\texttt{cheap}}) \cap \mathcal{LS}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	4.5	10.43	12.9	6.6	7.27	70.0
Rand_5	4.5	0	26.5	6.6	0	71.2
H_6	12.3	8.27	19.5	35.4	4.65	198.8
Rand_6	13.8	1.06	20.9	29.9	1.06	177.5
H_7	31.7	4.36	30.8	255.1	4.26	415.0
Rand_7	31.7	0	31.0	232.0	0	407.4
Instance	$(\texttt{SDP}_{\texttt{cheap}}) \cap \mathcal{M}$			$(\texttt{SDP}_{\texttt{full}})$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	8.3	7.27	92.9	9.6	7.14	144.2
Rand_5	7.9	0	98.4	7.9	0	141.3
H_6	70.4	2.42	339.8	130.3	2.42	477.4
Rand_6	58.2	0.53	310.5	94.2	0.53	424.8
H_7	1821.5	3.72	940.1	6177.5	3.69	1205.2
Rand_7	1006.0	0	780.4	2704.0	0	1016.0

Table 11.4: Study of the impact of constraint classes $\mathcal{M}$ and $\mathcal{LS}$

two-row case, we have:

$$\left| \sum_{\substack{i<j\in[n],\\ r(i)=r(j)=1}} c_{ij} - \sum_{\substack{i<j\in[n],\\ r(i)=r(j)=2}} c_{ij} \right| + \left| \sum_{\substack{i<j\in[n],\\ r(i)=r(j)=1}} c_{ij} - \sum_{\substack{i,j\in[n],\\ r(i)=1,r(j)=2}} c_{ij} \right| + \left| \sum_{\substack{i<j\in[n],\\ r(i)=r(j)=2}} c_{ij} - \sum_{\substack{i,j\in[n],\\ r(i)=1,r(j)=2}} c_{ij} \right|.$$

The intuition behind this heuristic is that small differences in both cases are generally good: it is desirable that the sum of lengths of departments in the rows should be equal, and that connectivities should be spread equally.

Table 11.6 summarizes the results obtained for the enumeration using bound information for instances with $n \leq 15$ departments. Comparing with the performance of using $(\texttt{SDP}_{\texttt{cheap}})$ without bound information documented in Table 11.5, we see that using the bounds makes a dramatic reduction in the running time without any effect on the quality of the results. As a consequence, we are able to compute bounds for instances with up to $n = 15$ within the 24-hour time limit. Nevertheless, the running time still increases by a factor of 3 for each unit increase in n. For the instances with $n = 15$, the rapid growth of the computational effort required to handle the 3-cycle equations is clear.

We point out that for the results in Table 11.6, we change our strategy for $n \geq 10$ by doing 20 function evaluations (instead of 10) and 5 constraint updates (instead of 3) in the bundle method. Not only do the Sedumi iterations become more expensive compared to bundle iterations for $n \geq 10$, but also because in these tests we use the bounds for pruning, running the bundle method longer reduces the overall computation time as we often can prune the lower bound computation before switching to Sedumi. As a consequence we do not have to go to Sedumi for many assignments (see the sixth and seventh columns in Table 11.6).

We checked the quality of the order of the row assignments obtained by our heuristic using the data obtained by the complete enumeration approach above. The results are summarized in the last two columns of Table 11.6 which give the average position of the best assignment with respect to the lower and the upper bound respectively. The impact of the heuristic is measured by comparing the average percentages we obtain with the expected value of 50% for a random ordering; our smaller percentages show that the

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems			Computational statistics	
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time w/ bundle (sec)
H_5	407.5	450	10.43	10.43	2.52	5	12.9	4.5
Rand_5	52.5	52.5	0	8.93	1.78	7	15.1	4.5
H_6	665	720	8.27	16.95	5.78	4	19.5	12.3
Rand_6	188.5	190.5	1.06	6.65	2.59	3	20.9	13.8
H_7	1629	1700	4.36	15.17	3.75	4	30.8	31.7
Rand_7	166	166	0	13.82	3.16	11	31.0	31.7
H_8	2351	2385	1.45	21.22	5.80	1	42.9	86.8
S_8	380.5	408	7.23	20.10	5.87	2	44.1	91.6
SH_8	990.5	1135.5	14.64	17.00	10.85	4	56.1	87.4
Rand_8	192	205	6.77	28.10	4.77	6	42.8	82.2
S_9	1163	1181.5	1.59	13.63	3.42	6	64.2	253.4
SH_9	1974.5	2294.5	16.21	18.87	11.31	0	80.6	251.9
Rand_9	447.5	492.5	10.06	19.73	5.61	3	55.8	252.6
S_10	1314	1374.5	4.60	10.77	4.20	7	82.7	713.0
Rand_10	779	838	7.57	15.16	5.68	0	78.7	698.2
S_11	3325.5	3439.5	3.43	14.92	5.16	6	106.8	2127.0
Rand_11	643.5	708	10.02	23.95	5.76	9	103.4	2048.5
H_12	8446.5	8995	6.49	17.31	6.19	0	125.1	6189.5
Rand_12	775.5	799	3.03	17.69	6.21	0	128.8	6389.5
Rand_13	1058	1070	1.13	19.11	5.98	0	159.5	20636.9
Rand_14	1335.5	1393.5	4.34	20.25	6.59	1	172.5	60845.6

Table 11.5: Computational results for $(\texttt{SDP}_{\texttt{cheap}})$

heuristic generally has the desired effect. Note that the quality of the heuristic cannot be evaluated for instances with more than 14 departments since the exact lower bounds for all row assignments of these instances could not be computed in the previous subsection.

Let us also compare the above results with the results reported by Amaral [10]. His MILP formulation allows him to solve instances with up to 13 departments to optimality within a few hours of computing time but he is not able to give good lower bounds for larger instances. In particular Amaral considered the instance P_15: The layout given by cplex when it was aborted (incumbent solution) had a cost of 3263.0. Hence he proposed two heuristics (based on 2-opt and 3-opt) that gave a layout with costs of 3195, hence with the same costs as the layout obtained by our SDP rounding heuristic. Furthermore Amaral showed that these heuristics can handle larger instances with up to 30 departments.

We also reimplemented the approach by Amaral using the cplex solver on our machine and obtained the same results as reported in [10] for instances with up to 13 departments. But for our instances with 14 and 15 departments the gaps obtained by the MILP formulation where always larger than 25 % after 24h computing time.

11.6.5 A Strategy for Further Improvement of the Bound Quality

The relaxation $(\texttt{SDP}_{\texttt{cheap}})$ is more efficient than $(\texttt{SDP}_{\texttt{full}})$ but is also weaker. We can often improve the quality of the global lower bounds for an instance by finding the row assignment with the weakest lower bound; taking the optimal solution of its $(\texttt{SDP}_{\texttt{cheap}})$ relaxation as reported in Table 11.5; adding to the relaxation the violated inequalities from those present in $(\texttt{SDP}_{\texttt{full}})$ and resolving with Sedumi until we get the optimal solution of $(\texttt{SDP}_{\texttt{full}})$ for the selected row assignment; update the global lower bound of the

Instance	Global bounds (over all row assignments)			Computational statistics			Validation of the ordering heuristic	
	Lower bound	Upper bound	Gap (%)	Total time w/ bundle (sec)	% of instances stopped early by: bundle only	% of instances stopped early by: bundle or Sedumi	Position of best lower bound	Position of best upper bound
H_5	407.5	450	10.43	2.6	66.7	80.0	46.67	26.67
Rand_5	52.5	52.5	0	2.0	86.7	86.7	13.33	13.33
H_6	665	720	8.27	5.6	83.9	87.1	41.94	41.94
Rand_6	188.5	190.5	1.06	6.4	83.9	90.3	12.90	16.13
H_7	1629	1700	4.36	13.5	90.5	95.2	7.94	7.94
Rand_7	166	166	0	14.9	76.0	90.5	22.22	22.22
H_8	2351	2385	1.45	44.2	66.1	96.1	22.05	5.51
S_8	380.5	408	7.23	47.8	60.6	96.1	18.11	21.26
SH_8	990.5	1135.5	14.64	47.9	60.0	99.2	0.79	10.24
Rand_8	192	205	6.77	47.2	49.6	93.7	36.22	36.22
S_9	1163	1181.5	1.59	126.2	60.4	98.8	1.57	1.57
SH_9	1974.5	2294.5	16.21	135.9	49.4	99.6	3.92	3.53
Rand_9	447.5	492.5	10.06	109.1	80.4	98.0	10.20	2.35
S_10	1314	1374.5	4.60	333.2	82.6	98.8	2.35	4.50
Rand_10	779	838	7.57	398.4	71.4	98.8	11.74	4.31
S_11	3325.5	3439.5	3.43	1221.2	54.5	99.4	25.22	12.51
Rand_11	643.5	708	10.02	556.6	96.6	99.2	20.14	20.14
H_12	8446.5	8995	6.49	3245.2	52.2	99.7	24.72	5.18
Rand_12	775.5	799	3.03	1428.3	97.8	99.7	16.27	23.50
Rand_13	1058	1070	1.13	3444.5	98.5	99.8	30.99	30.99
Rand_14	1335.5	1393.5	4.34	9941.4	97.4	99.9	39.58	39.58
H_15	16066	16640	3.57	69181.3	44.2	99.9	-	-
P_15	3046	3195	4.89	69622.8	42.8	99.9	-	-
Rand_15	2461	2643.5	7.42	64097.9	48.9	99.9	-	-

Table 11.6: Results using $(\mathtt{SDP_{cheap}})$ and using bounds for pruning in enumeration

instance accordingly. We repeat this process until the weakest lower bound comes from a row assignment for which we have already improved the $(\mathtt{SDP_{cheap}})$ relaxation. We also use the current overall lower bound to stop the lower bound computation when it becomes irrelevant.

The results we obtained using this improvement approach on instances with $n = 7, 8, 9$ are reported in Table 11.7. (We omit Rand_7 since $(\mathtt{SDP_{cheap}})$ is already optimal for it.) As the direct solution for instances with 8 or more departments is far too expensive, the improvement strategy proves to be a very valuable tool. For instance, for the H_7 instance, we were able to compute the global lower bound from $(\mathtt{SDP_{full}})$ in only $31.7 + 80.1 < 112$ seconds instead of the 6177.5 seconds needed in Table 11.2. Furthermore, for the H_8 instance, we closed the gap and hence proved global optimality. Overall we see that most of the gaps are reduced by between 0.2% and 1.5% with respect to those in Table 11.6. But there is significant variability: while for the H_8 instance we closed the gap and hence proved global optimality, the lower bound for SH_8 was not reduced at all. Similarly the computational times vary significantly even for instances of the same size. For instance, computing the $(\mathtt{SDP_{full}})$ bound for Rand_9 in this manner required only 77.7 seconds (less time than it took to compute the $(\mathtt{SDP_{cheap}})$ bound for it in Table 11.6) while the $(\mathtt{SDP_{full}})$ bound for SH_9 took over 9000 seconds.

We can control the computational effort involved in improving the lower bounds by considering a relaxation between $(\mathtt{SDP_{cheap}})$ and $(\mathtt{SDP_{full}})$ in the sense of setting a limit on the total number of inequality constraints that can be present in the relaxation. Motivated by the results from Tables 11.3 and 11.4, we consider the inequalities in the following order: (11.24), $\mathcal{DV}$, $\mathcal{LS}$, $\mathcal{M}$, $\mathcal{SI}$. We summarize the computational results, where we use at most 2000, 4000 or 6000 inequality constraints in Table 11.8. Whenever the fourth column of Table 11.8 has a zero, this means that we effectively solved $(\mathtt{SDP_{full}})$. Therefore we do not test those instances for larger values of the maximum number of inequalities.

Comparing the results for the instances with 9 departments in Tables 11.7 and 11.8 shows that limiting the number of constraints helps to reduce the computation time considerably without too much impact on the quality of the lower bounds. Changing the limit from 2000 to 4000 and from 4000 to 6000 constraints

	Improvement statistics		Global bounds (over all row assignments)			Computational statistics	
Instance	Nbr of instances with gap > 0	Nbr of instances improved	Lower bound	Upper bound	Gap (%)	Total time (sec)	Average final nbr of ineq constraints
H_7	3	1	1639.5	1700	3.69	80.1	1320.0
H_8	4	4	2385	2385	0	911.2	2457.5
S_8	26	1	380.5	408	7.23	1.9	336.0
SH_8	58	2	999.0	1135.5	13.66	20.3	1013.5
Rand_8	17	1	193	205	6.22	1713.0	3360.0
S_9	6	2	1168.0	1181.5	1.16	432.9	2888.0
SH_9	131	9	2009.5	2294.5	14.18	8806.9	2599.7
Rand_9	22	1	448.5	492.5	9.81	77.7	1516.0

Table 11.7: Computing the ($\mathtt{SDP_{full}}$) bounds starting with the ($\mathtt{SDP_{cheap}}$) relaxation

we observe that while the lower bound improves a little, the computation time grows significantly. When allowing at most 6000 constraints the computation times get already quite large and hence we do not consider adding even more constraints.

Starting the improvement strategy with ($\mathtt{SDP_{basic}}$) is not an attractive option because the bounds are much weaker than the ($\mathtt{SDP_{cheap}}$) bounds, and the number of relevant inequalities in $\mathcal{DV}$ is very small compared to the $\binom{n}{3}$ 3-cycle equalities.

11.6.6 Comparison of the Results for the SF-MRFLP with 2 to 5 rows

Next we compare the SF-MRFLP results for 2 rows with them for 3 to 5 rows. The number of different row assignments is given by the recursive formula

$$R(m,n) = \frac{m^n}{m!} - \sum_{k=1}^{m-1} \frac{R(k,n)}{(m-k)!},$$

where m is the number of rows and n is the number of departments. Applying $m \in \{2,3,4,5\}$ to the above formula yields:

$$R(2,n) = 2^{n-1} - 1, \qquad R(3,n) = \frac{3^{n-1} - 2^n + 1}{2},$$

$$R(4,n) = \frac{4^{n-1} - 3^n + 3 \cdot 2^{n-1} - 1}{6}, \qquad R(5,n) = \frac{5^{n-1} - 4^n + 2 \cdot 3^n - 2^{n+1} + 1}{24}.$$

We use the algorithmic approach and setup proposed in Subsections 11.6.1 and 11.6.3 and again restrict the running time per instance to 26 hours. In Tables 11.9, 11.10 and 11.11 we summarize the computational results obtained by this approach using ($\mathtt{SDP_{full}}$) and ($\mathtt{SDP_{cheap}}$) respectively. We also restate the according results for the DRFLP to facilitate the comparison.

The average number of active inequalities increases for increasing m for all our instances. This has a negative effect on the computing time and an overall positive influence on the bound quality when considering more than 2 rows. Additionally the number of row assignments is monotonically increasing for growing $m \in \{1,2,3,4,5\}$ and $n \geq 10$ which is another reason for the increase of the computing time for layouts with a larger number of rows.

Finally we compare the direct distance calculation (used so far in this section) with the "indirect" one proposed at the end of Section 11.3, for an illustration see Figure 11.1. The results obtained for 3 to 5 rows are summarized in Tables 11.12 and 11.13 applying ($\mathtt{SDP_{full}}$) and ($\mathtt{SDP_{cheap}}$) respectively.

Instance	Nbr of instances with gap > 0	Nbr of instances improved	Nbr of instances for which max nbr of ineqs added	Lower bound	Upper bound	Gap (%)	Total time (sec)	Average final nbr of ineq constraints
Maximum of 2000 inequality constraints								
S_9	6	2	1	1168.0	1181.5	1.16	262.7	1150.0
SH_9	131	8	4	2008	2294.5	14.27	822.5	1129.5
S_10	19	1	0	1319.5	1374.5	4.17	115.1	1760.0
Rand_10	84	1	1	788	838	6.35	128.1	2000.0
S_11	62	2	2	3334	3439.5	3.16	560.8	2000.0
Rand_11	5	1	1	649.5	708	9.01	155.0	2000.0
H_12	478	10	3	8482	8995	6.05	710.9	1079.4
Rand_12	3	1	1	777.5	799	2.77	216.3	2000.0
Rand_13	3	1	1	1061	1070	0.85	368.5	2000.0
Rand_14	6	2	1	1349	1393.5	3.30	307.0	1546.5
Maximum of 4000 inequality constraints								
Rand_10	84	2	1	792.5	838	5.74	1779.4	2305.0
S_11	62	2	0	3337.5	3439.5	3.06	9932.8	3761.0
Rand_11	5	1	1	652.5	708	8.51	1366.9	4000.0
H_12	478	10	3	8485	8995	6.01	5086.4	1694.3
Rand_12	3	1	1	783.5	799	1.98	1465.3	4000.0
Rand_13	3	1	1	1062.5	1070	0.71	2054.0	4000.0
Rand_14	6	2	1	1353.5	1393.5	2.96	1880.7	2546.5
Maximum of 6000 inequality constraints								
Rand_10	84	2	1	793	838	5.68	9891.8	3306.0
Rand_11	5	1	1	655.5	708	8.01	4173.0	6000.0
H_12	478	10	3	8485.5	8995	6.00	23844.3	2285.8
Rand_12	3	1	1	785.5	799	1.72	5968.7	6000.0
Rand_13	3	1	1	1063	1070	0.66	6905.5	6000.0
Rand_14	6	2	1	1355.5	1393.5	2.80	5588.9	3537.0

Table 11.8: Improved bounds starting from ($\texttt{SDP}_{\texttt{cheap}}$) and with limits on the number of inequality constraints

Again the computing time clearly grows with the instance size. The distance calculation chosen has no significant influence on the running time but a considerable effect on the gaps regardless if we apply ($\texttt{SDP}_{\texttt{full}}$) or ($\texttt{SDP}_{\texttt{cheap}}$):

1. The gaps are essentially smaller for the "indirect" distance calculation, i.e. in the direct distance calculation the objective function contains more quadratic terms in the inferior constrained variables V and hence the corresponding SDP relaxation provides less tight lower bounds. The difference between the two distance calculation types clearly grows with an increasing number of rows and also with an increasing number of departments.

2. For "indirect" distance calculation the average gaps decrease considerably for an increasing number of rows as this causes a decreasing number of quadratic terms in V.

11.7 Conclusions and Future Research

We proposed a new semidefinite programming approach for the space-free multi-row facility layout problem. This is the special case of multi-row layout in which all the rows have a common left origin and no empty space is allowed between departments. Our computational results show that for space-free double-row instances the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for instances with up to 15 departments and 5 rows.

The next step in extending our approach are the incorporation of spacing within the rows in the optimization process, and the use of the SDP approach within a suitable enumeration scheme to globally optimize instances of double-row and multi-row layout. The development of a post-processing algorithm or mathematical program for MRFLP based on the results of the SF-MRFLP would be another interesting direction of research.

			Global bounds (over all row assignments)			Statistics for the R(m,n) subproblems			Computational statistics	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time (sec)
2	15	H_5	420	450	7.14	7.14	1.54	7	130.3	15.8
	15	Rand_5	52.5	52.5	0	7.65	1.03	10	108.3	15.2
	31	H_6	703	720	2.42	10.96	3.30	7	385.7	118.7
	31	Rand_6	189.5	190.5	0.53	6.18	1.70	7	336.6	89.1
	63	H_7	1639.5	1700	3.69	14.05	2.69	6	954.0	2325.5
	63	Rand_7	166	166	0	13.21	2.20	16	774.4	1237.4
3	25	H_5	290	290	0	11.40	1.35	17	142.2	14.7
	25	Rand_5	34.5	34.5	0	11.64	1.04	17	144.6	14.2
	90	H_6	560	560	0	15.25	2.17	44	365.6	230.1
	90	Rand_6	107	110.5	3.27	9.54	1.25	38	335.7	208.0
	301	H_7	1135	1190	4.85	16.07	3.93	58	876.8	8120.0
	301	Rand_7	108	108	0	11.17	0.92	171	754.4	4642.1
4	10	H_5	140	140	0	2.22	0.22	9	271.3	7.5
	10	Rand_5	48.5	48.5	0	1.97	0.20	9	263.7	4.5
	65	H_6	410	410	0	20.35	0.67	55	459.6	102.6
	65	Rand_6	98.5	98.5	0	5.85	0.58	45	489.2	122.7
	350	H_7	820	820	0	19.32	2.32	142	1023.2	7422.4
	350	Rand_7	77	77	0	6.29	0.35	290	1040.7	3769.7
5	1	H_5	200	200	0	0	0	1	295.0	0.2
	1	Rand_5	44.5	44.5	0	0	0	1	337.0	0.2
	15	H_6	260	260	0	1.20	0.08	14	757.7	33.9
	15	Rand_6	94.5	94.5	0	0.91	0.10	13	776.0	25.1
	140	H_7	650	650	0	18.57	1.06	88	1493.3	5397.9
	140	Rand_7	61	61	0	3.48	0.18	128	1521.8	2377.9

Table 11.9: Computational results for ($\mathtt{SDP_{full}}$) for 2 to 5 rows

			Global bounds (over all row assignments)			Statistics for the R(m,n) subproblems			Computational statistics	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time (sec)
	15	H_5	407.5	450	10.43	10.43	2.52	5	12.9	4.5
	15	Rand_5	52.5	52.5	0	8.93	1.78	7	15.1	4.5
	31	H_6	665	720	8.27	16.95	5.78	4	19.5	12.3
	31	Rand_6	188.5	190.5	1.06	6.65	2.59	3	20.9	13.8
	63	H_7	1629	1700	4.36	15.17	3.75	4	30.8	31.7
	63	Rand_7	166	166	0	13.82	3.16	11	31.0	31.7
	127	H_8	2351	2385	1.45	21.22	5.80	1	42.9	86.8
	127	S_8	380.5	408	7.23	20.10	5.87	2	44.1	91.6
	127	SH_8	990.5	1135.5	14.64	17.00	10.85	4	56.1	87.4
	127	Rand_8	192	205	6.77	28.10	4.77	6	42.8	82.2
2	255	S_9	1163	1181.5	1.59	13.63	3.42	6	64.2	253.4
	255	SH_9	1974.5	2294.5	16.21	18.87	11.31	0	80.6	251.9
	255	Rand_9	447.5	492.5	10.06	19.73	5.61	3	55.8	252.6
	511	S_10	1314	1374.5	4.60	10.77	4.20	7	82.7	713.0
	511	Rand_10	779	838	7.57	15.16	5.68	0	78.7	698.2
	1023	S_11	3325.5	3439.5	3.43	14.92	5.16	6	106.8	2127.0
	1023	Rand_11	643.5	708	10.02	23.95	5.76	9	103.4	2048.5
	2047	H_12	8446.5	8995	6.49	17.31	6.19	0	125.1	6189.5
	2047	Rand_12	775.5	799	3.03	17.69	6.21	0	128.8	6389.5
	4095	Rand_13	1058	1070	1.13	19.11	5.98	0	159.5	20636.9
	8191	Rand_14	1335.5	1393.5	4.34	20.25	6.59	1	172.5	60845.6
	25	H_5	284.5	290	1.93	13.76	1.81	14	17.9	8.0
	25	Rand_5	34.5	34.5	0	11.64	1.46	15	17.8	7.6
	90	H_6	489.5	560	14.40	43.30	6.68	25	28.1	32.8
	90	Rand_6	101	110.5	9.41	9.54	2.13	21	30.0	36.4
	301	H_7	1074.5	1190	10.75	23.52	5.94	32	41.0	139.2
	301	Rand_7	108	108	0	13.79	1.92	117	47.0	157.0
	966	H_8	1393	1515	8.76	27.20	6.71	67	55.7	638.0
	966	S_8	244	260	6.56	17.79	5.66	9	56.7	709.1
	966	SH_8	647	740.5	14.45	21.01	10.17	4	60.8	633.6
3	966	Rand_8	128	138	7.81	21.61	4.71	144	60.7	653.3
	3025	S_9	758	771.5	1.78	16.15	4.51	91	81.6	3074.5
	3025	SH_9	1257.5	1431.5	13.84	21.80	11.78	5	85.0	2883.7
	3025	Rand_9	291.5	317.5	8.92	23.47	5.30	69	72.1	3055.0
	9330	S_10	847	887.5	4.78	20.01	5.73	38	104.2	13881.4
	9330	Rand_10	490.5	541	10.30	23.72	6.31	19	99.2	13384.4
	28501	S_11	2191	2308.5	5.36	20.38	6.83	63	184.1	64116.5
	28501	Rand_11	402	435.5	8.33	22.13	5.93	258	162.8	67056.2

Table 11.10: Computational results for $(\texttt{SDP}_{\texttt{cheap}})$ for 2 and 3 rows

			Global bounds (over all row assignments)			Statistics for the R(m,n) subproblems			Computational statistics	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time (sec)
4	10	H_5	140	140	0	2.22	0.22	9	21.5	3.5
	10	Rand_5	48.5	48.5	0	2.02	0.58	7	19.1	4.5
	65	H_6	410	410	0	25.71	2.73	36	32.2	34.9
	65	Rand_6	98.5	98.5	0	5.85	0.83	38	37.9	40.0
	350	H_7	820	820	0	28.89	4.44	78	47.4	232.2
	350	Rand_7	77	77	0	8.21	0.89	238	56.8	190.1
	1701	H_8	1135	1135	0	39.04	4.96	279	62.5	1211.1
	1701	S_8	177.5	188	5.92	20.80	5.19	43	61.6	1461.4
	1701	SH_8	467	491.5	5.25	14.25	6.75	9	65.5	1292.7
	1701	Rand_8	98	98	0	19.84	2.90	553	71.9	1411.1
	7770	S_9	551.5	587.5	6.53	18.70	5.06	171	128.9	8961.0
	7770	SH_9	951.5	1062.5	11.67	17.73	9.61	13	136.1	8276.2
	7770	Rand_9	200	212.5	6.25	22.10	4.02	530	101.9	9136.7
	34105	S_10	584.5	630.5	7.87	23.02	5.71	361	111.5	52796.6
	34105	Rand_10	352	363	3.13	22.28	5.03	320	134.9	59391.1
5	1	H_5	200	200	0	0	0	1	19.0	0.2
	1	Rand_5	44.5	44.5	0	0	0	1	16.0	0.2
	15	H_6	260	260	0	2.46	0.37	11	33.2	7.0
	15	Rand_6	94.5	94.5	0	0.91	0.10	13	39.5	6.7
	140	H_7	650	650	0	18.84	2.04	63	54.2	87.6
	140	Rand_7	61	61	0	6.14	0.38	121	61.8	83.3
	1050	H_8	875	875	0	29.53	3.28	336	68.9	981.6
	1050	S_8	141.5	146	3.18	13.81	3.51	78	64.2	1194.8
	1050	SH_8	413.5	413.5	0	9.52	3.25	54	71.8	911.8
	1050	Rand_8	82	82	0	14.29	1.36	638	83.8	967.2
	6951	S_9	472.5	472.5	0	17.42	4.84	230	132.5	8692.1
	6951	SH_9	792.5	817.5	3.15	12.41	6.33	8	139.6	8429.5
	6951	Rand_9	156.5	157.5	0.64	16.47	2.56	1202	109.2	9888.4
	42525	S_10	494	499.5	1.11	21.81	5.14	969	161.9	78783.0
	42525	Rand_10	263	272	3.42	17.11	3.60	2045	142.5	92148.8

Table 11.11: Computational results for ($\mathtt{SDP}_{\mathtt{cheap}}$) for 4 and 5 rows

			Global bounds (over all row assignments)			Statistics for the R(m,n) subproblems			Computational statistics	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time (sec)
3	25	H_5	380	390	2.63	5.26	0.85	12	130.5	12.8
	25	Rand_5	40.5	40.5	0	10.49	0.67	21	134.0	9.6
	90	H_6	610	610	0	10.57	1.28	51	317.9	164.1
	90	Rand_6	174.5	174.5	0	5.08	0.46	57	290.4	143.6
	301	H_7	1395	1440	3.23	13.14	1.60	119	811.5	9689.2
	301	Rand_7	156	156	0	10.28	0.47	225	566.4	3357.5
4	10	H_5	430	430	0	0.58	0.12	7	230.6	3.1
	10	Rand_5	64.5	64.5	0	1.63	0.16	9	224.1	2.2
	65	H_6	610	610	0	14.97	0.27	50	410.1	49.4
	65	Rand_6	195.5	195.5	0	2.81	0.12	56	439.8	72.2
	350	H_7	1388	1390	0.14	10.74	0.65	246	725.6	2793.6
	350	Rand_7	164	164	0	5.98	0.11	322	693.6	1547.5
5	1	H_5	430	430	0	0	0	1	343.0	0.1
	1	Rand_5	100.5	100.5	0	0	0	1	372.0	0.1
	15	H_6	637	640	0.47	0.47	0.09	11	624.9	15.0
	15	Rand_6	207.5	207.5	0	0	0	15	666.1	16.9
	140	H_7	1370	1370	0	7.37	0.25	111	1049.7	1334.7
	140	Rand_7	165	165	0	1.72	0.02	138	1083.7	973.6

Table 11.12: Computational results for ($\mathtt{SDP}_{\mathtt{full}}$) for 3 to 5 rows using the “indirect” distance calculation

			Global bounds (over all row assignments)			Statistics for the R(m,n) subproblems			Computational statistics	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time (sec)
3	25	H_5	380	390	2.63	9.05	1.62	10	18.5	6.6
	25	Rand_5	40.5	40.5	0	10.49	0.77	20	18.4	6.8
	90	H_6	567.5	610	7.49	33.33	3.99	41	31.9	31.8
	90	Rand_6	174.5	174.5	0	5.36	0.72	42	34.3	35.7
	301	H_7	1369.5	1440	5.15	14.86	2.75	66	50.1	135.1
	301	Rand_7	156	156	0	11.38	0.87	192	50.1	135.4
	966	H_8	2070.5	2225	7.46	15.82	3.06	135	61.9	701.9
	966	S_8	324	345	6.48	12.69	2.82	90	66.9	736.6
	966	SH_8	926	1041.5	12.47	13.83	5.39	32	90.3	749.1
	966	Rand_8	146.5	155	5.80	14.49	1.97	390	68.0	692.1
	3025	S_9	1044	1063.5	1.87	11.14	2.23	304	96.0	3558.0
	3025	SH_9	1852.5	2043.5	10.31	15.58	6.20	45	130.3	3284.1
	3025	Rand_9	347.5	354.5	2.01	14.47	2.19	439	78.8	2836.5
	9330	S_10	1134	1180.5	4.10	12.92	2.08	870	114.0	12499.7
	9330	Rand_10	645	670	3.88	14.74	2.53	708	105.0	12651.4
	28501	S_11	2806	2090.5	3.01	14.74	2.89	1433	143.0	67607.7
	28501	Rand_11	580	580	0	14.44	2.38	2344	135.0	67687.8
4	10	H_5	430	430	0	1.01	0.32	6	19.9	2.3
	10	Rand_5	64.5	64.5	0	1.63	0.31	8	20.7	2.2
	65	H_6	610	610	0	15.80	1.38	45	31.3	24.8
	65	Rand_6	194.5	195.5	0.51	2.81	0.18	50	33.5	28.4
	350	H_7	1336	1390	4.04	18.33	1.50	157	48.0	254.5
	350	Rand_7	164	164	0	7.76	0.22	294	48.9	262.7
	1701	H_8	1897	1945	2.53	12.27	1.39	635	63.1	1575.1
	1701	S_8	337.5	350	3.70	10.43	1.13	554	68.6	1366.1
	1701	SH_8	891.5	942.5	5.72	9.44	2.25	139	82.7	1334.8
	1701	Rand_8	144	144	0	11.07	0.53	1249	72.2	1581.3
	7770	S_9	967	1004.5	3.88	9.87	1.12	1917	93.0	8710.2
	7770	SH_9	1815.5	1920.5	5.78	9.56	2.94	180	112.3	9051.8
	7770	Rand_9	333.5	339.5	1.80	9.13	0.90	3112	81.6	8215.1
	34105	S_10	1177	1205.5	2.42	8.83	1.04	7677	111.7	52384.4
	34105	Rand_10	617	622	0.81	12.42	1.03	8278	106.9	55120.0
5	1	H_5	430	430	0	0	0	1	23.0	0.2
	1	Rand_5	100.5	100.5	0	0	0	1	21.0	0.1
	15	H_6	638.5	640	0.23	0.29	0.04	12	33.4	6.2
	15	Rand_6	207.5	207.5	0	0.39	0.03	14	36.1	8.3
	140	H_7	1370	1370	0	9.91	0.66	94	50.2	97.2
	140	Rand_7	165	165	0	1.72	0.06	133	49.8	93.6
	1050	H_8	1845	1845	0	10.01	0.70	654	65.5	856.7
	1050	S_8	314	323	2.87	5.13	0.34	682	76	821.1
	1050	SH_8	861	863.5	0.29	4.30	0.70	295	80.6	1020.5
	1050	Rand_8	144	144	0	5.19	0.13	948	75.4	799.5
	6951	S_9	904	912.5	0.94	5.97	0.44	3643	100.4	7373.8
	6951	SH_9	1673	1724.5	3.08	5.75	1.21	493	96.0	8534.8
	6951	Rand_9	348.5	348.5	0	7.95	0.34	4628	86.5	7427.0
	42525	S_10	1147.5	1147.5	0	8.19	0.52	17752	117.0	64136.0
	42525	Rand_10	625.5	645	3.12	8.11	0.50	19364	112.7	73729.2

Table 11.13: Computational results for ($\mathtt{SDP_{cheap}}$) for 3 to 5 rows using the "indirect" distance calculation

Chapter 12

A Semidefinite Optimization-Based Approach for Global Optimization of Multi-Row Facility Layout

Authors: Philipp Hungerländer and Miguel F. Anjos

Abstract: This paper is concerned with the Multi-Row Facility Layout Problem. Given a set of rectangular facilities, a fixed number of rows, and weights for each pair of departments, the problem consists of finding an assignment of departments to rows and the positions of the departments in each row so that the total weighted sum of the center-to-center distances between all pairs of departments is minimized. We show how to extend our recent approach for the Space-Free Multi-Row Facility Layout Problem to general Multi-Row Facility Layout as well as some special cases thereof. To the best of our knowledge this is the first global optimization approach for multi-row layout that is applicable beyond the double-row case. A key aspect of our proposed approach is a model for multi-row layout that expresses the problem as a discrete optimization problem, and thus makes it possible to exploit the underlying combinatorial structure. In particular we can explicitly control the number and size of the spaces between departments. We construct a semidefinite relaxation of the discrete optimization formulation and present computational results showing that the proposed approach gives promising results for several variants of multi-row layout problems on a variety of benchmark instances.

Keywords: Facilities planning and design; Flexible manufacturing systems; Semidefinite Programming; Combinatorial Optimization; Global Optimization

12.1 Introduction

The general facility layout design problem is concerned with placing departments of given areas within a given facility. To each possible placement is assigned a cost based on the interactions between each pair of departments. These costs reflect an appropriate measure of adjacency preferences between departments. In some versions of the problem, the dimensions of the departments are also given. When this is not the case, finding their optimal shape is also a part of the problem. This problem is known to be NP-hard in general.

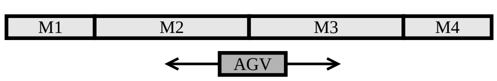
Versions of the facility layout problem occur in many practical contexts, not only in the planning of production and logistics facilities but also in applications such as VLSI chip design. A thorough survey of the facility layout problem is given in Meller and Gau (1996), where the papers on facility layout are divided into three broad areas. The first is concerned with algorithms for tackling the layout problem as defined above. The second area is concerned with extensions of the problem in order to account for additional issues that arise in applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and achieving layouts that optimize two or more objectives simultaneously. The third area is concerned with specially structured instances of the problem, such as the linear layout of machines along production lines. This paper is concerned with one such structured instance, namely the Multi-Row Facility Layout Problem (MRFLP).

An instance of the Multi-Row Facility Layout Problem (MRFLP) consists of a set of rectangular facilities, a given number of rows, and weights for each pair of departments. We assume without loss of generality that each department can be assigned to any of the given rows. The problem is to find an assignment of departments to rows and the positions of the departments in each row so that the total weighted sum of the center-to-center distances between all pairs of departments is minimized. There has not been much research done on metaheuristic approaches to row layout problems with spacing, see e.g. the recent paper of Murray et al. [230]. In this paper, we focus on mathematical programming approaches that can certify global optimality of solutions, or at least provide a guaranteed bound on the gap to optimality.

Row layout problems in general are of special interest for optimizing flexible manufacturing systems (FMSs). FMSs are automated production systems, typically consisting of numerically controlled machines and material handling devices under computer control, which are designed to produce a variety of parts. In FMSs the layout of the machines has a significant impact on the materials handling cost and time, on throughput, and on productivity of the facility. A poor layout may also negate some of the flexibilities of an FMS [133]. The type of material-handling devices used such as handling robots, automated guided vehicles (AGVs), and gantry robots typically determines machine layout in an FMS [232].

Possible row layout types are single-row (Figure 12.1), double-row and multi-row layout (Figure 12.2). The Single-Row Facility Layout Problem (SRFLP) requires that the departments be placed next to each other along a single row; this simplifies the problem significantly. In particular, there is no need to assign each department to a row, and the optimal solution will not have any empty space between departments. Therefore solving the (SRFLP) consists of finding the optimal permutation of the departments. This problem arises for example as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle (AGV) travelling in both directions on a straight-line path [150] (see Figure 12.1). Several heuristic algorithms have been suggested to tackle large instances; the best ones to date are Datta et al. [80], Kothari and Ghosh [188] and Samarghandi and Eshghi [267].

a.)



b.)

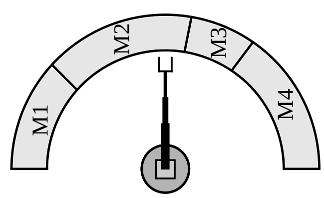


Figure 12.1: In a.) an AGV transports parts between the machines moving in both directions along a straight line. In b.) a material-handling industrial robot carries parts between the machines.

Three problems closely related to the (SRFLP) are the Single-Row Equidistant Facility Layout Problem (SREFLP), the Linear Arrangement Problem (LA), and the k-Parallel Row Ordering Problem (kPROP). The (SREFLP) is the special case of the (SRFLP) with all departments equal in shape and the positions where they can be placed on the rows fixed in advance; it is sometimes also called one-dimensional machine location problem [269] or linear machine-cell location problem [308]. On the other hand, the (kPROP) is the an extension of the (SRFLP) that considers arrangements of the departments along more than one row but with each department being assigned to a specific row in advance; hence the objective of the (kPROP) is to find a permutation of the departments within each row so that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized.

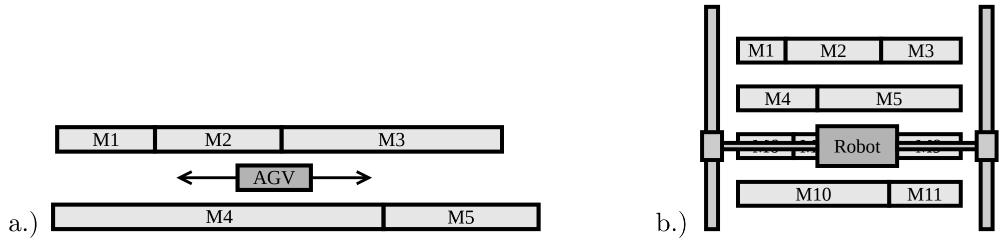


Figure 12.2: In a.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. In b.) a gantry robot is used when the space is limited.

If the (kPROP) is restricted to two rows we simply call it (PROP). Applications of the (kPROP) are the arrangement of departments along two or more parallel straight lines on a floor plan, the construction of multi-floor buildings, and the layout of machines in FMSs. We mention that even the (LA), which is a special case of the (SREFLP) where all weights are binary, is already an NP-hard problem [112], and it remains so even if the underlying graph is bipartite [111].

In layout problems, the pairwise connectivities are usually assumed to be non-negative to ensure boundedness of the objective value of the optimal layout. For the (SRFLP) and the other problems closely related to it, this further guarantees that all departments are placed next to each other without spacing. More generally, the (kPROP) can be further extended to the Space-Free Multi-Row Facility Layout Problem (SF-MRFLP) in which the optimization is also carried out with respect to the row assignments. This is a particular version of the (MRFLP) in which all the rows have a common left origin and no empty space is allowed between departments [164]. If we restrict the (SF-MRFLP) to two rows we obtain the Space-Free Double-Row Facility Layout Problem (SF-DRFLP) as a special case that has been applied in spine layout design. Spine layouts, introduced by Tompkins [295], require departments to be located along both sides of specified corridors along which all the traffic between departments takes place. Although in general some spacing is allowed, layouts with no spacing are much preferable since spacing often translates into higher construction costs for the facility.

Finally, the (MRFLP) is a generalization of the (SF-MRFLP) in which the rows may not have a common left origin and space is allowed between departments. This is the most general version of row layout problems. The (MRFLP) has many applications such as computer backboard wiring [288], campus planning [90], scheduling [115], typewriter keyboard design [253], hospital layout [95], the layout of machines in an automated manufacturing system [151], balancing hydraulic turbine runners [198], numerical analysis [46] and optimal digital signal processors memory layout generation [302].

The special case of the Double-Row Facility Layout Problem (DRFLP) can be viewed as a natural extension of the (SRFLP) in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel (see Figure 12.2). This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the (DRFLP) and its space-free counterpart (SF-DRFLP), sometimes called the corridor allocation problem, are especially relevant for real-world applications.

Toy Example. Next let us further clarify the workings and differences of the (SRFLP), the (kPROP), the (SF-MRFLP) and the (MRFLP) with the help of a toy example: We consider 4 departments with lengths $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4$. Additionally we are given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$ for the row layout problems. Figure 12.3 illustrates the optimal layouts and the associated costs for the four different combinatorial optimization problems:

- In a.) we display the optimal layout for the (SRFLP) with corresponding total cost of $3 \cdot 2 + 2.5 \cdot 1 +$

$2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5.$

- In b.) we show the optimal layout for the (PROP) with departments 2 and 3 assigned to row 1 and departments 1 and 4 assigned to row 2. The corresponding cost is $3.5{\cdot}1{+}1{\cdot}2{+}1{\cdot}2{+}1.5{\cdot}1{+}2.5{\cdot}1 = 11.5$.

- In c.) we display the optimal layout for the (SF-DRFLP). The associated cost is $1.5 \cdot 1 + 2 \cdot 2 + 0.5 \cdot 1 + 1 \cdot 2 + 2.5 \cdot 1 = 10.5$.

- Finally we depict the optimal layout for the (DRFLP) in d.). The corresponding cost is $2.5 \cdot 1 + 2.5 \cdot 1 + 2.5 \cdot 1 = 7.5$.

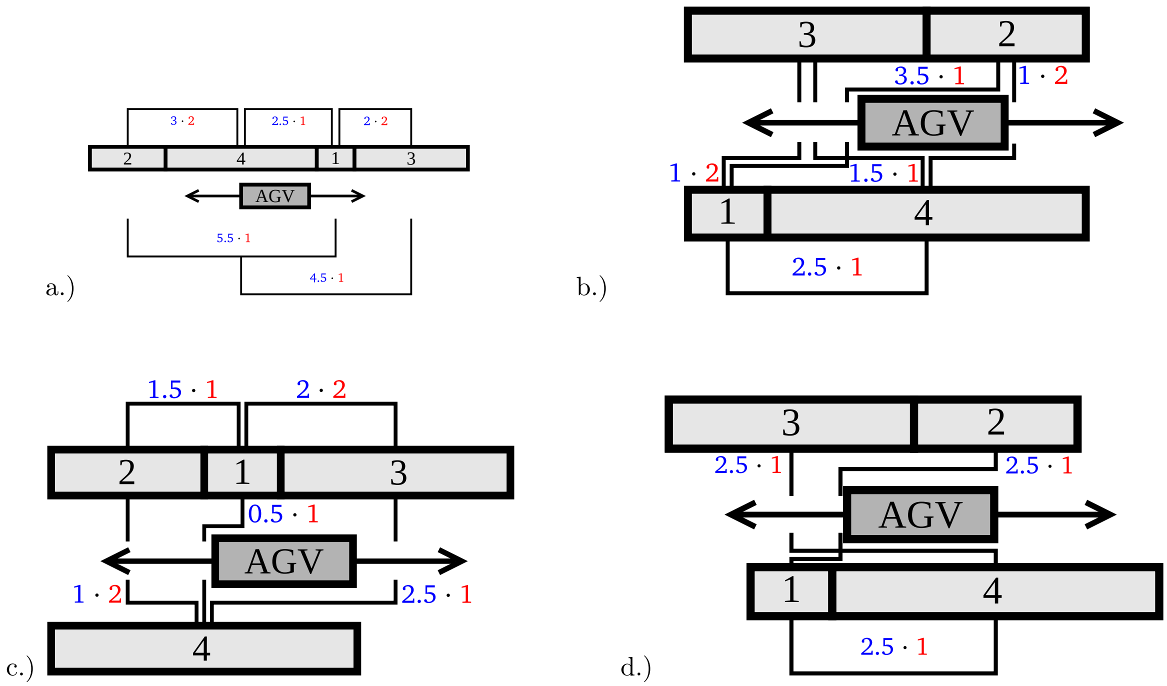


Figure 12.3: Optimal layouts for four different row layout problems.

The contribution of this paper is a new model for (MRFLP) that expresses the problem as a discrete optimization problem. This makes it possible to exploit the underlying combinatorial structure, and in particular to explicitly control the number and size of the spaces between departments. This discrete optimization model is an extension of our recent approach for the (SF-MRFLP) to the more general (MRFLP) and also captures some of its important special cases. We then construct a semidefinite relaxation of the discrete optimization formulation and present computational results showing that the proposed approach gives promising results for several variants of multi-row layout problems on a variety of benchmark instances. To the best of our knowledge this is the first global optimization approach for (MRFLP) that is applicable beyond the double-row case.

The paper is structured as follows. In Section 12.2 we summarize the various global optimization approaches in the literature for row layout problems closely related to multi-row layout, including the (SF-MRFLP). Section 12.3 presents the theoretical results underpinning our new discrete optimization model for (MRFLP). The key result is Theorem 12.2 showing that if all departments lengths are integer then there is always an optimal solution to the (MRFLP) on the half grid. Section 12.4 covers the main techniques used for solving the SDP relaxations, and detailed computational results are reported in Section 12.5.

12.2 Previous Exact Approaches for Row Layout Problems

In this section we review the literature on global optimization approaches for the single-row, double-row and space-free versions of the (MRFLP), with an emphasis on the most recent developments. This sets the foundations for the methodology we propose in Section 12.3 for the general (MRFLP).

12.2.1 Single-Row Facility Layout

The (SRFLP) is one of the few layout problems for which strong global lower bounds and even optimal solutions can be computed for instances of reasonable size. Pioneering approaches for (SRFLP) were based on dynamic programming [252], integer linear optimization (ILO) [215] and nonlinear optimization [150]. We outline here the more recent global optimization approaches that are based on either ILO or semidefinite optimization (SDO) relaxations.

Let $\pi = (\pi_1, \dots, \pi_n)$ denote a permutation of the indices $[n] := \{1, 2, \dots, n\}$ of the departments. Given π and two distinct departments i and j, the center-to-center distance between i and j with respect to this permutation is $\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j$, where $D_\pi(i,j)$ denotes the sum of the lengths of the departments between i and j in the ordering defined by π. In other words, the (SRFLP) is:

$$\min_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} \left[\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j \right],$$

where Π_n denotes the set of all permutations of $[n]$, and the weights c_{ij} are usually assumed to be non-negative to ensure boundedness of the objective value of the optimal layout. It was already observed by Simmons [283] that the crux of the problem is to minimize $\sum\limits_{i<j} c_{ij} D_\pi(i,j)$ over all permutations $\pi \in \Pi_n$. It is also clear that $D_\pi(i,j) = D_{\pi'}(i,j)$, where π' denotes the permutation symmetric to π, defined by $\pi'_i = \pi_{n+1-i},\ i \in [n]$. Hence, it is possible to simplify the problem by considering only the permutations for which, say, department 1 is on the left half of the arrangement. This type of symmetry-breaking strategy is important for reducing the computational requirements of most algorithms, including those based on LO or dynamic programming.

It is also clear that the key concept required for modelling the (SRFLP) is *betweenness*. This is because $D_\pi(i,j)$ is precisely the sum of the lengths of the departments between i and j. We thus define the betweenness variables ζ_{ijk} for triplets i, j, k of departments

$$\zeta_{ijk} = \begin{cases} 1, & \text{if department } k \text{ lies between departments } i \text{ and } j, \\ 0, & \text{otherwise} \end{cases}$$

and the betweenness polytope as the convex hull of the incidence vectors ζ_{ijk} corresponding to feasible solutions:

$$\mathcal{P}_{Btw} = \operatorname{conv}\left\{ \zeta \in \{0,1\}^{n(n-1)(n-2)} : \zeta \text{ arises from a permutation of } [n] \right\}.$$

This polytope is well understood, see e.g. Christof [66]. Using betweenness variables, one can express $D_\pi(i,j) = \sum_k \ell_k \zeta_{ijk}$ and hence the (SRFLP) as a linear optimization problem over the betweenness polytope.

The ILO model proposed by Amaral [8] uses precisely this idea. Since an exact description of $\mathcal{P}_{Btw}$ is not known, Amaral proposes to optimize over a partial description of it. The starting LO relaxation is:

$$\begin{array}{lll}
\min & \sum_{i<j} c_{ij} \left[\frac{1}{2}(\ell_i + \ell_j) + \sum_{k \neq i,j} \ell_k \zeta_{ijk} \right] & \\
\text{s.t.} & \zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, & \text{for all } \{i,j,k\} \subseteq [n], \\
& \zeta_{ijd} + \zeta_{jkd} - \zeta_{ikd} \geq 0, & \text{for all } \{i,j,k,d\} \subseteq [n], \\
& \zeta_{ijd} + \zeta_{jkd} - \zeta_{ikd} \geq 0, & \text{for all } \{i,j,k,d\} \subseteq [n], \\
& \zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \leq 2, & \text{for all } \{i,j,k,d\} \subseteq [n], \\
& 0 \leq \zeta_{ijk} \leq 1, & \text{for all } \{i,j,k\} \subseteq [n],
\end{array}$$

where the indices i, j, k, d are always all different. Amaral also proposes a class of valid inequalities for $\mathcal{P}_{Btw}$ that can be used as cuts to improve the (LP) relaxation:

$$\sum_{t<q, t \in S_1, q \in S_1} \zeta_{tqr} + \sum_{t<q, t \in S_2, q \in S_2} \zeta_{tqr} - \sum_{t \in S_1, q \in S_2} \zeta_{\min\{t,q\},\max\{t,q\},r} \leq 0 \qquad (12.1)$$

where $\beta \leq n$ is a positive even integer, $S \subseteq [n]$ such that $|S| = \beta$, and (S_1, S_2) is a partition of $S \backslash \{r\}$ such that $r \in S$ and $|S_1| = \frac{1}{2}\beta$. It is easy to check that for $\beta = 4$, (12.1) is a triangle inequality already present in the LO relaxation, Amaral uses inequalities arising from $\beta = 6$ for computational purposes. His approach is able to solve instances with up to 35 departments to optimality.

There are many interesting and important connections between the betweenness polytope and the well-known cut polytope. For instance, the classes of inequalities above are known to be facet-defining for the cut polytope, and Sanjeevi and Kianfar [268] showed that they remain facet-inducing for the face of the cut polytope induced by $\mathcal{P}_{Btw}$.

A different approach to model the (SRFLP) is to use $\binom{n}{2}$ continuous distance variables $z_{ij}, i, j \in [n]$ corresponding to each pair of departments. The polytope containing the feasible positive distance variables z_{ij} for n departments is called the distance polytope:

$$\mathcal{P}^n_{Dis} := \text{conv}\left\{ z \in \mathbb{R}^{\binom{n}{2}} : \exists \pi \in \Pi \ : \ z_{ij} = z^{\pi}_{ij}, \ i, j \in \mathcal{N}, i < j \right\}.$$

Recently Amaral and Letchford [13] achieved significant progress in improving this approach by identifying several classes of valid inequalities and using them as cutting planes. In particular they show that the equation

$$\sum_{i,j \in \mathcal{N}, i<j} \ell_i \ell_j z_{ij} = \frac{1}{6}\left[\left(\sum_{i \in \mathcal{N}} \ell_i \right)^3 - \sum_{i \in \mathcal{N}} \ell_i^3 \right],$$

defines the smallest linear subspace that contains $\mathcal{P}^n_{Dis}$. They prove that clique inequalities, strengthened pure negative type inequalities and special types of hypermetric inequalities induce facets of $\mathcal{P}^n_{Dis}$. They further show the validity of rounded psd inequalities and star inequalities for $\mathcal{P}^n_{Dis}$ and use them together with the facet inducing inequalities as cutting planes in a Branch-and-Cut approach that can solve instances with up to 30 departments to optimality.

Alternatively, semidefinite programming (SDP) models have been proposed to obtain tight global bounds for the (SRFLP). (SDP) is the extension of (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. The handbooks edited by Wolkowicz et al. [303] and Anjos and Lasserre [18] provide a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where (SDP) has had a major impact.

The first (SDP) relaxation was proposed in Anjos et al. [22] and was used by Anjos and Vannelli [20] to compute globally optimal solutions for instances with up to 30 departments. A more compact relaxation that can compute useful bounds for instances with up to 100 departments was proposed by Anjos and Yen [21]. The state-of-the-art is the approach of Hungerländer and Rendl [169] that uses an improved (SDP) relaxation to compute globally optimal solutions for instances with up to 42 departments and bounds for instances with up to 100 departments. These approaches are all based on modelling betweenness using products of binary ± 1 variables y_{ij} defined as

$$y_{ij} := \begin{cases} 1, & \text{if } \pi_j < \pi_i, \\ -1, & \text{if } \pi_i < \pi_j. \end{cases}$$

for each pair of integers ij with $1 \leq i < j \leq n$, and for a given permutation π of $[n]$. The order of the subscripts matters, and $y_{ij} = -y_{ji}$. Thus, department k is between i and j if and only if $y_{ik}y_{jk} = -1$. Furthermore, given a particular assignment of ± 1 values to the y_{ij} variables, this assignment represents a permutation of $[n]$, if the transitivity condition

if i is to the right of j and j is to the right of k, then i is to the right of k

is fulfilled. This necessary condition can be formulated as a set of quadratic constraints:

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1 \text{ for all triples } 1 \leq i < j < k \leq n. \tag{12.2}$$

The resulting formulation of the (SRFLP) is:

$$\begin{aligned} \min \quad & \sum_{i<j} c_{ij} \left[\tfrac{1}{2}(\ell_i + \ell_j) + \sum_{k<i} \ell_k \left(\tfrac{1 - y_{ki}y_{kj}}{2} \right) + \sum_{i<k<j} \ell_k \left(\tfrac{1 + y_{ik}y_{kj}}{2} \right) + \sum_{j<k} \ell_k \left(\tfrac{1 - y_{ik}y_{jk}}{2} \right) \right] \\ \text{s.t.} \quad & \\ & y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1 \text{ for all triples } i < j < k, \\ & y_{ij}^2 = 1 \text{ for all } i < j. \end{aligned} \tag{12.3}$$

One advantage of the SDO-based approaches is that they implicitly account for the symmetry in the (SRFLP) because if every y_{ij} variable is replaced by its negative, then there is no change whatsoever to the formulation. (We note in passing that Buchheim et al. [48] proved (12.2) is the minimum equation system for the quadratic linear ordering problem, and we refer the reader to the survey chapter by Anjos and Liers [19] for more details concerning the problem and corresponding global optimization approaches.)

To obtain SDO relaxations we collect the ordering variables in a vector y and define the matrix $Y := yy^\top$. The main diagonal entries of Y correspond to y_{ij}^2 and hence $\operatorname{diag}(Y) = e$, the vector of all ones. Now we can formulate the (SRFLP) as a semidefinite optimization program as was first proposed in Anjos et al. [22]:

$$\min \{ \langle C, Y \rangle + K : \; Y \text{ satisfies (12.2)}, \; \operatorname{diag}(Y) = e, \; \operatorname{rank}(Y) = 1, \; Y \succcurlyeq 0 \}, \tag{12.4}$$

where the cost matrix C and the constant K are deduced from (12.3). Dropping the rank constraint yields the basic semidefinite relaxation of the (SRFLP)

$$\min \{ \langle C, Y \rangle + K : \; Y \text{ satisfies (12.2)}, \; \operatorname{diag}(Y) = e, \; Y \succcurlyeq 0 \}, \tag{SDP$_1$}$$

providing a lower bound on the optimal value of the (SRFLP).

Because the matrices Y with rank equal to 1 have only $\{-1, 1\}$ entries, Anjos and Vannelli [20] proposed to tighten (SDP$_1$) by adding the triangle inequalities. These inequalities are known to be facet-defining for the cut polytope, see e.g. Deza and Laurent [88], and they define the metric polytope $\mathcal{M}$:

$$\mathcal{M} = \left\{ Y : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{i,j} \\ Y_{j,k} \\ Y_{i,k} \end{pmatrix} \leq e, \; i < j < k \in \binom{n}{2} \right\}. \tag{12.5}$$

Adding the triangle inequalities to $(\mathtt{SDP}_1)$, we obtain the following relaxation of the (SRFLP)

$$\min \{ \; \langle C, Y \rangle + K : \; Y \text{ satisfies (12.2)}, \; Y \in \mathcal{M}, \; \operatorname{diag}(Y) = e, \; Y \succcurlyeq 0 \; \}. \tag{$\mathtt{SDP}_2$}$$

To reduce the computational burden, Anjos and Vannelli [20] do not solve $(\mathtt{SDP}_2)$ directly but rather use the $\approx \frac{1}{12}n^6$ triangle inequalities as cutting planes. This approach made it possible to solve instances of (SRFLP) with up to 30 departments to global optimality.

Hungerländer and Rendl [169] suggested a further strengthening of $(\mathtt{SDP}_2)$ as well as an alternative algorithmic approach to handle the resulting large number of inequalities. They use the matrix

$$Z := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{12.6}$$

and relax the equation $Y - yy^\top = 0$ to

$$Y - yy^T \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0,$$

which is convex due to the Schur-complement lemma. Note that $Z \succcurlyeq 0$ is in general a stronger constraint than $Y \succcurlyeq 0$. Additionally they use an approach suggested by Lovász and Schrijver [214] to further improve the relaxation by adding the following inequalities:

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \qquad & i < j < k \in [n], \; l < m \in [n] \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \qquad & i < j < k \in [n], \; l < m \in [n] \end{aligned} \tag{12.7}$$

The inequalities (12.7) are generated by multiplying the valid 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0,$$

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. These constraints define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{ \; Z : Z \text{ satisfies (12.7)} \; \}, \tag{12.8}$$

consisting of $\approx \frac{1}{3}n^5$ constraints. The result is the following relaxation of the (SRFLP):

$$\min \{ \; \langle C, Y \rangle + K : \; Y \text{ satisfies (12.2)}, \; Z \in (\mathcal{M} \cap \mathcal{LS}), \; \operatorname{diag}(Z) = e, \; Z \succcurlyeq 0 \; \}. \tag{$\mathtt{SDP}_{\mathtt{standard}}$}$$

To make $(\mathtt{SDP}_{\mathtt{standard}})$ computationally tractable, Hungerländer and Rendl [169] handle the triangle inequalities (12.5) and LS-cuts (12.7) through Lagrangian duality. This allowed them to solve instances with up to 42 departments to optimality and obtain tight global bounds for instances with up to 110 departments.

The semidefinite approach we have described is also the best method known for the (SREFLP) [163] and one of the best methods for the (LA) on dense graphs Hungerländer and Rendl [168]. We will use a similar approach in Section 12.4.

12.2.2 Parallel Row and Space-Free Multi-Row Layout

None of the approaches for the (SRFLP) can be applied directly to the (MRFLP) because when two or more rows are available, the betweenness variables do not suffice to model the problem. There are three additional modeling issues that arise in the (MRFLP) but not in the (SRFLP):

1. Expressing the center-to-center distance between departments assigned to different rows;
2. Assigning each department to exactly one row;
3. Handling the possibility of empty space between departments.

We summarize in this section the approaches that have been proposed to handle 1 and 2 by extending the models for the (SRFLP) in Subsection 12.2.1.

On the ILO side, Amaral [12] extended the approach for the (SRFLP) based on distance variables [6] to the (PROP) and was able to provide optimal solutions for instances with up to 23 departments. Amaral [10] also suggested mixed integer linear programming formulation based on distance variables for the (SF-DRFLP) that allows him to solve instances with up to 13 departments to optimality.

On the (SDP) side, the authors recently extended the (SDP)-based methodology for (SRFLP) to the (kPROP) and the (SF-MRFLP). The recent paper by Hungerländer and Anjos [164] includes results suggesting that the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for instances with up to 15 departments and 5 rows. For the (kPROP) tight global bounds can be computed for instances with up to 100 departments and for an arbitrary number of rows [159].

To generalize the (SDP) approach from the (SRFLP) to the (kPROP) and the (SF-MRFLP) we introduce the function $r:\ [n] \to \mathcal{R}$ that assigns each department to one of the m rows $\mathcal{R} := \{1, \dots, m\}$. Now the center-to-center distances of departments from different rows are expressed as quadratic terms in ordering variables:

$$z_{ij} = \frac{1}{2}(\ell_i + \ell_j) + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ki}y_{kj}}{2} + \sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k \frac{1 + y_{ik}y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ik}y_{jk}}{2},\ r(i) = r(j), \tag{12.9a}$$

$$\begin{aligned} z_{ij} = y_{ij} \Bigg[& \Bigg(\frac{\ell_j}{2} + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k \frac{1 + y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k \frac{1 - y_{jk}}{2} \Bigg) \\ & - \Bigg(\frac{\ell_i}{2} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1 + y_{ki}}{2} + \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k \frac{1 - y_{ik}}{2} \Bigg) \Bigg], \qquad r(i) \neq r(j). \end{aligned} \tag{12.9b}$$

where additionally the distances between pairs of departments in non-adjacent rows have to be non-negative:

$$z_{ij} \geq 0, \qquad i, j \in [n],\ i < j,\ r(i) \neq r(j). \tag{12.10}$$

Now building on the (SDP) formulations of the (SRFLP) we can formulate the (kPROP) as a semidefinite optimization problem.

Theorem 12.1 *The following optimization problem is equivalent to the* (kPROP)*:*

$$\min\ \{\ \langle C_d, Z\rangle : Z \text{ satisfies } (12.2) \text{ and } (12.10)\ ,\ Z \in \mathcal{E},\ y \in \{-1, 1\}\ \}$$

where the cost matrix C_d is deduced by equating the coefficients of the following equation

$$
\begin{aligned}
2\langle C_d, Z\rangle \overset{!}{=} & \sum_{\substack{i<j\in[n],\\ r(i)=r(j)}} c_{ij}\left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik} y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj}\right) \\
& + \sum_{\substack{i<j\in[n],\\ r(i)\neq r(j)}} c_{ij} y_{ij} \left(L_{r(i)} - L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} \right. \\
& \left. - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right) + \sum_{h\in\mathcal{R}} \left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij} \right) \left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i \right) \right],
\end{aligned}
$$

and L_i denotes the sum of the length of the departments on row i

$$L_i = \sum_{\substack{k\in[n],\\ r(k)=i}} \ell_k, \quad i\in\mathcal{R}.$$

Proof. Since $y_i^2 = 1,\ i\in\{1,\dots,\Delta-1\}$ we have $\operatorname{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. The equations (12.2) model transitivity for $y \in \{-1, 1\}$ [48]. Hence the integrality on Y together with (12.2) and (12.10) suffice to describe all feasible layouts of the (kPROP) and the objective function $\langle C_d, Z\rangle$ gives the costs for feasible layouts. □

As we are able to model the (kPROP) using the same variables as the (SRFLP), namely products of ordering variables, it makes sense to adopt the strongest (SDP) relaxation from the previous section:

$$\min\{\ \langle C_d, Z\rangle + K:\ Y \text{ satisfies (12.2) and (12.10)},\ Z\in(\mathcal{M}\cap\mathcal{LS}),\ \operatorname{diag}(Z)=e,\ Z\succcurlyeq 0\ \}. \quad (\texttt{SDP}_{\texttt{standard}})$$

However, because the objective function of the (kPROP) is more complex than that of the (SRFLP), additional valid inequalities are necessary to tighten the relaxation. Specifically we may consider triangle inequalities relating the distances between three departments:

$$z_{ij} + z_{ik} \geq z_{jk}, \qquad z_{ij} + z_{ik} \geq z_{jk}, \qquad z_{ik} + z_{jk} \geq z_{ij}, \qquad i<j<k\in[n]. \tag{12.11}$$

An easy inductive argument further shows that the above constraints imply the associated constraints for more than three departments. Hence we define the polytope $\mathcal{DV}$ containing the $3\binom{n}{3}$ triangle inequalities relating the distances between three or more departments:

$$\mathcal{DV} := \{\ Z : Z \text{ satisfies (12.11)}\ \}. \tag{12.12}$$

Adding these constraints to ($\texttt{SDP}_{\texttt{standard}}$) yields ($\texttt{SDP}_4$):

$$\min\{\ \langle C_d, Z\rangle + K:\ Y \text{ satisfies (12.2) and (12.10)},\ Z\in(\mathcal{M}\cap\mathcal{LS}\cap\mathcal{DV}),\ \operatorname{diag}(Z)=e,\ Z\succcurlyeq 0\ \}. \quad (\texttt{SDP}_4)$$

It was demonstrated in Hungerländer and Rendl [168] that using $\mathcal{LS}$ in the semidefinite relaxation pays off in practice for several ordering problems including the (SRFLP). However for the (kPROP) and further extensions, these additional tightening constraints did not pay off in terms of practical performance. For this reason, it is preferable to work with the following SDP relaxation

$$\min \{ \langle C_d, Z\rangle + K : \ Y \text{ satisfies (12.2) and (12.10)}, \ Z \in (\mathcal{M} \cap \mathcal{DV}), \ \operatorname{diag}(Z) = e, \ Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_3)$$

The relaxation ($\texttt{SDP}_3$) can be solved with the same algorithmic tools as used for the most competitive approach for the (SRFLP) (see Section 12.4 below for details). To extend this approach from the (kPROP) to the (SF-MRFLP), Hungerländer and Anjos [164] solve ($\texttt{SDP}_3$) for all possible row assignments (e.g. $2^{n-1} - 1$ in the double-row case). From the obtained bounds, one can deduce global upper and lower bounds: these are the minima of all upper and lower bounds respectively.

12.2.3 Double-Row Facility Layout

The (DRFLP) has been much less studied in the literature than the (SRFLP). We summarize here the two LO appraoches that extend in different ways the LO models proposed for the (SRFLP). The key difference between the two approaches is in the number of binary variables: the first approach uses $O(n^3)$ binary variables while the second uses only $O(n^2)$. As a consequence, the structure of the two models is very different.

The first approach [72, 310] defines the binary variables as follows:

$$y_{ik} = \begin{cases} 1, & \text{if department } i \text{ is placed in row } k, \\ 0, & \text{otherwise.} \end{cases}$$

$$z_{kij} = \begin{cases} 1, & \text{if departments } i \text{ and } j \text{ are both placed in row } k \text{ and } j \text{ is located to the right of } i, \\ 0, & \text{otherwise.} \end{cases}$$

In other words, the variables y encode the assignment of departments to rows and the variables z encode the relative position of two departments assigned to the same row. The resulting model is:

$$\begin{array}{lll}
\min & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij}(v_{ij}^{+} + v_{ij}^{-}) & \\
\text{s.t.} & x_{ik} \leq M y_{ik} & \text{for } i = 1, \ldots, n, \ k = 1, \ldots, m \\
& \sum_{k=i}^{m} y_{ik} = 1 & \text{for } i = 1, \ldots, n \\
& \frac{\ell_i y_{ik} + \ell_j y_{jk}}{2} + a_{ij} z_{kji} \leq x_{ik} - x_{jk} + M(1 - z_{kji}) & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n, \ k = 1, \ldots, m \\
& \frac{\ell_i y_{ik} + \ell_j y_{jk}}{2} + a_{ij} z_{kij} \leq -x_{ik} + x_{jk} + M(1 - z_{kij}) & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n, \ k = 1, \ldots, m \\
& \sum_{k=i}^{m} x_{ik} - \sum_{k=i}^{m} x_{jk} + v_{ij}^{+} - v_{ij}^{-} = 0 & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n \\
& z_{kij} + z_{kji} \leq \frac{1}{2}(y_{ik} + y_{jk}) & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n, \ k = 1, \ldots, m \\
& z_{kij} + z_{kji} + 1 \geq y_{ik} + y_{jk} & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n, \ k = 1, \ldots, m \\
& x_{ik} \geq 0 & \text{for } i = 1, \ldots, n-1, \ k = 1, \ldots, m \\
& v_{ij}^{+}, v_{ij}^{-} \geq 0 & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n \\
& y_{ik} \in \{0, 1\} & \text{for } i = 1, \ldots, n-1, \ k = 1, \ldots, m \\
& z_{kij} \in \{0, 1\} & \text{for } i = 1, \ldots, n, \ j = 1, \ldots, n, \ i \neq j, \ k = 1, \ldots, m,
\end{array}$$

where M is a sufficiently large number. Note that there are two z variables for each triplet i, j, k. Furthermore, there is one continuous variable x_{ik} for each pair department-row; this variable equals zero whenever department i is not assigned to row k, and moreover if two departments i and j are both assigned to the same row k, then $v_{ij}^{+} + v_{ij}^{-}$ represents the distance between i and j. We observe that the presence of the variables x_{ik} causes redundancy in the encoding, as the assignment of rows to departments is already encoded in the y_{ik} variables. This is not necessarily a negative feature; there are situations where models with similar "redundancy" in the binary variables perform better in practice than alternate models with fewer binaries. For instance, this behaviour was recently observed for the unit commitment problem in power systems [243].

In this case, however, an alternate approach that eliminates redundancy achieves a better performance. The approach of Amaral [11] encodes all the information in only $O(n^2)$ binary variables defined as

$$\lambda_{ij} = \begin{cases} 1, & \text{if departments } i \text{ and } j \text{ are both placed in the same row and } j \text{ is located to the right of } i \\ 0, & \text{otherwise} \end{cases}$$

The model is:

$$\begin{array}{lll} \min & \sum\limits_{i=1}^{n-1} \sum\limits_{j=i+1}^{n} c_{ij} d_{ij} & \\ \text{s.t.} & d_{ij} \geq x_i - x_j & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n \\ & d_{ij} \geq x_j - x_i & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n \\ & d_{ij} - \frac{1}{2}(\ell_i + \ell_j)\lambda_{ij} - \frac{1}{2}(\ell_i + \ell_j)\lambda_{ji} \geq 0 & \text{for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n \\ & x_i + \frac{1}{2}(\ell_i + \ell_j) \leq x_j + L(1 - \lambda_{ij}) & \text{for } i = 1, \ldots, n, \ j = 1, \ldots, n, \ i \neq j \\ & x_{\tilde{i}} \leq x_{\tilde{j}} & \text{where } (\tilde{i}, \tilde{j}) = \underset{(i,j)}{\operatorname{argmin}}\ c_{ij} \\ & \frac{\ell_i}{2} \leq x_i \leq L - \frac{\ell_i}{2} & \text{for } i = 1, \ldots, n \\ & \lambda_{ij} \in \{0, 1\} & \text{for } i = 1, \ldots, n, \ j = 1, \ldots, n, \ i \neq j \\ & \lambda \in \Lambda_n & \end{array}$$

where λ denotes the vector containing all the λ_{ij} variables associated to a fixed ordering, and

$$\begin{aligned} \Lambda = \{\lambda \in \mathbb{R}^{n(n-1)} : & -\lambda_{ij} + \lambda_{ik} + \lambda_{jk} - \lambda_{ji} + \lambda_{ki} + \lambda_{kj} \leq 1 \text{ for } i = 1, \ldots, n-1, \ j = i+1, \ldots, n, \ k \neq i, \ k \neq j \\ & -\lambda_{ij} + \lambda_{ik} - \lambda_{jk} + \lambda_{ji} - \lambda_{ki} + \lambda_{kj} \leq 1 \text{ for } j = 1, \ldots, n, \ i = 1, \ldots, j-1, \ k = 1, \ldots, j-1, \ i \neq k \\ & \lambda_{ij} + \lambda_{ik} + \lambda_{jk} + \lambda_{ji} + \lambda_{ki} + \lambda_{kj} \geq 1 \text{ for } i = 1, \ldots, n-2, \ j = i+1, \ldots, n-1, \ k = j+1, \ldots, n \\ & 0 \leq \lambda_{ij}, \lambda_{ji} \leq 1 \text{ for } i = 1, \ldots, n, \ j = 1, \ldots, n, \ i \neq j\} \end{aligned}$$

where the continuous variable x_i gives the position of department i within one of the two rows, and d_{ij} represents the distance between i and j. Furthermore Λ is a relaxation of the convex hull of the 0-1 incidence vectors λ of all possible orderings. This second model has a reduced number of variables, both binary and continuous, and eliminates some constraints relating them. On the other hand, it involves a large number of inequalities in the description of Λ.

The differences between these models are reflected in their running times. We implemented both approaches using the CPLEX solver on our machine (for details see the first paragraph of Section 12.5) and obtained the following results: The model of Amaral [11] is at least 100 times faster for instances of challenging size with 9 or more departments. Thus the method of Amaral [11] can solve instances with up 13 departments in a 12 hours time limit, whereas the approach by Chung and Tanchoco [72] can only handle instances with up to 12 departments within the same time limit. This difference in running time is mainly due to the tighter relaxations provided by the formulation of Amaral [11], and the consequent smaller number of branch-and-bound nodes. We further note that the corrections provided by Zhang and Murray [310], and the corresponding additional variables and constraints, result only in marginal increases of the running times compared to the original version of Chung and Tanchoco [72].

12.3 A Discrete Model for Space Between Departments

This section presents the theoretical results used to extend the semidefinite approach outlined in Subsections 12.2.1 and 12.2.2 to handle the possibility of empty space between departments. The key result is Theorem 12.2 showing that if all departments lengths are integer then there is always an optimal solution to the `(MRFLP)` on the half grid. As a consequence, the semidefinite approach from Hungerländer and

Anjos [164] will provide an optimal layout for the `(MRFLP)` if we add a sufficient number of *spacing departments* of length 0.5 and with all connectivities equal to zero. Because the number of spacing departments needed will normally be too large for practical computation, this section also proposes several strategies to reduce the number of spacing departments needed.

12.3.1 Theoretical Foundations for a Discrete Optimization Model

From the description of the `(MRFLP)` in the Introduction, it seems intuitive that the optimal distances between departments can be arbitrary. However, Theorem 12.2 shows that the `(MRFLP)` with integer department lengths can be modelled as a purely discrete optimization problem.

Theorem 12.2 *If all the department lengths are integer, then there is always an optimal solution to the* `(MRFLP)` *on the half grid.*

Proof. Let an optimal multi-row layout of the departments be given. We define a half-integer grid such that the centers of the departments with the most left centers are on a grid point. Let S be the set containing every department such that its center is on this grid, and let T be the set containing the remaining departments. Without loss of generality we assume that the indices of the departments in S are all smaller than the indices of the departments in T, i.e., $i < j, \ \forall\, i \in S, \ j \in T$.

Observe that there exists $\varepsilon > 0$ such that we can move all the departments in T simultaneously by ε either to the left or to the right without overlapping with departments in S. The existence of ε follows from the integrality of the department lengths and the definitions of S and T. The change in the objective function from shifting all the departments in T by ε is equal to

$$\delta = \sum_{i \in T} \left(\varepsilon \sum_{j \in S,\ j \dot{<} i} c_{ij} - \varepsilon \sum_{j \in S,\ i \dot{<} j} c_{ij} \right)$$

for a shift to the left, and to $-\delta$ for a shift to the right, where $i \dot{<} j$ means that the center of j is to the right of the center of i, and ε is chosen small enough such that no center of a department in T traverses a half grid point. Due to the optimality of the given layout, δ must to be 0 (otherwise we could improve the objective value by a shift of the departments in T in the appropriate direction, contradicting the assumption that the current layout is optimal). Hence such a shift does not change the objective value.

Choosing ε such that the center of at least one department in T lies on a half grid point after the shifting operation (to the left or right), and repeating this process allows us to arrange all departments on the grid in at most $n - 1$ steps without changing the objective value. □

Corollary 12.3 *If all the department lengths are integer, then for each instance of the* `(MRFLP)` *we obtain an equivalent instance by adding spacing departments of length 0.5 such that the length of each row becomes equal to* $M := \sum_{i=1}^{n} \ell_i$.

Proof. After adding the spacing departments we know that all the department centers will lie on the half-grid. Theorem 12.2 guarantees that there is always an optimal solution on the half-grid for every instance of the `(MRFLP)`. Moreover the row length M is sufficient: If there exists an interval that contains only spaces on all parallel rows, the objective value can only improve if we remove the respective spaces from each row because the pairwise center-to-center distances of the departments stay the same or are reduced by the removal. □

Finally we show by construction that the bound M on the lengths of the rows is tight for any given number of rows. Let us start with the case of 2 rows; consider 5 departments with lengths $\ell_1 = \ell_2 = \ell_3 = \ell_4 = 1, \ \ell_5 = n$ and connectivities $c_{12} = c_{34} = n, \ c_{15} = c_{35} = 1$. Part a.) of Figure 12.4 illustrates the optimal layout for this instance. We have $M = n + 4$ and

$$\lim_{n\to\infty} \frac{\text{Length of the shortest row in the optimal layout}}{M} = \frac{n+2}{n+4} = 1.$$

This construction can be generalized to an arbitrary number of rows by simply adding for each additional row 2 departments of length 1 and connectivity n with 2 other departments of length 1 that are not connected with each other. Specifically for 3 rows we add 2 departments with lengths $\ell_6 = \ell_7 = 1$ and connectivities $c_{26} = c_{47} = n$; the optimal layout for this extended instance is depicted in part b.) Figure 12.4. The cost of the optimal layout is $n + 1$ regardless of the number of rows considered.

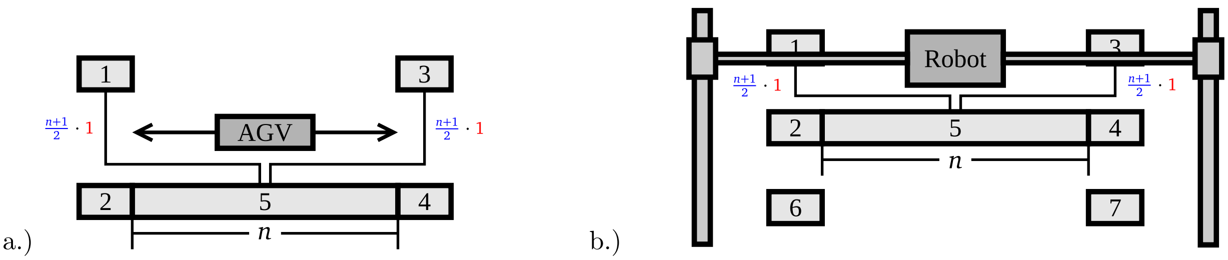


Figure 12.4: Examples for 2 and 3 rows showing that the choice of M is best possible.

12.3.2 Strategies for Reducing the Number of Spaces

The results in this section show that it is possible under certain assumptions to reduce the number and size of the spacing departments.

Lemma 12.4 applies to instances with equidistant department locations.

Lemma 12.4 *If all departments have the same length w, then spaces of size w are sufficient to preserve an optimal solution.*

Next we illustrate that M remains a tight bound using Figures 12.4 a.) and 12.5. While this illustration is for two rows, it is straightforward to generalize it to an arbitrary number of rows in an identical manner to that in Figure 12.4. First observe that the bound M from Theorem 12.2 on the lengths of the rows is tight if we further assume that the row assignment is fixed. (This is a reasonable assumption because our proposed (SDP) approach enumerates the row assignments.) For two rows we can simply replace department 5 from Figure 12.4 a.) by n departments of length 1 with connectivities $c_{i(i+1)} = 1,\ 5 \le i \le n+3$. Thus we obtain the following data: $\ell_1 = \ell_2 = \ldots = \ell_{n+1} = 1,\ c_{12} = c_{34} = c_{15} = c_{3(n+4)} = 1,\ c_{i(i+1)} = 1,\ 5 \le i \le n+3$, with departments 1 and 3 assigned to row 1, and the remaining departments assigned to row 2. The corresponding optimal double-row layout, with total cost $1 \cdot 1 \cdot (n + 1) = n + 1$, is depicted in Figure 12.5.

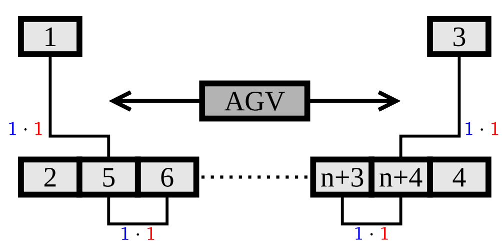


Figure 12.5: Optimal layout for the reformulation of the problem in Figure 12.4 a.) according to Lemma 12.4.

Lemma 12.5 shows when it is allowed to enlarge the grid size and thus reduce the number of spaces.

Lemma 12.5 *If the lengths of the departments have a greatest common divisor d, then we can divide all lengths by d, solve the resulting problem (where all department lengths are still integer) and multiply the bounds obtained by d to get bounds of the original problem.*

Lemma 12.6 indicates how to further reduce the number of spaces by suitably modeling the possible distances between the departments.

Lemma 12.6 *Let p_i be the number of departments assigned to row i. Then we preserve an optimal solution to the* (MRFLP) *if we use p_i spaces of lengths $2^{k-1},\ k = 0, 1, 2, \dots, h_i$, where h_i is chosen such that*

$$p_i \sum_{k=0}^{h_i} 2^{k-1} + \sum_{k \in [n],\ r(k)=i} \ell_k \geq M, \qquad i \in [m].$$

Proof. We can model all feasible distances between all pairs of departments (all feasible layouts of the departments on the half grid) with the suggested spaces. Note that we also need spaces to allow a gap between the left origin the first department on each row. □

Finally Lemma 12.7 builds on Lemma 12.6 and examines when additional spaces have no marginal value.

Corollary 12.7 *Let a row assignment be given. If we add to each row (as per Lemma 12.6) k_1 spaces of the minimal necessary lengths to get the lower bound v_{k_1}, and if we also add k_2 $(> k_1)$ spaces of minimal necessary lengths to each row to get the lower bound v_{k_2}, then it follows from $v_{k_1} = v_{k_2}$ that v_{k_1} is a lower bound for the* (MRFLP) *for the given row assignment.*

Proof. If adding additional spaces has no effect on the value of the (SDP) relaxation then adding even more (possibly larger) spaces also does not have any effect. □

Corollary 12.7 leads to another way of reducing the computational effort. Indeed in Section 12.5.2 we propose a computational strategy that does not solve the (MRFLP) directly but rather iteratively adds spaces of minimal necessary length to the (SF-MRFLP). Unlike the (DRFLP) approaches reviewed in Section 12.2.3, our semidefinite approach allows us to control both the number and the sizes of the spaces considered. This is a valuable feature in practice because layout with less space are normally preferable. For computational experiments with a controlled number of spaces see Sections 12.5.3 – 12.5.5.

12.4 Solving the (SDP) Relaxations

The core of our (SDP) approach is to solve the relaxation (SDP_3) for every row assignment with sufficient space added. The resulting fractional solutions provide lower bounds on the costs of the optimal layout for the given row assignment. By rounding the fractional solutions, we can also obtain upper bounds, i.e., integer solutions that describe feasible layouts for the given row assignment. By computing these lower and upper bounds for evry row assignement, we can obtain global bounds for the (MRFLP). In the following two subsections we discuss how to solve (SDP_3) and how to round the fractional solutions.

12.4.1 Computing Lower Bounds

Looking at the constraint classes and their sizes in the relaxation (SDP_3), it is clear that explicitly maintaining $O(n^3)$ or more constraints is not an attractive option. We therefore adapt an approach originally suggested in Fischer et al. [103] that was successful for the max-cut problem [262] and several ordering problems [62, 168].

The idea is to combine the bundle method and an interior-point method. Initially, we only explicitly ensure satisfaction of the constraints $\operatorname{diag}(Z) = e$ and $Z \succcurlyeq 0$; this can be achieved with standard interior-point methods, see e.g. Helmberg et al. [146]. All other constraints are handled using Lagrangian duality. However this makes the objective function f non-smooth, hence the need for the bundle method. The bundle method iteratively evaluates f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to again solving an (SDP) with the constraints $\operatorname{diag}(Z) = e$ and $Z \succcurlyeq 0$, which can be done efficiently using an interior-point method. In this way we obtain an approximate minimizer of f that is guaranteed to yield a lower bound to the optimal solution of ($\texttt{SDP}_3$). In some experiments we also use the bundle method to obtain an initial set of constraints to add to the relaxation ($\texttt{SDP}_2$) and then solved the resulting relaxation exactly using Sedumi [289].

12.4.2 Obtaining Feasible Layouts

To obtain feasible (MRFLP) layouts we use a rounding heuristic originally proposed by Hungerländer and Anjos [164]. First we apply the hyperplane rounding algorithm of Goemans and Williamson [121] to the solution of the (SDP) relaxation, then we take the resulting vector $\overline{w}$ and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities

$$-1 \leq \overline{y}_{ij} + \overline{y}_{jk} - \overline{y}_{ik} \leq 1 \tag{12.13}$$

within the rows, and with respect to the inequalities (12.10) for the inter-row variables.

Let us give a more detailed description of the implementation of our heuristic. Let W'' be the current fractional (primal) solution of the semidefinite relaxation, and w' be an initial vector that encodes a feasible layout of the departments in all rows.

1. Compute the convex combination $R := \lambda(w'w'^\top) + (1-\lambda)W''$ using a randomly generated $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^\top$ of R.

2. Apply the Goemans-Williamson hyperplane rounding to D and obtain a $-1/{+1}$ vector $\overline{w}$ (cf. Rendl et al. [262]).

3. Compute the induced objective value $z(\overline{w}) := \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}^\top C_Z \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}$. If $z(\overline{w}) \geq z(w')$: go to step 2.

4. If $\overline{w}$ satisfies (12.13) and (12.10): set $w' := \overline{w}$ and go to 2.
   Otherwise: modify $\overline{w}$ by first changing the signs of one of three variables in all violated 3-cycle inequalities, afterwards flipping signs of the inter-row ordering variables to satisfy (12.10) and go to step 3.

The algorithm stops after 100 executions of step 2. (Note that before the 51st execution of step 2, we perform step 1 again. As step 1 is quite expensive, we refrain from executing it too often.) The final w' is the heuristic solution. If the duality gap is not closed after the heuristic, we return to the (SDP) optimization algorithm, and then reapply the heuristic (retaining the last vector w').

12.5 Computational Results

We report the results for different computational experiments with the proposed semidefinite relaxation ($\texttt{SDP}_3$). All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 24 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. We enforced a time limit of 12 hours for all the experiments.

We solve the (MRFLP) on (SRFLP) instances from the literature. Additionally we generated new equidistant instances (instances with uniform department length = 1), denoted by E_*, with connectivities varying randomly between 1 and 10 and density of either 50 % or 100 %. Table 12.1 gives the characteristics of the small (SRFLP) instances together with their optimal (SRFLP) solutions and the computational time to solve them using the SDP approach of Hungerländer and Rendl [169] run on our computer. Table 12.8 does the same for the larger instances. All the instances can be downloaded from http://anjos.mgi.polymtl.ca/flplib. In general, while for the (SRFLP) we can round to the nearest integer because 0.5 can only occur in the constant term, for the (MRFLP) we can round the lower bound only to 0.5 as the constant term is different for distinct row assignments.

First we apply our exact semidefinite approach to the (DRFLP) in Subsections 12.5.1 - 12.5.3. Then we expand our experiments to the case of 3 and more rows in Subsection 12.5.4. Finally we examine the effects of adding space to medium and larges instances with fixed row assignments in Subsection 12.5.5.

12.5.1 First Results for the (DRFLP)

For small (DRFLP) instances, the relaxation (SDP_3) can be solved for each of the $2^{n-1} - 1$ possible row assignments. As per Corollary 12.3 we add sufficiently many spaces to the (DRFLP) instances and solve the resulting (PROP) for each row assignment. From the obtained bounds, we can deduce global upper and lower bounds: these are the minima of all upper and lower bounds respectively. The upper bounds were obtained using the SDP rounding heuristic described in Subsection 12.4.2.

For reasons of efficiency we used 10 function evaluations of the bundle method applied to (SDP_3) to obtain an initial set of constraints to add to the relaxation (SDP_2). We then solved the resulting relaxation using Sedumi [289]; added all violated inequality constraints from (SDP_3); solved again using Sedumi; and repeated this process until no more violations were found. We also tried to solve (SDP_3) directly but the running times were at least one order of magnitude greater. Solving (SDP_4) instead of (SDP_3) resulted in a few slightly improved lower bounds but tremendously larger running times because many of the $O(n^5)$ LS-cuts are active at the optimum.

The computational results are summarized in Table 12.2. We exploit Lemma 12.5 to reduce the number of spaces and hence speed up our algorithmic framework. For the equidistant instances we additionally use Lemma 12.6. Looking at the running times and their growth rates, we deduce that this approach is realistic only for instances with up to 10 departments within the 12-hour time limit.

The ILP approaches by Chung and Tanchoco [72] and Amaral [11] focused exclusively on the (DRFLP) and allowed them to solve instances with up to 13 departments to optimality with better results achieved by Amaral [11]. Hence our semidefinite approach exhibits a weaker practical performance for the (DRFLP), especially when we allow arbitrary lengths of the departments. Nevertheless the strengths of our approach are that it allows us to control the number and sizes of spaces considered (for details see Subsections 12.5.3 - 12.5.5 below) and that it is applicable to the (MRFLP) with more than two rows (for details see Subsection 12.5.4 below).

Finally let us compare our results with the ones obtained for the (SF-DRFLP) using the same algorithmic framework on the same machine. We adopted part of the (SF-DRFLP) results from Hungerländer and Anjos [164] and summarized all the computational details in Table 12.3.

We observe that the global lower and upper bounds (especially for the E_*-instances) are often not improved. But nonetheless the average lower and upper bounds are always significantly improved. Hence space especially helps to improve the bounds for unbalanced ("bad") row assignments. The computation times show that including space is quite expensive, especially if the lengths of the departments are diverse. Without using Lemmas 12.5 and 12.6 we could only compute bounds to the E_*-instances for up to 7 departments and the other instances for up to 5 departments within the time limit.

Instance	Source	Size (n)	(SRFLP) Optimal (SRFLP) solution	(SRFLP) Time (sec) Hungerländer and Rendl [169]
H_5	[151]	5	800	0.1
H_6	[151]	6	1480	0.1
H_7	[151]	7	3680	0.6
H_8	[151]	8	4725	0.4
H_12	[151]	12	17945	7.9
Rand_5	[164]	5	147.5	0.1
Rand_6	[164]	6	420	0.4
Rand_7	[164]	7	344	0.3
Rand_8	[164]	8	382	1.3
Rand_9	[164]	9	1024.5	2.2
Rand_10	[164]	10	1697	3.1
Rand_11	[164]	11	1564	2.0
Rand_12	[164]	12	2088	8.4
Rand_13	[164]	13	3101.5	7.8
S_8	[283]	8	801	0.6
SH_8	[283]	8	2324.5	2.3
S_9	[283]	9	2469.5	0.7
SH_9	[283]	9	4695.5	9.2
S_10	[283]	10	2781.5	0.6
S_11	[283]	11	6933.5	1.3
E_5_50	This paper	5	30	0.1
E_5_100	This paper	5	95	0.2
E_6_50	This paper	6	100	0.1
E_6_100	This paper	6	216	0.2
E_7_50	This paper	7	106	0.2
E_7_100	This paper	7	252	0.2
E_8_50	This paper	8	136	0.3
E_8_100	This paper	8	397	0.3
E_9_50	This paper	9	240	0.4
E_9_100	This paper	9	618	1.3
E_10_50	This paper	10	387	0.6
E_10_100	This paper	10	873	0.7
E_11_100	This paper	11	1085	1.4

Table 12.1: Characteristics and optimal (SRFLP) results for the small instances.

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems					
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Average lower bound	Average upper bound	Average nbr spaces	Total time (sec)
E_5_50	13	13	0	0	0	17.6	17.6	4.7	12.0
E_5_100	44	46	4.55	4.62	1.15	53.3	53.9	4.7	11.1
E_6_50	45	45	0	3.77	0.18	58.3	58.4	5.1	40.2
E_6_100	91.5	99	8.20	8.20	2.54	119.5	122.1	5.1	35.1
E_7_50	51	51	0	3.23	0.23	63.5	63.7	6.3	177.7
E_7_100	119.5	126	5.44	5.44	1.32	145.2	147.0	6.3	167.2
E_8_50	64	64	0	6.67	1.00	79.3	80.0	6.9	716.4
E_8_100	175	191	9.14	12.47	3.73	218.6	226.3	6.9	538.5
E_9_50	112	118	5.36	13.96	2.55	137.1	140.4	8.2	4748.6
E_9_100	268	306	14.18	16.42	7.18	328.3	350.5	8.2	3530.9
E_10_50	181.5	191	5.23	16.79	2.97	219.8	226.1	8.8	24855.9
E_10_100	374.5	427	14.02	16.63	7.59	463.5	496.8	8.8	17109.2
H_5	350	350	0	1.94	0.20	501.0	502.0	7.7	411.8
Rand_5	52.5	52.5	0	10.00	1.21	88.0	89.1	9.9	944.0
H_6	640	640	0	14.63	2.22	938.1	957.7	9.9	6609.7
Rand_6	188	190.5	1.33	8.43	2.37	253.9	259.5	11.1	24850.6

Table 12.2: Computational results for the (DRFLP) obtained by applying our semidefinite approach exploiting Lemmas 12.5 and 12.6 to (SDP_3).

12.5.2 Exploiting the Marginal Value of Spaces for the (DRFLP)

In this subsection we exploit Corollary 12.7 algorithmically and examine if this alternative strategy to handle space allows us to improve on the computational results from the previous subsection.

First let us give a description of our approach: We start with computing bounds for the space free case for a given row assignment and then

1. we add the smallest necessary space to the shortest row until the lower bound does not worsen any more,
2. and after that we add the smallest necessary spaces to all other rows. If the lower bound does not worsen during step 2, we have found a valid lower bound for the (MRFLP) for the given row assignment due to Corollary 12.7, otherwise we go back to step 1.

We summarize the results obtained by this alternative approach in Table 12.4.

For some instances, namely the E_*-instances with density 100 % and the Rand_*-instances, we are able to reduce the running times compared with the results from the previous subsection summarized in Table 12.2. This reduction in the running time is due to the reduction of the average number of spaces needed. Thus we solve more but cheaper relaxations to obtain the same lower bounds. For some other instances we cannot significantly reduce the number of spaces needed (see e.g. H_6) and thus the computationally effort grows. Further note that the average upper bounds are better than in the previous section because we work with smaller SDPs (with fewer spaces) for which the upper bound heuristic yields stronger feasible layouts.

12.5.3 Solving the (DRFLP) with Space Restrictions

Let us recall two findings concerning the usage space in our semidefinite approach from the previous subsections:

1. For promising balanced row assignments we need less space than for unbalanced ones, hence the global bounds are similar and sometimes even the same for the cases with and without space.

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems				
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Average lower bound	Average upper bound	Total time (sec)
E_5_50	13	13	0	0	0	18.7	18.7	3.0
E_5_100	44	46	4.55	4.55	0.79	57.1	57.5	5.0
E_6_50	45	45	0	3.64	0.36	61.2	61.5	9.5
E_6_100	91.5	99	8.20	9.87	4.54	126.0	131.2	12.9
E_7_50	51	51	0	2.70	0.13	65.4	65.5	30.2
E_7_100	119.5	126	5.44	5.44	1.83	151.1	153.7	32.0
E_8_50	64	64	0	8.91	1.31	83.6	84.7	78.3
E_8_100	175	191	9.14	17.89	5.93	229.6	242.0	86.2
E_9_50	112	118	5.36	9.49	2.51	143.2	146.8	223.4
E_9_100	268	306	14.18	16.42	7.70	344.0	368.8	251.5
E_10_50	181.5	191	5.23	15.93	4.18	229.4	238.7	649.9
E_10_100	374.5	427	14.02	18.84	8.44	482.7	520.4	815.2
E_11_100	481	539	12.06	17.07	7.69	598.8	642.4	1742.3
H_5	410	450	9.76	9.76	2.31	564.7	576.0	4.8
Rand_5	52.5	52.5	0	8.93	1.78	97.9	99.6	4.5
H_6	665	720	8.27	16.58	5.55	1050.6	1103.5	11.1
Rand_6	188.5	190.5	1.06	6.65	2.59	280.0	286.7	13.8
H_7	1630	1700	4.29	15.17	3.65	2334.9	2413.0	25.2
Rand_7	166	166	0	13.82	3.16	239.4	246.3	31.7
H_8	2355	2385	1.27	21.15	5.72	3185.6	3355.6	91.3
S_8	380.5	408	7.23	20.10	5.87	494.8	521.8	91.6
SH_8	990.5	1135.5	14.64	17.00	10.85	1299.6	1424.2	87.4
Rand_8	192	205	6.77	28.10	4.77	246.7	258.0	82.2
S_9	1163	1181.5	1.59	13.63	3.42	1518.9	1567.5	253.4
SH_9	1974.5	2294.5	16.21	18.87	11.31	2556.1	2814.9	251.9
Rand_9	447.5	492.5	10.06	19.73	5.61	629.5	662.3	252.6
S_10	1314	1374.5	4.60	10.77	4.20	1726.7	1796.3	713.0
Rand_10	779	838	7.57	15.16	5.68	1023.0	1077.8	698.2
S_11	3325.5	3439.5	3.43	14.92	5.16	4094.5	4297.2	2127.0
Rand_11	643.5	708	10.02	23.95	5.76	1054.6	1113.4	2048.5
H_12	8450	8995	6.45	17.30	6.25	10611.8	11247.7	5943.3
Rand_12	775.5	799	3.03	17.69	6.21	1327.8	1408.1	6389.5
Rand_13	1058	1070	1.13	19.11	5.98	1865.5	1974.1	20636.9

Table 12.3: Computational results for the (SF-DRFLP) by applying our semidefinite approach to (SDP_3).

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems					
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Average lower bound	Average upper bound	Average nbr spaces	Total time (sec)
E_5_50	13	13	0	0	0	17.6	17.6	2.3	20.0
E_5_100	44	46	4.55	4.62	1.15	53.3	53.9	2.4	21.3
E_6_50	45	45	0	3.77	0.18	58.3	58.4	2.6	68.8
E_6_100	91.5	99	8.20	8.20	2.54	119.5	122.1	2.9	65.7
E_7_50	51	51	0	3.23	0.22	63.5	63.7	3.4	288.5
E_7_100	119.5	126	5.44	5.44	1.20	145.2	146.8	3.0	212.8
E_8_50	64	64	0	5.97	0.98	79.3	80.0	4.4	1809.6
E_8_100	175	191	9.14	10.29	3.40	218.6	225.6	3.0	481.0
E_9_50	112	118	5.36	7.88	1.77	137.1	139.4	4.8	6714.8
E_9_100	268	306	14.18	14.93	6.12	328.3	347.3	2.7	1650.6
E_10_50	181.5	191	5.23	9.71	2.21	219.8	224.5	5.0	38993.4
E_10_100	374.5	427	14.02	14.02	6.47	463.5	491.8	2.6	4313.6
E_11_100	481	539	12.06	13.72	5.30	578.2	607.2	2.9	13869.1
H_5	350	350	0	0	0	501.0	501.0	6.5	323.3
Rand_5	52.5	52.5	0	0	0	88.0	88.0	4.9	66.2
H_6	640	640	0	7.58	0.71	938.1	943.5	9.9	16014.4
Rand_6	188	190.5	1.33	4.22	0.63	253.9	255.3	7.6	4190.2

Table 12.4: Computational results for the (DRFLP) using (SDP_3) and Corollary 12.7.

2. The average number of spaces needed to find and prove the optimal layout is significantly lower than the theoretical bound on their number (compare the "average nbr spaces" columns of Tables 12.2 and 12.4).

These points motivate us to examine how much the lower and upper bounds change if we allow only a few spaces of predefined sizes and how much the running times can be reduced through this restriction. This question is also interesting becasue in practice layouts with fewer spaces are usually preferable.

In our experiments we added four spaces with lengths 2^k, $k \in \{-1, 0, 1, 2\}$, to the shorter row for each row assignment. The results for the (DCFLP) are reported in Table 12.5. Here we do not consider the equidistant E_*-instances because space has no effect on the global lower and upper bounds of these instances (compare Tables 12.3 and 12.4).

The global bounds for the instances with space restrictions coincide with the global (DRFLP) bounds as far as they are available, i.e., for instances with at most 6 departments. For larger instances we compared the global bounds with the exact solutions obtained by the ILP approach of Amaral: the global lower and upper bounds are only slightly higher (and often the same). The results suggest that restricting the number of spaces is a very good strategy to approximate the (DRFLP) for larger instances and also for obtaining layouts that yield a good compromise between the number and sizes of spaces used and the corresponding layout costs induced. Additionally our approach can easily handle particular requirements arising in practice regarding the assignment of the departments and the spacing between them.

12.5.4 The (MRFLP) with more than 2 Rows

Next we extend the results from the previous subsections by considering 3 to 5 rows. The number of different row assignments is given by the recursive formula

$$R(m,n) = \frac{m^n}{m!} - \sum_{k=1}^{m-1} \frac{R(k,n)}{(m-k)!},$$

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems				
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Average lower bound	Average upper bound	Total time (sec)
H_5	350	350	0	6.25	1.08	501.7	506.3	11.0
Rand_5	52.5	52.5	0	4.27	0.32	88.4	88.7	11.5
H_6	640	640	0	7.58	0.90	938.7	946.0	33.3
Rand_6	188	190.5	1.33	4.22	0.74	254.0	255.7	34.9
H_7	1600	1660	3.75	6.48	1.47	2204.0	2234.2	111.5
Rand_7	159	159	0	11.38	2.19	216.6	221.4	118.3
H_8	2265	2265	0	14.86	2.49	2870.0	2945.3	423.7
S_8	380.5	396	4.07	11.43	3.24	462.5	477.1	409.1
SH_8	990.5	1125.5	13.63	14.90	9.52	1218.8	1329.7	406.6
Rand_8	189.5	189.5	0	7.51	1.88	229.0	233.3	416.3
S_9	1162	1179	1.46	10.08	2.31	1445.6	1479	1146.1
SH_9	1974.5	2294.5	16.21	17.10	11.52	2409.6	2675.7	1199.2
Rand_9	447.5	486.5	8.72	16.72	3.93	591.0	613.9	1071.3
S_10	1314	1353.5	3.01	10.89	2.81	1642.5	1689.2	3663.3
Rand_10	778.5	821	5.46	11.52	4.15	962.3	1002.3	3392.7
S_11	3325.5	3424.5	2.98	13.59	4.23	3916.5	4081.6	11553.8
Rand_11	633.5	697	10.02	14.61	3.92	989.2	1028.1	11212.1
H_12	8385	8875	5.84	13.94	4.93	10002.6	10493	39927.9
Rand_12	768	788	2.60	15.10	5.15	1242.3	1306.9	39409.3

Table 12.5: Computational results for the (DRFLP) using (SDP_3) and adding four spaces to the shorter row.

where m is the number of rows and n is the number of departments. Applying $m \in \{2,3,4,5\}$ to the above formula yields:

$$R(2,n) = 2^{n-1} - 1, \qquad R(3,n) = \frac{3^{n-1} - 2^n + 1}{2},$$

$$R(4,n) = \frac{4^{n-1} - 3^n + 3 \cdot 2^{n-1} - 1}{6}, \qquad R(5,n) = \frac{5^{n-1} - 4^n + 2 \cdot 3^n - 2^{n+1} + 1}{24}.$$

We use the same algorithmic approach and setup as above and again restrict the running time per instance to 12 hours. Firstly in Table 12.6 we summarize the computational results obtained for the general (MRFLP). We also restate the corresponding results for the (DRFLP) to facilitate the comparison. Note that both strategies "Direct Approach" and "Marginal value" yield the same global bounds for all instances, whereas the average upper bounds are again slightly better for the approach exploiting the marginal value of spaces.

Clearly the computation times grow for an increasing number of rows, especially as the average number of spaces per row assignment increases (and also the overall number of row assignments for larger n). Interestingly the global bounds become tighter as m increases. We also observe that exploiting the marginal value of spaces becomes more effective for a larger number of rows as the absolute gap increases between the number of spaces theoretically needed and the number relevant in practice; this is observed by comparing the first columns of "Direct approach" and "Marginal value" in Table 12.6.

Next we extend the experiments from Subsection 12.5.3 by considering up to 5 rows. We add four spaces with lengths 2^k, $k \in \{-1, 0, 1, 2\}$, to the shortest row and summarize the corresponding results in Table 12.7.

The results clearly show that adding only a few spaces to the shortest row help to considerably improve the optimal objective value without increasing the running time too much. Hence the layouts obtained again yield a good compromise between the number and size of spaces used and the corresponding layout costs induced. We further observe that for a growing number of rows the average impact of spaces added

			Global bounds (over all row assignments)			Direct approach		Marginal value	
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Average nbr spaces	Total time (sec)	Average nbr spaces	Total time (sec)
2	15	E_5_50	13	13	0	4.7	12.0	2.3	20.0
	15	E_5_100	44	46	4.55	4.7	11.1	2.4	21.3
	31	E_6_50	45	45	0	5.1	40.2	2.6	68.8
	31	E_6_100	91.5	99	8.20	5.1	35.1	2.9	65.7
	63	E_7_50	51	51	0	6.3	177.7	3.4	288.5
	63	E_7_100	119.5	126	5.44	6.3	167.2	3.0	212.8
	127	E_8_50	64	64	0	6.9	716.4	4.4	1809.6
	127	E_8_100	175	191	9.14	6.9	538.5	3.0	481.0
	255	E_9_50	112	118	5.36	8.2	4748.6	4.8	6714.8
	255	E_9_100	268	306	14.18	8.2	3530.9	2.7	1650.6
	511	E_10_50	181.5	191	5.23	8.8	24855.9	5.0	38993.4
	511	E_10_100	374.5	427	14.02	8.8	17109.2	2.6	4313.6
	15	H_5	350	350	0	7.7	411.8	6.5	323.3
	15	Rand_5	52.5	52.5	0	9.9	944.0	4.9	66.2
	31	H_6	640	640	0	9.9	6609.7	9.9	16014.4
	31	Rand_6	188	190.5	1.33	11.1	24850.6	7.6	4190.2
3	25	E_5_50	6	6	0	8.6	70.9	4.7	64.4
	25	E_5_100	27	27	0	8.6	70.8	4.0	62.8
	90	E_6_50	29	29	0	8.8	380.8	4.0	309.5
	90	E_6_100	56	56	0	8.8	479.0	4.3	419.3
	301	E_7_50	31	31	0	10.8	7982.0	5.0	4427.2
	301	E_7_100	76	79	3.95	10.8	7770.5	4.6	2900.3
	966	E_8_50	37	37	0	11.2	49437.5	5.2	32955.9
	966	E_8_100	116	125	7.76	11.2	51625.2	4.6	12088.3
	25	H_5	175	175	0	13.2	9566.7	10.3	680.6
	25	Rand_5	18	18	0	16.1	22475.0	10.4	893.2
4	10	E_5_50	4	4	0	12.0	257.8	6.4	44.0
	10	E_5_100	17	17	0	12.0	188.5	5.3	37.9
	65	E_6_50	22	22	0	12.0	2665.9	4.9	335.9
	65	E_6_100	49	49	0	12.0	2736.0	5.0	386.0
	350	E_7_50	17	17	0	14.1	72976.5	6.1	16082.9
	350	E_7_100	50	50	0	14.1	72174.1	5.8	5961.8
	10	H_5	105	105	0	16.9	11796.5	14.1	2287.4
	10	Rand_5	10	10	0	20.4	32734.4	15.1	4360.1
5	1	E_5_50	0	0	0	15.0	39.5	7.0	5.1
	1	E_5_100	0	0	0	15.0	80.9	7.0	6.2
	15	E_6_50	12	12	0	15.0	1585.6	5.7	103.7
	15	E_6_100	29	29	0	15.0	1755.8	5.0	81.6
	1	H_5	0	0	0	20.0	2517.9	15.0	271.3
	1	Rand_5	0	0	0	24.0	7275.6	22.0	6252.4

Table 12.6: Computational results for the (MRFLP) obtained by applying our semidefinite approach to $(\mathtt{SDP}_3)$.

			Results without space						Results with spaces added					
			Global bounds (over all row assignments)			Statistics for the $R(m,n)$ subproblems			Global bounds (over all row assignments)			Statistics for the $R(m,n)$ subproblems		
Nbr rows	Nbr row assignm.	Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Total time (sec)	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Total time (sec)
3	25	H_5	284.5	290	1.93	13.76	1.81	8.0	235	250	6.38	13.04	2.70	28.8
	25	Rand_5	34.5	34.5	0	11.64	1.46	7.6	19	19	0	4.86	0.70	22.8
	90	H_6	489.5	560	14.40	43.30	6.68	32.8	425	440	3.53	47.25	6.92	117.5
	90	Rand_6	101	110.5	9.41	9.54	2.13	36.4	96.5	100.5	4.15	12.18	2.78	117.6
	301	H_7	1074.5	1190	10.75	23.52	5.94	139.2	1060	1155	8.96	16.32	4.73	585.0
	301	Rand_7	108	108	0	13.79	1.92	157.0	105	105	0	13.35	2.47	566.1
	966	H_8	1393	1515	8.76	27.20	6.71	638.0	1350	1430	5.93	25.84	6.38	3283.1
	966	S_8	244	260	6.56	17.79	5.66	709.1	239	250	4.60	20.13	6.03	3251.5
	966	SH_8	647	740.5	14.45	21.01	10.17	633.6	647	739.5	14.30	20.49	11.81	2564.2
	966	Rand_8	128	138	7.81	21.61	4.71	653.3	123.5	132.5	7.29	24.15	5.66	3087.1
	3025	S_9	758	771.5	1.78	16.15	4.51	3074.5	757.5	770	1.65	15.94	4.88	17025.5
	3025	SH_9	1257.5	1431.5	13.84	21.80	11.78	2883.7	1257.5	1413.5	12.41	21.94	13.01	11039.1
	3025	Rand_9	291.5	317.5	8.92	23.47	5.30	3055.0	290.5	298.5	2.75	20.35	5.83	17631.1
4	10	H_5	140	140	0	2.22	0.22	3.5	140	140	0	0	0	6.3
	10	Rand_5	48.5	48.5	0	2.02	0.58	4.5	38.5	38.5	0	2.14	0.37	5.9
	65	H_6	410	410	0	25.71	2.73	34.9	340	340	0	28.00	4.07	56.8
	65	Rand_6	98.5	98.5	0	5.85	0.83	40.0	82	82	0	7.10	1.29	66.1
	350	H_7	820	820	0	28.89	4.44	232.2	765	780	1.96	32.34	5.86	457.1
	350	Rand_7	77	77	0	8.21	0.89	190.1	73	73	0	13.93	1.60	455.7
	1701	H_8	1135	1135	0	39.04	4.96	1211.1	1050	1125	7.14	29.39	6.78	3403.5
	1701	S_8	177.5	188	5.92	20.80	5.19	1461.4	172	182	5.81	17.85	6.14	3343.7
	1701	SH_8	467	491.5	5.25	14.25	6.75	1292.7	466.0	483.5	3.76	15.83	8.26	3240.5
	1701	Rand_8	98	98	0	19.84	2.90	1411.1	94	98	4.26	24.41	4.35	3209.9
5	1	H_5	200	200	0	0	0	0.2	140	140	0	0	0	0.6
	1	Rand_5	44.5	44.5	0	0	0	0.2	38.5	38.5	0	0	0	0.6
	15	H_6	260	260	0	2.46	0.37	7.0	190	190	0	5.26	0.57	11.4
	15	Rand_6	94.5	94.5	0	0.91	0.10	6.7	83.5	83.5	0	6.09	0.47	13.3
	140	H_7	650	650	0	18.84	2.04	87.6	600	600	0	20.43	2.76	220.9
	140	Rand_7	61	61	0	6.14	0.38	83.3	61	61	0	6.25	0.54	170.2
	1050	H_8	875	875	0	29.53	3.28	981.6	865	865	0	33.46	4.55	5331.4
	1050	S_8	141.5	146	3.18	13.81	3.51	1194.8	80.5	82	1.86	17.49	2.55	3793.2
	1050	SH_8	413.5	413.5	0	9.52	3.25	911.8	133	136	2.26	16.06	4.94	4259.8
	1050	Rand_8	82	82	0	14.29	1.36	967.2	413.5	413.5	0	11.06	4.24	3832.9

Table 12.7: Computational results for (SDP_3) for 3 to 5 rows without space and with specified spaces.

on the objective function increases. On the other hand there seems to be no clear connection between the spaces added and the tightness of $(\texttt{SDP}_3)$.

12.5.5 The (PROP) with Additional Spaces on Medium and Large Instances

For our final set of experimentswe consider the (PROP) using the data from selected (SRFLP) instances with between 20 and 70 departments. For each value of n we choose one instance from the literature. Table 12.8 lists the characteristics of the instances together with their optimal (SRFLP) solutions and corresponding running times.

We suggest to select balanced row assignments because similar row lengths are often of interest in the design of layouts in practice, see e.g. Langevin et al. [197]. To do so we use the following simple heuristic: First we randomly assign 25% of the departments to each of the two rows; then the remaining 50% of the departments are added one at a time by taking the longest remaining department and adding it to the shorter row. Our heuristic quickly yields assignments for which the total row lengths are very close, in average they differ by less than length 2.

To examine the effect of spaces for medium and large instances we add few spaces to each row (as they are balanced). More precisely we add five spaces with lengths $2^k, k = -1, 0, 1, 2, 3$ to each row. As the considered row assignments (which are the most promising ones) are balanced, the results from Subsection 12.5.3 suggest that we do not have to insert a lot of spaces to find a strong layout for the given row assignment.

We again apply our algorithmic framework to the semidefinite relaxation $(\texttt{SDP}_3)$ for the selected row assignments. To be able to solve this relaxation for medium and large instances we have to slightly adapt our algorithmic approach: We use only the bundle method (without Sedumi) to obtain reasonable bounds as solving the relaxations exactly with Sedumi becomes way too time-consuming due to the large number of constraints involved. Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations to 125 to control the overall computational effort. In fact these evaluations constitute the computational bottleneck as they are responsible for more than 95% of the required running time. This limitation of the number of function evaluations sacrifices some possible incremental improvement of the bounds. Similar experiments showed [164] that the lower bounds of the bundle method quickly get close to the exact $(\texttt{SDP}_3)$ bounds even though the number of function evaluations is capped at 125. (Note that the results in Table 12.8 required a higher number of function evaluations and hence higher computation times.) We summarize the results averaged over 10 row assignments selected by our heuristic in Table 12.9.

Examining the results in Table 12.9 we observe that the running times are dependent on the number of departments plus the number of spaces. The lower bound slightly decreases when we add the spaces. Nonetheless the results clearly suggest that especially for large instances the space-free layout yields a very good approximation of the (MRFLP) layout. In contrast to the previous subsections spaces do not always help to reduce the costs of the best layout found because the spaces enlarge the size of the instances and hence can have a bad effect on the performance of the rounding heuristic.

We conclude with a mention that we extended the computations to instances of (kPROP) with 3 to 5 rows, but as the results exhibited the same trends as for the double-row case, we chose not to report them.

12.6 Summary and Future Research Directions

This paper proposed a new model for the (MRFLP) that expresses the problem as a discrete optimization problem. This makes it possible to exploit the underlying combinatorial structure, and in particular to explicitly control the number and size of the spaces between departments. This discrete optimization model is an extension of our recent approach for the (SF-MRFLP) to the more general (MRFLP) and also captures some of its important special cases. We constructed a semidefinite relaxation of the discrete

Instance	Source	Size (n)	(SRFLP)		
			Best lower bound	Best layout	Time (sec) Hungerländer and Rendl [169]
H_20	[151]	20	15549		54
N25_05	[20]	25	15623		211
H_30	[151]	30	44965		547
N30_05	[20]	30	115268		1110
Am33_03	[8]	33	69942.5		2193
Am35_03	[8]	35	69002.5		3194
ste36.5	[21]	36	91651.5		1078
N40_5	[169]	40	103009		8409
sko42-5	[21]	42	248238.5		4122
sko49-5	[21]	49	666130	666143	34222
sko56-5	[21]	56	591915.5	592335.5	64006
AKV-60-05	[22]	60	318801	318805	99106
sko64-5	[21]	64	501342.5	502063.5	119158
AKV-70-05	[22]	70	4213774.5	4218002.5	101765

Table 12.8: Characteristics and optimal solutions for larger (SRFLP) instances.

	Results without spaces					Results with spaces added				
Instance	Lower bound	Upper bound	Average lower bound	Average upper bound	Average time (sec)	Lower bound	Upper bound	Average lower bound	Average upper bound	Average time (sec)
H_20	7402	8149	7591.0	8352.3	48.5	7247.5	8220	7467.0	8490.9	220.6
N25_05	7254	7945	7377.9	8148.2	128.7	7114	7977	7245.9	8297.1	524.4
H_30	20659.5	22801	21040.8	23846.8	313.2	20395	23678	20781.1	24230.3	1096.9
N30_05	52756.5	58425	54613.3	60318.2	310.4	51959.5	59274	54082.8	61617.6	1072.4
Am33_03	32058	35958.5	32429.3	37417.1	554.3	31569.5	36315	32016.9	37292.0	1697.6
Am35_03	31521	34794.5	31886.8	36614.9	720.1	31199.5	35492.5	31554.2	36324.3	2098.4
ste36.5	41409.5	47259.5	45397.9	51239.1	808.2	40691.5	50182.5	44254.9	55691.9	2458.6
N40_5	46877.5	55220	49809.6	58530.2	1464.4	46256	55107	49177.5	58539.1	4207.6
sko42-5	113606	127639.5	114862.9	132717.8	1959.6	112687	129293.5	113984.5	134034.7	5185.2
sko49-5	291004.5	349137	295443.0	355099.0	4904.0	289939.5	351509	293602.6	358216.5	12804.1
sko56-5	261686	306133.5	263678.4	315512.4	11849.1	259337.5	314842.5	261857	318948	24854.1
AKV-60-05	145702	171280	147436.4	176056.3	17162.7	144718	175098	146352.5	178835.7	37687.4
sko64-5	219646	261257.5	220945.6	268591.9	22828.3	217810	264242.5	219601.9	270922.1	49010.7
AKV-70-05	1861211	2196942.5	1870414.3	2237296.7	45232.4	1852731	2208733.5	1864956.8	2243198.2	81161.7

Table 12.9: Computational results for the (PROP) averaged over 10 row assignments. We again apply our semidefinite approach to ($\texttt{SDP}_3$) and compare space-free layouts with layouts where five spaces with lengths 2^k, $k \in \{-1, 0, 1, 2, 3\}$, are present in both rows.

optimization formulation and presented computational results showing that the proposed approach gives promising results for several variants of multi-row layout problems on a variety of benchmark instances.

There is a large variety of important layout problems both in the literature and in practice. Future research will study the application of the discrete modelling approach proposed in this paper, particularly Theorem 12.2, to other classes of layout problems.

Chapter 13

Solution Approaches for Equidistant Double- and Multi-Row Facility Layout Problems[1]

Authors: Miguel F. Anjos, Anja Fischer and Philipp Hungerländer

Abstract: The facility layout problem is a well-known operations research problem that arises in multiple applications. This paper is concerned with the multi-row layout problem in which one-dimensional departments are to be placed on a given number of rows so that the sum of the weighted center-to-center distances is minimized. While the optimal solution for a single-row problem will normally have no spaces between departments, for multi-row layout problems it is necessary to allow for the presence of spaces of arbitrary lengths between departments. We consider the special case of equidistant row layout problems in which all departments have the same length, taken to be unity without loss of generality. For this class of problems we prove two theoretical results that facilitate the handling of spaces. First we show that although the lengths of the spaces are in general continuous quantities, every multi-row equidistant problem has an optimal solution on the grid. This implies that only spaces of unit length need to be used when modeling the problem, and hence that the problem can be formulated as a purely discrete optimization problem. Second we state and prove exact expressions for the minimum number of spaces that need to be added so as to preserve at least one optimal solution. One important consequence of these results is that multi-row equidistant layout problems can be modeled using only binary variables; this has a significant impact for a computational perspective. These results are used to formulate two new models for the equidistant problem, an integer linear optimization model and a semidefinite optimization model. Special attention is paid to the double-row layout case that has received much attention recently and is particularly important in practice. Our computational results with the new formulations as well as with a recent formulation by Amaral show that the semidefinite approach dominates for medium- to large-sized instances and that it is well-suited for providing high-quality lower bounds for large-scale instances in reasonable computation time. Specifically for double-row instances, we attain global optimality for some instances with up to 25 departments, and achieve optimality gaps smaller than 1% for instances with up to 50 departments.

Keywords: Facilities planning and design; Flexible manufacturing systems; Row layout; Semidefinite Programming; Global Optimization; Combinatorial Optimization

[1] A short paper with a brief outline of the ideas as specialized to the double-row problem and without any technical details, was accepted for the proceedings of OR 2014.

13.1 Introduction

The facility layout problem is a well-known operations research problem that arises in multiple applications. The problem consists in finding an optimal location of departments inside a plant according to a given objective function. In general, the objective function may reflect transportation costs, the construction cost of a material-handling system, or simply adjacency preferences among departments. For example, the placement of machines that form a production line inside a plant is a layout problem in which one wishes to minimize the total cost of the material flow between the machines.

The variety of applications means that facility layout encompasses a broad class of optimization problems. This paper is concerned with the Multi-Row Equidistant Facility Layout Problem `(MREFLP)`. This is one of several row-layout problems that are of interest in the design of flexible manufacturing systems (FMSs). FMSs are automated production systems that typically consist of numerically controlled machines and material handling devices under computer control with the materials handled by devices such as automated guided vehicles (AGVs). It is well-known that the layout of the machines of an FMS has a significant impact on the productivity of the facility, and furthermore that a poor layout is likely to reduce the flexibility of an FMS [133]. Among most frequently encountered layout types in practice are the single-row and multi-row layouts (Figure 13.1). If all the departments are to be placed in only one row, then we have an instance of the Single-Row Facility Layout Problem `(SRFLP)`, while if more than one row can be used, then the problem is a Multi-Row Facility Layout Problem `(MRFLP)`. We are interested here in the `(SREFLP)` and the `(MREFLP)` which are respectively the special cases of the `(SRFLP)` and `(MRFLP)` in which all the machines have the same length.

Figure 13.1: AGV handling materials in a single-row layout (a) and a double-row layout (b).

Single-Row Layout. Arguably the simplest layout problem is that of a single-row layout. An instance of the Single-Row Facility Layout Problem `(SRFLP)` consists of n one-dimensional machines, with given positive lengths $l_1, \ldots, l_n$, and pairwise weights w_{ij} often referred to as connectivities. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ i<j}} w_{ij} z_{ij}^{\pi}, \tag{13.1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \ldots, n\}$ and z_{ij}^{π} is the center-to-center distance between machines i and j with respect to a particular permutation $\pi \in \Pi_n$.

Under the assumption that the weights w_{ij} are non-negative, the optimal solution will have no empty spaces between departments. Hence the `(SRFLP)` consists of finding a permutation of the departments that minimizes the total weighted sum of the center-to-center distances. Note that the assumption that $w_{ij} \geq 0$ also ensures boundedness of the objective value of the optimal layout.

Beyond the arrangement of machines in FMSs [150], practical applications of the `(SRFLP)` include the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], and the assignment of airplanes to gates in an airport terminal [291]. Accordingly several heuristic algorithms have been suggested for the `(SRFLP)`; among the best ones to date are [80, 188, 267].

Global optimization approaches for the `(SRFLP)` are based on relaxations of integer linear programming (ILP) and semidefinite programming (SDP) formulations. The strongest ILP approach is an LP-based cutting plane algorithm using betweenness variables that can solve instances with up to 35 departments within a few hours [8]. The strongest SDP approach to date using products of ordering variables is even stronger and can solve instances with up to 42 departments within a few hours [169].

In this context let us recall that SDP is the extension of LP from the set of non-negative vectors to the cone of symmetric positive semidefinite matrices. For further information on SDP we refer the reader to the handbooks [18, 303]. In particular, a survey of global optimization approaches for the (SRFLP) can be found in [19].

This leads us to one of the problems addressed in this paper. The Single-Row Equidistant Facility Layout Problem (SREFLP) is the special case of the (SRFLP) in which the department lengths are all equal. The (SREFLP) arises in several applications, including sheet-metal fabrication [15], printed circuit board and disk drive assembly [65], and the optimal design of a flowline in a manufacturing system [308]. Furthermore Bhasker and Sahni [38] applied the (SREFLP) to minimize the total wire length needed when arranging circuit components on a straight line.

The (SREFLP) is also a special case of the Quadratic Assignment Problem (QAP) (see e. g. [50, 211]). While exact methods and heuristics especially designed for the (SREFLP) clearly outperform general methods for the (QAP), this is not the case for recent approaches to the (SRFLP) [163]. Indeed the most effective global optimization approaches for the (SRFLP) are also the best ones for the (SREFLP). Hence to date the (SREFLP) can also be solved to optimality for instances with up to 42 departments within a few hours.

In contexts other than manufacturing, the (SREFLP) is usually called weighted Linear Arrangement (LA). This problem was originally proposed by Harper [131, 132] to develop error-correcting codes with minimal average absolute errors. It is NP-hard [112], and remains so even if all weights are binary and the underlying graph is bipartite [111]. It follows that all the problems considered in this paper are also NP-hard, as they are extensions of (LA).

Multi-Row Layout. The Double-Row Facility Layout Problem (DRFLP) is a natural extension of the (SRFLP) in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel (see Figure 13.1). The (DRFLP) is especially relevant for real-world applications because this is a common approach in practice for improved material handling and space usage, and thus real factory layouts most often reduce to a combination of single-row and double-row problems.

Row layout can be further generalized to the Multi-Row Facility Layout Problem (MRFLP) where the departments are arranged along several parallel rows. An instance of the (MRFLP) consists of n one-dimensional departments with given positive lengths $l_1, \dots, l_n$, pairwise non-negative weights w_{ij} between the departments, and a set $\mathcal{R} := \{1, \dots, m\}$ of rows available for placing the departments. The objective is to find an assignment $r\colon [n] \to \mathcal{R}$ of departments to rows, and feasible horizontal positions for the centers of the departments within the assigned rows, i. e., a function

$$p\colon [n] \to \mathbb{R} \text{ satisfying } \frac{l_i + l_j}{2} \leq |p(i) - p(j)| \text{ if } r(i) = r(j),$$

such that the total weighted sum of the center-to-center distances between all pairs of departments is minimized. Our formulation of the (MRFLP) is thus:

$$\min_{r,p} \quad \sum_{\substack{i,j \in [n] \\ i<j}} w_{ij} |p(i) - p(j)| \tag{13.2}$$

$$\text{s. t.} \quad \frac{l_i + l_j}{2} \leq |p(i) - p(j)|, \; i \neq j, \text{ and } \; r(i) = r(j). \tag{13.3}$$

The (MRFLP) has numerous applications such as computer backboard wiring [288], campus planning [90], scheduling [115], typewriter keyboard design [253], hospital layout [95], the layout of machines in an automated manufacturing system [151], balancing hydraulic turbine runners [198], numerical analysis [46], optimal digital signal processors memory layout generation [302]. Different extensions of the (MRFLP) like considering a clearance between any two adjacent machines given as a fuzzy set [114] or the design of an FMS in one or multiple rows [101] have been proposed and tackled with genetic algorithms.

Somewhat surprisingly, the development of exact algorithms for the (DRFLP) and the (MRFLP) has received only limited attention in the literature. In the 1980s Heragu and Kusiak [150] proposed a non-linear programming model and obtained locally optimal solutions to the (SRFLP) and the (DRFLP). Recently Chung and Tanchoco [72] (see also Zhang and Murray [310]) focused exclusively on the (DRFLP) and proposed a mixed integer linear programming (MILP) formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Their approach was able to solve instances with up to 10 departments. Amaral [11] proposed an improved MILP formulation that allowed him to solve instances with up to 12 departments to optimality. Most recently Hungerländer and Anjos [167] proposed an SDP approach for the general (MRFLP) that is however only applicable to small instances with less than 12 departments.

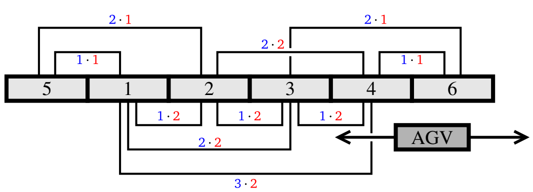
Again our interest in this paper is on the case of (MREFLP) that has all the department lengths equal. With respect to this special case, Amaral [9] proposed an MILP formulation tailored to the Minimum Duplex Arrangement Problem, which in our terminology corresponds to the (DREFLP). His approach allows to exploit the sparsity of the instances considered and is able to solve randomly generated instances with at most 10 (very dense instances) to 20 (very sparse instances) departments. For more details on this formulation we refer to Subsection 13.3.1.

A Toy Example. We illustrate the (SREFLP) and the (MREFLP) with the help of a toy example. We consider 6 equidistant departments and the following given pairwise weights:

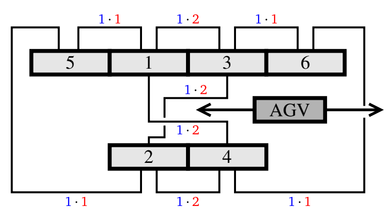
$$w_{12} = w_{13} = w_{14} = w_{23} = w_{24} = w_{34} = 2, \ w_{15} = w_{25} = w_{36} = w_{46} = 1.$$

Figure 13.2 illustrates the optimal layouts and corresponding total costs when using between one and four rows.

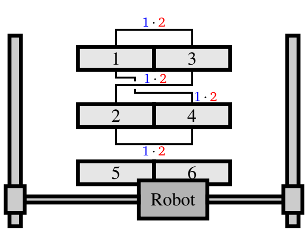
a.)



b.)



c.)



d.)

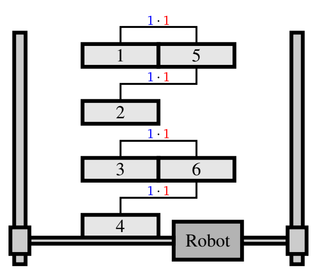


Figure 13.2: a.) Optimal (SREFLP) solution with total cost of 26.
b.) Optimal (DREFLP) solution with total cost of 12.
c.) Optimal solution for the (MREFLP) with 3 rows with total cost of 8.
d.) Optimal solution for the (MREFLP) with 4 rows with total cost of 4.

Outline. This paper is structured as follows. In Section 13.2 we state and prove our theoretical results on the structure of optimal layouts. Namely that the (MREFLP) always has an optimal solution on the grid, and that we can give exact expressions for the minimum number of spaces that need to be added to an instance of the (MREFLP) so that from an optimal solution to the resulting "space-free" problem we can recover at least one optimal solution for the (MREFLP) instance. One consequence of these results is that the (MREFLP) can be modeled using solely binary variables. In Section 13.3 we focus on the double-row case, recalling an MILP formulation for the (DREFLP) by Amaral [9] and presenting two new models for that problem, one based on ILP and the other on SDP. In Section 13.4 we show how the three models can be extended to the general multi-row case, and in Section 13.5 we describe a suitable combination of optimization methods to obtain strong lower bounds and feasible layouts using the presented models. Section 13.6 reports the results of our computational experiments to assess the practical performance of the different approaches. Finally, Section 13.7 concludes the paper and summarizes some directions for future research.

13.2 The Structure of Optimal Layouts

The definition of the (MREFLP) implicitly allows the spaces between two departments to be of arbitrary length. For this reason most optimization models in the literature use continuous variables to model the distances between departments.

In this section we prove two theoretical results about the structure of optimal layouts. We first show that the (MREFLP) always has an optimal solution on the grid. The key insight here is that restricting the spacing between departments to be formed using *spacing departments* preserves at least one optimal solution. Second we give exact expressions for the minimum number of such spacing departments required for each combination of numbers of departments and rows so as to preserve at least one optimal solution. One consequence of these results is that the (MREFLP) can be modeled using solely binary variables.

These results are of intrinsic theoretical interest because they reveal hitherto hidden structural properties of the (MREFLP). Moreover they are of use to improve the practical performance of the optimization models that we propose in Sections 13.3 and 13.4.

13.2.1 A Combinatorial Property of Multi-Row Layouts

Theorem 13.1 is a special case of [164, Theorem 2]:

Theorem 13.1 *There is always an optimal solution to the (MREFLP) on the grid.*

Proof. Let an optimal solution of the (MREFLP) be given. We define an integer grid such that the centers of the departments with the leftmost centers are on a grid point. Next we divide the departments into two sets, a set S containing those with their centers already on the grid, and a set T containing the others. We assume w. l. o. g. that the indices of the departments in S are all smaller than the indices of the departments in T: $i < j, \; \forall \, i \in S, \; j \in T$.

Observe that there exists $\varepsilon > 0$ sufficiently small so that we can move all the departments in T simultaneously, either to the left or to the right, by a distance ε. This holds because all departments have (the same) integer length, and because the departments in S are arranged on the grid. The change in the objective function from any such shift of the departments in T is given by

$$\delta = \sum_{i \in T} \left(\varepsilon \sum_{j \in S, \; j \dot{<} i} w_{ij} - \varepsilon \sum_{j \in S, \; i \dot{<} j} w_{ij} \right)$$

for a shift to the left, and by $-\delta$ for a shift to the right, where $i \dot{<} j$ means that the center of j is to the right of the center of i, and ε is chosen small enough such that no department in T traverses a grid point.

Due to the optimality of the given layout, δ has to be equal to zero because otherwise a shift either to the left (for $\delta < 0$) or to right (for $\delta > 0$) would improve the objective value. Hence the proposed shifting operation does not change the objective value.

Let us choose ε as the largest value such that the center of at least one department in T lies on a grid point after the shifting operation (to the left or right). If we apply this shifting to the given optimal solution, we can now move that department to the set S. Repeatedly applying this operation to the remaining departments allows us to arrange all departments on the grid in at most $n-1$ steps without changing the objective value. □

Theorem 13.1 is illustrated in Figure 13.3. For layouts fulfilling the grid property, we say that department i lies in column j if the center of i is located at the j^{th} grid point. For example department 5 lies in column 4 in Figure 13.3.

Note that the grid property is automatically fulfilled for layouts corresponding to the graph version of the (MREFLP), i. e., an extension of (LA) where two or more nodes can be assigned to the same position. Hence by Theorem 13.1 the Minimum Duplex Arrangement Problem considered in [9] is a special case of the (DREFLP).

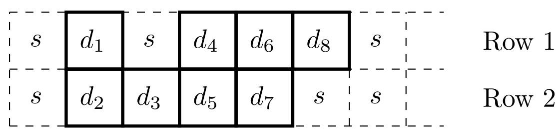


Figure 13.3: Illustration of the grid property of layouts. Note that for such layouts all departments and spaces have equal size.

From now on we restrict our attention to layouts fulfilling the grid property. This restriction is clearly advantageous from both a theoretical as well as a practical point of view.

13.2.2 Bounds on the Number of Spaces

In this subsection we are interested in the minimum number of spacing departments, or simply spaces, that must be added to an instance of the (MREFLP) so that we can recover at least one optimal solution for the original (MREFLP) instance from the optimal solution to the resulting problem. Clearly this number is a function of the number of departments and the number of rows, but since we do not have a priori knowledge about the structure of optimal solutions for given cost coefficients, it does not depend on the weights w_{ij} (other than assuming their non-negativity).

In the following theorem we make three additional assumptions that allow us to reduce the number of spaces needed. Note that at least one optimal layout is preserved under these assumptions (see Lemma 13.1 below for a formal proof of this statement).

Assumption 1 Columns that contain only spaces can be deleted. Equivalently, if we number the columns from 1 to n there exists $k' \in [n]$ such that each column with index at most k' contains at least one department.

Assumption 2 If two non-empty neighboring columns contain altogether no more than m departments, then all corresponding departments can be assigned to the left column and the right column can be deleted.

Thus with k' as in Assumption 1 we know that columns i and $i+1$ with $i \in [k'-1]$ contain at least $m+1$ departments.

Assumption 3 If $d > 2m$ and the first column and the third column contain in total at most m departments, then all corresponding departments can be assigned to the third column and the first column can be deleted.

A similar argument holds for columns $k' - 2$ and k' with k' as in Assumption 1.

These assumptions are illustrated in Figure 13.4 where the left-hand side depicts a feasible layout and the right-hand side depicts the adaptation of that layout so that the respective assumption holds. Note that the adaptations cannot worsen the objective value of the layout.

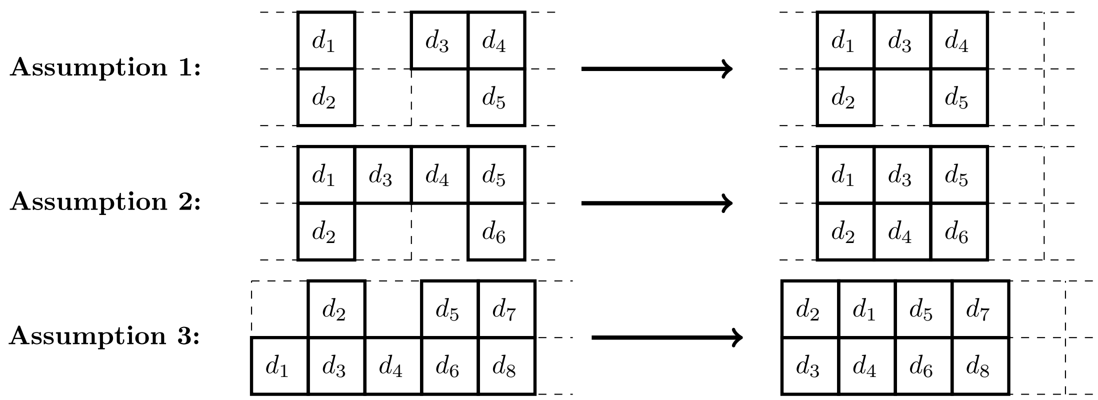


Figure 13.4: Illustration of Assumptions 1, 2 and 3.

We can now state the second theorem.

Theorem 13.2 *The number of columns sufficient to preserve at least one optimal layout for an instance with d departments is*

1. *equal to* 1 *if $d \leq m$, and equal to* 2 *if $m < d < \frac{3}{2}m + \frac{3}{2}$;*
2. *equal to $\lceil \frac{2d}{3} \rceil - 1$ for the* `(DREFLP)` *with $d \geq 9$;*
3. *equal to $\left\lfloor \frac{2d}{m+1} \right\rfloor$ for the* `(MREFLP)` *with an odd number of rows m; and*
4. *equal to $2l + 1$ for the* `(MREFLP)` *with an even number of rows m and $d \in \{\frac{m}{2} + 2 + (m+1)(l-1), \dots, \frac{m}{2} + 1 + (m+1)l\}$ for some $l \in \mathbb{N}$.*

To prove Theorem 13.2, we begin by using the fact that Theorem 13.1 allows us to assume that the departments of the `(MREFLP)` are arranged on a grid. Hence we can represent an optimal solution of the `(MREFLP)` by an assignment $\alpha\colon [d] \to [d]$ of the d departments to d different columns with the interpretation

$$\alpha(i) = j, \quad \text{if department } i \in [d] \text{ lies in column } j \in [d], \quad (i, j \in [d]) \tag{13.4}$$

and at most m departments are assigned to each column $j \in [d]$, i. e.,

$$|\{i \in [d] \colon \alpha(i) = j\}| \leq m.$$

Indeed, the modeling approach in [9] directly reflects the assignment (13.4) (see Subsection 13.3.1 for details). Furthermore, there always exists an optimal solution $\alpha^*\colon [d] \to [d]$ that fulfills additional structural properties that we already depicted in Figure 13.4 and now formally describe and prove in the next lemma.

Lemma 13.1 *Let $d, m \in \mathbb{N}$. Then there always exists an optimal solution $\alpha^*\colon [d] \to [d]$ of the* `(MREFLP)` *(fulfilling the grid structure) that assigns each department $i \in [d]$ to a column $\alpha^*(i) \in [d]$ which fulfills the following properties:*

1. *There exists a* $k' \in [d]$ *such that* $|\{i \in [d] \colon \alpha^*(i) = l\}| \geq 1$ *for all* $l \in [d]$, $l \leq k'$, *and* $|\{i \in [d] \colon \alpha^*(i) \geq k'+1\}| = 0$.
2. *If* $|\{i \in [d] \colon \alpha^*(i) = j\}| > 0$ *and* $|\{i \in [d] \colon \alpha^*(i) = j+1\}| > 0$ *for some* $j \in [d]$, $j < d$, *then* $|\{i \in [d] \colon \alpha^*(i) = j\}| + |\{i \in [d] \colon \alpha^*(i) = j+1\}| \geq m+1$.
3. *Let* $d > 2m$. *Then* $|\{i \in [d] \colon \alpha^*(i) \geq k'+1\}| = 0$ *and* $|\{i \in [d] \colon \alpha^*(i) = k'\}| > 0$ *for some* $k' \in [d]$ *imply* $|\{i \in [d] \colon \alpha^*(i) = k'-2\}| + |\{i \in [d] \colon \alpha^*(i) = k'\}| \geq m+1$. *Furthermore* $|\{i \in [d] \colon \alpha^*(i) = 1\}| + |\{i \in [d] \colon \alpha^*(i) = 3\}| \geq m+1$.

Proof. Let $d, m \in \mathbb{N}$ and α^* be an optimal solution of the `(MREFLP)` fulfilling the grid structure.

1. If $|\{i \in [d] \colon \alpha^*(i) = j-1\}| = 0$ and $|\{i \in [d] \colon \alpha^*(i) = j\}| \geq 1$ for some $j \in [d]$, then the assignment α' with

$$\alpha'(l) = \begin{cases} \alpha^*(l), & \alpha^*(l) < j, \\ \alpha^*(l) - 1, & \text{otherwise}, \end{cases}$$

for $l \in [d]$ is optimal for the `(MREFLP)`, too, because the distances between departments are not enlarged. The repeated "deletion" of empty columns proves the statement.

2. Assume that $|\{i \in [d] \colon \alpha^*(i) = j\}| + |\{i \in [d] \colon \alpha^*(i) = j+1\}| \leq m$ for some $j \in [d], j < d$. Then α' with

$$\alpha'(l) = \begin{cases} \alpha^*(l), & \alpha^*(l) \leq j, \\ \alpha^*(l) - 1, & \text{otherwise}, \end{cases}$$

for $l \in [d]$ is a feasible multi-row assignment and it is even optimal, because all distances are not enlarged (some are even shortened) and there are at most m departments in each row. Applying this approach repeatedly we get an optimal assignment $\bar{\alpha}$ such that $|\{i \in [d] \colon \bar{\alpha}(i) = j\}| > 0$ and $|\{i \in [d] \colon \bar{\alpha}(i) = j+1\}| > 0$ for some $j \in [d-1]$ imply $|\{i \in [d] \colon \bar{\alpha}(i) \in \{j, j+1\}\}| > m$.

3. Now assume, w. l. o. g., that there exists an optimal solution α^* of the `(MREFLP)` and $k' \in [d]$ such that $|\{i \in [d] \colon \alpha^*(i) = k'\}| > 1, |\{i \in [d] \colon \alpha^*(i) \geq k'+1\}| = 0$. By the previous statements we may assume $|\{i \in [d] \colon \alpha^*(i) = k'-1\}| > 0$ and $|\{i \in [d] \colon \alpha^*(i) \in \{k'-1, k'\}\}| > m$. If, additionally $|\{i \in [d] \colon \alpha^*(i) \in \{k'-2, k\}\}| \leq m$, the solution α' with

$$\alpha'(l) = \begin{cases} \alpha^*(l) - 2, & \alpha^*(l) = k', \\ \alpha^*(l), & \text{otherwise}, \end{cases}$$

for $l \in [d]$ is optimal, too, because all distances between departments are not enlarged.

□

We are now ready to prove Theorem 13.2.

Proof. (of Theorem 13.2) We prove each of the claims of Theorem 13.2 in turn.

- Proof of 1: Let $d, m \in \mathbb{N}$ be given. If $d \leq m$, it is clear that arranging all departments in one column leads to costs of zero. Furthermore, as long as $m < d < \frac{3}{2}m + \frac{3}{2}$ there exists an arrangement such that only two columns are used because, w. l. o. g., we can assume that the first two columns contain $m+1$ departments and that the second column contains maximal $\lceil \frac{m}{2} \rceil$ of these departments. Then, the remaining departments could also be included in one of the first two columns, either all in the second column or also some of them in the first column.

- Proof of 2: Let $m = 2$, $d \geq 9$ and let α^* be an optimal solution of the (DREFLP) fulfilling the grid structure as well as the properties described in Lemma 13.1. So we might assume that there exists a $k' \in [d]$ such that $|\{i \in [d]: \alpha^*(i) = l\}| \geq 1$ for all $l \in [d],\ l \leq k'$ and $|\{i \in [d]: \alpha^*(i) > k'\}| = 0$. (Note, $d \geq 9$ implies $k' \geq 5$.) By Lemma 13.1 the solution α^* fulfills $|\{i \in [d]: \alpha^*(i) \in \{j, j+1\}\}| \geq 3$ for all $j \in [d],\ j < k'$, as well as $|\{i \in [d]: \alpha^*(i) \in \{1,2,3\}\}| \geq 5$ and $|\{i \in [d]: \alpha^*(i) \in \{k'-2, k'-1, k'\}\}| \geq 5$. We consider two cases for k'. If $(k'-6) \mod 2 \equiv 0$, then the first k' columns contain at least $10 + (k'-6)\frac{3}{2} = \frac{3}{2}k' + 1$ departments. Otherwise, if $(k'-6) \mod 2 \equiv 1$, then the first k' columns contain at least $5 + (k'-3)\frac{3}{2} = \frac{3}{2}k' + \frac{1}{2}$ departments. Now, assume for a contradiction, that $k' \geq \lceil \frac{2d}{3} \rceil$. Then the first k' columns contain at least $\lceil (\frac{3}{2} \lceil \frac{2d}{3} \rceil + \frac{1}{2}) \rceil > d$ departments, a contradiction. So the statement follows.

- Proof of 3: Let m be odd and $d > 2m$. Let α^* be an optimal solution of the (MREFLP) that fulfills all properties described in Lemma 13.1. We might assume by Lemma 13.1 that there exists $k' \in [d]$ such that $|\{i \in [d]: \alpha^*(i) = l\}| \geq 1$ for all $l \in [d],\ l \leq k'$ and $|\{i \in [d]: \alpha^*(i) \geq k'+1\}| = 0$. Then we know by Lemma 13.1 that $|\{i \in [d]: \alpha^*(i) = j\}| + |\{i \in [d]: \alpha^*(i) = j+1\}| \geq m+1$ for all $j \in [d],\ j < k'$. Assume now, for a contradiction, that $k' > \left\lfloor \frac{2d}{m+1} \right\rfloor$, then the k' columns contain at least $\frac{m+1}{2} \cdot k' \geq \frac{m+1}{2}(\left\lfloor \frac{2d}{m+1} \right\rfloor + 1) > d$ departments, a contradiction.

- Proof of 4: Let m be even and $d > 2m$. Let α^* be an optimal solution of the (MREFLP) that fulfills all properties described in Lemma 13.1. Assume $d \in \{\frac{m}{2}+2+(m+1)(l-1), \ldots, \frac{m}{2}+1+(m+1)l\}$ for some $l \in \mathbb{N}$. We might assume by Lemma 13.1 that there is a $k' \in [d]$ such that $|\{i \in [d]: \alpha^*(i) = m\}| \geq 1$ for all $m \in [d],\ m \leq k'$ and $|\{i \in [d]: \alpha^*(i) \geq k'+1\}| = 0$. Then we know by Lemma 13.1 that $|\{i \in [d]: \alpha^*(i) = j\}| + |\{i \in [d]: \alpha^*(i) = j+1\}| \geq m+1$ for all $j \in [d],\ j < k'$. Assume now, for a contradiction, that $k' \geq 2l+2$. Then the first k' columns contain at least $\frac{2l+2}{2}(m+1) = (m+1)l + m + 1 > (m+1)l + \frac{m}{2} + 1 \geq d$ departments, a contradiction.

□

Table 13.1 gives exact values for the minimum number of columns for small values of d and problems with two to four rows.

d	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2 rows	1	1	2	2	3	4	4	5	5	6	7	7	8	9	9	10
3 rows	1	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
4 rows	1	1	1	1	2	2	2	3	3	4	4	4	5	5	5	6

Table 13.1: Minimum number of columns needed for instances of (MREFLP) with $d \leq 16$ and $m = 2, 3, 4$.

Let us give small toy examples for which the optimal layout contains many spaces and hence the number of columns given in Table 13.1 is necessary: consider problems with $m = 3$ rows, $d = 2l$ departments for some $l \in \mathbb{N}$, and with weights $w_{i(i+1)} = 1, i = 1, 3, 5, \ldots, 2l-1$, and $w_{ij} = \varepsilon$ otherwise. For ε sufficiently small, the optimal solution contains exactly one space in each column; the case with $d = 10$ is shown on the left-hand side of Figure 13.5. Note that in this example the objective value is not worsened if we reduce the number of rows from three to two.

Next let us point out that for the (MREFLP) with an even number of rows > 2, the exact calculation of the bounds is quite involved and might be slightly improved if d cannot be written as $\frac{m}{2} + 1 + (m+1)l$ for some $l \in \mathbb{N}$. Nevertheless, although the number of spaces seems large, sometimes no improvement is possible if we want to preserve an optimal solution. To see this consider a problem with four rows and 13 departments with $w_{12} = w_{13} = w_{45} = w_{67} = w_{68} = w_{9\,10} = w_{11\,12} = w_{11\,13} = 1$ and all other weights equal to a small $\varepsilon > 0$, then all optimal solutions have a structure like the one visualized on the right-hand side of Figure 13.5. In this case $d = 13 = \frac{4}{2} + 1 + 5l$ with $l = 2$.

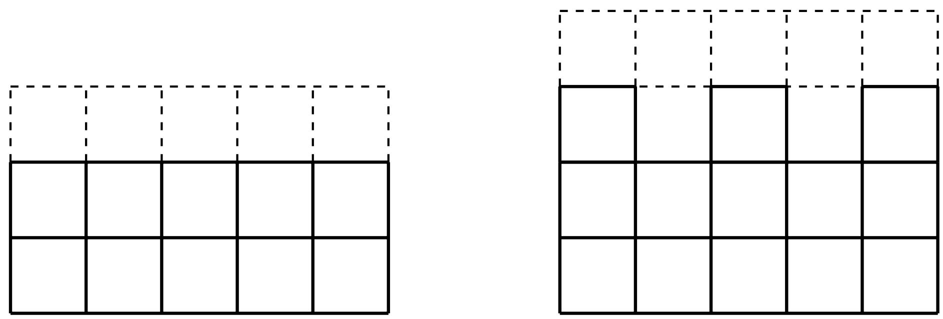


Figure 13.5: Worst-case examples for Theorem 13.2

Theorem 13.2 allows us to reduce the number of spaces, and hence of variables, both in the MILP model from Amaral [9] and in the new ILP and SDP formulations proposed in Section 13.3. This theorem also helps to eliminate some of the symmetries in the problem, for example the position of empty columns, and hence to obtain stronger global bounds from all the relaxations. The computational results in Section 13.6 demonstrate the practical impact of Theorem 13.2.

13.3 Three Modeling Approaches for Double-Row Layouts

In this section we focus on the double-row case. First we recall a MILP formulation for the `(DREFLP)` by Amaral [9]. Second, we present two new models for the `(DREFLP)`: the first one is an ILP formulation that uses betweenness variables together with variables modeling whether pairs of departments are assigned to the same column, and the second one is an SDP formulation based on products of ordering variables.

We note that in the approaches discussed below we do not assign the departments to a specific row (as was done for instance in recent SDP-based approaches to the `(MRFLP)` [167]). We instead ensure that at most m departments are assigned to each column.

13.3.1 A MILP Formulation Related to the Quadratic Assignment Problem

To the best of our knowledge the paper by Amaral [9] contains the only approach tailored specifically for the `(DREFLP)`. Let us briefly outline his ILP formulation. We introduce the binary variables $z_{ip} \in \{0,1\},\ i \in [d],\ p \in [c]$ (c the number of columns), with the interpretation

$$y_{ip} = \begin{cases} 1, & \text{department } i \text{ is assigned to column } p, \\ 0, & \text{otherwise.} \end{cases}$$

Using these variables we can rewrite the objective function (13.2) for the `(DREFLP)` as

$$\sum_{\substack{i,j \in [d], \\ i<j}} \sum_{\substack{p,q \in [c], \\ p<q}} w_{ij}(q-p) y_{ip} y_{jq}.$$

This quadratic objective function is linearized by introducing the binary variables[2]

$$z_{ipjq} = \begin{cases} 1, & \text{if department } i \text{ is assigned to column } p \text{ and department } j \text{ is assigned to column } q, \\ 0, & \text{otherwise,} \end{cases}$$

[2]Note that Amaral [9] introduced more binary variables y and z than we do, as he set the number of columns c to d.

with $i, j \in [d],\ p, q \in [c]$ and $(i \neq j, p < q) \vee (i < j, p = q),\ w_{ij} > 0$. Hence the (DREFLP) can be formulated as the following MILP:

$$\min \sum_{\substack{i,j\in[d],\ i<j \\ w_{ij}>0}} \sum_{\substack{p,q\in[c], \\ p<q}} w_{ij}(q-p)z_{ipjq} \tag{13.5}$$

$$\text{s.t.} \sum_{p\in[c]} y_{ip} = 1, \qquad i \in [d], \tag{13.6}$$

$$\sum_{i\in[d]} y_{ip} \leq 2, \qquad p \in [c], \tag{13.7}$$

$$y_{ip} + y_{jq} - z_{ipjq} \leq 1, \qquad i, j \in [d],\ p, q \in [c],\ (i \neq j, p < q) \vee (i < j, p = q),\ w_{ij} > 0, \tag{13.8}$$

$$y_{ip} \in \{0, 1\}, \qquad i \in [d],\ p \in [c], \tag{13.9}$$

$$z_{ipjq} \in [0, 1], \qquad i, j \in [d],\ p, q \in [c], (i \neq j, p < q) \vee (i < j, p = q),\ w_{ij} > 0. \tag{13.10}$$

Note that the constraints

$$z_{ipjq} \leq y_{ip},\ z_{ipjq} \leq y_{jq},\ i, j \in [d],\ p, q \in [c],\ (i \neq j, p < q) \vee (i < j, p = q),\ w_{ij} > 0$$

of the standard linearization can be omitted because the weights $w_{ij},\ i, j \in [d],\ i \neq j$, are assumed to be non-negative. In order to tighten this formulation Amaral [9] also applied the techniques of Sherali-Adams [278, 280]. This results in the following tighter MILP formulation:

$$\min \sum_{\substack{i,j\in[d],\ i<j, \\ w_{ij}>0}} \sum_{\substack{p,q\in[c], \\ p<q}} w_{ij}(q-p)z_{ipjq}$$

$$\text{s.t.} \quad (13.6)\text{–}(13.10),$$

$$\sum_{\substack{j\in[d],\ j>i, \\ w_{ij}>0}} z_{ipjp} \leq y_{ip}, \qquad i \in [d],\ p \in [c], \tag{13.11}$$

$$\sum_{\substack{j\in[d],\ j<i, \\ w_{ij}>0}} z_{jpip} \leq y_{ip}, \qquad i \in [d],\ p \in [c], \tag{13.12}$$

$$\sum_{\substack{q\in[c], \\ q>p}} z_{ipjq} + \sum_{\substack{q\in[c], \\ q<p}} z_{jqip} + z_{ipjp} \leq y_{ip}, \qquad i, j \in [d],\ i < j,\ w_{ij} > 0,\ p \in [c], \tag{13.13}$$

$$\sum_{\substack{j\in[d],\ j\neq i, \\ w_{ij}>0}} z_{ipjq} \leq 2y_{ip}, \qquad i \in [d],\ p, q \in [c],\ p < q, \tag{13.14}$$

$$\sum_{\substack{j\in[d],\ j\neq i, \\ w_{ij}>0}} z_{jqip} \leq 2y_{ip}, \qquad i \in [d],\ p, q \in [c],\ q < p. \tag{13.15}$$

13.3.2 An ILP Formulation Related to the Linear Ordering Problem

In this subsection we present a new ILP formulation for the (DREFLP). This formulation is an extension of the model proposed in [8] for the (SRFLP). We use additional variables to model that two departments can be assigned to the same column and additionally fill up the c columns with spaces (i. e., departments of length 1 and weights of zero). We collect all these spaces in a set S. To simplify notation we set the total number of departments (original ones plus spaces) to $n := 2c$ and the number of spaces is thus $s = n - d$. After the insertion of spaces we deal in fact with a space-free problem, and by Theorems 13.1 and 13.2 the optimal solution of the corresponding optimization problem is ensured to be an optimal solution of the (DREFLP).

Our model makes use of binary betweenness variables

$$b_{ijk} = b_{kji} \in \{0,1\},\ i,j,k \in [n],\ i < k,\ i \neq j \neq k,$$

and of binary column overlap variables

$$a_{ij} = a_{ji} \in \{0,1\}, i,j \in [n], i < j.$$

These two sets of variables have the following interpretations:

$$b_{ijk} = \begin{cases} 1, & \text{if department } j \text{ lies between departments } i \text{ and } k, \\ 0, & \text{otherwise,} \end{cases}$$

$$a_{ij} = \begin{cases} 1, & \text{if departments } i \text{ and } j \text{ are assigned to the same column,} \\ 0, & \text{otherwise.} \end{cases}$$

Our resulting formulation of the (DREFLP) is

$$\begin{aligned}
\min \quad & \sum_{i,j \in [n],\ i<j} \tfrac{w_{ij}}{2} \cdot \left(\sum_{k \in [n] \setminus \{i,j\}} b_{ikj} + 2(1 - a_{ij}) \right) && && (13.16)\\
\text{s.t.} \quad & a_{ij} + a_{ik} + a_{jk} + b_{ijk} + b_{ikj} + b_{jik} = 1, && i,j,k \in [n],\ i<j<k && (13.17)\\
& \sum_{j \in [n] \setminus \{i\}} a_{ij} = 1, && i \in [n], && (13.18)\\
& b_{ihj} + b_{ihk} + b_{jhk} \leq 2, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.19)\\
& - b_{ihj} + b_{ihk} + b_{jhk} + b_{ikj} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.20)\\
& + b_{ihj} - b_{ihk} + b_{jhk} + b_{ikj} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.21)\\
& + b_{ihj} + b_{ihk} - b_{jhk} + b_{ikj} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.22)\\
& - b_{ihj} + b_{ihk} + b_{jhk} + a_{hk} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.23)\\
& + b_{ihj} - b_{ihk} + b_{jhk} + a_{hk} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.24)\\
& + b_{ihj} + b_{ihk} - b_{jhk} + a_{hk} \geq 0, && i,j,k,h \in [n],\ i<j<k \neq h,\ i \neq h \neq j, && (13.25)\\
& b_{ijk} \in \{0,1\}, && i,j,k \in [n],\ i<j,\ i \neq k \neq j, && (13.26)\\
& a_{ij} \in \{0,1\}, && i,j \in [n],\ i<j. && (13.27)
\end{aligned}$$

The objective function (13.16) counts all departments that lie between the departments i and j, and because of the double-row structure the corresponding sum is divided by two. We count w_{ij} towards the cost if departments i and j do not lie in the same column.

Equations (13.17) express that three different departments lie either in three different columns such that one of the betweenness variables equals one or that exactly two of the three departments lie in the same column such that the associated overlap variable is one. With equations (13.18) we ensure that each department $i \in [n]$ lies in the same column with exactly one other department. Inequalities (13.19) to (13.25) are extensions of the inequalities in [8] for the SRFLP: inequality (13.19) ensures that a department h cannot lie between each two of the three departments $i,j,k \in [n] \setminus \{h\},\ i<j<k$, and inequalities (13.20)–(13.25) ensure that if department h lies between departments i and j, then h lies also between i,k or j,k or it lies in the same column as k, which also implies that k lies between i and j.

Due to the introduction of spaces our model contains some symmetries that should be broken to improve the practical performance of the model. The following constraints enforce an order of the s spaces such that space i lies left of space j or is in the same column as j iff $i < j,\ i, j \in S$:

$$a_{ij} = 0, \qquad i, j \in S,\ i + 2 \leq j, \tag{13.28}$$
$$b_{ijk} = 1, \qquad i, j, k \in S,\ i + 4 \leq j + 2 \leq k, \tag{13.29}$$
$$b_{ijk} = 0, \qquad i, j, k \in S,\ i \neq k,\ (j > \max\{i, k\} \vee j < \min\{i, k\}). \tag{13.30}$$

A further way to improve the presented model is to include an adapted variant of certain inequalities for the (SRFLP) proposed by Amaral [8].

Observation 13.2 *Let $\beta \in \mathbb{N}, \beta \geq 4$, be even and let $T \subseteq [n]$ with $|T| = \beta$. For a partition of T in $T_1, T_2, \{k\}$ such that $T = T_1 \dot\cup T_2 \dot\cup \{k\}, (T_1 \cap T_2 = \emptyset, k \notin T_1, k \notin T_2)$ and $|T_1| = \frac{\beta}{2}$ the following inequalities are valid for the (DREFLP)*

$$\sum_{\substack{p,q \in T_1,\\ p<q}} b_{pkq} + \sum_{\substack{p,q \in T_2,\\ p<q}} b_{pkq} - \sum_{\substack{p \in T_1,\\ q \in T_2}} b_{pkq} \leq \sum_{p \in T_2} a_{kp}, \tag{13.31}$$

$$\sum_{\substack{p,q \in T_1,\\ p<q}} b_{pkq} + \sum_{\substack{p,q \in T_2,\\ p<q}} b_{pkq} - \sum_{\substack{p \in T_1,\\ q \in T_2}} b_{pkq} \leq \sum_{\substack{p,q \in T_1,\ o \in T_2,\\ p<q}} b_{poq}. \tag{13.32}$$

Proof. Let $\beta \in \mathbb{N}, \beta \geq 4$, even and $T \subseteq [n]$ with $|T| = \beta$ be given. We consider a partition of T into $T_1, T_2, \{k\}$ such that $T = T_1 \dot\cup T_2 \dot\cup \{k\}$ and $|T_1| = \frac{\beta}{2}$ (so $|T_2| = \frac{\beta}{2} - 1$). In order to prove that inequalities (13.31) and (13.32) are valid for the (DREFLP) we consider a fixed double-row assignment $\alpha \colon [n] \to [\frac{n}{2}]$ that assigns each of the n departments (original and spaces) to one of the columns. We define $\sigma_1^1 := |\{i \in T_1 \colon \alpha(i) < \alpha(k)\}|$, $\sigma_1^2 := |\{i \in T_2 \colon \alpha(i) < \alpha(k)\}|$, $\sigma_2^1 := |\{i \in T_1 \colon \alpha(i) > \alpha(k)\}|$, $\sigma_2^2 := |\{i \in T_2 \colon \alpha(i) > \alpha(k)\}|$, $\sigma_3^1 := |\{i \in T_1 \colon \alpha(i) = \alpha(k)\}|$, $\sigma_3^2 := |\{i \in T_2 \colon \alpha(i) = \alpha(k)\}|$. Then $\sigma_1^1 + \sigma_2^1 + \sigma_3^1 = \frac{\beta}{2}$, $\sigma_1^2 + \sigma_2^2 + \sigma_3^2 = \frac{\beta}{2} - 1$ and $\sigma_3^1 + \sigma_3^2 \leq 1$. The left-hand side of (13.31) and (13.32) calculates to

$$\sigma_1^1\sigma_2^1 + \sigma_1^2\sigma_2^2 - \sigma_1^1\sigma_2^2 - \sigma_2^1\sigma_1^2 = -(\sigma_1^1 - \sigma_1^2)^2 + \sigma_1^1 - \sigma_1^2 - \sigma_1^1\sigma_3^1 + \sigma_1^1\sigma_3^2 + \sigma_3^1\sigma_1^2 - \sigma_1^2\sigma_3^2 =: \gamma.$$

We consider three cases:

- $\sigma_3^1 = \sigma_3^2 = 0$: Then $\gamma = -(\sigma_1^1 - \sigma_1^2)^2 + \sigma_1^1 - \sigma_1^2 = -(\sigma_1^1 - \sigma_1^2)(\sigma_1^1 - \sigma_1^2 - 1) \leq 0$ and with $a_{ij} \geq 0,\ i, j \in [n],\ i < j,\ b_{ijk}, i, j, k \in [n],\ i < k,\ |\{i, j, k\}| = 3$, the validity follows in this case.
- $\sigma_3^1 = 1, \sigma_3^2 = 0$: Then $\gamma = -(\sigma_1^1 - \sigma_1^2)^2 + \sigma_1^1 - \sigma_1^2 - \sigma_1^1 + \sigma_1^2 = -(\sigma_1^1 - \sigma_1^2)^2$ and with $a_{ij} \geq 0,\ i, j \in [n],\ i < j,\ b_{ijk},\ i, j, k \in [n],\ i < k,\ |\{i, j, k\}| = 3$, the validity follows in this case.
- $\sigma_3^1 = 0, \sigma_3^2 = 1$: Then $\gamma = -(\sigma_1^1 - \sigma_1^2)^2 + \sigma_1^1 - \sigma_1^2 + \sigma_1^1 - \sigma_1^2 = -(\sigma_1^1 - \sigma_1^2)(\sigma_1^1 - \sigma_1^2 - 2)$. This term is positive if and only if $\sigma_1^1 - \sigma_1^2 = 1$ by the integrality of the σ_i^j.

So, it suffices to show that the right-hand sides of (13.31) and (13.32) are at least one if $\sigma_3^1 = 0,\ \sigma_3^2 = 1$ and $\sigma_1^1 - \sigma_1^2 = 1$. For (13.31) the term $\sigma_3^2 = 1$ implies the existence of an $o \in T_2$ that lies in the same column as k. Considering (13.32), $\sigma_3^2 = 1$ and $\sigma_1^1 - \sigma_1^2 = 1$ imply $\sigma_1^1 > 0,\ \sigma_2^1 > 0$ and so there exist $p, q \in T_1,\ p \neq q$, and $o \in T_2$ such that o lies between p, q. □

Taking $\beta = 4$ we obtain exactly (13.20)–(13.25). In comparison to the variant for the (SRFLP) we added $\sum_{p \in T_2} a_{rp}$ or $\sum_{p,q \in T_1, o \in T_2, p<q} b_{poq}$ to the previous right-hand side with value zero, respectively.

13.3.3 An SDP Formulation Related to the Linear Ordering Problem

We now present another new formulation for the (DREFLP). This formulation is based on a quadratic formulation using ordering variables that we rewrite using symmetric matrices. The matrix-based formulation is then relaxed into an SDP problem, and this SDP relaxation can be tightened using several classes of valid constraints.

Our quadratic formulation is based on the ordering variables $x_{ij},\ i,j \in [n],\ i \neq j$, defined as:

$$x_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies left of department } j, \\ -1, & \text{otherwise.} \end{cases} \tag{13.33}$$

We observed in Subsection 13.3.2 that the center-to-center distances between departments can be encoded using betweenness variables and column overlap variables. Because we are willing to work with quadratic terms, we can express both of these variables using the ordering variables:

$$\begin{aligned} b_{ikj} &= \tfrac{1}{4}(x_{ik}x_{kj} + x_{jk}x_{ki} + x_{ik} + x_{kj} + x_{jk} + x_{ki}) + \frac{1}{2}, \quad i,j,k \in [n],\ i < j, \\ a_{ij} &= -\tfrac{1}{2}(x_{ij} + x_{ji}), \quad i,j \in [n],\ i < j. \end{aligned} \tag{13.34}$$

It directly follows that we can rewrite the objective function (13.16) as a linear-quadratic function of the ordering variables:

$$K + \sum_{\substack{i,j \in [n] \\ i<j}} \frac{w_{ij}}{8} \left(\sum_{\substack{k \in [n] \\ k \neq i,\ k \neq j}} (x_{ik}x_{kj} + x_{jk}x_{ki}) \right) + \sum_{\substack{i,j \in [n] \\ i<j}} \frac{w_{ij}}{4}(x_{ij} + x_{ji}), \tag{13.35}$$

where K is a constant defined as

$$K := n \left(\sum_{\substack{i,j \in [n] \\ i<j}} \frac{w_{ij}}{4} \right). \tag{13.36}$$

Any feasible ordering of the departments has to satisfy the 3-cycle inequalities

$$-1 \leq x_{ij} + x_{jk} - x_{ik} \leq 1, \qquad i,j,k \in [n],\ i \neq j \neq k,\ i \neq k. \tag{13.37}$$

It is well known that the 3-cycle inequalities together with integrality conditions on the ordering variables suffice to describe feasible orderings, see e.g. [296, 306]. In the present context we need the following additional constraints

$$x_{ij} + x_{ji} \leq 0, \quad i,j \in [n],\ i < j, \tag{13.38}$$

that model the fact that:

- either department i lies to the left of department j;
- or department j lies to the left of department i;
- or both departments are assigned to the same column.

Note from the definition of the ordering variables that if two departments i and j are placed in different columns then $x_{ij} + x_{ji}$ equals zero, while if they are assigned to the same column the sum is -2. This observation is often used in models using ordering variables, such as the ones for the `(SRFLP)`, to halve the number of variables because they require that $x_{ij} + x_{ji} = 0$, i. e., no two departments can overlap. While some overlap is allowed here, we can ensure that exactly two departments are assigned to each column using the constraints

$$\sum_{j\in[n]\setminus\{i\}} (x_{ij} + x_{ji}) = -2, \quad i \in [n]. \tag{13.39}$$

Next we collect the ordering variables in a vector x and reformulate the (DREFLP) as a quadratic program in ordering variables.

Theorem 13.3 *Minimizing the objective function* (13.35) *over* $x \in \{-1,1\}^{n(n-1)}$ *and* (13.37)–(13.39) *solves the (DREFLP).*

Proof. The constraints (13.37)–(13.39) together with the integrality conditions on x suffice to induce feasible double-row layouts and the definition of the objective function ensures that the distances between departments are computed correctly. □

We can rewrite the quadratic objective function (13.35) in matrix notation to obtain:

$$\min \{\langle C_X, X\rangle + c_x^\top x + K : x \in \{-1,1\}^{n(n-1)} \text{ satisfies (13.37)–(13.39)}\}, \tag{DREFLP}$$

where $X := xx^\top$ and the cost matrix C_X and the cost vector c_x are deduced from (13.35):

$$\langle C_X, X\rangle = \sum_{\substack{i,j\in[n]\\ i<j}} \frac{w_{ij}}{8} \sum_{\substack{k\in[n]\\ i\neq k\neq j}} (x_{ik}x_{kj} + x_{jk}x_{ki}),$$

$$c_x^\top x = \sum_{\substack{i,j\in[n]\\ i<j}} \frac{w_{ij}}{4}(x_{ij} + x_{ji}).$$

We can further rewrite the above formulation as an SDP by relaxing the nonconvex equation $X - xx^\top = 0$ to the positive semidefinite constraint

$$X - xx^\top \succcurlyeq 0.$$

Moreover, the main diagonal entries of X correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(X) = e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(x, X) := \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix}, \tag{13.40}$$

where $\dim(Z) = n(n-1)+1$. By the Schur complement lemma [43, Appendix A.5.5], $X - xx^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. Hence any feasible layout is contained in the elliptope

$$\mathcal{E} := \{Z : \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\}.$$

In order to express constraints on x in terms of X, they have to be reformulated as quadratic conditions in x. A natural way to do this for the 3-cycle inequalities $|x_{ij} + x_{jk} - x_{ik}| = 1$ consists in squaring both sides. Additionally using $x_{ij}^2 = 1$, we obtain

$$x_{ij,jk} - x_{ij,ik} - x_{ik,jk} = -1, \qquad i,j,k \in [n],\ i \neq j \neq k,\ i \neq k. \tag{13.41}$$

Now we can formulate the (DREFLP) as a semidefinite optimization problem in binary variables.

Theorem 13.4 *The problem*

$$\min \left\{K + \langle C_Z, Z\rangle : Z \text{ satisfies (13.41)},\ Z \in \mathcal{E},\ x \in \{-1,1\}^{n(n-1)} \text{ satisfies (13.38) and (13.39)}\right\}$$

where Z is given by (13.40)*, K is defined in* (13.36) *and the cost matrix C_Z is given by*

$$C_Z := \begin{pmatrix} 0 & \frac{1}{2}c_x \\ \frac{1}{2}c_x & C_X \end{pmatrix},$$

is equivalent to the (DREFLP)*.*

Proof. Since $x_i^2 = 1,\ i \in \{1, \ldots, n(n-1)\}$ we have $\operatorname{diag}(X - xx^\top) = 0$, which together with $X - xx^\top \succcurlyeq 0$ shows that in fact $X = xx^\top$ is integral. Hence the 3-cycle equations (13.41) ensure that $|x_{ij} + x_{jk} - x_{ik}| = 1$ holds. But the constraints (13.37)–(13.39) together with the integrality of x suffice to induce feasible double-row layouts due to Theorem 13.3. Finally the definition of K and C_Z ensures that the distances between departments are computed correctly. □

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of the (DREFLP):

$$\min\ \{K + \langle C_Z, Z\rangle : Z \text{ satisfies (13.41)},\ Z \in \mathcal{E},\ x \text{ satisfies (13.38) and (13.39)}\}. \qquad (\texttt{SDP}_2)$$

There are several ways to tighten the above relaxation. First we will concentrate on finding further valid equality constraints. Let us start with showing that the equations (13.17) from our ILP model are already described via 3-cycle equations (13.41).

Lemma 13.3 *The equations* (13.17)*,*

$$a_{ij} + a_{ik} + a_{jk} + b_{ijk} + b_{ikj} + b_{jik} = 1, \qquad i, j, k \in [n],\ i < j < k,$$

can be expressed as the sum of two equations of the form (13.41)*.*

Proof. Applying (13.34) to (13.17) gives

$$x_{ik,kj} + x_{jk,ki} + x_{ij,jk} + x_{kj,ji} + x_{ji,ik} + x_{ki,ij} = -2,\ i, j, k \in [n],\ i < j < k,$$

which is the sum of the following two equations from (13.41):

$$x_{ij,jk} + x_{ki,ij} + x_{jk,ki} = -1, \qquad x_{ik,kj} + x_{kj,ji} + x_{ji,ik} = -1.$$

□

Next we add symmetry-breaking constraints arising from the addition of spaces (as already seen in Subsection 13.3.2):

$$x_{21} = -1, \qquad (13.42)$$

$$x_{ij} = 1, \qquad i, j \in S,\ i + 2 \leq j, \qquad (13.43)$$

$$x_{ij} = -1, \qquad i, j \in S,\ j < i, \qquad (13.44)$$

$$\left.\begin{aligned} x_{i(i+1),ki} - x_{ki} - x_{i(i+1)} &= -1, \\ x_{i(i+1),k(i+1)} - x_{k(i+1)} - x_{i(i+1)} &= -1, \end{aligned}\right\} \quad i \in S,\ i \neq n,\ k \in [d]. \qquad (13.45)$$

Constraint (13.42) breaks the symmetry of the overall arrangement. Constraints (13.43) ensure that two spaces i and j can only be assigned to the same column if $i + 1 = j$. Equations (13.44) guarantee that in all layouts considered the spaces have increasing labels when going from left to right. Finally constraints (13.45) are related to Assumption 1 in Subsection 13.2.2: if two spaces $i, j \in S$ lie in the same column, then each department $k \in [d]$ has to lie left to them (see also Figure 13.4).

Lemma 13.4 *The ILP symmetry-breaking equations* (13.28)–(13.30) *can be derived from* (13.43)–(13.45).

Proof. Using equations (13.34) that relate the variables of the ILP and SDP models, we get:

- Let $i, j \in S, i + 2 \leq 2$, then $a_{ij} = -\frac{1}{2}(x_{ij} + x_{ji}) = -\frac{1}{2}(1 - 1) = 0$.
- Let $i, j, k \in S, i + 4 \leq j + 2 \leq k$, then $b_{ijk} = \frac{1}{4}(x_{ij}x_{jk} + x_{kj}x_{ji} + x_{ij} + x_{jk} + x_{kj} + x_{ji}) + \frac{1}{2} = \frac{1}{4}(1 \cdot 1 + (-1) \cdot (-1) + 1 + 1 - 1 - 1) + \frac{1}{2} = 1$.
- Let $i, j, k \in S, i \neq k, j > \max\{i, k\}$, then $b_{ijk} = \frac{1}{4}(x_{ij}x_{jk} + x_{kj}x_{ji} + x_{ij} + x_{jk} + x_{kj} + x_{ji}) + \frac{1}{2} = \frac{1}{4}(-x_{ij} - x_{kj} + x_{ij} - 1 + x_{kj} - 1) + \frac{1}{2} = 0$.
- Let $i, j, k \in S, i \neq k, j < \min\{i, k\}$, then $b_{ijk} = \frac{1}{4}(x_{ij}x_{jk} + x_{kj}x_{ji} + x_{ij} + x_{jk} + x_{kj} + x_{ji}) + \frac{1}{2} = \frac{1}{4}(-x_{jk} - x_{ji} - 1 + x_{jk} - 1 + x_{ji}) + \frac{1}{2} = 0$.

□

Equations (13.43) and (13.44) allow us to reduce the size of the semidefinite problem when it comes to the computational experiments in Section 13.6. However this requires all constraints containing the relevant variables to be transformed accordingly. While this is a straightforward exercise, it involves much technical detail that does not provide further insights. For this reason, we do not include the details of this transformation or of the resulting constraints. (For the same reason, we also chose not to exploit (13.42) though this could be done in principle.)

Again because we allow quadratic terms, we can express the inequalities (13.38) as equations:

$$x_{ij}x_{ji} + x_{ij} + x_{ji} = -1, \qquad i, j \in [n],\ i < j. \tag{13.46}$$

Equation (13.46) is valid because either $x_{ij} = x_{ji} = -1$ (both departments lie in the same column) or $x_{ij} + x_{ji} = 0$ and $x_{ij}x_{ji} = -1$ (they lie in different columns).

The theoretically smoothest way to deal with equations (13.39) would be to use them to reduce the dimension of the problem by n (for details see [156, Proposition 4.4]). Unfortunately this would make the practical implementation much more complicated. An alternative is to lift (13.39) into quadratic space via multiplication by an arbitrary ordering variable $x_{lm},\ l, m \in [n],\ l \neq m$, and the addition of the resulting linear-quadratic equations to the semidefinite relaxation:

$$\sum_{\substack{j \in [n] \\ j \neq i}} (x_{ij}x_{lm} + x_{ji}x_{lm}) = -2x_{lm}, \qquad i, l, m \in [n],\ l \neq m. \tag{13.47}$$

Another class of valid inequalities for our model are the triangle inequalities of the max-cut polytope, see e.g. [88]: Since Z is generated as the outer product of the vector $\begin{pmatrix} 1 & x \end{pmatrix}^\top$ that has merely $\{-1, 1\}$ entries in the non-relaxed SDP formulation, any feasible layout also belongs to the metric polytope $\mathcal{M}$:

$$\mathcal{M} = \left\{ Z\colon \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \leq e,\ \ 1 \leq i < j < k \leq n(n-1) + 1 \right\}. \tag{13.48}$$

We note that $\mathcal{M}$ is defined through $\approx 4n^6$ facets.

In summary we get the following tractable semidefinite relaxation of the (DREFLP):

$$\min \left\{ K + \langle C_Z, Z \rangle \colon Z \in \mathcal{E} \cap \mathcal{M} \text{ satisfies (13.41)–(13.47)} \right\}. \tag{SDP$_4$}$$

All variables in Z with cost coefficient greater than zero appear in a 3-cycle equality (13.41) or in equations (13.47) and thus are tightly constrained in the relaxation. Such tightly constrained variables are also the reason why the various linear and semidefinite relaxations for the (SRFLP) that are based on betweenness or ordering variables produce very tight bounds in the root node relaxation even for large instances.

13.4 Extending the Models to the Multi-Row Case

In this section we generalize the approaches for double-row problems in the previous section to multi-row problems. Given the set $[d]$ of departments with unit length, we seek an arrangement of these in $m \in \mathbb{N}$ rows such that the weighted sum of the pairwise distances is minimized. Based on the study of optimal solutions, we present adapted versions of the MILP, ILP and SDP models from the double-row case.

We again use Theorem 13.2 to reduce the (MREFLP) to a space-free version by introducing enough spacing departments. Let c be the minimal number of columns needed in order to preserve at least one of the original optimal solutions. Then our transformed problem has $n := cm$ departments, where $s = n - d$ are spaces.

Extending the MILP Formulation from Subsection 13.3.1. There is a straightforward way to extend the model by Amaral [9] to the multi-row case using the same variables. It suffices to change constraint (13.7) to

$$\sum_{i\in[d]} y_{ip} \leq m, \quad p \in [c], \tag{13.49}$$

and we obtain a formulation for the m-row case by optimizing (13.5) subject to (13.6), (13.49), and (13.8)–(13.10). For the tighter MILP formulation, (13.14) and (13.15) can be adjusted similarly.

Extending the ILP Formulation from Subsection 13.3.2. A formulation of the space-free (MREFLP) is given by

$$\min \sum_{\substack{i,j\in[n],\\ i<j}} \tfrac{w_{ij}}{m} \cdot \left(\sum_{k\in[n]\setminus\{i,j\}} b_{ikj} + m(1-a_{ij}) \right) \tag{13.50}$$

$$\begin{aligned}
\text{s.t. } & (13.19)\text{–}(13.27) \\
& a_{ij} + b_{ijk} + b_{ikj} + b_{jik} \leq 1, && i,j,k \in [n],\ i<j<k && (13.51)\\
& a_{ik} + b_{ijk} + b_{ikj} + b_{jik} \leq 1, && i,j,k \in [n],\ i<j<k && (13.52)\\
& a_{jk} + b_{ijk} + b_{ikj} + b_{jik} \leq 1, && i,j,k \in [n],\ i<j<k && (13.53)\\
& a_{ij} + a_{jk} - a_{ik} \leq 1, && i,j,k \in [n],\ i<k,\ i\neq j\neq k, && (13.54)\\
& \sum_{j\in[n]\setminus\{i\}} a_{ij} = m-1, && i \in [n], && (13.55)\\
& \sum_{\substack{i,j,k\in[n],\\ i<k,\ j\neq k,\ j\neq i}} b_{ijk} = m^3 \binom{c}{3}. && && (13.56)
\end{aligned}$$

Inequalities (13.51)–(13.53) express that three departments $i, j, k \in [n],\ i < j < k$, either lie next to each other or at least two of them are in the same column. Note, in the double-row case we could use the strengthened version (13.17). The inequalities (13.54) enforce the transitivity property that if departments i and j as well as j and k lie in the same column, then i and k also lie in the same column. Equations (13.55) are the generalization of (13.18) for the (DREFLP): each i lies in the same column as $m - 1$ other departments (possibly spaces). Finally, we know exactly how many betweenness variables equal 1 in a feasible solution: let $c_1, c_2, c_3 \in \{1, \ldots, c\}$ be three different columns of a solution, then for each choice of one department from each of the three columns we count 1 towards the left-hand side of (13.56).

Extending the SDP Formulation from Subsection 13.3.3. The starting point for our semidefinite relaxation for the (MREFLP) is again a quadratic problem in ordering variables. We use the x-variables defined in (13.33) and the 3-cycle inequalities (13.37) as well as (13.38). We change (13.39) to

$$\sum_{j \in [n] \setminus \{i\}} (x_{ij} + x_{ji}) = -2m + 2, \quad i \in [n], \tag{13.57}$$

and adjust the objective function (13.35)

$$K^m + \sum_{\substack{i,j \in [d], \\ i<j}} \frac{w_{ij}}{4m} \sum_{k \in [n] \setminus \{i,j\}} (x_{ik}x_{kj} + x_{jk}x_{ki}) + \sum_{\substack{i,j \in [d], \\ i<j}} \frac{(m-1)w_{ij}}{2m} (x_{ij} + x_{ji}), \tag{13.58}$$

where $K^m = \frac{n}{2m} \sum_{\substack{i,j \in [d], \\ i<j}} w_{ij}$.

The following result for the (MREFLP) follows directly from Theorem 13.3.

Corollary 13.5 *Minimizing* (13.58) *over* $x \in \{0,1\}^{n(n-1)}$ *and* (13.37), (13.38), (13.57) *solves the (MREFLP).*

In analogy to the double-row case, we can rewrite the (MREFLP) in matrix notation as

$$\min \left\{ \langle C_X^m, X \rangle + c_x^m x + K^m \colon x \in \{-1,1\}^{n(n-1)} \text{ satisfies (13.37),(13.38) and (13.57)} \right\}, \tag{MREFLP}$$

where $X := xx^\top$ and the cost matrix C_X^m and cost vector c_x^m are deduced from (13.58). Rewriting the above formulation along the lines of the double-row case gives

$$\min \left\{ K^m + \langle C_Z^m, Z \rangle \colon Z \text{ satisfies (13.41)},\ Z \in \mathcal{E},\ x \in \{-1,1\}^{n(n-1)} \text{ satisfies (13.38) and (13.57)} \right\}$$

where Z is given by (13.40), K^m is defined in (13.58) and the cost matrix C_Z^m is given by

$$C_Z^m := \begin{pmatrix} 0 & \frac{1}{2} c_x^m \\ \frac{1}{2} c_x^m & C_X^m \end{pmatrix}.$$

The basic semidefinite relaxation of the (MREFLP) reads

$$\min \left\{ K^m + \langle C_Z^m, Z \rangle \colon Z \text{ satisfies (13.41)},\ Z \in \mathcal{E},\ x \text{ satisfies (13.38) and (13.57)} \right\}. \tag{$\mathtt{SDP}^m_{\mathtt{basic}}$}$$

In order to strengthen this relaxation we use

$$\sum_{j \in [n] \setminus \{i\}} (x_{ij}x_{kl} + x_{ji}x_{kl}) = (-2m+2)x_{kl}, \qquad i,k,l \in [n],\ k \neq l, \tag{13.59}$$

which can be derived by multiplying (13.57) for fixed i with an x-variable $x_{kl},\ k,l \in [n],\ k \neq l$. Furthermore we can use (13.46) instead of (13.38).

Finally, we add constraints to break the symmetry of the spaces S:

$$x_{ij} = 1, \qquad i,j \in S,\ i+m \leq j, \tag{13.60}$$
$$x_{ij} = -1, \qquad i,j \in S,\ j < i, \tag{13.61}$$
$$x_{ij,ki} - x_{ki} - x_{ij} = -1, \qquad i,j \in S,\ j = i+m-1,\ k \in [d], \tag{13.62}$$
$$x_{ij,kj} - x_{kj} - x_{ij} = -1, \qquad i,j \in S,\ j = i+m-1,\ k \in [d], \tag{13.63}$$
$$-x_{i(i+j),i(i+k)} + x_{i(i+k)} - x_{i(i+j)} = -1, \qquad i \in S,\ j,k \in \mathbb{N},\ k < j < m,\ i+j \leq n. \tag{13.64}$$

The constraints (13.60) and (13.61) express that two spaces $i, j \in S,\ i < j$, can only lie in the same column if $i+m > j$. If two spaces $i, (i+m-1) \in S$ lie in the same column each of the original departments $k \in [d]$ lies left to them, see (13.62)–(13.63). Furthermore, if two spaces $i, (i+j) \in S$ lie in the same column, then all spaces $i+1, \ldots, i+j-1$ also lie in this column by equation (13.64). Additionally, we can use equation (13.42) and the triangle inequalities described in (13.48).

In summary we obtain the following tractable semidefinite relaxation of the (MREFLP):

$$\min\left\{K^m + \langle C_Z^m, Z\rangle \colon Z \in \mathcal{E} \cap \mathcal{M} \text{ satisfies (13.41), (13.42), (13.46) and (13.59)–(13.64)}\right\}. \tag{13.65}$$

13.5 Implementation

In this section we give details on our implementation of exact approaches based on each of the three formulations proposed for the (DREFLP) and (MREFLP). In Subsection 13.5.1 we discuss how we computationally solve the respective linear and semidefinite optimization problems to obtain lower bounds on the optimal solution. In Subsection 13.5.2 we describe heuristics for the semidefinite approach that yield feasible layouts and hence upper bounds to the optimal solution. For the MILP and ILP approaches, the standard solvers provide upper bounds. The combination of upper and lower bounds gives for each instance both a feasible solution and a proof of how far this solution is (at most) from the true optimum.

13.5.1 Computing the Lower Bounds

For the Amaral model (13.6)–(13.15) we included all the constraints directly and used Gurobi 5.6 [126] as ILP solver. We considered two different versions. Given d departments, we tested both the model allowing d columns per row (as suggested originally by Amaral [9]) and the one with a reduced number of positions according to Theorem 13.2.

For our new ILP model (13.17)–(13.27), tests with Gurobi showed that one should not add all equations at once, but should separate inequalities (13.19)–(13.25). We separate (13.19)–(13.25) dynamically in a branch-and-cut approach for linear 0-1 problems. These inequalities contain the inequalities (13.20)–(13.25) that are important for the success of the approach in [8]. We decided to not additionally separate (13.31), (13.32) because only handling (13.19)–(13.25) is already computationally challenging. Indeed, we do not add all violated inequalities in each step but rather restrict the number of cutting planes to 1000 in each iteration to keep the computational effort reasonable. The same separation procedure was applied in the multi-row case.

For the SDP approach, we solve our new SDP relaxation using a spectral bundle method [142, 144] in conjunction with primal cutting plane generation [141]. In general the solution of the relaxation is not integer but nevertheless we obtain a lower bound on the layout problem (see [26, 141] for the application of a spectral bundle method in the solution of the max-cut problem and the bisection problem).

One of the main advantages of the spectral bundle method is the ability to exploit the sparsity of the semidefinite relaxation [141]. In the objective function all the entries $x_{ij,kl}$ with $|\{i,j,k,l\}| = 4$ have value zero, and the support of equations (13.41)–(13.46) is also small. However (13.47) as well as the triangle inequalities of the metric polytope $\mathcal{M}$ have a larger support. In order to keep the small support consisting of the first row and column and the entries $x_{ij,kl}$ with $i, j, k, l \in [n], i \neq j, k \neq l, |\{i, j, k, l\}| \leq 3$, we restrict to inequalities (13.47) with $i \in \{l, m\}$, i. e., we only multiply (13.39) for $i \in [n]$ fixed with $x_{lm}, l, m \in [n], l \neq m$, if $i \in \{l, m\}$. Additionally we do not include the triangle inequalities and instead add the odd-cycle inequalities [32] (transformed to the -1/1-setting) on the small support of the objective function, where the coefficient matrix is interpreted as the adjacency matrix of a graph. In our experiments we used a separator by C. Helmberg that is an adapted variant of the one by M. Jünger. Note that if we worked with the full support (and thus on a complete graph) and exactly separated the triangle inequalities, then there is no need for an additional odd-cycle separator because all odd-cycles with length at least five are not chordless and are so implied by the other constraints [32].

As mentioned before, (13.43)–(13.44) for `(DREFLP)` and (13.60)–(13.61) for `(MREFLP)` are used to reduce the size of the semidefinite relaxations. In our implementation we add all the equations of ($\mathtt{SDP}_4$) and (13.65) respectively from the beginning (except the ones with large support mentioned above), and then iteratively include the odd-cycle inequalities. After 50 (null or descent) steps of the spectral bundle method we determine violated odd-cycle inequalities and restrict the separation to at most 100 additional constraints. In order to speed up the implementation we also delete constraints if they are not important anymore, see e.g. [26].

13.5.2 Details on the Heuristics Used

Gurobi provides upper bounds while solving the MILP and the ILP formulations. We describe here how we derive feasible layouts using SDP primal information. Let $\begin{pmatrix}1 & \tilde{x}\end{pmatrix}$ denote the first row of the SDP matrix Z. Hence $\tilde{x}$ gives the values of the x-variables in the relaxation. Given a partial solution consisting of k completely filled columns, $k \in \{0, \ldots, \frac{n}{m}\}$ (we arrange departments and spaces simultaneously), we determine and position the next column in a greedy fashion. First, we determine for each subset T of the remaining departments and spaces with $|T| = m$ the sum $\tau_T = \sum_{i,j \in T, i \neq j} \tilde{x}_{ij}$. A small value of τ_T indicates that the m elements of T should be arranged in the same column.[3] Hence we choose the smallest τ_T and arrange the according departments that we denote by C in the same column. Finally we decide on the position of the new column using again the information encoded in $\tilde{x}$.

More precisely let $N \subset [n]$ denote the set of all departments and spaces that have already been assigned and set $l = \frac{|N|}{m}$. The function $\alpha_{part} \colon N \to \left[\frac{|N|}{m}\right]$ gives an assignment of the elements of N to the $\frac{|N|}{m}$ columns. Now we calculate for the departments in C

$$\gamma_p = \sum_{\substack{i \in C,\ j \in N \\ \alpha_{part}(j) < p}} \tilde{x}_{ji} + \sum_{\substack{i \in C,\ j \in N \\ \alpha_{part}(j) \geq p}} \tilde{x}_{ij},$$

for all possible positions $p \in [l+1]$. Finally we determine $\hat{p} = \operatorname{argmax}_{p \in [l+1]} \gamma_p$, update α_{part} by

$$\alpha_{part}(i) \leftarrow \begin{cases} \hat{p}, & i \in C, \\ \alpha_{part}(i), & i \in N, \alpha_{part}(i) < \hat{p}, \\ \alpha_{part}(i) + 1, & i \in N, \hat{p} \leq \alpha_{part}(i), \end{cases}$$

and set $N \leftarrow N \cup C$.

After the layout is complete, we try to improve it using a 3-OPT heuristic that searches for advantageous exchanges of two or three departments in a greedy fashion. We also test if the solution can be improved by reallocation of any column or by exchanging two or three columns.

Unfortunately, the heuristic described above is only useful in practice if the number of rows m is small, say $m \leq 3$, because of significant memory requirements for larger m. For this reason, we propose a closely

[3] Note that if all departments in T lie in the same column, then $\sum_{i,j \in T, i \neq j} \tilde{x}_{ij} = -m(m-1)$.

related heuristic for larger m.

To reduce memory requirements we determine the departments that lie in the same row in an alternative way. We start with the pair $\{i,j\} \subset [n] \setminus N$ of the currently unassigned departments that minimizes $\tilde{x}_{ij} + \tilde{x}_{ji}$ and set $D = \{i,j\}$. Next we add the department $k \in [n] \setminus (N \cup D)$ to D that minimizes the sum $\sum_{l \in D}(\tilde{x}_{kl} + \tilde{x}_{lk})$. We iterate until $|D| = m$ and set $N \leftarrow N \cup D$. If every department has been assigned to a column, we finally determine the order of the columns in the same way as above. For $m = 2$ the two heuristics are exactly the same. Additionally we used the first heuristic for $m = 3$ and the second cheaper one for $m \in \{3,4,5\}$.

13.6 Computational Experiments

In this section we present some computational results for `DRFLP` instances from the literature as well as instances originally studied for the `SREFLP`. All experiments were conducted on an Intel Core i7 CPU 920 with 2.67 GHz and 12 GB RAM in single processor mode using openSUSE Linux 12.2.

We test the instances used for the `SREFLP` in [163], all instances proposed for the `(DREFLP)` by Amaral [9] (denoted by A-d-{edge probability}), where the pairwise weights w_{ij} are either zero or one because of the underlying graph problem, and the small instances constructed by Hungerländer and Anjos [167] (denoted by E-d-{edge probability}). All instances together with current best upper and lower bounds are available at `http://www.miguelanjos.com/flplib`.

In Tables 13.8–13.11 in the appendix we state the source, the density and the best `(SREFLP)` value of each instance as well as the best upper and lower bounds for the considered single- and multi-row problems. This also shows the effect on the optimal value (or on the value of the best known solution) of a growing number of rows. Also note that most benchmark instances from the layout literature are very dense.

In the following we compare the computation times and final gaps, calculated by the solution approaches presented above. We set the time limit to one hour and extend it to five hours for some larger instances. We calculate the gap between the best layout found[4] and the best lower bound obtained by $\frac{\text{upper bound}}{\text{lower bound}} - 1$, given in percent. We denote the original approach proposed by Amaral [9] using exactly d columns by `MILP I` and the approach to solve the same model using the reduced number of columns preserving at least one optimal solution according to Theorem 13.2 by `MILP II`. The results for our double- and multi-row ILP models can be found in the columns `ILP` and for our semidefinite programming model in the columns `SDP`.

The performance of the integer linear programming models changes significantly depending on the number of rows considered. The `ILP` is the best approach for small- to medium-sized instances with $d \leq 17$ in the case $m = 2$, see Tables 13.2–13.6. But for larger m, the solution times are much higher than in the case $m = 2$ and the obtained lower bounds are rather weak, see Tables 13.3–13.5. Often the ILP solver has problems finding a good upper bound. Hence even small instances with $d = 10$ could not be solved within the time limit of one hour. One explanation for this behavior could be that equations (13.17) for $m = 2$ are rather strong in comparison to inequalities (13.51)–(13.53) for $m \geq 3$. Furthermore, `ILP` and `SDP` both suffer from the fact that for larger d, the number of spaces (additional departments) needed grows with m, see also Table 13.1 for large d. Thus we present the results of the `ILP` only for the instances where it is a possible alternative to `SDP` ($m = 2,\ d \in \{16,\dots,20\}$ and $d \leq 15,\ m$ arbitrary).

Considering the results for `MILP I` and `MILP II`, we observe that reducing the number of columns in the model by Amaral [9] helps to reduce the computation times considerably. Nevertheless our computational results suggest that this improvement is not enough to create a competitive solution approach in general. For quite small instances with $d \leq 15$ one hour is sometimes not enough to solve all instances to optimality using `MILP II`. For this reason we do not present any further experiments for `MILP I` and `MILP II` with

[4]We always use only the layouts found by the heuristic of the respective approach, except for one case: For the instance "A-14-20" with $m = 2$ the SDP heuristic does not find the optimal layout after one hour, although the lower bound is already tight after less then 6 minutes.

$d \geq 16$. Comparing different numbers of rows, we observe that the bounds and the solution times of `MILP II` improve if m increases. One explanation for this behavior is that for larger m the number of columns needed is reduced, and hence the number of variables is also smaller than for $m = 2$. In summary `MILP II` is in most cases the worst approach for instances with $m = 2$ and $d \geq 13$, especially if the density of the instances is high, but it is also sometimes the best algorithm if m is large and d is small ($d \leq 13$).

The lower bounds derived with the `SDP` approach are often very strong, independent of the number of rows. To demonstrate this, we tested the `SDP` approach with a time limit of five hours for all instances with $d \geq 16$. Because of the long computation times and high gaps we do not enlarge the time limit for the `ILP` and the `MILP`. Table 13.7 shows that especially if $d \geq 30$ then major improvements in the gaps are achieved with the increased time limit: most gaps are below three percent. Looking at the upper bound, our SDP construction heuristics yields significantly better solutions (especially for large d) than the MILP and the ILP.

In summary, we conclude that MILP II is the overall best choice for instances with $d \leq 13$ and $m \geq 3$, the ILP approach is overall the best choice for $d \leq 17$ and $m = 2$, and for all other instances the SDP is the best choice. Moreover the SDP approach is well-suited for providing high-quality lower bounds for large-scale instances in reasonable computation time. Hence it seems promising to use the SDP relaxation within a branch-and-bound scheme. Because there is at present no commercial or other software that does this automatically for SDP relaxations, this possibility is a non-trivial computational project beyond the scope of this paper, and is thus left for future research.

13.7 Conclusions and Future Work

We considered the special case of equidistant row layout problems in which all departments have the same length. We showed that only spaces of unit length need to be used when modeling the problem, and we stated and proved exact expressions for the minimum number of spaces that need to be added so as to preserve at least one optimal solution. These results show that the multi-row equidistant layout can be modeled using only binary variables; this has a significant impact for a computational perspective. Using these results we proposed two new models for the equidistant problem, an ILP model and an SDP model. Our computational results show that the SDP approach dominates for medium- to large-sized instances and that it is well-suited for providing high-quality lower bounds for large-scale instances in reasonable computation time. Specifically for double-row instances, we attain global optimality for some instances with up to 25 departments, and achieve optimality gaps smaller than 1% for instances with up to 50 departments.

On the theoretical side, it remains an open question to extend the theoretical results to general double-row or multi-row problems. From the computational perspective, one direction for future research is the use of the SDP relaxation within a branch-and-bound scheme. While there is at present no commercial or other generally available software that does this automatically for SDP relaxations, this possibility is well worth exploring given the high-quality lower bounds provided by the SDP approach.

Acknowledgement

The authors thank A. R. S. Amaral for providing us with his instances and C. Helmberg for making us his bundle solver `ConicBundle` as well as his odd-cycle separator, an adapted version of the one by M. Jünger, available. The work of the first author was partially supported by a Discovery Grant from the National Science and Engineering Research Council (NSERC) of Canada. The second author was partially supported by the European Union and the Free State of Saxony funding the cluster eniPROD at Chemnitz University of Technology.

	$m=2$					$m=3$					$m=4$					$m=5$				
Instance	opt	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP
A-9-10	2	00:01	00:00	00:00	00:00	0	00:00	00:00	00:01	00:00	0	00:00	00:00	00:09	00:01	0	00:00	00:00	10:58	00:04
A-9-20	9	00:02	00:00	00:00	00:00	6	00:02	00:00	00:17	00:06	3	00:00	00:00	00:00	00:01	3	00:01	00:00	00:06	00:12
A-9-30	3	00:00	00:00	00:00	00:00	2	00:00	00:00	00:03	00:03	1	00:00	00:00	00:03	00:01	1	00:00	00:00	00:21	00:06
A-9-40	11	00:03	00:00	00:01	00:01	7	00:02	00:00	00:22	00:09	5	00:01	00:00	00:03	00:03	4	00:01	00:00	00:07	00:16
A-9-50	18	00:16	00:01	00:01	00:02	11	00:07	00:00	00:07	00:06	7	00:03	00:00	00:00	00:01	5	00:02	00:00	00:02	00:05
A-9-60	23	00:19	00:01	00:01	00:02	15	00:10	00:00	00:25	00:49	9	00:03	00:00	00:00	00:01	7	00:02	00:00	00:01	00:04
A-9-70	38	00:57	00:12	00:00	00:21	23	00:20	00:02	00:34	00:03	18	00:13	00:00	00:54	02:27	13	00:07	00:00	00:07	00:31
A-9-80	51	02:10	00:20	00:01	01:08	30	00:48	00:02	02:27	00:01	24	00:23	00:01	03:25	03:58	17	00:11	00:00	00:01	00:31
A-9-90	45	01:43	00:19	00:00	00:18	27	00:18	00:02	01:45	00:01	21	00:17	00:01	01:17	00:58	15	00:05	00:00	00:01	00:18
A-10-10	2	00:01	00:00	00:01	00:01	1	00:00	00:00	03:20	00:07	0	00:00	00:00	13:09	00:04	0	00:00	00:00	00:27	00:04
A-10-20	3	00:01	00:00	00:00	00:00	3	00:01	00:00	00:21	03:03	1	00:01	00:00	00:19	00:05	1	00:01	00:00	00:11	00:05
A-10-30	7	00:04	00:01	00:00	00:01	5	00:04	00:00	00:08	00:55	3	00:01	00:00	00:06	00:19	1	00:01	00:00	00:00	00:02
A-10-50	28	00:44	00:09	00:00	00:01	19	00:37	00:05	51:29	02:10	13	00:13	00:01	06:14	00:35	9	00:04	00:00	00:02	00:13
A-10-60	25	00:47	00:06	00:02	04:10	15	00:15	00:02	32:11	00:25	11	00:09	00:01	08:53	00:38	8	00:03	00:00	00:01	00:14
E-5-50	13	00:00	00:00	00:00	00:00	6	00:00	00:00	00:00	00:00	4	00:00	00:00	00:00	00:00	0	00:00	00:00	00:00	00:00
E-5-100	46	00:00	00:00	00:00	00:00	27	00:00	00:00	00:00	00:00	17	00:00	00:00	00:00	00:00	0	00:00	00:00	00:00	00:00
E-6-50	45	00:00	00:00	00:00	00:00	29	00:00	00:00	00:00	00:09	22	00:00	00:00	00:00	00:03	12	00:00	00:00	00:00	00:03
E-6-100	99	00:01	00:00	00:00	00:00	56	00:00	00:00	00:00	00:00	49	00:00	00:00	00:00	00:19	29	00:00	00:00	00:00	00:22
E-7-50	51	00:01	00:00	00:00	00:07	31	00:01	00:00	00:00	00:23	17	00:00	00:00	00:00	00:00	9	00:00	00:00	00:00	00:01
E-7-100	126	00:06	00:00	00:00	00:47	79	00:04	00:00	00:01	00:33	50	00:01	00:00	00:00	00:01	40	00:01	00:00	00:00	00:09
E-8-50	64	00:02	00:00	00:00	00:03	37	00:01	00:00	00:02	00:09	26	00:01	00:00	00:00	00:10	25	00:01	00:00	00:00	00:36
E-8-100	191	00:23	00:04	00:00	00:11	125	00:11	00:01	01:30	00:33	74	00:06	00:00	00:00	00:01	70	00:04	00:00	00:00	01:49
E-9-50	118	00:18	00:02	00:00	02:12	70	00:09	00:00	00:15	00:49	55	00:06	00:00	00:32	04:09	40	00:02	00:00	00:03	02:21
E-9-100	306	03:09	00:20	00:00	02:50	181	00:42	00:03	00:49	00:10	140	00:28	00:01	01:30	03:03	100	00:12	00:00	00:01	00:41
O-5	70	00:00	00:00	00:00	00:00	38	00:00	00:00	00:00	00:00	32	00:00	00:00	00:00	00:05	0	00:00	00:00	00:00	00:00
O-6	136	00:01	00:00	00:00	00:00	72	00:00	00:00	00:00	00:00	64	00:00	00:00	00:00	00:02	28	00:00	00:00	00:00	00:01
O-7	236	00:05	00:00	00:00	00:13	144	00:02	00:00	00:01	00:13	102	00:01	00:00	00:00	00:03	76	00:01	00:00	00:00	00:07
O-8	366	00:20	00:03	00:00	00:01	250	00:12	00:01	02:20	01:46	148	00:04	00:00	00:00	00:03	138	00:05	00:00	00:00	02:34
O-9	508	02:33	00:19	00:00	00:28	302	00:41	00:02	00:09	00:20	238	00:20	00:01	00:46	02:44	168	00:13	00:00	00:01	01:01
Y-6	630	00:01	00:00	00:00	00:02	350	00:00	00:00	00:00	00:00	315	00:00	00:00	00:00	02:07	193	00:00	00:00	00:00	02:18
Y-7	899	00:08	00:01	00:00	01:15	577	00:04	00:00	00:03	02:01	383	00:02	00:00	00:00	00:09	311	00:01	00:00	00:00	00:41
Y-8	1095	00:39	00:08	00:00	00:27	728	00:17	00:02	05:39	01:17	430	00:05	00:00	00:00	00:01	394	00:06	00:00	00:00	00:23
Y-9	1401	04:03	00:25	00:00	01:37	848	00:59	00:05	01:19	00:04	658	00:39	00:01	02:23	10:17	476	00:13	00:00	00:01	01:16

Table 13.2: Computation times (in mm:ss) for small instances with up to 9 departments and between 2 and 5 rows. All of these instances were solved to optimality within one hour of computation time by all four solution approaches.

		Gap (%)				Time (hh:mm:ss)					Gap (%)				Time(hh:mm:ss)			
Instance	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP
	$m = 2$									$m = 3$								
E-10-50	191	0.0	0.0	0.0	0.0	00:02:02	00:00:20	00:00:05	00:05:52	114	0.0	0.0	0.0	0.0	00:00:37	00:00:04	00:01:52	00:00:32
E-10-100	427	0.0	0.0	0.0	0.0	00:19:58	00:01:38	00:00:01	00:01:39	277	0.0	0.0	17.4	0.0	00:04:52	00:00:32	01:00:00	00:01:42
E-11-100	539	23.9	0.0	0.0	0.0	01:00:00	00:31:05	00:00:02	00:04:08	351	0.0	0.0	13.6	0.0	00:26:48	00:01:55	01:00:00	00:06:48
N-15	1064	153.3	68.4	0.0	0.0	01:00:00	01:00:00	00:13:16	00:08:37	668	76.7	16.8	32.3	0.0	01:00:00	01:00:00	01:00:00	00:11:24
O-10	670	0.0	0.0	0.0	0.0	00:07:24	00:01:21	00:00:00	00:00:20	450	0.0	0.0	16.0	0.0	00:02:02	00:00:23	01:00:00	00:02:12
O-15	2556	296.9	139.3	0.0	0.0	01:00:00	01:00:00	00:02:49	00:10:06	1660	165.2	38.6	12.2	0.0	01:00:00	01:00:00	01:00:00	00:53:27
S-12	2167	84.7	17.4	0.0	0.0	01:00:00	01:00:00	00:00:02	00:01:40	1404	48.9	0.0	19.4	0.1	01:00:00	00:37:31	01:00:00	01:00:00
S-13	2940	155.9	58.1	0.0	0.0	01:00:00	01:00:00	00:00:29	00:17:55	1938	92.1	12.7	19.4	0.6	01:00:00	01:00:00	01:00:00	01:00:00
S-14	3608	187.9	74.6	0.0	0.0	01:00:00	01:00:00	00:00:56	00:55:57	2408	159.2	43.4	23.2	0.2	01:00:00	01:00:00	01:00:00	01:00:00
S-15	4466	391.9	131.2	0.0	0.3	01:00:00	01:00:00	00:02:54	01:00:00	2883	192.1	44.8	15.6	0.0	01:00:00	01:00:00	01:00:00	00:51:32
Y-10	1697	0.0	0.0	0.0	0.0	00:27:02	00:03:08	00:00:00	00:00:57	1140	0.0	0.0	25.1	0.0	00:08:37	00:00:53	01:00:00	00:14:28
Y-11	2008	41.4	3.3	0.0	0.0	01:00:00	01:00:00	00:00:07	00:03:57	1314	3.7	0.0	17.0	0.0	01:00:00	00:02:59	01:00:00	00:09:21
Y-12	2342	92.6	22.2	0.0	0.0	01:00:00	01:00:00	00:00:02	00:01:33	1510	33.5	0.0	20.3	0.0	01:00:00	00:39:51	01:00:00	00:06:34
Y-13	2730	138.2	49.5	0.0	0.0	01:00:00	01:00:00	00:00:19	00:25:06	1798	89.7	13.6	17.0	0.0	01:00:00	01:00:00	01:00:00	00:50:55
Y-14	3164	196.3	113.1	0.0	0.0	01:00:00	01:00:00	00:02:22	00:15:15	2110	157.6	48.9	22.3	0.1	01:00:00	01:00:00	01:00:00	01:00:00
Y-15	3676	357.2	114.3	0.0	0.3	01:00:00	01:00:00	00:02:33	01:00:00	2357	204.1	45.0	13.6	0.0	01:00:00	01:00:00	01:00:00	00:06:18
	$m = 4$									$m = 5$								
E-10-50	89	0.0	0.0	9.9	0.0	00:00:20	00:00:01	01:00:00	00:03:35	59	0.0	0.0	0.0	0.0	00:00:09	00:00:00	00:00:00	00:00:32
E-10-100	209	0.0	0.0	17.4	0.0	00:01:55	00:00:11	01:00:00	00:13:20	133	0.0	0.0	0.0	0.0	00:00:26	00:00:00	00:00:00	00:00:13
E-11-100	256	0.0	0.0	15.8	0.4	00:06:45	00:00:22	01:00:00	01:00:00	191	0.0	0.0	0.0	0.0	00:02:16	00:00:02	00:10:32	00:13:18
N-15	500	38.9	0.0	41.2	0.6	01:00:00	00:39:50	01:00:00	01:00:00	382	0.0	0.0	24.0	1.3	00:41:45	00:02:20	01:00:00	01:00:00
O-10	334	0.0	0.0	14.8	0.0	00:01:03	00:00:06	01:00:00	00:03:17	222	0.0	0.0	0.0	0.0	00:00:20	00:00:00	00:00:00	00:02:02
O-15	1250	106.3	26.3	31.4	1.4	01:00:00	01:00:00	01:00:00	01:00:00	914	62.1	0.0	17.8	0.0	01:00:00	00:09:26	01:00:00	00:11:10
S-12	995	0.0	0.0	14.9	0.0	00:45:02	00:01:06	01:00:00	00:08:40	841	0.0	0.0	19.1	0.0	00:27:09	00:00:58	01:00:00	00:33:11
S-13	1413	48.7	0.0	30.2	0.0	01:00:00	00:54:59	01:00:00	00:42:47	1132	36.4	0.0	24.0	0.3	01:00:00	00:02:58	01:00:00	01:00:00
S-14	1794	114.1	41.1	40.0	2.1	01:00:00	01:00:00	01:00:00	01:00:00	1369	59.0	0.0	22.3	0.0	01:00:00	00:06:39	01:00:00	00:37:36
S-15	2175	149.7	37.3	40.7	1.2	01:00:00	01:00:00	01:00:00	01:00:00	1612	98.0	10.9	22.0	0.3	01:00:00	01:00:00	01:00:00	01:00:00
Y-10	845	0.0	0.0	21.8	0.0	00:02:49	00:00:16	01:00:00	00:19:12	530	0.0	0.0	0.0	0.0	00:00:32	00:00:01	00:00:00	00:01:35
Y-11	947	0.0	0.0	19.4	0.0	00:12:57	00:00:33	01:00:00	00:05:45	724	0.0	0.0	0.0	0.0	00:03:37	00:00:06	00:42:39	00:18:17
Y-12	1070	0.0	0.0	15.3	0.0	00:35:45	00:01:02	01:00:00	00:01:25	908	0.0	0.0	19.3	0.0	00:28:25	00:00:53	01:00:00	00:27:49
Y-13	1314	63.2	0.0	30.1	0.0	01:00:00	00:20:12	01:00:00	00:45:32	1048	24.3	0.0	23.7	0.6	01:00:00	00:02:50	01:00:00	01:00:00
Y-14	1574	113.0	41.2	41.3	1.8	01:00:00	01:00:00	01:00:00	01:00:00	1201	76.9	0.0	23.3	0.6	01:00:00	00:10:28	01:00:00	01:00:00
Y-15	1782	180.6	41.8	41.4	1.1	01:00:00	01:00:00	01:00:00	01:00:00	1322	101.5	13.2	22.5	0.4	01:00:00	01:00:00	01:00:00	01:00:00

Table 13.3: Computation times and gaps for instances with between 10 and 15 departments. Not all methods were able to solve these instances to optimality in the time limit of one hour.

		Gap (%)				Time (hh:mm:ss)					Gap (%)				Time (hh:mm:ss)			
Instance	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP
	$m = 2$									$m = 3$								
A-10-40	30	0.0	0.0	0.0	0.0	00:02:21	00:00:09	00:00:01	00:00:17	20	0.0	0.0	5.3	0.0	00:00:38	00:00:06	01:00:00	00:01:43
A-10-70	49	0.0	0.0	0.0	0.0	00:03:55	00:00:31	00:00:01	00:00:02	33	0.0	0.0	10.0	0.0	00:01:43	00:00:17	01:00:00	00:01:24
A-10-80	65	0.0	0.0	0.0	0.0	00:13:43	00:01:51	00:00:00	00:00:04	44	0.0	0.0	25.7	0.0	00:03:28	00:00:42	01:00:00	00:07:09
A-10-90	65	0.0	0.0	0.0	0.0	00:11:28	00:01:52	00:00:00	00:00:05	44	0.0	0.0	25.7	0.0	00:03:32	00:00:33	01:00:00	00:04:38
A-11-10	0	-	-	-	-	00:00:00	00:00:00	00:00:15	00:00:01	0	-	-	-	-	00:00:00	00:00:00	00:01:58	00:00:07
A-11-20	17	0.0	0.0	0.0	0.0	00:00:42	00:00:12	00:00:02	00:00:31	11	0.0	0.0	0.0	0.0	00:00:28	00:00:04	00:49:57	00:06:41
A-11-30	25	0.0	0.0	0.0	0.0	00:01:45	00:00:19	00:00:02	00:00:13	16	0.0	0.0	0.0	0.0	00:00:50	00:00:06	00:01:47	00:00:23
A-11-40	30	0.0	0.0	0.0	0.0	00:01:53	00:00:26	00:00:09	00:00:26	20	0.0	0.0	05.3	0.0	00:00:36	00:00:06	01:00:00	00:11:32
A-11-50	51	0.0	0.0	0.0	0.0	00:06:05	00:01:17	00:00:07	00:00:49	34	0.0	0.0	13.3	0.0	00:01:59	00:00:27	01:00:00	00:04:60
A-11-60	37	0.0	0.0	0.0	0.0	00:01:22	00:00:29	00:00:01	00:00:07	24	0.0	0.0	0.0	0.0	00:00:44	00:00:09	00:01:50	00:00:12
A-11-70	54	0.0	0.0	0.0	0.0	00:16:12	00:02:25	00:00:03	00:01:57	35	0.0	0.0	6.1	0.0	00:02:60	00:00:25	01:00:00	00:01:30
A-11-80	74	0.0	0.0	0.0	0.0	00:35:21	00:09:54	00:00:02	00:00:55	49	0.0	0.0	16.7	0.0	00:07:56	00:01:07	01:00:00	00:03:29
A-11-90	101	36.5	16.1	0.0	0.0	01:00:00	01:00:00	00:00:04	00:07:14	66	8.2	0.0	20.0	0.0	01:00:00	00:04:04	01:00:00	00:04:16
A-12-10	1	0.0	0.0	0.0	0.0	00:00:00	00:00:00	00:00:10	00:00:02	1	0.0	0.0	0.0	0.0	00:00:00	00:00:00	00:12:04	00:01:23
A-12-20	11	0.0	0.0	0.0	0.0	00:00:15	00:00:03	00:00:03	00:00:04	7	0.0	0.0	0.0	0.0	00:00:02	00:00:02	01:00:00	00:01:11
A-12-30	13	0.0	0.0	0.0	0.0	00:00:41	00:00:07	00:00:03	00:00:16	8	0.0	0.0	0.0	0.0	00:00:22	00:00:02	01:00:00	00:01:51
A-12-40	37	0.0	0.0	0.0	0.0	00:35:05	00:00:54	00:00:03	00:00:12	24	0.0	0.0	4.3	0.0	00:03:08	00:00:29	01:00:00	00:01:43
A-12-50	43	0.0	0.0	0.0	0.0	00:07:19	00:01:08	00:00:04	00:00:38	27	0.0	0.0	0.0	0.0	00:01:41	00:00:30	01:00:00	00:00:43
A-12-60	53	0.0	0.0	0.0	0.0	00:51:40	00:02:11	00:00:02	00:00:49	33	0.0	0.0	3.1	0.0	00:04:25	00:00:46	01:00:00	00:00:25
A-12-70	77	24.2	0.0	0.0	0.0	01:00:00	00:27:12	00:00:04	00:00:57	49	0.0	0.0	11.4	0.0	00:40:15	00:01:43	01:00:00	00:00:60
A-12-80	102	61.9	9.7	0.0	0.0	01:00:00	01:00:00	00:00:01	00:00:17	65	10.2	0.0	16.1	0.0	01:00:00	00:05:31	01:00:00	00:00:12
A-12-90	108	52.1	24.1	0.0	0.0	01:00:00	01:00:00	00:00:04	00:00:26	70	29.6	0.0	22.8	0.0	01:00:00	00:13:01	01:00:00	00:01:04
A-13-10	2	0.0	0.0	0.0	0.0	00:00:01	00:00:00	00:00:17	00:00:05	1	0.0	0.0	0.0	0.0	00:00:00	00:00:00	00:50:47	00:00:38
A-13-20	24	0.0	0.0	0.0	0.0	00:05:16	00:00:29	00:00:38	00:03:22	15	0.0	0.0	0.0	0.0	00:01:01	00:00:07	01:00:00	00:03:03
A-13-30	38	0.0	0.0	0.0	0.0	00:13:42	00:01:30	00:00:53	00:01:16	25	0.0	0.0	8.7	0.0	00:04:51	00:00:33	01:00:00	00:03:02
A-13-40	42	31.2	0.0	0.0	0.0	01:00:00	00:06:38	00:00:07	00:02:03	27	0.0	0.0	3.8	0.0	00:10:59	00:00:33	01:00:00	00:02:57
A-13-50	68	44.7	0.0	0.0	0.0	01:00:00	00:18:59	00:00:42	00:04:54	44	12.8	0.0	7.3	0.0	01:00:00	00:02:48	01:00:00	00:03:52
A-13-60	70	29.6	0.0	0.0	0.0	01:00:00	00:17:23	00:00:11	00:02:05	46	0.0	0.0	4.5	0.0	00:44:06	00:02:14	01:00:00	00:06:27
A-13-70	105	72.1	12.9	0.0	0.0	01:00:00	01:00:00	00:06:56	00:04:55	69	38.0	0.0	16.9	0.0	01:00:00	00:12:26	01:00:00	00:11:16
A-13-80	138	126.2	43.8	0.0	0.0	01:00:00	01:00:00	00:00:03	00:04:08	90	76.5	0.0	15.4	0.0	01:00:00	00:51:03	01:00:00	00:03:32
A-13-90	153	146.8	56.1	0.0	0.0	01:00:00	01:00:00	00:00:09	00:10:42	101	110.4	11.0	20.2	0.0	01:00:00	01:00:00	01:00:00	00:53:40
A-14-10	4	0.0	0.0	0.0	0.0	00:00:02	00:00:01	00:00:31	00:00:18	3	0.0	0.0	0.0	0.0	00:00:02	00:00:01	01:00:00	00:06:15
A-14-20	24	0.0	0.0	0.0	0.0	00:03:45	00:00:36	00:19:20	00:05:44	16	0.0	0.0	14.3	6.7	00:01:21	00:00:10	01:00:00	01:00:00
A-14-30	36	0.0	0.0	0.0	0.0	00:40:52	00:07:46	00:02:03	00:03:39	24	0.0	0.0	33.3	0.0	00:12:18	00:02:01	01:00:00	00:19:60
A-14-40	43	26.5	0.0	0.0	0.0	01:00:00	00:15:18	00:01:41	00:02:40	28	0.0	0.0	33.3	0.0	00:21:14	00:01:30	01:00:00	00:08:31
A-14-50	94	118.6	46.9	0.0	0.0	01:00:00	01:00:00	00:19:16	00:05:02	63	70.3	0.0	18.9	1.6	01:00:00	00:21:16	01:00:00	01:00:00
A-14-60	99	86.8	39.4	0.0	0.0	01:00:00	01:00:00	00:10:18	00:01:12	65	54.8	0.0	12.1	0.0	01:00:00	00:17:05	01:00:00	00:03:19
A-14-70	138	133.9	86.5	0.0	0.0	01:00:00	01:00:00	00:08:36	00:08:57	92	114.0	22.7	22.7	0.0	01:00:00	01:00:00	01:00:00	00:35:22
A-14-80	167	153.0	74.0	0.0	0.0	01:00:00	01:00:00	00:00:21	00:04:32	111	122.0	22.0	24.7	0.0	01:00:00	01:00:00	01:00:00	00:13:02
A-14-90	187	179.1	96.8	0.0	0.0	01:00:00	01:00:00	00:14:15	00:03:17	125	160.4	42.0	31.6	0.8	01:00:00	01:00:00	01:00:00	01:00:00

Table 13.4: Computation times and gaps for small instances from Amaral [9] with $d \in \{10, 11, 12, 13, 14\}, m \in \{2, 3\}$. Not all methods were able to solve these instances to optimality in the time limit of one hour.

		Gap (%)				Time (hh:mm:ss)					Gap (%)				Time (hh:mm:ss)			
Instance	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP	opt	MILP I	MILP II	ILP	SDP	MILP I	MILP II	ILP	SDP
	$m = 4$									$m = 5$								
A-10-40	13	0.0	0.0	0.0	0.0	00:00:12	00:00:01	00:04:16	00:00:07	10	0.0	0.0	0.0	0.0	00:00:10	00:00:00	00:00:04	00:00:12
A-10-70	24	0.0	0.0	9.1	0.0	00:00:35	00:00:06	01:00:00	00:00:59	16	0.0	0.0	0.0	0.0	00:00:15	00:00:00	00:00:01	00:00:07
A-10-80	32	0.0	0.0	14.3	0.0	00:01:33	00:00:08	01:00:00	00:03:03	21	0.0	0.0	0.0	0.0	00:00:27	00:00:01	00:00:00	00:00:06
A-10-90	32	0.0	0.0	14.3	0.0	00:01:22	00:00:05	01:00:00	00:02:12	21	0.0	0.0	0.0	0.0	00:00:25	00:00:00	00:00:01	00:00:04
A-11-10	0	-	-	-	-	00:00:00	00:00:00	00:37:50	00:00:08	0	-	-	-	-	00:00:00	00:00:00	00:06:01	00:00:04
A-11-20	8	0.0	0.0	14.3	0.0	00:00:12	00:00:00	01:00:00	00:01:09	5	0.0	0.0	0.0	0.0	00:00:06	00:00:00	00:00:01	00:00:04
A-11-30	12	0.0	0.0	0.0	0.0	00:00:27	00:00:01	01:00:00	00:00:49	8	0.0	0.0	0.0	0.0	00:00:13	00:00:00	00:00:05	00:00:06
A-11-40	14	0.0	0.0	0.0	0.0	00:00:22	00:00:01	01:00:00	00:00:47	11	0.0	0.0	0.0	0.0	00:00:16	00:00:00	00:07:14	00:01:38
A-11-50	24	0.0	0.0	9.1	0.0	00:00:48	00:00:05	01:00:00	00:01:11	18	0.0	0.0	0.0	0.0	00:00:31	00:00:00	00:05:39	00:03:10
A-11-60	18	0.0	0.0	0.0	0.0	00:00:26	00:00:01	00:05:02	00:00:37	14	0.0	0.0	0.0	0.0	00:00:16	00:00:00	00:00:55	00:00:52
A-11-70	26	0.0	0.0	13.0	0.0	00:01:38	00:00:06	01:00:00	00:04:40	19	0.0	0.0	0.0	0.0	00:00:36	00:00:01	00:03:42	00:01:13
A-11-80	36	0.0	0.0	12.5	0.0	00:05:23	00:00:16	01:00:00	00:03:54	27	0.0	0.0	0.0	0.0	00:01:27	00:00:02	00:20:52	00:03:11
A-11-90	48	0.0	0.0	20.0	0.0	00:13:44	00:00:40	01:00:00	00:03:46	37	0.0	0.0	8.8	0.0	00:04:15	00:00:06	01:00:00	00:10:08
A-12-10	0	-	-	-	-	00:00:00	00:00:00	00:03:54	00:00:06	0	-	-	-	-	00:00:00	00:00:00	01:00:00	00:00:27
A-12-20	5	0.0	0.0	0.0	0.0	00:00:04	00:00:00	00:00:18	00:00:27	4	0.0	0.0	0.0	0.0	00:00:02	00:00:00	00:01:24	00:02:03
A-12-30	5	0.0	0.0	0.0	0.0	00:00:07	00:00:00	00:02:48	00:00:21	4	0.0	0.0	0.0	0.0	00:00:08	00:00:00	01:00:00	00:01:25
A-12-40	16	0.0	0.0	0.0	0.0	00:01:07	00:00:01	00:06:57	00:00:09	15	0.0	0.0	15.4	0.0	00:00:54	00:00:02	01:00:00	00:15:13
A-12-50	20	0.0	0.0	0.0	0.0	00:00:45	00:00:02	00:25:15	00:01:21	17	0.0	0.0	13.3	0.0	00:00:34	00:00:02	01:00:00	00:08:46
A-12-60	24	0.0	0.0	0.0	0.0	00:01:08	00:00:03	01:00:00	00:00:35	21	0.0	0.0	16.7	0.0	00:01:34	00:00:07	01:00:00	00:17:52
A-12-70	34	0.0	0.0	0.0	0.0	00:02:49	00:00:08	01:00:00	00:00:06	30	0.0	0.0	15.4	0.0	00:03:04	00:00:12	01:00:00	00:26:15
A-12-80	47	0.0	0.0	9.3	0.0	00:15:20	00:00:26	01:00:00	00:00:13	40	0.0	0.0	14.3	0.0	00:12:02	00:00:28	01:00:00	00:15:27
A-12-90	50	0.0	0.0	13.6	0.0	00:16:58	00:00:38	01:00:00	00:00:22	42	0.0	0.0	13.5	0.0	00:10:45	00:00:29	01:00:00	00:20:46
A-13-10	1	0.0	0.0	0.0	0.0	00:00:00	00:00:00	01:00:00	00:01:28	1	0.0	0.0	0.0	0.0	00:00:00	00:00:00	01:00:00	00:01:36
A-13-20	11	0.0	0.0	10.0	0.0	00:00:35	00:00:02	01:00:00	00:04:13	9	0.0	0.0	12.5	0.0	00:00:22	00:00:01	01:00:00	00:12:14
A-13-30	19	0.0	0.0	18.8	0.0	00:01:58	00:00:08	01:00:00	00:19:49	14	0.0	0.0	7.7	0.0	00:00:45	00:00:02	01:00:00	00:01:40
A-13-40	20	0.0	0.0	17.6	0.0	00:02:21	00:00:12	01:00:00	00:07:48	16	0.0	0.0	14.3	0.0	00:01:56	00:00:02	01:00:00	00:05:03
A-13-50	32	0.0	0.0	23.1	0.0	00:10:45	00:00:49	01:00:00	00:05:22	26	0.0	0.0	23.8	0.0	00:04:03	00:00:13	01:00:00	00:09:35
A-13-60	33	0.0	0.0	13.8	0.0	00:12:30	00:00:43	01:00:00	00:04:53	26	0.0	0.0	13.0	0.0	00:03:16	00:00:11	01:00:00	00:07:40
A-13-70	50	0.0	0.0	25.0	0.0	00:50:12	00:01:49	01:00:00	00:12:21	41	0.0	0.0	20.6	2.5	00:31:39	00:00:29	01:00:00	01:00:00
A-13-80	66	46.7	0.0	22.2	0.0	01:00:00	00:05:15	01:00:00	00:16:37	53	15.2	0.0	15.2	0.0	01:00:00	00:01:06	01:00:00	00:11:55
A-13-90	74	39.6	0.0	27.6	0.0	01:00:00	00:14:12	01:00:00	00:48:06	58	18.4	0.0	18.4	0.0	01:00:00	00:01:50	01:00:00	00:09:02
A-14-10	1	0.0	0.0	0.0	0.0	00:00:01	00:00:00	01:00:00	00:02:57	1	0.0	0.0	0.0	0.0	00:00:01	00:00:00	00:05:53	00:00:45
A-14-20	11	0.0	0.0	10.0	0.0	00:01:21	00:00:07	01:00:00	00:16:15	8	0.0	0.0	0.0	0.0	00:00:23	00:00:00	01:00:00	00:01:28
A-14-30	18	0.0	0.0	28.6	5.9	00:03:34	00:00:40	01:00:00	01:00:00	13	0.0	0.0	8.3	0.0	00:01:56	00:00:05	01:00:00	00:03:06
A-14-40	21	0.0	0.0	31.2	0.0	00:07:14	00:01:17	01:00:00	00:53:19	16	0.0	0.0	14.3	0.0	00:02:36	00:00:02	01:00:00	00:11:06
A-14-50	47	42.4	0.0	46.9	2.2	01:00:00	00:07:14	01:00:00	01:00:00	35	0.0	0.0	20.7	0.0	00:12:46	00:00:20	01:00:00	00:11:11
A-14-60	49	19.5	0.0	32.4	0.0	01:00:00	00:05:46	01:00:00	00:43:05	37	0.0	0.0	12.1	0.0	00:34:09	00:00:24	01:00:00	00:03:30
A-14-70	68	54.5	0.0	38.8	1.5	01:00:00	00:41:36	01:00:00	01:00:00	52	15.6	0.0	15.6	0.0	01:00:00	00:01:22	01:00:00	00:06:24
A-14-80	83	62.7	10.7	38.3	1.2	01:00:00	01:00:00	01:00:00	01:00:00	64	28.0	0.0	18.5	0.0	01:00:00	00:03:41	01:00:00	00:09:16
A-14-90	93	93.8	31.0	38.8	1.1	01:00:00	01:00:00	01:00:00	01:00:00	71	65.1	0.0	18.3	0.0	01:00:00	00:07:01	01:00:00	00:14:20

Table 13.5: Computation times and gaps for small instances from Amaral [9] with $d \in \{10, 11, 12, 13, 14\}, m \in \{4, 5\}$. Not all methods were able to solve these instances to optimality in the time limit of one hour.

	$m = 2$							$m = 3$					$m = 4$					$m = 5$				
		ILP 1h		SDP 1h		SDP 5h			SDP 1h		SDP 5h			SDP 1h		SDP 5h			SDP 1h		SDP 5h	
instance	ub	gap	time	gap	time	gap	time	ub	gap	time	gap	time	ub	gap	time	gap	time	ub	gap	time	gap	time
A-20-10	12	20.0	01:00:00	9.1	01:00:00	9.1	05:00:00	7	16.7	01:00:00	16.7	05:00:00	4	33.3	01:00:00	0.0	03:50:42	3	0.0	00:26:33		
A-20-20	73	87.2	01:00:00	0.0	00:33:30			49	4.3	01:00:00	2.1	05:00:00	34	3.0	01:00:00	0.0	01:21:34	27	3.8	01:00:00	0.0	01:36:52
A-20-30	111	68.2	01:00:00	0.0	00:20:13			74	1.4	01:00:00	1.4	05:00:00	54	3.8	01:00:00	1.9	05:00:00	42	2.4	01:00:00	0.0	02:28:39
A-20-40	149	50.5	01:00:00	0.0	00:35:02			98	1.0	01:00:00	0.0	01:13:53	73	2.8	01:00:00	1.4	05:00:00	58	3.6	01:00:00	1.8	05:00:00
A-20-50	249	11.2	01:00:00	0.0	00:52:37			166	1.2	01:00:00	0.6	05:00:00	122	1.7	01:00:00	0.8	05:00:00	96	2.1	01:00:00	1.1	05:00:00
A-20-60	345	28.7	01:00:00	0.3	01:00:00	0.0	01:21:58	229	0.9	01:00:00	0.4	05:00:00	167	0.6	01:00:00	0.0	01:51:21	132	0.8	01:00:00	0.8	05:00:00
A-20-70	385	25.0	01:00:00	0.0	00:20:58			258	1.6	01:00:00	0.4	05:00:00	187	0.5	01:00:00	0.5	05:00:00	146	0.7	01:00:00	0.0	01:20:43
A-20-80	434	17.0	01:00:00	0.2	01:00:00	0.0	01:17:34	290	1.4	01:00:00	0.7	05:00:00	211	1.0	01:00:00	0.5	05:00:00	165	0.6	01:00:00	0.0	03:47:42
A-20-90	521	6.8	01:00:00	0.2	01:00:00	0.2	02:12:30	347	0.9	01:00:00	0.0	03:51:36	252	0.0	00:41:05			197	0.0	00:23:50		
N-16a	1496	0.0	00:21:48	0.0	00:09:33			1002	0.9	01:00:00	0.0	02:15:36	706	0.0	00:16:06			584	2.3	01:00:00	0.0	03:11:45
N-16b	1168	0.0	00:06:02	0.0	00:06:15			792	1.7	01:00:00	0.8	05:00:00	570	1.6	01:00:00	0.0	03:25:47	462	2.7	01:00:00	0.9	05:00:00
N-17	1678	0.0	00:35:31	0.0	00:27:20			1114	1.3	01:00:00	0.0	02:23:52	808	0.0	00:28:04			662	2.8	01:00:00	0.5	05:00:00
N-18	1970	0.0	00:34:44	0.0	00:45:05			1292	0.6	01:00:00	0.4	05:00:00	972	1.7	01:00:00	0.0	01:19:23	772	2.3	01:00:00	0.0	03:44:02
N-20	2782	38.8	01:00:00	0.0	00:31:52			1856	1.6	01:00:00	0.5	05:00:00	1360	2.0	01:00:00	1.0	05:00:00	1068	1.9	01:00:00	0.7	05:00:00
O-20	6414	20.9	01:00:00	0.0	00:54:27			4284	1.3	01:00:00	0.4	05:00:00	3118	0.8	01:00:00	0.1	05:00:00	2444	1.0	01:00:00	0.5	05:00:00
S-16	5446	0.0	00:20:56	0.0	00:12:41			3638	1.0	01:00:00	0.0	04:55:13	2600	0.0	00:48:46			2094	1.5	01:00:00	0.0	03:40:35
S-17	6577	0.0	00:35:02	0.3	01:00:00	0.0	01:49:13	4354	0.6	01:00:00	0.3	05:00:00	3225	2.0	01:00:00	0.9	05:00:00	2577	1.7	01:00:00	0.4	05:00:00
S-18	7788	0.2	01:00:00	0.1	01:00:00	0.0	05:00:00	5110	0.2	01:00:00	0.1	05:00:00	3892	1.9	01:00:00	1.0	05:00:00	3083	2.8	01:00:00	1.1	05:00:00
S-19	9343	1.1	01:00:00	0.5	01:00:00	0.3	05:00:00	6190	1.2	01:00:00	0.7	05:00:00	4599	2.2	01:00:00	1.2	05:00:00	3614	1.4	01:00:00	0.6	05:00:00
S-20	10841	8.5	01:00:00	0.1	01:00:00	0.0	02:58:26	7227	1.2	01:00:00	0.6	05:00:00	5260	0.4	01:00:00	0.2	05:00:00	4105	0.0	01:00:00	0.0	01:03:54
Y-20	6046	9.3	01:00:00	0.0	01:00:00	0.0	01:34:03	4033	0.9	01:00:00	0.4	05:00:00	2934	0.4	01:00:00	0.1	05:00:00	2282	0.0	00:17:29		

Table 13.6: Computation times (in hh:mm:ss) and gaps (in percent) for medium-sized instances solved with SDP and with ILP in the case $m = 2$.

	$m = 2$					$m = 3$					$m = 4$					$m = 5$				
		SDP 1h		SDP 5h			SDP 1h		SDP 5h			SDP 1h		SDP 5h			SDP 1h		SDP 5h	
instance	best ub	gap	time	gap	time	best ub	gap	time	gap	time	best ub	gap	time	gap	time	best ub	gap	time	gap	time
A-25-10	41	2.5	01:00:00	0.0	01:05:06	27	8.0	01:00:00	3.8	05:00:00	20	25.0	01:00:00	11.1	05:00:00	15	7.1	01:00:00	0.0	03:16:36
A-25-20	110	0.9	01:00:00	0.0	03:08:29	74	7.2	01:00:00	4.2	05:00:00	55	14.6	01:00:00	5.8	05:00:00	40	2.6	01:00:00	0.0	02:10:41
A-25-30	222	0.5	01:00:00	0.0	03:01:22	146	1.4	01:00:00	0.0	04:01:29	110	5.8	01:00:00	1.9	04:01:29	87	4.8	01:00:00	1.2	05:00:00
A-25-40	400	1.0	01:00:00	0.3	05:00:00	265	1.9	01:00:00	0.8	05:00:00	198	4.2	01:00:00	2.1	05:00:00	156	4.7	01:00:00	2.6	05:00:00
A-25-50	511	0.4	01:00:00	0.4	05:00:00	340	2.1	01:00:00	1.2	05:00:00	254	4.1	01:00:00	2.4	05:00:00	196	1.6	01:00:00	0.5	05:00:00
A-25-60	549	0.9	01:00:00	0.4	05:00:00	364	1.7	01:00:00	1.1	05:00:00	271	3.0	01:00:00	1.9	05:00:00	212	2.4	01:00:00	1.4	05:00:00
A-25-70	660	0.9	01:00:00	0.3	05:00:00	438	1.6	01:00:00	0.7	05:00:00	325	2.8	01:00:00	1.2	05:00:00	255	2.0	01:00:00	1.2	05:00:00
A-25-80	910	0.4	01:00:00	0.3	05:00:00	604	1.0	01:00:00	0.7	05:00:00	450	2.0	01:00:00	1.4	05:00:00	350	0.6	01:00:00	0.3	05:00:00
A-25-90	1084	0.4	01:00:00	0.4	05:00:00	721	1.1	01:00:00	0.7	05:00:00	537	1.7	01:00:00	1.3	05:00:00	417	0.2	01:00:00	0.0	01:54:59
N-21	2512	0.5	01:00:00	0.0	01:04:30	1664	2.0	01:00:00	1.3	05:00:00	1248	3.7	01:00:00	1.9	05:00:00	972	3.3	01:00:00	0.7	05:00:00
N-22	3064	0.9	01:00:00	0.2	05:00:00	2034	2.8	01:00:00	0.8	05:00:00	1510	2.2	01:00:00	0.9	05:00:00	1188	2.8	01:00:00	0.0	04:08:02
N-24	4120	2.4	01:00:00	0.6	05:00:00	2712	1.5	01:00:00	0.6	05:00:00	2010	3.4	01:00:00	1.3	05:00:00	1624	4.3	01:00:00	2.1	05:00:00
N-25	4604	1.9	01:00:00	0.0	04:41:49	3062	1.8	01:00:00	0.7	05:00:00	2286	4.4	01:00:00	2.0	05:00:00	1796	3.2	01:00:00	1.3	05:00:00
N-30	8230	1.6	01:00:00	0.4	05:00:00	5442	2.4	01:00:00	0.7	05:00:00	4086	4.3	01:00:00	2.1	05:00:00	3232	4.5	01:00:00	2.1	05:00:00
S-21	12431	0.6	01:00:00	0.2	05:00:00	8144	0.2	01:00:00	0.0	03:26:10	6136	2.1	01:00:00	1.6	05:00:00	4849	2.7	01:00:00	1.8	05:00:00
S-22	14208	0.1	01:00:00	0.0	05:00:00	9484	1.4	01:00:00	0.8	05:00:00	7082	2.1	01:00:00	1.2	05:00:00	5623	2.9	01:00:00	1.3	05:00:00
S-23	16521	0.8	01:00:00	0.4	05:00:00	10974	1.3	01:00:00	0.7	05:00:00	8159	1.5	01:00:00	0.9	05:00:00	6523	2.7	01:00:00	1.1	05:00:00
S-24	18658	0.3	01:00:00	0.1	05:00:00	12349	0.5	01:00:00	0.2	05:00:00	9147	0.8	01:00:00	0.3	05:00:00	7342	1.9	01:00:00	1.1	05:00:00
S-25	21172	0.8	01:00:00	0.4	05:00:00	14070	1.2	01:00:00	0.9	05:00:00	10487	2.2	01:00:00	1.5	05:00:00	8149	0.4	01:00:00	0.0	05:00:00
Y-25	10170	0.8	01:00:00	0.3	05:00:00	6761	1.3	01:00:00	0.8	05:00:00	5049	2.5	01:00:00	1.7	05:00:00	3930	1.0	01:00:00	0.5	05:00:00
Y-30	13790	0.5	01:00:00	0.1	05:00:00	9133	1.0	01:00:00	0.3	05:00:00	6889	2.8	01:00:00	1.8	05:00:00	5386	1.5	01:00:00	0.6	05:00:00
Y-35	19087	0.5	01:00:00	0.3	05:00:00	12705	1.7	01:00:00	0.5	05:00:00	9492	3.6	01:00:00	1.4	05:00:00	7504	2.1	01:00:00	0.8	05:00:00
Y-40	23749	0.9	01:00:00	0.4	05:00:00	15825	3.5	01:00:00	1.3	05:00:00	11785	4.9	01:00:00	1.5	05:00:00	9381	5.0	01:00:00	1.4	05:00:00
Y-45	31442	1.6	01:00:00	0.7	05:00:00	20896	4.5	01:00:00	1.5	05:00:00	15663	9.6	01:00:00	2.5	05:00:00	12442	7.8	01:00:00	2.1	05:00:00
Y-50	41517	3.2	01:00:00	0.9	05:00:00	27674	8.1	01:00:00	2.0	05:00:00	20809	15.7	01:00:00	4.5	05:00:00	16475	10.3	01:00:00	2.8	05:00:00
Y-60	55996	11.7	01:00:00	1.9	05:00:00	37279	18.3	01:00:00	4.1	05:00:00	27913	21.7	01:00:00	6.6	05:00:00	22370	24.8	01:00:00	6.5	05:00:00

Table 13.7: Computation times (in hh:mm:ss) and gaps (in percent) for large instances solved with SDP.

Appendix

Details on Benchmark Instances Used

The following tables state the source, the density and the optimal solution or best bounds for our benchmark instances with $m \in [5]$ rows.

Instance	Source	Size (n)	Density (%)	Optimal solution $m=1$	$m=2$	$m=3$	$m=4$	$m=5$
E-5-50	[167]	5	50	30	13	6	4	0
E-5-100		5	100	95	46	27	17	0
E-6-50		6	50	100	45	29	22	12
E-6-100		6	100	216	99	56	49	29
E-7-50		7	50	106	51	31	17	9
E-7-100		7	100	252	126	79	50	40
E-8-50		8	50	136	64	37	26	25
E-8-100		8	100	397	191	125	74	70
E-9-50		9	50	240	118	70	55	40
E-9-100		9	100	618	306	181	140	100
E-10-50		10	50	387	191	114	89	59
E-10-100		10	100	873	427	277	209	133
E-11-100		11	50	1085	539	351	256	191
N-15	[240]	15	71	2186	1064	668	500	382
O-5	[241]	5	100	150	70	38	32	0
O-6		6	100	292	136	72	64	28
O-7		7	100	472	236	144	102	76
O-8		8	100	784	366	250	148	138
O-9		9	100	1032	508	302	238	168
O-10		10	100	1402	670	450	334	222
O-15		15	100	5134	2556	1660	1250	914
S-12	[270]	12	100	4431	2167	1404	995	841
S-13		13	100	5897	2940	1938	1413	1132
S-14		14	100	7316	3608	2408	1794	1369
S-15		15	100	8942	4466	2883	2175	1612
Y-6	[308]	6	100	1372	630	350	315	193
Y-7		7	100	1801	899	577	383	311
Y-8		8	100	2302	1095	728	430	394
Y-9		9	100	2808	1401	848	658	476
Y-10		10	100	3508	1697	1140	845	530
Y-11		11	100	4022	2008	1314	947	724
Y-12		12	100	4793	2342	1510	1070	908
Y-13		13	100	5471	2730	1798	1314	1048
Y-14		14	100	6445	3164	2110	1574	1201
Y-15		15	100	7359	3676	2357	1782	1322

Table 13.8: Characteristics and optimal results for small instances with 5 to 15 departments.

Instance	Source	Size (d)	Density (%)	Optimal solution				
				$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
N-16a	[240]	16	78	3050	1494	1002	706	584
N-16b		16	70	2400	1168	[786,792]	570	462
N-17		17	74	3388	1678	1114	808	662
N-18		18	74	3986	1970	[1287,1292]	972	772
N-20		20	74	5642	2782	[1847,1856]	[1347,1360]	[1061,1068]
N-21		21	65	5084	2512	[1643,1664]	[1225,1248]	[965,972]
N-22		22	66	6184	[3059,3064]	[2018,2034]	[1497,1510]	1188
N-24		24	67	8270	[4097,4120]	[2696,2712]	[1985,2010]	[1590,1624]
N-25		25	67	9236	4604	[3040,3062]	[2242,2286]	[1773,1796]
N-30		30	67	16494	[8194,8230]	[5406,5442]	[4001,4086]	[3165,3232]
O-20	[241]	20	100	12924	6414	[4265,4284]	[3115,3118]	[2431,2444]
S-16	[270]	16	100	11019	5446	3638	2600	2094
S-17		17	100	13172	6577	[4343,4354]	[3196,3225]	[2568,2577]
S-18		18	100	15699	[7787,7788]	[5107,5110]	[3854,3892]	[3048,3083]
S-19		19	100	18700	[9311,9343]	[6149,6190]	[4545,4599]	[3593,3614]
S-20		20	100	21825	[10837,10841]	[7186,7227]	[5248,5260]	4105
S-21		21	100	24891	[12406,12431]	8144	[6042,6136]	[4762,4849]
S-22		22	100	28607	[14202,14208]	[9412,9484]	[6997,7082]	[5549,5623]
S-23		23	100	33046	[16448,16521]	[10900,10974]	[8086,8159]	[6450,6523]
S-24		24	100	37498	[18646,18658]	[12325,12349]	[9116,9147]	[7261,7342]
S-25		25	100	42349	[21091,21172]	[13951,14070]	[10332,10487]	[8148,8149]
Y-20	[308]	20	100	12185	6046	[4018,4033]	[2930,2934]	2282
Y-25		25	100	20357	[10139,10170]	[6709,6761]	[4967,5049]	[3912,3930]
Y-30		30	100	27673	[13771,13790]	[9107,9133]	[6764,6889]	[5355,5386]
Y-35		35	100	38194	[19025,19087]	[12636,12705]	[9357,9492]	[7447,7504]
Y-40		40	100	47561	[23648,23749]	[15616,15825]	[11615,11785]	[9253,9381]
Y-45		45	99	[62849,62904]	[31237,31442]	[20592,20896]	[15283,15663]	[12182,12442]
Y-50		50	99	[83086,83127]	[41156,41517]	[27129,27674]	[19915,20809]	[16032,16475]
Y-60		60	97	[111884,112126]	[54925,55996]	[35803,37279]	[26180,27913]	[21007,22370]

Table 13.9: Characteristics and optimal results for medium-sized to large instances with 16 to 60 departments.

Instance	Size (d)	Density (%)	Optimal solution				
			$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
A-9-10	9	14	5	2	0	0	0
A-9-20	9	31	19	9	6	3	3
A-9-30	9	17	7	3	2	1	1
A-9-40	9	33	23	11	7	5	4
A-9-50	9	47	36	18	11	7	5
A-9-60	9	56	48	23	15	9	7
A-9-70	9	78	76	38	23	18	13
A-9-80	9	92	102	51	30	24	17
A-9-90	9	86	90	45	27	21	15
A-10-10	10	11	6	2	1	0	0
A-10-20	10	16	9	3	3	1	1
A-10-30	10	24	16	7	5	3	1
A-10-40	10	53	62	30	20	13	10
A-10-50	10	53	59	28	19	13	9
A-10-60	10	47	50	25	15	11	8
A-10-70	10	76	101	49	33	24	16
A-10-80	10	91	134	65	44	32	21
A-10-90	10	91	134	65	44	32	21
A-11-10	11	5	3	0	0	0	0
A-11-20	11	29	36	17	11	8	5
A-11-30	11	40	51	25	16	12	8
A-11-40	11	44	62	30	20	14	11
A-11-50	11	62	103	51	34	24	18
A-11-60	11	55	75	37	24	18	14
A-11-70	11	65	108	54	35	26	19
A-11-80	11	82	149	74	49	36	27
A-11-90	11	96	202	101	66	48	37
A-12-10	12	8	5	1	1	0	0
A-12-20	12	20	24	11	7	5	4
A-12-30	12	23	28	13	8	5	4
A-12-40	12	42	76	37	24	16	15
A-12-50	12	48	88	43	27	20	17
A-12-60	12	55	108	53	33	24	21
A-12-70	12	70	158	77	49	34	30
A-12-80	12	85	208	102	65	47	40
A-12-90	12	88	218	108	70	50	42

Table 13.10: Characteristics and optimal results for the instances from Amaral [9] with $d \in \{9, 10, 11, 12\}$.

Instance	Size	Density	Optimal solution				
	(d)	(%)	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$
A-13-10	13	9	7	2	1	1	1
A-13-20	13	28	49	24	15	11	9
A-13-30	13	37	77	38	25	19	14
A-13-40	13	40	85	42	27	20	16
A-13-50	13	53	136	68	44	32	26
A-13-60	13	56	141	70	46	33	26
A-13-70	13	72	211	105	69	50	41
A-13-80	13	87	277	138	90	66	53
A-13-90	13	92	306	153	101	74	58
A-14-10	14	9	10	4	3	1	1
A-14-20	14	25	49	24	16	11	8
A-14-30	14	30	74	36	24	18	13
A-14-40	14	35	87	43	28	21	16
A-14-50	14	56	191	94	63	47	35
A-14-60	14	62	201	99	65	49	37
A-14-70	14	75	279	138	92	68	52
A-14-80	14	86	336	167	111	83	64
A-14-90	14	91	380	187	125	93	71
A-20-10	20	9	25	12	7	4	3
A-20-20	20	22	148	73	[48,49]	34	27
A-20-30	20	30	225	111	[73,74]	[53,54]	42
A-20-40	20	37	300	149	98	[72,73]	[57,58]
A-20-50	20	53	502	249	[165,166]	[121,122]	[95,96]
A-20-60	20	67	693	345	[228,229]	167	[131,132]
A-20-70	20	71	777	385	[257,258]	[186,187]	146
A-20-80	20	77	873	434	[288,290]	[210,211]	165
A-20-90	20	88	1048	[520,521]	347	252	197
A-25-10	25	11	84	41	[26,27]	20	15
A-25-20	25	18	225	110	[71,74]	[52,55]	40
A-25-30	25	30	444	222	146	[108,110]	[86,87]
A-25-40	25	44	802	[399,400]	[263,265]	[194,198]	[152,156]
A-25-50	25	53	1023	[509,511]	[336,340]	[248,254]	[195,196]
A-25-60	25	55	1098	[547,549]	[360,364]	[266,271]	[209,212]
A-25-70	25	64	1322	[658,660]	[435,438]	[321,325]	[252,255]
A-25-80	25	81	1820	[907,910]	[600,604]	[444,450]	[349,350]
A-25-90	25	91	2169	[1080,1084]	[716,721]	[530,537]	417

Table 13.11: Characteristics and optimal results for the instances from Amaral [9] with $d \in \{13, 14, 20, 25\}$.

Chapter 14

The Checkpoint Ordering Problem

Authors: Philipp Hungerländer
Abstract: We suggest a new combinatorial optimization problem: Find an ordering of n departments with given lengths such that the total weighted sum of their distances to a given checkpoint is minimized. The Checkpoint Ordering Problem (COP) is both of theoretical and practical interest. It has several applications and is conceptually related to some well-studied combinatorial optimization problems, namely the Single-Row Facility Layout Problem, the Multi-Row Facility Layout Problem (MRFLP) and the Linear Ordering Problem.

In this paper we study the complexity of the (COP) and its special cases. The general version of the (COP) is NP-hard and can be modeled as a Quadratic Ordering Problem (QOP). We propose an exact semidefinite approach that is highly competitive for several (QOPs) to tackle the (COP). Computational experiments indicate that the (COP) is very hard to solve in practice. Supported by this finding we additionally suggest a new approach to obtain global bounds for the (MRFLP). We show the practical applicability and benefits of these bounds by providing computational experience on a variety of well-known benchmark instances.
Keywords: Combinatorial Optimization; Quadratic Ordering Problem; SDP and ILP approaches; Global Optimization; Facilities planning and design

14.1 Introduction

In this paper we introduce and analyze a new combinatorial optimization problem. The Checkpoint Ordering Problem (COP) has several interesting, partly counter-intuitive properties and relations to other combinatorial optimization problems.

An instance of the (COP) consists of n one-dimensional departments, with given positive lengths $\ell_1, \dots, \ell_n$ and connectivities $c_1, \dots, c_n$, and a checkpoint on a fixed position, e.g. left-aligned or at the center position. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{i \in [n]} c_i z_i^{\pi}, \tag{14.1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \dots, n\}$ and z_i^{π} is the distance between the center of department i and the checkpoint with respect to a particular permutation $\pi \in \Pi_n$.

Next we will elaborate on the connections of the (COP) to the Linear Ordering Problem (LOP) and related facility layout planning problems (for a thorough survey see [222]), namely the Single-Row Facility Layout Problem (SRFLP) and the Multi-Row Facility Layout Problem (MRFLP). We will also define and briefly review the main application areas of and the best algorithmic approaches to these three combinatorial optimization problems.

Row layout problems are e.g. of special interest for optimizing flexible manufacturing systems (FMSs).

FMSs are automated production systems, typically consisting of numerically controlled machines[1] and material handling devices under computer control, which are designed to produce a variety of parts. In FMSs the layout of the machines has a significant impact on the material handling cost and time, on throughput, and on productivity of the facility. A poor layout may also adulterate some of the flexibilities of an FMS [133]. The type of material-handling devices used such as handling robots, automated guided vehicles (AGVs), and gantry robots typically determines machine layout in an FMS [232]. In practice, two of the most frequently encountered layout types are the single-row layout (Figure 14.1) and multi-row layouts (Figure 14.2).

Figure 14.1: In a.) an AGV transports parts between the machines moving in both directions along a straight line. In b.) a material-handling industrial robot carries parts between the machines.

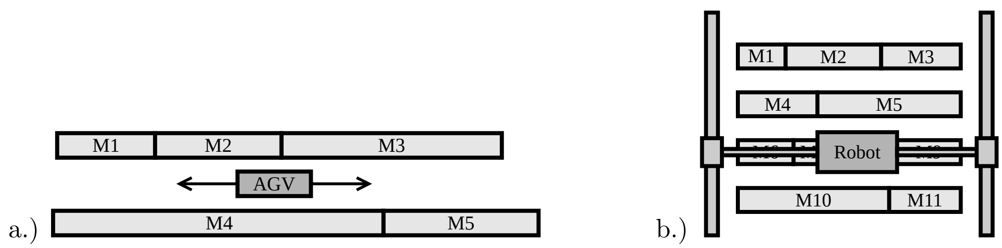


Figure 14.2: In a.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. In b.) a gantry robot is used when the space is limited.

The Single-Row Facility Layout Problem (SRFLP) The easiest known layout type is single-row layout. It arises as the problem of ordering stations on a production line where the material flow is handled by an AGV in both directions on a straight-line path [150]. An instance of the (SRFLP) consists of n one-dimensional departments, with given positive lengths $\ell_1, \dots, \ell_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi}, \tag{14.2}$$

where Π_n is the set of permutations of the indices $[n]$ and z_{ij}^{π} is the center-to-center distance between departments i and j with respect to a particular permutation $\pi \in \Pi_n$. Note that the connectivities are assumed to be non-negative for all row layout problems to ensure boundedness of the objective value of the optimal layout. For the (SRFLP) $c_{ij} \geq 0$ further guarantees that all departments are placed next to each other without spacing. For the (COP) we do not need this restriction on the connectivities as space between the departments is not allowed.

The (SRFLP) is one of the few layout problems for which strong global lower bounds and even optimal solutions can be computed for instances of reasonable size. The global optimization approaches for the

[1]Departments are more conveniently denoted as machines in the manufacturing context.

(SRFLP) are based on relaxations of integer linear programming (ILP) and semidefinite programming (SDP) formulations.

The strongest ILP approach for the (SRFLP) is an LP-based cutting plane algorithm using betweenness variables [8] that can solve instances with up to 35 departments within a few hours. The strongest SDP approach to date using products of ordering variables [169] is even stronger and can solve instances with up to 42 departments within a few hours. Also several heuristic algorithms have been suggested to tackle the (SRFLP), the best ones to date are [80, 188, 267].

Several practical applications of the (SRFLP) have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [283], the assignment of airplanes to gates in an airport terminal [291], the arrangement of machines in flexible manufacturing systems [150], the arrangement of books on a shelf and the assignment of disk cylinders to files [252].

Similar applications are conceivable for the (COP), e.g. the rooms on a corridor could be arranged such that the weighted sum of their distances with the office of the head is minimized or planes could be assigned to gates such that the weighted sum of their distances from the entrance of the airport terminal is minimized. When comparing the (SRFLP) with the (COP), we observe that the problems are quite similar. One difference is that an (SRFLP) instance has $\binom{n}{2}$ connectivities while an (COP) instance has only n connectivities. On first sight the (SRFLP) seems more difficult than the (COP) (at least this was the first impression of the author). We will show in the following sections that the opposite is true.

The Multi-Row Layout Problem (MRFLP). The Double-Row Facility Layout Problem (DRFLP) is a natural extension of the (SRFLP) in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel (see Figure 14.2). This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the (DRFLP) is especially relevant for real-world applications. The (DRFLP) can be further generalized to the (MRFLP), where the departments are arranged along k parallel rows. If

1. the assignment of the departments to the rows is fixed,
2. all the rows have a common left origin and
3. no empty space is allowed between the departments,

the (MRFLP) simplifies to the k-Parallel Row Ordering Problem (kPROP). For $k = 2$ rows we simply call it Parallel Ordering Problem (PROP).

The (MRFLP) has many applications such as computer backboard wiring [288], campus planning [90], scheduling [115], typewriter keyboard design [253], hospital layout [95], the layout of machines in an automated manufacturing system [151], balancing hydraulic turbine runners [198], numerical analysis [46], optimal digital signal processors memory layout generation [302]. Different extensions of the (MRFLP) like considering a clearance between any two adjacent machines given as a fuzzy set [114] or the design of a FMS in one or multiple rows [101] have been proposed and tackled with genetic algorithms.

Somewhat surprisingly, the development of exact algorithms for the (DRFLP) and the (MRFLP) has received only limited attention in the literature. In the 1980s Heragu and Kusiak [150] proposed a non-linear programming model and obtained locally optimal solutions to the (SRFLP) and the (DRFLP). Recently Chung and Tanchoco [72] (see also [310]) focused exclusively on the (DRFLP) and proposed a mixed integer programming (MIP) formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Amaral [11] proposed an improved MIP formulation that allowed him to solve instances with up to 12 departments to optimality.

For the (PROP) a MIP formulation was proposed by Amaral [12]. The MIP approach allows to solve instances with up to 23 departments to optimality within a few days of computing time. Hungerländer [159] suggested a semidefinite programming approach that yields tight global bounds for instances of the same size within a few minutes on a similar machine. Additionally the SDP algorithm is able to produce reasonable global bounds for instances with up to 100 departments and it is also applicable for the (kPROP) with $k \geq 3$ rows and even yields better computational results for a larger number of rows.

Note that the (COP) can be interpreted as a very special (PROP). Let us assume w.l.o.g. that the checkpoint is centered. Then the (PROP) simplifies to the (COP) if we

1. assign all but one of the departments to row 1,
2. set all connectivities between the departments on row 1 to zero and
3. set the length of the department on row 2 to be the sum of the lengths of the departments on row 1.

In summary the (COP) is a lot easier than (MRFLP) as it contains 6 assumptions that are part of the optimization problem when considering the (MRFLP). Nonetheless we will show that the (COP) is still a very difficult combinatorial optimization problem and in some sense even harder than the (MRFLP). Further note that we could possibly improve the optimal objective value of a (COP) if we would allow space between the departments. But we refrain from allowing space because we want keep the (definition of the) (COP) preferably simple.

The Linear Ordering Problem (LOP). Ordering problems associate to each ordering (or permutation) of the set $[n]$ a profit and the goal is to find an ordering of maximum profit. In the simplest case of the Linear Ordering Problem (LOP), this profit is determined by those pairs $(u, v) \in [n] \times [n]$, where u comes before v in the ordering. Thus in its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that

$$\sum_{\substack{i,j \in [n] \\ i<j}} w_{\pi(i),\pi(j)},$$

is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of nodes i and j either arc (i, j) or arc (j, i), but not both. Then the (LOP) consists of finding an acyclic tournament, i.e. a tournament without directed cycles, of G of maximum total edge weight.

The (LOP) is equivalent to the acyclic subdigraph problem and the feedback arc set problem. It is well known to be NP-hard [112] and it is even NP-hard to approximate (LOP) within the factor $\frac{65}{66}$ [238]. Surprisingly there is not much known about heuristics with approximation guarantees. If all entries of W are nonnegative, a $\frac{1}{2}$-approximation is trivial, but no better polynomial time approximation is known. To narrow this quite large gap $[\frac{1}{2}, \frac{65}{66}]$ is a challenging open problem.

The (LOP) arises in a large number of applications in such diverse fields as economy, sociology (determination of ancestry relationships [119]), graph drawing (one sided crossing minimization [173]), archaeology, scheduling [39], assessment of corruption perception [1] and ranking in sports tournaments. In 1959 Kemeny [182] posed the first application of the (LOP) (Kemeny's problem) concerning the aggregation of individual orderings to a common one in the best possible way. Later on further problems were proposed in the context of mathematical psychology and the theory of social choice that can be formulated as linear ordering problems [104, 286]. The probably most established application of the (LOP) is the triangulation of input-output matrices of an economy. Leontief [205, 206] was awarded the Nobel Prize in 1973 for his research on input-output analysis.

Currently available exact algorithms include a Branch-and-Bound algorithm that uses a linear programming based lower bound by Kaas [175], a Branch-and-Cut algorithm proposed by Grötschel, Jünger and Reinelt [124] and a combined interior-point cutting-plane algorithm by Mitchell and Borchers [225] who explore polyhedral relaxations of the problem and also provide computational results using Branch-and-Cut. The current state-of-the-art Branch-and-Cut algorithm was developed by the working group of

Prof. Reinelt in Heidelberg and is based on sophisticated cut generation procedures (for details see [219]). It can solve large instances from specific instance classes with up to 150 objects, while it fails on other much smaller instances with only 50 or less objects. There exist many heuristics and methaheuristics for the (LOP) and some of them are quite good in finding the optimal solution for large instances in reasonable time. For a recent survey and comparison see [219, 220].

Although the (LOP) and the (COP) have apparently a similar structure, it is harder to directly relate these two problems. We will show in Section 14.2 that the (COP) with left-aligned or right-aligned checkpoint is in fact a (LOP) with some additional structure and can be solved efficiently by a greedy heuristic. But if the checkpoint is located at some arbitrary position the (COP) is also NP-hard and practically much more difficult to solve to optimality than the (LOP) (for details see Section 14.5).

Toy Examples. Now let us further clarify the workings and differences of the (SRFLP), the (PROP), the (MRFLP) and the (COP) with the help of a toy example: We consider 4 departments with lengths $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4$. Additionally we are given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$. For the (COP) we assign department 1 to row 2 and all other departments to row 1 and hence disregard the connectivity $c_{24} = 2$. Figure 14.3 illustrates the optimal layouts and the associated costs for the four combinatorial optimization problems.

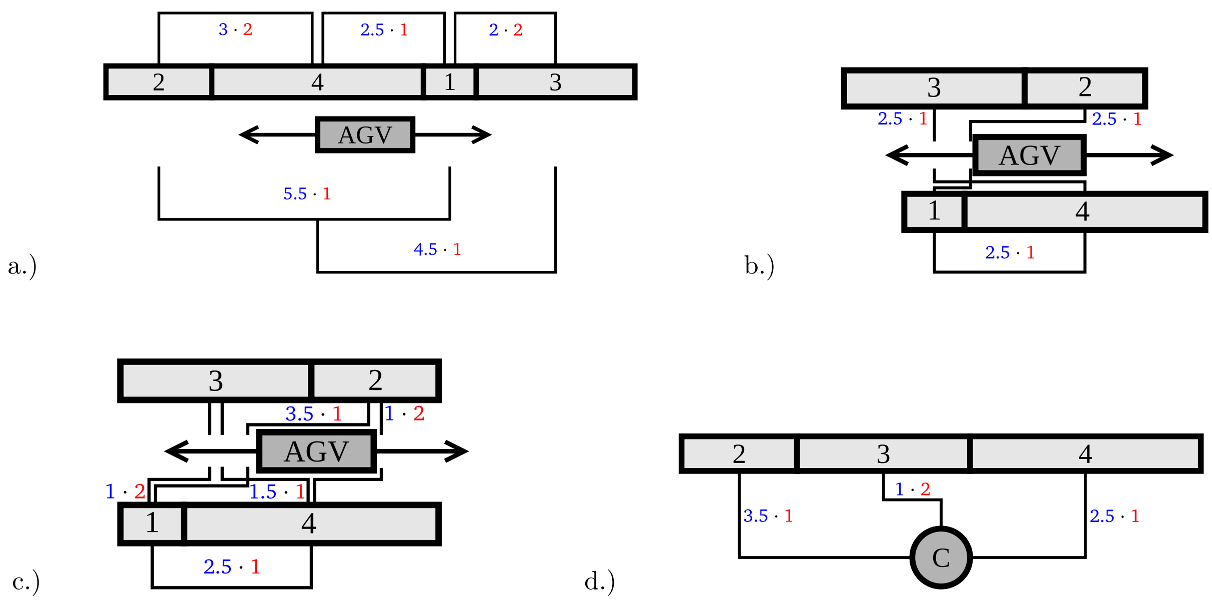


Figure 14.3: We are given the following data: $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4,\ c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$. In a.) we display the optimal layout for the (SRFLP) with corresponding costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we show the optimal layout for the (DRFLP). The corresponding costs are $2.5 \cdot 1 + 2.5 \cdot 1 + 2.5 \cdot 1 = 7.5$. In c.) we display the optimal layout for the (PROP) with departments 2 and 3 assign to row 1 and departments 1 and 4 assign to row 2. The corresponding costs are $3.5 \cdot 1 + 1 \cdot 2 + 1 \cdot 2 + 1.5 \cdot 1 + 2.5 \cdot 1 = 11.5$. Finally in d.) we depict the optimal layout for the (COP) with department 1 assigned to row 2 and all other departments assigned to row 1, disregarding the connectivity $c_{24} = 2$. Further we assume that the checkpoint lies at the center. The costs of the optimal (COP) layout are $3.5 \cdot 1 + 1 \cdot 2 + 2.5 \cdot 1 = 8$.

Finally we also want to clarify the workings of the (LOP) with the help of a toy example. We consider 4 objects and the weights $w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. Figure 14.4 illustrates the optimal ordering of the objects and the corresponding benefit.

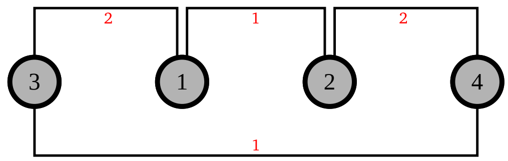


Figure 14.4: We are given 4 objects and the weights $w_{12} = w_{41} = w_{34} = 1$, $w_{31} = w_{24} = 2$. We display the optimal (LOP) solution with the corresponding benefit of $1 + 1 + 2 + 2 = 6$.

Outline. The main contributions of this paper are the following:

- We propose a new combinatorial optimization problem that is both of theoretical and practical interest.
- We study the complexity of the (COP) and its special cases, pointing out several connections to related problems.
- We model the (COP) as a Quadratic Ordering Problem (QOP). Building on this model we propose an exact semidefinite optimization approach that is highly competitive for several (QOPs) to tackle the (COP).
- We introduce the Single-Row bound that is the key for obtaining reasonably strong global bounds for the (MRFLP).
- We demonstrate in a computational study the practical difficulty of the (COP) and analyse the associated reasons in some detail.
- We showcase the applicability and benefits of the Single-Row bound on a large variety of benchmark instances.

The paper is structured as follows. In Section 14.2 we study the complexity of the (COP) and its special cases. In Section 14.3 we review matrix-based formulations and semidefinite relaxations proposed for the (SRFLP) and the (LOP) and then show that these models can be extended for the (COP). The Single-Row bound, which is a lower bound to the optimal (MRFLP) solution value, is introduced in Section 14.4. Finally in Section 14.5 we conduct several computational experiments on a large variety of well-known benchmark instances, indicating the practical applicability and benefits of the approaches suggested. Section 14.6 concludes the paper.

14.2 Complexity of the Checkpoint Ordering Problem

Consider the decision variant of the (COP): given some value M we ask whether there exists a permutation of the departments such that the obtained costs are at most M.:

Decision Checkpoint Ordering (DIRO):
Instance: n departments with given lengths $\ell \in \mathbb{N}$ and integer connectivities and a checkpoint on a fixed position.
Question: Is there an ordering π of the departments such that the total costs $\sum_{i\in[n]} c_i z_i^{\pi}$ are $\leq M$?

In the following proof we assume w.l.o.g. that the checkpoint is located at the center to simplify the presentation.

Theorem 14.1 *DIRO is NP-complete.*

Proof. It is clear that DIRO $\in$ NP since a nondeterministic algorithm need only guess an ordering π and check in polynomial time if the corresponding costs are $\leq M$.

To prove that DIRO is NP-complete, we give an NP-complete problem and a polynomial-time transformation to DIRO. The following problem is NP-complete (see Subsection 3.1.5 of [111], originally proven by [179]):

PARTITION:
Instance: A finite set A and a "size" $s(a) \in \mathbb{N}$ for each $a \in A$ such that $\sum_{a \in A} s(a) = 2B$ is even.
Question: Is there a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = B?$$

We transform an instance of PARTITION to an instance of DIRO as follows. We replace each element $a \in A$ with given size $s(a)$ by a department a with length $s(a)$ and connectivity $s(a)$. Additionally to this local replacement we use an enforcer[2] by introducing a further department t with length 1 and connectivity $2(B+1)^2$ on row 1. Clearly this DIRO instance can be constructed in polynomial time from the PARTITION instance.

If the ordering π is optimal then the center of department t on row 1 is located exactly above the center of the department on row 2 because of the large connectivity of department t.

Due to the definition of the connectivities of the departments $a \in A$, it does not change the objective value if we switch the positions of two departments both located left or right of t. Hence the only way to influence the objective value is to decide whether the departments should be located left or right of t. If we can find a subset $A' \subseteq A$ of the departments such that $\sum_{a \in A'} \ell_a = B$ and place them left of t and all other departments right of t then the corresponding ordering π is for sure optimal. If the sum of lengths of the departments left of t in the optimal ordering π does not give B then there exists no subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = B$.

In summary we have shown that there exists an ordering π of the departments on row 1 such that the total costs are $\leq M := B(B+1)$ if and only if there exists a subset $A' \subseteq A$ of the corresponding PARTITION instance such that $\sum_{a \in A'} s(a) = B$. □ □

In the layout context the above result can be interpreted as follows: The minimization of the inter-row costs that occur in all multi-row layout problems is NP-hard, even in its simplest version. This is not only an interesting theoretical insight on its own but also supports our definition of the Single-Row bound in Section 14.4. Next let us consider some more specialized versions of the `(COP)` that turn out to be solvable by a greedy heuristic, hence in particular in polynomial time.

First we assume that all departments have the same length. In this case the optimal permutation of the departments can be obtained by a simple greedy selection. We choose the permutation π with the following property: The higher the connectivity of a department the smaller is its distance from the checkpoint. Next we give a short formal argument for the above claim and refer to Figure 14.5 for a toy example of this variant of the `(COP)`.

Fact 14.2 *A permutation π is optimal for the* `(COP)` *with identical department lengths iff it ensures the inequalities*

$$(z_i - z_j)(c_i - c_j) \leq 0, \quad i, j \in [n], i < j, \tag{14.3}$$

where z_i, $i \in [n]$, denotes the distance of the center of department i from the checkpoint.

Proof. The change of the objective function caused by swapping two departments i and j is $-(z_i - z_j)(c_i - c_j)$. Hence in particular the change in the objective function caused by this swap is independent of the

[2] A picturesque term suggested by [292].

length and connectivities of all other departments $k \in [n],\ k \neq i,\ k \neq j$. Now assume that there exists an optimal permutation that does not ensure one inequality in (14.3), i.e. $-(z_i - z_j)(c_i - c_j) < 0$ for some departments i and j. Now swapping the two departments improves the objective value. As all other departments are not affected by pairwise swaps, it is not possible to improve the objective value by an arbitrary number of pairwise changes if (14.3) holds. Hence (14.3) is not only a necessary condition but also a sufficient one. □ □

Note that the special case of the `(SRFLP)` where all department lengths are equal and the connectivities are binary is still NP-hard [112]. This problem is called Minimum Linear Arrangement `(LA)`, belongs to the class of graph layout problems and is NP-hard even if the underlying graph G is bipartite [111]. `(LA)` was originally proposed by Harper [131, 132] to develop error-correcting codes with minimal average absolute errors and was since then applied to VLSI design [300], single machine job scheduling [2, 258] and computational biology [180, 226]. There exist approximation algorithms for `(LA)` with performance guarantee $O(\log n)$ [41, 257] and $O(\sqrt{\log n}\log\log n)$ [58, 99]. For further details on graph layout problems we refer to the survey paper of Díaz et al. [89].

Next we assume that the checkpoint is left-aligned or right-aligned. Also in this case the optimal permutation of the departments can be obtained by a simple greedy selection. We choose the permutation π with the following property: The higher the relative connectivity $\frac{c_i}{\ell_i}$ of a department the smaller is its distance from the checkpoint. Next we give a short formal argument for the above claim and refer to Figure 14.5 for a toy example of this variant of the `(COP)`. To facilitate the presentation of the proof, we assume w.l.o.g. that

1. the checkpoint is left-aligned and
2. the relative connectivities $\frac{c_i}{l_i},\ i \in [n]$ are all distinct.[3]

The following result is in fact known as Smith's Rule [287] in the scheduling context, where it describes a greedy algorithm to solve single-machine scheduling with the objective of minimizing the sum of completion times. For convenience we restate the proof in our notation.

Fact 14.3 *The permutation π is optimal for the* `(COP)` *with a left-aligned checkpoint iff it satisfies the conditions*

$$\frac{c_i}{\ell_i} > \frac{c_j}{\ell_j},\ i,j \in [n],\ \pi(i) < \pi(j). \tag{14.4}$$

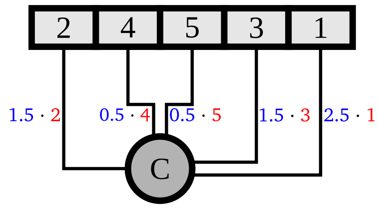
Proof. Assume that there exists an optimal permutation that does not satisfy condition (14.4) for two departments i and j: $\frac{c_i}{\ell_i} < \frac{c_j}{\ell_j}$ and $\pi(i) < \pi(j)$. Then there are also two neighboring departments k and l ($\pi(i) \leq \pi(k) < \pi(k) + 1 = \pi(l) \leq \pi(j)$) that do not satisfy condition (14.4): $\frac{c_k}{\ell_k} < \frac{c_l}{\ell_l}$ and $\pi(k) + 1 = \pi(l)$. Now if we swap k and l, the value of the objective function changes by the term $c_k\ell_l - c_l\ell_k$. But this term is negative as $\frac{c_k}{\ell_k} < \frac{c_l}{\ell_l}$ holds which yields a contradiction to the assumption that the permutation was optimal. Finally note that condition (14.4) defines a unique permutation, hence it is not only necessary but also sufficient. □ □

We can interpret the `(COP)` with left-aligned checkpoint also as a `(LOP)` with special structure. We collect the lengths of the departments in a column vector ℓ and the connectivities of the departments in a column vector c. Now we aim to find a simultaneous permutation π of the rows and columns of $W = \ell c^\top$ such that

$$\sum_{\substack{i,j\in[n]\\ i<j}} w_{\pi(i),\pi(j)},$$

[3] The proof is analogous for a right-aligned checkpoint and two departments i and j with the same relative connectivities can be combined to one department with length $l_i + l_j$ and connectivity $c_i + c_j$.

is minimized. Hence contrary to the (LOP) the matrix entries of W are not independent but determined by an outer product of two vectors. This additional structure ensures that the (COP) with left-aligned checkpoint can be solved in polynomial time.

a.) 

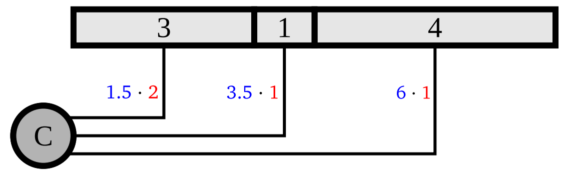
b.) 

Figure 14.5: In a.) we display the optimal layout for the (COP) with identical department lengths equal to 1 and connectivities $c_1 = 1,\ c_2 = 2,\ c_3 = 3,\ c_4 = 4,\ c_5 = 5$. The corresponding layout costs are $0.5{\cdot}5+0.5{\cdot}4+1.5{\cdot}3+1.5{\cdot}2+2.5{\cdot}1 = 14.5$. In b.) we show the optimal layout for the (COP) with left-aligned checkpoint on the following instance with 3 departments: $\ell_1 = 1, \ell_2 = 2,\ \ell_3 = 3,\ c_1 = 1,\ c_2 = 3,\ c_3 = 2$. The associated layout costs are $1 \cdot 3 + 2.5 \cdot 1 + 4.5 \cdot 2 = 14.5$.

We can summarize the above results as follows: The (COP) is NP-hard and its "hard" part is to determine which department goes on which side of the checkpoint in the optimal solution.

14.3 A Mathematical Model for the Checkpoint Ordering Problem

In the following we show that the (COP) in its general version can be modelled as a Quadratic Ordering Problem (QOP) and we suggest an exact approach based on semidefinite programming to solve it. The proposed SDP approach proved to be highly competitive for other (QOP), e.g. it is the method of choice for the (SRFLP) [169], Multi-Level Crossing Minimization [64] and Multi-Level Verticality Optimization [62]. For the (LOP) and the (MRFLP) our approach has proved to be competitive with the respective state-of-the-art ILP algorithms, for a detailed analysis of the strengths and weaknesses of the SDP and ILP approaches see [168] for the (LOP) and [167] for the (MRFLP). Hence we will be able to obtain global bounds for the (LOP), the (SRFLP), the (MRFLP) and the (COP) using the same algorithm. This allows a fair comparison of the practical difficulty of the four problems in Section 14.5.

In the following two subsections we will show that the (SRFLP), the (LOP) and the (COP) can all be modeled as optimization problems on the same variables with the same set of linear constraints and different (linear or quadratic) objective functions. Additionally we will discuss how to use these models to obtain tight but still computationally tractable relaxations for the problems. Modelling the (MRFLP) as a (QOP) and solving it with our algorithmic framework is a bit more involved. We will elaborate on necessary extensions in Section 14.4 and provide a review of corresponding computational results in Section 14.5.

14.3.1 Matrix-Based Formulations and SDP Relaxations for the (SRFLP) and the (LOP)

We can deduce an SDP formulation for the (SRFLP) by introducing bivalent ordering variables $y_{ij},\ i, j \in [n],\ i < j$,

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise,} \end{cases} \tag{14.5}$$

With their help we can easily rewrite the (SRFLP) from (14.2) in terms of ordering variables (for a detailed derivation see e.g. [169])

$$\min_{\substack{y_{ij}\in\{-1,1\}\\ i,j\in[n],\ i<j}} \quad K - \sum_{\substack{i,j\in[n]\\ i<j}} \frac{c_{ij}}{2}\left(\sum_{\substack{k\in[n]\\ k<i}} \ell_k y_{ki} y_{kj} - \sum_{\substack{k\in[n]\\ i<k<j}} \ell_k y_{ik} y_{kj} + \sum_{\substack{k\in[n]\\ k>j}} \ell_k y_{ik} y_{jk}\right), \tag{14.6}$$

$$\text{subject to:} \quad y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k\in[n],\ i<j<k, \tag{14.7}$$

with

$$K = \left(\sum_{\substack{i,j\in[n]\\ i<j}} \frac{c_{ij}}{2}\right) L, \qquad L = \sum_{k\in[n]} \ell_k.$$

In [48] it is shown that the equations (14.7) formulated in a $\{0,1\}$ model describe the smallest linear subspace that contains the quadratic ordering polytope

$$\mathcal{P}_{QO}^n := \operatorname{conv}\{\ yy^\top\ :\ y\in\{-1,1\},\ |y_{ij}+y_{jk}-y_{ik}| = 1\ \}.$$

To obtain matrix-based relaxations we collect the ordering variables in a vector y and consider the matrix $Y = yy^\top$. The main diagonal entries of Y correspond to y_{ij}^2 and hence $\operatorname{diag}(Y) = e$, the vector of all ones. Now we can formulate the (SRFLP) as a semidefinite program, first proposed in [22]

$$\min\{\ \langle C,Y\rangle + K:\ Y \text{ satisfies (14.7)},\ \ \operatorname{diag}(Y)=e,\ \operatorname{rank}(Y)=1,\ Y\succcurlyeq 0\ \}, \tag{SRFLP}$$

where the cost matrix C is deduced from (14.6). Dropping the rank constraint yields the basic semidefinite relaxation of the (SRFLP)

$$\min\{\ \langle C,Y\rangle + K:\ Y \text{ satisfies (14.7)},\ \ \operatorname{diag}(Y)=e,\ Y\succcurlyeq 0\ \}, \tag{$\mathtt{SDP_{trivial}}$}$$

providing a lower bound on the optimal value of the (SRFLP).

As Y is actually a matrix with $\{-1,1\}$ entries in the original (SRFLP) formulation, Anjos and Vanelli [20] proposed to further tighten ($\mathtt{SDP_{trivial}}$) by adding the triangle inequalities, defining the metric polytope $\mathcal{M}$ and known to be facet-defining for the cut polytope, see e.g. [88]

$$\mathcal{M} = \left\{ Y:\begin{pmatrix} -1 & -1 & -1\\ -1 & 1 & 1\\ 1 & -1 & 1\\ 1 & 1 & -1\end{pmatrix}\begin{pmatrix} Y_{i,j}\\ Y_{j,k}\\ Y_{i,k}\end{pmatrix} \le e,\ i<j<k\in\binom{n}{2}\right\}. \tag{14.8}$$

Adding the triangle inequalities to ($\mathtt{SDP_{trivial}}$), we obtain the following relaxation of the (SRFLP)

$$\min\{\ \langle C,Y\rangle + K:\ Y \text{ satisfies (14.7)},\ Y\in\mathcal{M},\ \operatorname{diag}(Y)=e,\ Y\succcurlyeq 0\ \}. \tag{$\mathtt{SDP_{basic}}$}$$

As solving ($\mathtt{SDP_{basic}}$) directly with an interior-point solver like CSDP gets far too expensive, Anjos and Vannelli [20] suggest to use the $\approx \frac{1}{12}n^6$ triangle inequalities as cutting planes in their algorithmic framework.

Recently Hungerländer and Rendl [169] suggested a further strengthening of ($\mathtt{SDP_{basic}}$) and an alternative algorithmic approach to solve large SDP relaxations. To this end we introduce the matrix

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{14.9}$$

and relax the equation $Y - yy^\top = 0$ to

$$Y - yy^T \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0,$$

which is convex due to the Schur-complement lemma. Note that $Z \succcurlyeq 0$ is in general a stronger constraint than $Y \succcurlyeq 0$. Additionally we use an approach suggested by Lovász and Schrijver in [214] to further improve on the strength of the semidefinite relaxation. This yields the following inequalities

$$\begin{aligned} -1 - y_{lm} \le y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \le 1 + y_{lm}, \qquad & i < j < k \in [n],\ l < m \in [n] \\ -1 + y_{lm} \le y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \le 1 - y_{lm}, \qquad & i < j < k \in [n],\ l < m \in [n] \end{aligned} \tag{14.10}$$

that are generated by multiplying the 3-cycle inequalities valid for the ordering problem

$$1 - y_{ij} - y_{jk} + y_{ik} \ge 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \ge 0,$$

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. These constraints define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (14.10)\ \}, \tag{14.11}$$

consisting of $\approx \frac{1}{3}n^5$ constraints. In summary, we come up with the following relaxation of the (SRFLP)

$$\min \{\ \langle C, Y \rangle + K :\ Y \text{ satisfies } (14.7),\ Z \in (\mathcal{M} \cap \mathcal{LS}),\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \tag{$\texttt{SDP}_{\texttt{standard}}$}$$

Note that both the formulations and relaxations discussed above also work for the (LOP) if we simply replace $\langle C, Y \rangle + K$ by the objective function $(w_{ij} - w_{ji})\frac{y_{ij}+1}{2} + w_{ji}$ that corresponds to the (LOP). In the following subsection we show how to generalize the semidefinite models and relaxations for the (SRFLP) and the (LOP) to the (COP).

14.3.2 A Matrix-Based Formulation and Several SDP Relaxations for the (COP)

To simplify the notation, we consider the (COP) on $n-1$ departments. First we introduce additional ordering variables $y_{in},\ i \in [n-1]$ to relate the positions of the $n-1$ departments to the checkpoint that is located w.l.o.g. at the center of the row. Now the distances of the departments from the checkpoint can also be expressed as quadratic terms in ordering variables: For department $i \in [n-1]$, we sum up the lengths of the departments left of i plus $\ell_i/2$ and denote it by d_i. Furthermore we compute the position of the checkpoint d_c as $L/2$, i.e. the total length of the departments divided by 2. Then we subtract d_i from d_c. This difference gives the distance of the center of department i from the checkpoint, if department i is located to the left of the checkpoint. If department i is located to the right of the checkpoint, this difference is minus the distance of the center of department i from the checkpoint. Finally we multiply this difference by the ordering variable y_{in} that is 1 if department i is located to the left of the checkpoint and -1 if department i is located to the right of the checkpoint:

$$z_{in} = y_{in}\left(d_k - d_i\right), \qquad i \in [n-1], \tag{14.12}$$

with

$$d_i = \frac{\ell_i}{2} + \sum_{k\in[n],\ k<i} \ell_k \frac{1+y_{ki}}{2} + \sum_{k\in[n],\ k>i} \ell_k \frac{1-y_{ik}}{2},\ i \in [n-1], \qquad d_k = \frac{L}{2}.$$

The additional multiplication with y_{in} ensures a correct calculation of all distances through the following constraints:

$$z_{in} \geq 0, \qquad i \in [n-1]. \tag{14.13}$$

Also note that (14.12) simplifies to

$$z_{in} = \frac{y_{in}}{2}\left(\sum_{k\in[n],\ k>i} \ell_k y_{ik} - \sum_{k\in[n],\ k<i} \ell_k y_{ki}\right), \quad i \in [n-1].$$

In summary we are able to rewrite (14.1) using ordering variables.

Theorem 14.4 *Minimizing* $\sum_{i\in[n-1]} c_i z_{in}$ *over* $y \in \{-1,1\}^{\binom{n}{2}}$, *(14.7), (14.12) and (14.13) solves the* (COP).

Proof. The equations (14.7) model transitivity for $y \in \{-1,1\}^{\binom{n}{2}}$ [48] and hence suffice together with the integrality conditions on y and (14.13) to induce all feasible layouts. Thus by definition of the distances z_i in (14.12), the objective value $\sum_{i\in[n-1]} c_i z_i$ gives the costs of a feasible layout. □ □

Next we rewrite the objective function in terms of matrices and obtain a matrix-based formulation:

$$\min\left\{\langle C_d, Z\rangle \ :\ y \in \{-1,1\}^{\binom{n}{2}},\ y \text{ satisfies (14.7) and (14.13)}\right\}, \tag{COP}$$

where the cost matrix C_d is deduced by equating the coefficients of the following equation:

$$2\langle C_d, Z\rangle \stackrel{!}{=} \sum_{i\in[n-1]} c_i y_{in}\left(\sum_{k\in[n],\ k>i} \ell_k y_{ik} - \sum_{k\in[n],\ k<i} \ell_k y_{ki}\right).$$

Finally we can further rewrite the above matrix-based formulation as an SDP:

Theorem 14.5 *The problem*

$$\min\left\{\ \langle C_d, Z\rangle : Z \textit{ satisfies } (14.7) \textit{ and } (14.13)\ ,\ \textit{diag}(Z) = e,\ Z \succcurlyeq 0,\ y \in \{-1,1\}^{\binom{n}{2}}\ \right\},$$

is equivalent to the (COP).

Proof. Since $y_{ij}^2 = 1,\ i,j \in [n],\ i < j$ we have $\operatorname{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. By Theorem 14.4, integrality on Y together with (14.7) and (14.13) suffice to induce all feasible layouts of the (COP) and the objective function $\langle C_d, Z\rangle$ gives the correct costs for all feasible layouts. □ □

As we are able to model the (COP) on the same variables as the (SRFLP) and the (LOP), namely products of ordering variables, we can adopt the strongest SDP relaxation from the previous section:

$$\min\ \{\ \langle C_d, Z\rangle :\ Y \text{ satisfies (14.7) and (14.13)},\ Z \in (\mathcal{M} \cap \mathcal{LS})\ ,\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \tag{SDP$_{\texttt{standard}}$}$$

But the objective function of the (COP) is more complex than the one of the (SRFLP) because it contains not only products of ordering variables with three different indices that are tightly constraint through the minimal equation system (14.7) but all kinds of products of ordering variables. Hence we suggest additional valid inequalities to tighten the above relaxation that will prove to be very important in practice, for details see Subsection 14.5.2. We propose triangle inequalities relating the distances between three departments or two departments and the checkpoint:

$$z_{ij} + z_{ik} \geq z_{jk}, \qquad z_{ij} + z_{ik} \geq z_{jk}, \qquad z_{ik} + z_{jk} \geq z_{ij}, \qquad i,j,k \in [n],\ i < j < k. \tag{14.14}$$

By an inductive argument it is very easy to see that the above constraints imply similar constraints for more than three departments. Hence let us define the polytope

$$\mathcal{DV} := \{\ Z : Z \text{ satisfies } (14.14)\ \}, \tag{14.15}$$

containing the $3\binom{n}{3}$ triangle inequalities. Adding these constraints to (SDP$_{\texttt{standard}}$) yields

$$\min \{\ \langle C_d, Z \rangle :\ Y \text{ satisfies (14.7) and (14.13)},\ Z \in (\mathcal{M} \cap \mathcal{LS} \cap \mathcal{DV}),\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \qquad (\texttt{SDP}_{\texttt{full}})$$

It was demonstrated in [168] that using $\mathcal{M}$ and $\mathcal{LS}$ in the semidefinite relaxation pays off for several ordering problems including the (SRFLP) in practice. But for the (COP) we will refrain from using $\mathcal{M} \cap \mathcal{LS}$ in some experiments due to reasons that we will be discussed in Section 14.5 in detail. Then we will work with the following SDP relaxation instead:

$$\min \{\ \langle C_d, Z \rangle :\ Y \text{ satisfies (14.7) and (14.13)},\ Z \in \mathcal{DV},\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \qquad (\texttt{SDP}_{\texttt{cheap}})$$

14.4 A Lower Bound for the Multi-Row Facility Layout Problem

In this section we propose a cheap but reasonably tight lower bound for the (MRFLP). To begin with we give an outline of an algorithmic approach for solving the (MRFLP) using SDP relaxations: We solve an appropriate SDP relaxation for each possible assignment of the departments to the m rows given by the function $r :\ [n] \to \mathcal{R}$ with $\mathcal{R} := \{1, \ldots, m\}$. From the bounds obtained, we can deduce global upper and lower bounds for the (MRFLP) that are the minima of all upper and lower bounds respectively.

The following three observations should motivate the consideration of cheap, reasonably tight lower bound for the (MRFLP):

1. While the (SRFLP) can be solved to optimality for up to 42 departments and can be tightly approximated for up to 110 departments (for detailed computational results see Subsection 14.5.1), the (DRFLP) can be solved to optimality only for up to 12 departments and for larger instances there are no reasonable global bounds known. For the (MRFLP) with more than 2 rows global bounds are known only for instances with ≤ 6 departments to date.

2. Layout problems with exclusively inner-row costs like the (SRFLP) posses significantly tighter, practically computable SDP relaxations than problems of the same size where inter-row costs are present, like e.g. the (PROP) (for detailed computational results again Subsection 14.5.1).

3. The theoretical (see Section 14.2) and especially the practical (see Subsection 14.5.2) difficulty of the (COP) fall in line with the above observation, because the (COP) is practically even more difficult than the (PROP) and its objective function is the first one to contain only inter-row costs.

The three observations above support the following definition: For a given row assignment of n departments to m rows, we sort the m rows with respect to the sum of lengths of the departments located on each row, i.e. the longest row is the first one and so on. Then we define the Single-Row bound to the optimal (MRFLP) solution value for a given row assignment as the sum of the lower bounds of m associated

(SRFLPs) with new adapted pairwise connectivities $\overline{c}_{ij}$ given by:

$$\overline{c}_{ij} = c_{ij} + \sum_{k=r(i)+1}^{m} \sum_{\substack{l \in [n] \\ r(l)=k}} d_{ijl}, \qquad i,j \in [n],\ i<j,\ r(i)=r(j), \tag{14.16}$$

with

$$d_{ijl} = \begin{cases} c_{il}, & \text{if } c_{jl} = \max_{p\in[n],\ r(p)=r(i)} c_{pl}, \\ c_{jl}, & \text{if } c_{il} = \max_{p\in[n],\ r(p)=r(i)} c_{pl}, \\ 0, & \text{otherwise.} \end{cases}$$

Hence for a given row assignment we aim to incorporate the inter-row costs from the (MRFLP) in several associated (SRFLPs) by adding inter-row connectivities that are unavoidable to the inner-row connectivities. The idea is to position the center of each department l on row $k \in \{r(i)+1,\dots,m\}$ below the center of the department in row $r(i)$ that has the highest connectivity with l and then define the unavoidable inter-row connectivities d_{ijl}.[4] To obtain a global lower bound to the (MRFLP) that we denote as Single-Row bound we compute lower bounds for all possible row assignments.

Let us further clarify the workings of the Single-Row bound with the help of several toy examples. First we are given 2 rows and 4 departments with lengths $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4$, and connectivities $c_{12} = 2,\ c_{13} = 8,\ c_{14} = 1,\ c_{24} = 5,\ c_{25} = 1,\ c_{34} = 4$. Figure 14.6 illustrates the optimal double-row layout with costs of 17.5 and the corresponding (SRFLPs) needed to determine the Single-Row bound with the same value.

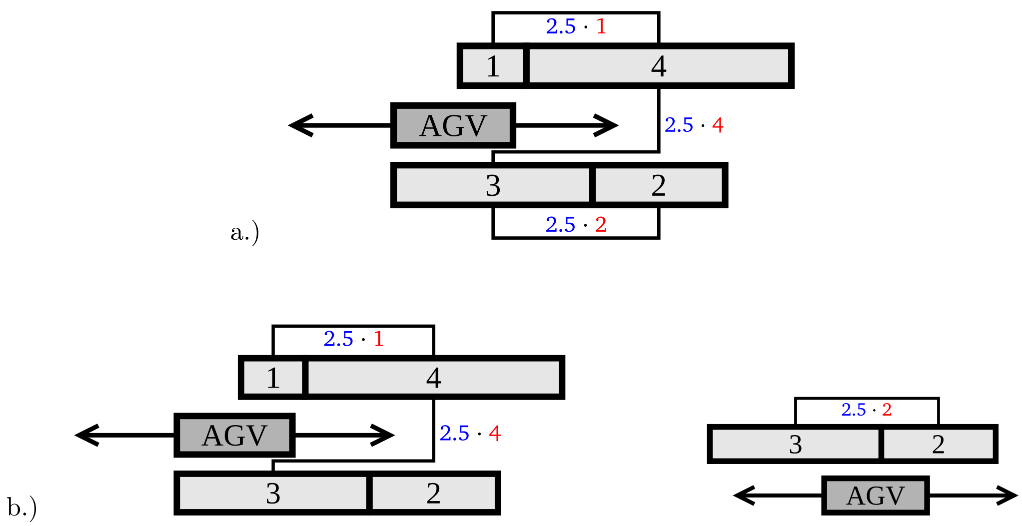


Figure 14.6: We are given the following data: $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4,\ c_{12} = 2,\ c_{13} = 8,\ c_{14} = 1,\ c_{24} = 5,\ c_{25} = 1,\ c_{34} = 4$. In a.) we display the optimal double-row layout with corresponding costs of 17.5. In b.) we depict the optimal single-row layouts needed for computing the Single-Row bound with value $12.5 + 5 = 17.5$.

Let us give another, more involved illustrating example: Now we are given 3 rows and 5 departments with lengths $\ell_1 = 1,\ \ell_2 = 2,\ \ell_3 = 3,\ \ell_4 = 4,\ \ell_5 = 5$ and connectivities $c_{12} = 2,\ c_{13} = 8,\ c_{14} = 1,\ c_{24} = 5,\ c_{25} = 1,\ c_{34} = c_{35} = 4$. Figure 14.7 illustrates the optimal multi-row layout with costs of 20 and the 3 corresponding (SRFLPs) needed to determine the Single-Row bound with value 16. As these two values

[4] If the highest connectivity is non-unique, we take the one with the smallest first index.

are obtained from different row assignments, we depict the optimal multi-row layout and the (SRFLPs) corresponding to the Single-Row bound for both row assignments. Note that for the same instance solved as a double-row layout problem the costs of the optimal layout and the Single-Row bound are again equal (both 35).

In the following theorem we give a technical argument showing that the Single-Row bound is a valid lower bound for the (MRFLP).

Theorem 14.6 *The Single-Row bound is a lower bound to the optimal* (MRFLP) *solution value.*

Proof. There exists an optimal solution without space for any (SRFLP). For a given row assignment the sum of the optimal (SRFLP) solution values with the original pairwise connectivities c_{ij} is equivalent to the optimal solution value of the (MRFLP) if all inter-row connectivities are 0. Hence this sum yields a lower bound for the (MRFLP) as $c_{ij} \geq 0$ holds for $i, j \in [n],\ i < j$.

We additionally add the term $\sum_{k=r(i)+1}^{m} \sum_{\substack{l \in [n] \\ r(l)=k}} d_{ijl}$ to the connectivities c_{ij}, see (14.16). In this way we define the inter-row connectivities as if each department from a different row is placed on its ideal position, hence making two relaxations with respect to the optimal (MRFLP) layout:

1. We disregard possible overlapping and
2. we allow different placements of the same department when considering distinct rows.

See Figure 14.7 c.) for an example where both 1 and 2 occur. Hence the definition of the new connectivities $\bar{c}_{ij}$ guarantees that the associated optimal (SRFLP) solution values are lower bounds to the intra-row costs of row $r(i)$ plus the inter-row costs between row $r(i)$ and row $k, \in \{r(i)+1, \ldots, m\}$ for the given row assignment. Hence the Single-Row bound is a lower bound to the costs of the optimal multi-row layout. □ □

In our computational experiments (see Subsection 14.5.3) we use the Single-Row bound to compute quite reasonably tight global lower bounds to the (MRFLP) for instances with up to 5 rows and 16 departments.

14.5 Computational Experiments

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. To generate (COP) instances we use layout benchmark instances from the literature by simply choosing one department i as checkpoint and deducing the (COP) connectivities from the pairwise connectivities as follows: $c_j = c_{ij},\ j \in [n],\ j \neq i$. This definition ensures that the (COP) instance and the corresponding (SRFLP), (PROP) and (MRFLP) instances have the same size, i.e. the same number of ordering variables in our SDP model. All the instances can be downloaded from `http://anjos.mgi.polymtl.ca/flplib`. Let us also mention that we can round the non-constant part of the objective functions of all problems expect the (MRFLP) to the nearest integer. For the (SRFLP) and the (PROP) 0.5 can only occur in the constant term K of the objective function. For the (MRFLP) we can round the lower bound only to 0.5 as the constant term is different for distinct row assignments.

First we review the best known practical results for the (SRFLP), the (PROP), the (MRFLP) and the (LOP) in Subsection 14.5.1. Then we apply our algorithmic framework to the (COP) and summarize the results obtained in Subsection 14.5.2. Finally we provide evidence of the practical benefit of the Single-Row bound in Subsection 14.5.3.

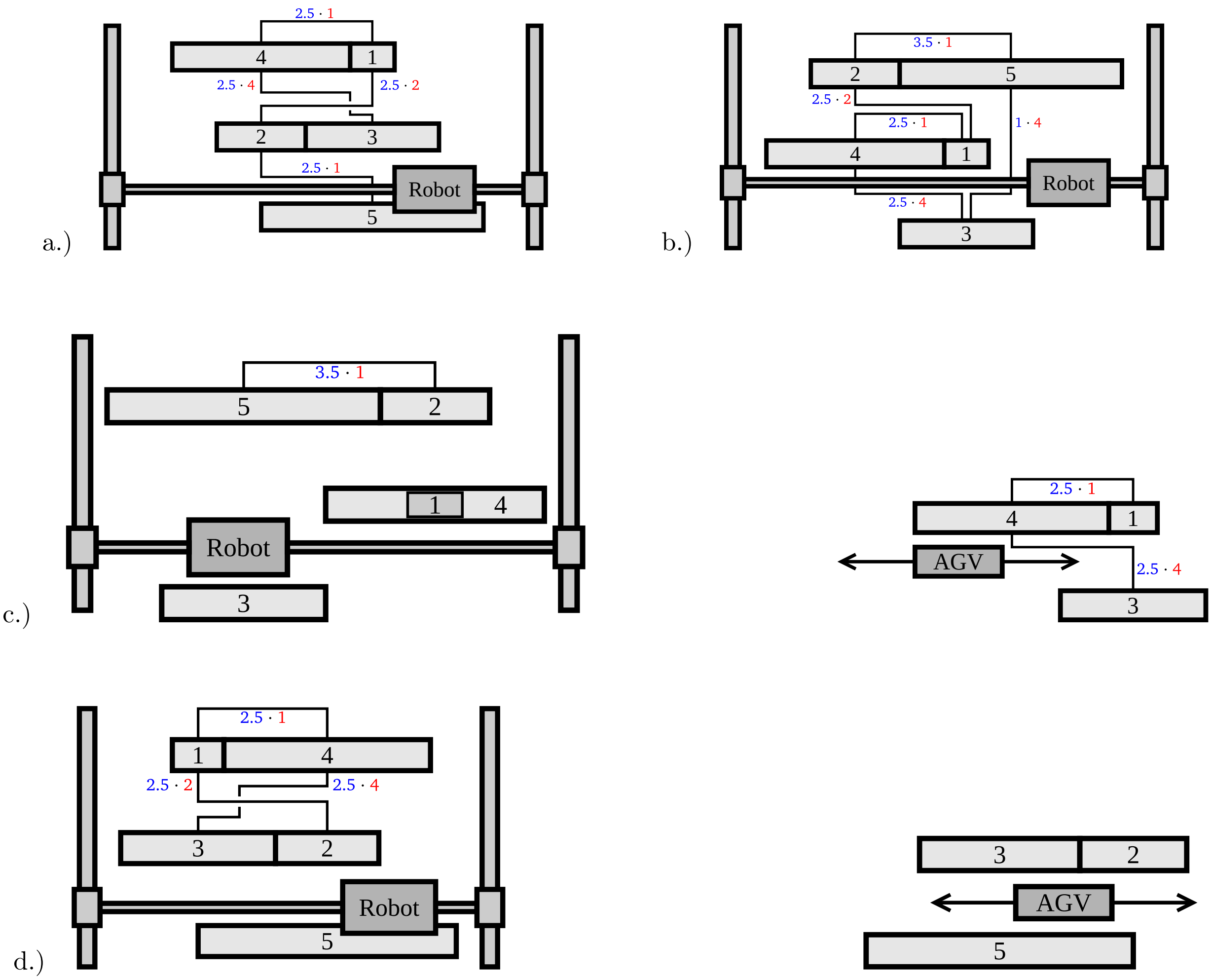


Figure 14.7: We are given the following data: $\ell_1 = 1$, $\ell_2 = 2$, $\ell_3 = 3$, $\ell_4 = 4$, $\ell_5 = 5$, $c_{12} = 2$, $c_{13} = 8$, $c_{14} = 1$, $c_{24} = 5$, $c_{25} = 1$, $c_{34} = c_{35} = 4$. In a.) we display the optimal multi-row layout with corresponding costs of $2.5 \cdot 1 + 2.5 \cdot 4 + 2.5 \cdot 2 + 2.5 \cdot 1 = 20$. And in b.) we depict the optimal multi-row layout for the row assignment for which the overall Single-Row bound is obtained. The corresponding costs are $3.5 \cdot 1 + 2.5 \cdot 2 + 2.5 \cdot 1 + 2.5 \cdot 4 + 1 \cdot 4 = 25$. In c.) we depict the single-row layouts needed for computing the Single-Row bound with value $3.5 + 12.5 = 16$ that is obtained for the row assignment from b.). Finally in d.) we display the single-row layouts needed for computing the Single-Row bound corresponding to the row assignment from a.) with value $17.5 + 0 = 17.5$.

14.5.1 Review: (LOP), (SRFLP), (PROP) and (MRFLP) Instances and Results

Let us start with giving the characteristics and the optimal (SRFLP), (PROP) and (MRFLP) solutions of the instances considered in our computational study in Tables 14.1–14.3.

The (SRFLP) results are provided by using the following algorithmic framework on our machine: To make ($\mathtt{SDP}_{\mathtt{standard}}$) computationally tractable Hungerländer and Rendl [169] suggest to deal with the triangle inequalities (14.8) and LS-cuts (14.10) through Lagrangian duality and then they use the bundle method [103] in conjunction with interior point methods to solve the resulting optimization problem with non-smooth objective function f. The bundle method iteratively evaluates f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to solving an SDP with the constraints $\operatorname{diag}(Z) = e$ and $Z \succcurlyeq 0$ that can be solved efficiently by using interior point methods. Finally we obtain an approximate minimizer of f that is guaranteed to yield a lower bound to the optimal solution of ($\mathtt{SDP}_{\mathtt{standard}}$). Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations to control the overall computational effort. This limitation of the number of function evaluations leaves some room for further incremental improvement. For a detailed description on how to tailor the bundle method to ordering problems we refer to [168].

The fractional solutions of ($\mathtt{SDP}_{\mathtt{standard}}$) constitute lower bounds for the exact SDP formulation of the (SRFLP). By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe feasible layouts of the departments. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum. For more details on global optimization approaches for the (SRFLP) we refer to the survey article by Anjos and Liers [19].

Let us also give a short outline of the heuristic mentioned above. We apply the hyperplane rounding algorithm of Goemans-Williamson [121] to the fractional solution of ($\mathtt{SDP}_{\mathtt{standard}}$), take the resulting vector $\overline{y}$ and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1. \tag{14.17}$$

that are well-known [296, 306] to ensure feasible orderings of n objects. Computational experiments demonstrated that repair strategies of this type are not as critical as one might assume. For example, in multi-level crossing minimization this SDP rounding heuristic clearly dominates traditional heuristic approaches [64] and for the (SRFLP) it is competitive with the best known local search methods and evolutionary algorithms [80, 188, 267].

To tackle the (PROP) with an SDP approach we suggest to apply the algorithmic framework described above to an SDP relaxation that differs only in two minor points from ($\mathtt{SDP}_{\mathtt{cheap}}$): we use another objective function and a generalization of the triangle inequalities (14.14). Additionally we tried to solve a relaxation of type ($\mathtt{SDP}_{\mathtt{full}}$) instead of one of type ($\mathtt{SDP}_{\mathtt{cheap}}$) which resulted in slightly improved lower bounds but tremendously larger running times because many of the $O(n^6)$ triangle inequalities and $O(n^5)$ LS-cuts are active at the optimum. E.g. even for the small instance *S11* with only 11 departments and an optimal solution value of 3895.5 we need more than 10 minutes to compute ($\mathtt{SDP}_{\mathtt{full}}$), we have to consider 2745 constraints and the lower bound is 3600.5. For comparison we need only 1 second to compute ($\mathtt{SDP}_{\mathtt{cheap}}$) as we have to consider only 113 constraints and the lower bound obtained is 3563.5. We will observe a very similar effect when considering different SDP relaxations for the (COP) below. Finally the rounding strategy for the (SRFLP) can be easily adapted for the (PROP) by not only considering (14.17) but also (14.13). For a more detailed description of our SDP approach to the (PROP) we refer to [159].

To further extend our algorithmic framework from the (PROP) to the (MRFLP) we have to apply the exact approach described above to several SDP relaxations of type ($\mathtt{SDP}_{\mathtt{cheap}}$), one for each possible assignment of the departments to the rows, e.g. $2^{n-1} - 1$ assignments in the double-row case. From the bounds obtained, we can deduce global lower and upper bounds that are the minima of all lower and upper bounds respectively. The SDP relaxations considered are substantially larger for the same number

Instance	Source	Size (n)	(SRFLP) Best lower bound	Best layout	Gap (%)	Time (sec) [169]
H_5	[151]	5	800		0	0
Rand_5	[164]	5	147.5		0	0
H_6	[151]	6	1480		0	0
Rand_6	[164]	6	420		0	0
H_7	[151]	7	3680		0	1
Rand_7	[164]	7	344		0	0
H_8	[151]	8	4725		0	0
Rand_8	[164]	8	382		0	1
S8	[283]	8	801		0	1
SH8	[283]	8	2324.5		0	2
Rand_9	[164]	9	1024.5		0	2
S9	[283]	9	2469.5		0	1
SH9	[283]	9	4695.5		0	9
Rand_10	[164]	10	1697		0	3
S10	[283]	10	2781.5		0	1
Rand_11	[164]	11	1564		0	2
S11	[283]	11	6933.5		0	1
H_12	[151]	12	17945		0	8
Rand_12	[164]	12	2088		0	8
Rand_13	[164]	13	3101.5		0	8
Rand_14	[164]	14	3653		0	18
P15	[6]	15	6305		0	20
Rand_15	[164]	15	5345.5		0	19
Rand_16	[164]	16	6212.5		0	13
P17	[7]	17	9254		0	35
P18	[7]	18	10650.5		0	33
H_20	[151]	20	15549		0	54
N25_05	[20]	25	15623		0	3:31
H_30	[151]	30	44965		0	14:17
N30_05	[20]	30	115268		0	18:30
Am33_03	[8]	33	69942.5		0	36:33
Am35_03	[8]	35	69002.5		0	53:14
ste36.5	[21]	36	91651.5		0	17:58
N40_5	[169]	40	103009		0	2:20:09
sko42-5	[21]	42	248238.5		0	1:08:42
sko49-5	[21]	49	666130	666143	0.002	9:30:22
sko56-5	[21]	56	591915.5	592335.5	0.07	17:46:46
AKV-60-05	[22]	60	318792	318805	0.004	12:39:37
sko64-5	[21]	64	501059.5	502063.5	0.20	13:53:04
AKV-70-05	[22]	70	4213774.5	4218002.5	0.10	28:16:05
sko72-5	[21]	72	426224.5	430288.5	0.95	31:39:43
AKV-75-05	[22]	75	1786154	1791469	0.30	41:10:37
AKV-80-05	[22]	80	1585491	1590847	0.34	58:30:30
sko81-5	[21]	81	1293905	1311166	1.33	58:59:28
sko100-5	[21]	100	1021584.5	1040929.5	1.89	201:29:27
AL110-3	[13]	110	2211923.5	2234854.5	1.04	370:36:41

Table 14.1: Characteristics and best known (SRFLP) results for instances with between 5 and 110 departments. The results were obtained by the exact SDP approach described above applied to ($\mathtt{SDP}_{\mathtt{standard}}$) on our machine. The bundle method is restricted to 500 function evaluations for $42 \leq n \leq 56$ and 250 function evaluations for $n \geq 60$. The running times are given in sec or min:sec or in h:min:sec respectively. The results are partly adapted from [169].

Instance	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average number of active inequalities	Average time
P17	4501.5	4722	2.68	10.05	5.82	265.7	41
P18	5153	5503.5	3.85	11.51	8.36	298.6	1:07
H_20	7520	8046	4.97	10.86	7.70	400.7	4:03
N25_05	7385	7986	5.62	11.56	8.79	659.1	23:06
H_30	21028	22848	6.64	13.74	9.63	1057.6	2:12:30
N30_05	53854	58221	5.89	13.46	9.27	1201.3	2:37:19
Am33_03	32847	35904.5	7.59	13.88	9.31	1580.7	5:52:13
Am35_03	32142	35273	8.64	12.89	9.74	1666.3	10:27:58
ste36.5	44786.5	46794.5	1.36	5.54	3.66	1633.6	12:40:15

Table 14.2: (PROP) results for an SDP relaxation of type ($\texttt{SDP}_{\texttt{cheap}}$) and given row assignments. The results were obtained on our machine by using the bundle method in conjunction with Sedumi, for details see Subsection 14.5.2 below. The results are averages over 10 row assignments. For the heuristically selected row assignments the total row lengths are very close. "Lower bound" gives the worst lower bound over the 10 instances and "Upper bound" states the best upper bound over the 10 instances. The running times are given in sec or min:sec or in h:min:sec respectively. The results are adapted from [169].

of departments as the possibility of spaces has to be incorporated in the model.

		Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems				
Instance	Number of rows	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Average lower bound	Average upper bound	Total time (sec)
H_5	2	350	350	0	1.94	0.20	501.0	502.0	5:52
Rand_5	2	52.5	52.5	0	10.00	1.21	88.0	89.1	15:44
H_6	2	640	640	0	14.63	2.22	938.1	957.7	1:50:10
Rand_6	2	188	190.5	1.33	8.43	2.37	253.9	259.5	6:54:11
H_5	3	175	210	20.00	25.71	6.58	330.0	348.4	2:40:10
Rand_5	3	18	23.5	30.56	39.06	11.48	54.7	59.7	5:19:46
H_6	3	315	350	11.11	39.68	10.96	657.0	723.9	51:18:05
Rand_6	3	96.0	100.5	4.69	32.08	8.28	178.1	191.7	81:23:04
H_5	5	0	85.0	-	-	-	0	85.0	41:11
Rand_5	5	0	31.5	-	-	-	0	31.5	2:00:09
H_6	5	135	190	40.74	173.33	67.17	268.7	425.3	83:59:40
Rand_6	5	24	61	154.17	279.17	89.14	67.7	105.8	101:41:15

Table 14.3: Computational results for the (MRFLP) obtained by the exact SDP approach outlined above applied to a relaxation of type ($\texttt{SDP}_{\texttt{cheap}}$) on our machine. The running times are given in min:sec or in h:min:sec respectively. The results are partly adapted from [167].

When applying our algorithmic framework to the (LOP), it is better to work with ($\texttt{SDP}_{\texttt{basic}}$) instead of ($\texttt{SDP}_{\texttt{standard}}$) because the LS-cuts (14.10) do not improve but deteriorate the lower bound provided by the bundle method. Hence the LS-cuts are only beneficial if the semidefinite relaxation is very tight like e.g. the one of the (SRFLP). For a detailed analysis of this effect we refer to [168]. In Table 14.4 we state the results for a few notoriously difficult (LOP) instances. Note that applying the current state-of-the-art ILP Branch-and-Cut algorithm to the paley graphs 31 and 43 results in bounds still beyond 300 respectively 600 after days of branching.

In summary the results given support the following ordering of the four combinatorial optimization problems with respect to their difficulty for our SDP approach: The easiest problem is the (SRFLP) as no instances are known with less than 42 departments that cannot be solved to optimality and also the gaps for larger instances are the smallest ones. Then comes the (LOP) followed by the (PROP) that is the first one to contain inter-row costs. Finally the most difficult problem is clearly the (MRFLP) that becomes even more difficult for a larger number of rows, where the fraction of inter-row costs (compared to inner-row

graph	n	best known solution	LP-bound	LP-gap	(SDP$_{\texttt{cheap}}$)	SDP-gap	SDP-time (sec)
pal31	31	285	310	8.77	297	4.21	17:45
pal43	43	543	602	10.87	569	4.79	1:24:09
pal55	55	1045	1084	3.73	1049	0.38	7:02:01
p50-10	50	43575	45346	4.06	44097	1.20	3:55:47
N-t1d100.01	100	106852	114468	7.13	110314	3.24	22:12:02

Table 14.4: Characteristics and best known root bounds for notoriously difficult (LOP) instances. The results were obtained by the exact SDP approach described above on our machine applied to (SDP$_{\texttt{basic}}$). The running times are given in min:sec or in h:min:sec respectively. The results are partly adapted from [168].

costs) in the objective function increases.

For ILP approaches the (LOP) is (in general) easier than the (SRFLP) as it has an integer programming formulation in $\binom{n}{2}$ binary ordering variables while the (SRFLP) has an integer programming formulation in $\binom{n}{3}$ binary betweenness variables. The (PROP) and the (DRFLP) are also the most difficult problems for the best ILP approaches. They are modelled as MIPs using continuous distance variables. Contrary to that the SDP approach models all problems on the same variables and also exploits the combinatorial structure of the (PROP) and the (MRFLP). Additionally the SDP approach does not use many different types of cutting planes for the various problems or a Branch & Bound framework that somehow covers the difficult of a problem by solving small instances also for extremely weak global bounds simply by enumeration. Hence the SDP approach is the more uniform and allows a fairer comparison of the different problems. Also note that ILP approaches perform worse compared to SDP approaches for (QOPs) with arbitrary products of ordering variables, see [62] for details.

14.5.2 Experiments for the (COP)

In this subsection we analyse which of the SDP relaxations from Section 14.3 is the best for tackling the (COP) and we aim for comparing the (COP) with the problems discussed in the previous subsection.

If we solve an SDP relaxation, e.g. (SDP$_{\texttt{full}}$), exactly in this subsection, we suggest the following algorithmic approach for reasons of efficiency: We use 10 function evaluations of the bundle method applied to (SDP$_{\texttt{full}}$) to obtain an initial set of constraints to add to the relaxation (SDP$_{\texttt{basic}}$). We then solve the resulting relaxation using Sedumi [289]; add all violated inequality constraints from (SDP$_{\texttt{full}}$); solve again using Sedumi; and repeat this process until no more violations are found. If we solve (SDP$_{\texttt{full}}$) directly instead, the running times are at least one order of magnitude slower.

To obtain upper bounds we slightly adapt and reuse the rounding heuristic for the (PROP) described in the previous subsection. Note that the performance of the rounding heuristic depends of course on the quality of the fractional starting solutions provided. If these starting solutions are already off a few percent (like it will be the case for the (COP)) the obtained layouts are good but not optimal in general.

Let us start with solving (SDP$_{\texttt{full}}$) exactly for the small instances from Table 14.3. We choose each of the departments as checkpoint to incorporate the whole data of the associated layout instances in our experiments and hence allow a preferably fair comparison with the results from the previous subsection. In Table 14.5 we state average results and outliers for our runs.

Comparing the results obtained with the (DRFLP) results from Table 14.3 we observe the following: on the one hand the (COP) can be solved a lot faster than the (DRFLP) but on the other hand the global bounds obtained are clearly worse for the (COP). This is somehow surprising as the (COP) is a very special case of the (DRFLP). Hence the (COP) contains one very hard part of the (DRFLP), namely optimizing the inter-row costs.

Next we examine the efficiency (impact on computation time) and effectiveness (impact on bound quality) of the various constraint classes. The aim is to find an SDP relaxation that contains the most

	Average results (over all n possible checkpoints)				Marginal results (over all n possible checkpoints)	
Instance	Average lower bound	Average upper bound	Average gap (%)	Total time (sec)	Largest gap (%)	Smallest gap (%)
H_5	143.7	148.0	3.21	5	9.00	0.00
Rand_5	27.6	29.2	10.04	4	44.44	0.00
H_6	216.5	238.0	9.81	9	18.07	0.00
Rand_6	65.0	65.9	1.84	6	9.18	0.00

Table 14.5: First computational results for small (COP) instances. The gaps were obtained by solving our strongest semidefinite relaxation ($\texttt{SDP}_{\texttt{full}}$) exactly using Sedumi.

important constraints with respect to bound quality and still can be solved for instances of reasonable size. For the remaining experiments in this section we choose w.l.o.g. department 1 of the respective layout instances as checkpoint.

Our starting relaxation is ($\texttt{SDP}_{\texttt{basic}}$). This model reflects the fundamental structure of the original problem in the sense that it would suffice to obtain the optimal solution if we additionally imposed integrality conditions on the ordering variables (see Theorem 14.4). For ($\texttt{SDP}_{\texttt{basic}}$) the lower bound always 0 and hence the gaps obtained are always ∞. If we add the constraint set $\mathcal{DV}$ and hence consider relaxation ($\texttt{SDP}_{\texttt{cheap}}$), the computation times are rather the same but we obtain reasonable gaps, see the first column of Table 14.6. Hence $\mathcal{DV}$ is clearly both effective and efficient.

Next we examine the effects of adding $\mathcal{LS}$ and $\mathcal{M}$ to ($\texttt{SDP}_{\texttt{cheap}}$). The results are also summarized in Table 14.6. Note that we stopped the computations of the relaxations ($\texttt{SDP}_{\texttt{cheap}}$) $\cap\ \mathcal{M}$ and ($\texttt{SDP}_{\texttt{full}}$) for the instances H_12 and $Rand$_12 after one day without result.

Instance	($\texttt{SDP}_{\texttt{cheap}}$)			($\texttt{SDP}_{\texttt{cheap}}$) $\cap\ \mathcal{LS}$		
	Time	Gap (%)	Number of active inequalities	Time	Gap (%)	Number of active inequalities
S_9	1	16.8	39	2	16.8	222
Rand_9	1	19.4	39	2	19.4	145
S_10	1	21.4	41	3	21.4	889
Rand_10	1	20.3	52	2	20.3	135
S_11	1	18.3	45	2	18.3	104
Rand_11	1	16.3	59	2	16.3	189
H_12	3	12.9	73	7:06	12.9	1852
Rand_12	1	18.9	83	3	18.9	139
Instance	($\texttt{SDP}_{\texttt{cheap}}$) $\cap\ \mathcal{M}$			($\texttt{SDP}_{\texttt{full}}$)		
	Time	Gap (%)	Number of active inequalities	Time	Gap (%)	Number of active inequalities
S_9	1:17	16.4	1483	2:10	16.4	1696
Rand_9	6:41	18.5	2750	13:31	18.5	3470
S_10	5:10	21.4	2044	10:22	21.4	2611
Rand_10	4:30	20.0	2140	7:54	20.0	2571
S_11	32:41	18.3	4003	20:14	18.3	3495
Rand_11	29:31	16.2	3184	29:29	16.2	3752

Table 14.6: Study of the impact of constraint classes $\mathcal{M}$ and $\mathcal{LS}$. The running times are given in sec or min:sec respectively.

We observe that $\mathcal{M}$ is slightly effective but really inefficient and $\mathcal{LS}$ is more or less efficient but totally ineffective. In summary the above results strongly suggest that ($\texttt{SDP}_{\texttt{cheap}}$) provides the best tradeoff between computational time and quality of the bounds. Thus next we solve ($\mathtt{SDP_{cheap}}$) exactly for larger instances and summarize the results in Table 14.7.

Instance	Lower bound	Upper bound	Gap (%)	Time	Number of active inequalities
P15	175	189	8.0	2	93
P17	549.5	675.5	22.9	5	145
P18	549.5	679.5	22.9	7	174
H_20	579	710	22.6	34	208
N25_05	306	369	20.6	2:24	307
H_30	1175	1452	23.6	22:53	401
N30_05	2640.5	3206.5	21.4	25:13	415
Am33_03	1577.5	1893.5	20.0	57:33	529
Am35_03	1705.5	2124.5	24.6	1:39:59	574
ste36.5	1431	1475	3.1	1:13:14	511
N40_5	2355	2749	16.7	12:21:57	831
sko42-5	3148	3706	17.7	21:50:14	929

Table 14.7: Computational experiments for larger (COP) instances. The results were obtained by solving ($\mathtt{SDP_{cheap}}$) exactly using Sedumi. The running times are given in sec or min:sec or in h:min:sec respectively.

Let us compare the above results with the ones for the (PROP) from Table 14.2: We used the exactly same algorithmic approach and the same type of SDP relaxation. The running times are smaller for the (COP) because of the smaller number of active inequalities at the optimal solution. The gaps are larger for the (COP) which supports our claim that optimizing the inner-row costs is easier than optimizing the inter-row costs because the (PROP) can be regarded as a combination of two (SRFLPs) and multiple, connected (COPs). For tackling even larger instances we can use only the bundle method without Sedumi like for the (SRFLP) results in Table 14.1. We refrain from stating detailed results and just briefly summarize our findings: The best relaxation is again ($\mathtt{SDP_{cheap}}$), the running times are in the same range as for the (SRFLP) (hence we could obtain reasonable bounds for instances with up to 110 departments within 400 hours of computing time) but the gaps are again clearly larger due to the difficulty of the (COP).

Finally note that we can easily extend our approach in two directions. Firstly we can easily combine the (COP) with the (SRFLP) in our algorithmic framework, i.e. we allow additional pairwise connectivities between departments in a (COP) instance. To tackle such extended instances we just have to add the (SRFLP) objective function to the objective function of our semidefinite relaxation ($\mathtt{SDP_{cheap}}$). Experiments on such combined instances yield the outcomes one might already expect: If we increase the fraction of the inter-row costs in the total objective function, the gaps decrease accordingly. Secondly we can consider more than one checkpoint. To clarify this suggested generalization we reuse the toy example from the introduction and add a second checkpoint, see Figure 14.8 for details. Rerunning the above experiments for two and more checkpoints yields the following main findings:

1. Considering another checkpoint is as expensive as considering another department, i.e. modelling instances with $[n-1]$ departments and one checkpoint needs as many variables, viz. $\left(\binom{n}{2}+1\right)^2-1$, as modelling instances with $[n-2]$ departments and two checkpoints and so on. Solving problems with the same number of variables requires more or less the same amount of running time.

2. The gaps obtained are also in the same range for problems with the same number of variables, independent of the number of checkpoints regarded.

3. Additional attention is needed when rounding the lower bounds if non-centered checkpoints are considered, i.e. in general we can round the non-constant part of the objective function only to 0.5 instead of rounding it to the nearest integer.

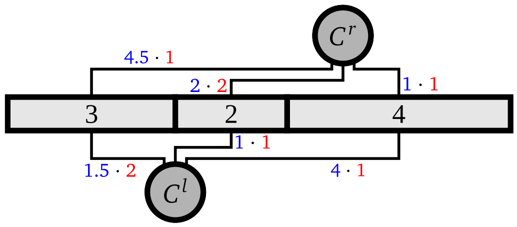


Figure 14.8: We are given 3 departments and 2 checkpoints c^l and c^r located at one third and two third of the sum of all department lengths L respectively. The corresponding connectivities are $c_2^l = c_4^l = c_3^r = c_4^r = 1$, $c_3^l = c_2^r = 2$. We display the optimal layout with the associated costs of $1.5{\cdot}2+1{\cdot}1+4{\cdot}1+4.5{\cdot}1+2{\cdot}2+1{\cdot}1 = 17.5$.

14.5.3 Using the Single-Row Bound to Obtain Global Bounds for (MRFLP) Instances

In this subsection we compute the Single-Row bound (see Theorem 14.6) for several benchmark instances from the literature. Let us recall that to date

1. only (DRFLP) instances with ≤ 12 departments can be solved to optimality,
2. for larger (DRFLP) instances there are no reasonable global bounds known,
3. for the (MRFLP) with more than 2 rows global bounds are only known for extremely small instances with ≤ 6 departments.

As the (MRFLP) has many applications global bounds for larger instances of at least reasonable quality are desirable. The computational results from the previous subsections suggest that considering space and optimizing over inter-row costs are the two most difficult aspects of the (MRFLP). The Single-Row bound introduced in Section 14.4 allows to incorporate these two aspects in a rash of (SRFLPs). To obtain (strong) feasible layouts and hence upper bounds we propose a simple heuristic building on the Single-Row bound below. We will demonstrate in this subsection that our approach yields global bounds of reasonable quality for (DRFLP) instances with up to 16 departments and for (MRFLP) instances with more than 2 rows and up to 12 departments.

We compute the Single-Row bound for various layout instances and summarize the results in Table 14.8 for the (DRFLP) and in Table 14.9 for the (MRFLP) with 3 and 5 rows. We use both the relaxations ($\mathtt{SDP}_{\mathtt{trivial}}$) and ($\mathtt{SDP}_{\mathtt{standard}}$) to solve the various (SRFLPs). We solve ($\mathtt{SDP}_{\mathtt{trivial}}$) for all instances and ($\mathtt{SDP}_{\mathtt{standard}}$) for (DRFLP) instances with ≤ 10 departments. For larger instances we solve ($\mathtt{SDP}_{\mathtt{standard}}$) with the help of the bundle method restricted to 100 function evaluations to speed up the computations.

As we enumerate over all row assignments when computing the Single-Row bound, we can use the lower bounds of the individual single row assignments to detect strong feasible layouts. In particular we suggest to take the ten instances with the lowest lower bounds and add four departments with lengths $2^k, k = -1, 0, 1, 2$ and no connectivities that serve as spaces to the shortest row. Then we solve the (kPROP) for the particular row assignments using a semidefinite relaxation of type ($\mathtt{SDP}_{\mathtt{trivial}}$) to obtain feasible layouts through our rounding heuristic.

The Single-Row bound can be computed very quickly and yields global bounds of reasonable quality for instances with up to 16 departments when considering 2 rows, for instances with up to 12 departments when considering 3 rows and for instances with up to 11 departments when considering 5 rows. For the stronger relaxation ($\mathtt{SDP}_{\mathtt{standard}}$) the lower bounds are slightly higher and in the majority of cases coincide with the optimal (SRFLP) solution (see column "Average (SRFLP) gap"). But solving ($\mathtt{SDP}_{\mathtt{standard}}$) is way more expensive, especially for larger instances. Hence we refrain from using ($\mathtt{SDP}_{\mathtt{standard}}$) in the multi-row case because

			Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems				
Instance	Optimal solution/ Global bound	SDP relaxation	Single-Row bound	Upper bound	Gap (%)	Average lower bound	Average (SRFLP) gap (%)	Average upper bound	Lower bound time	Upper bound time
H_5	350 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	305	350	14.8	472.3	0	454.0	1	7
H_5	350 [167]	$(\mathtt{SDP}_{\mathtt{standard}})$	305	350	14.8	472.3	0	454.0	1	7
Rand_5	52.5 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	49.5	52.5	6.1	87.3	0.1	75.9	1	7
Rand_5	52.5 [167]	$(\mathtt{SDP}_{\mathtt{standard}})$	49.5	52.5	6.1	87.4	0	75.9	1	7
H_6	640 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	530	640	20.8	880.0	0	763.5	4	11
H_6	640 [167]	$(\mathtt{SDP}_{\mathtt{standard}})$	530	640	20.8	880.0	0	763.5	4	11
Rand_6	188–190.5 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	145.5	190.5	30.9	250.2	0.2	208.3	5	10
Rand_6	188–190.5 [167]	$(\mathtt{SDP}_{\mathtt{standard}})$	145.5	190.5	30.9	250.4	0	208.3	7	10
H_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1495	1660	11.0	2164.9	0.4	1817.5	15	16
H_7	-	$(\mathtt{SDP}_{\mathtt{standard}})$	1495	1660	11.0	2168.7	0	1817.5	20	16
Rand_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	116.5	159	36.5	201.2	0.1	178.9	11	15
Rand_7	-	$(\mathtt{SDP}_{\mathtt{standard}})$	116.5	159	36.5	201.3	0	178.9	12	15
H_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1630	2325	42.6	2683.2	1.2	2485	38	28
H_8	-	$(\mathtt{SDP}_{\mathtt{standard}})$	1630	2325	42.6	2694.9	0	2485	47	28
S_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	321	398	24.0	452.7	1.1	413.2	51	34
S_8	-	$(\mathtt{SDP}_{\mathtt{standard}})$	322	398	23.6	453.8	0	413.2	1:01	34
SH_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1015.5	1123	10.6	1322.2	1.8	1137.0	1:03	30
SH_8	-	$(\mathtt{SDP}_{\mathtt{standard}})$	1028.5	1123	9.2	1324.1	0	1137.0	1:28	30
Rand_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	111.5	197	10.6	209.7	0.5	219.1	40	33
Rand_8	-	$(\mathtt{SDP}_{\mathtt{standard}})$	111.5	197	10.6	210.2	0	219.1	44	33
S_9	1179 [11]	$(\mathtt{SDP}_{\mathtt{trivial}})$	1109.5	1179	6.3	1483.9	4.2	1240.1	2:13	39
S_9	1179 [11]	$(\mathtt{SDP}_{\mathtt{standard}})$	1114	1179	5.8	1488.1	0	1240.1	3:26	39
SH_9	2293 [11]	$(\mathtt{SDP}_{\mathtt{trivial}})$	2012.5	2297.5	14.2	2592.9	5.5	2314.2	2:08	40
SH_9	2293 [11]	$(\mathtt{SDP}_{\mathtt{standard}})$	2014.5	2297.5	14.1	2598.4	0.1	2314.2	30:34	40
Rand_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	364.5	492.5	35.1	579.2	1.9	515.1	1:37	33
Rand_9	-	$(\mathtt{SDP}_{\mathtt{standard}})$	364.5	492.5	35.1	581.1	0.1	515.1	2:24	33
S_10	1351 [11]	$(\mathtt{SDP}_{\mathtt{trivial}})$	1214	1353.5	11.5	1627.0	4.5	1401.9	4:48	54
S_10	1351 [11]	$(\mathtt{SDP}_{\mathtt{standard}})$	1214	1353.5	11.5	1631.5	0.1	1401.9	5:57	54
Rand_10	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	683.5	827	21.0	943.5	5.2	859.1	4:12	1:04
Rand_10	-	$(\mathtt{SDP}_{\mathtt{standard}})$	683.5	827	21.0	948.6	0	859.1	8:52	1:04
S_11	3424.5 [11]	$(\mathtt{SDP}_{\mathtt{trivial}})$	2676.5	3475.5	29.9	3919.6	18.9	3581.7	11:29	1:00
S_11	3424.5 [11]	$(\mathtt{SDP}_{\mathtt{standard}})$	2689.5	3475.5	29.2	3938.5	0.0	3581.7	38:00	1:00
Rand_11	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	556	741	33.3	929.9	4.8	866.6	9:13	1:25
Rand_11	-	$(\mathtt{SDP}_{\mathtt{standard}})$	556	741	33.3	934.7	0.0	866.6	24:33	1:25
H_12	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	6615	9125	37.9	9618.9	8.5	9247.0	24:19	3:35
H_12	-	$(\mathtt{SDP}_{\mathtt{standard}})$	6615	9125	37.9	9702.1	0.1	9247.0	1:06:45	3:35
Rand_12	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	569	976.5	71.6	1124.2	9.2	1094.5	21:06	2:37
Rand_12	-	$(\mathtt{SDP}_{\mathtt{standard}})$	570	976.5	71.3	1133.3	0.0	1094.5	51:33	2:37
Rand_13	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	639.5	1454	127.4	1604.3	11.6	1645.7	54:15	3:01
Rand_13	-	$(\mathtt{SDP}_{\mathtt{standard}})$	642.5	1454	126.3	1615.8	0.0	1645.7	2:11:00	3:01
Rand_14	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1051.5	1458	38.7	1868.2	22.2	1629.8	2:02:14	5:58
Rand_14	-	$(\mathtt{SDP}_{\mathtt{standard}})$	1059.5	1458	37.6	1890.1	0.0	1629.8	8:17:34	5:58
H_15	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	13790	16910	22.6	18998.8	14.2	17385.5	4:37:30	9:38
P_15	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	2589	3357	29.7	3491.3	23.7	3452.1	4:49:57	10:38
Rand_15	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	2174.5	2691.5	23.8	2907.8	32.0	2756.2	4:38:48	9:29
Rand_16	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	2467	3130	26.9	3363.3	35.7	3241.8	10:40:22	16:21

Table 14.8: Global bounds for (DRFLP) instances with up to 16 departments provided by the Single-Row bound in conjunction with a SDP-based rounding heuristic. The running times are given in sec or min:sec or in h:min:sec respectively.

			Global bounds (over all row assignments)			Statistics for the $(3^{n-1} - 2^n + 1)/2$ subproblems				
Instance	Optimal solution/ Global bound	SDP relaxation	Single-Row bound	Upper bound	Gap (%)	Average lower bound	Average (SRFLP) gap (%)	Average upper bound	Lower bound time	Upper bound time
H_5	175–215 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	110	250	127.3	292.8	0	299.5	1	10
Rand_5	18–23.5 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	12	19	58.3	51.6	0	42.0	1	9
H_6	315-350 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	245	440	79.6	581.4	0	566.0	5	14
Rand_6	96–100.5 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	42	100.5	139.3	161.5	0	114.6	6	13
H_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	635	1155	81.9	1466.7	0	1196.0	24	20
Rand_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	48	108	125.0	137	0	122.0	22	18
H_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	745	1430	91.6	1879.4	0.2	1671.0	1:33	47
S_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	193	260	34.7	323.0	0.3	271.5	1:46	34
SH_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	635.5	739.5	16.4	954.2	0.9	756.5	2:20	20
Rand_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	61	142	132.8	147.4	0.1	171.1	1:27	44
S_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	690.5	770	11.5	1079.5	1.9	794.4	7:34	59
SH_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1303.5	1413.5	8.44	1880.4	2.5	1438.4	9:34	30
Rand_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	180	325	80.6	399.0	0.4	338.2	6:19	52
S_10	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	677.5	900.5	32.9	1180.0	1.6	925.8	25:07	1:38
Rand_10	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	370.5	529.5	42.9	678.2	1.3	561.5	24:46	1:07
S_11	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	1753	2335.5	33.2	2833.5	6.4	2491.4	1:40:22	1:01
Rand_11	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	260.5	481.5	84.8	686.1	1.8	598.9	1:27:14	1:18
H_12	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	3945	6075	54.0	6869.6	2.7	6347	5:23:12	1:54
Rand_12	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	289.5	637.5	120.2	815.5	3.0	810.1	5:12:10	1:42

			Global bounds (over all row assignments)			Statistics for the $(5^{n-1} - 4^n + 2 \cdot 3^n - 2^{n+1} + 1)/24$ subproblems				
Instance	Optimal solution/ Global bound	SDP relaxation	Single-Row bound	Upper bound	Gap (%)	Average lower bound	Average (SRFLP) gap (%)	Average upper bound	Lower bound time	Upper bound time
H_5	0–85 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	0	140	∞	0	0	140.0	0	1
Rand_5	0–31.5[167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	0	38.5	∞	0	0	38.5	0	1
H_6	135–190 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	30	190	533.3	177.0	0	395.0	1	10
Rand_6	24–61 [167]	$(\mathtt{SDP}_{\mathtt{trivial}})$	2	83.5	4075.0	45.1	0	101.7	1	13
H_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	225	650	188.9	635.4	0	770.5	8	32
Rand_7	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	6	61	916.7	59.7	0	76.1	8	15
H_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	230	1045	354.4	943.2	0	1256	1:12	48
S_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	64.5	136	110.9	173.1	0	153.2	1:16	29
SH_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	392.5	413.5	5.35	536.1	0.1	425.1	1:29	30
Rand_8	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	15	82	446.7	77.9	0	101.2	1:13	53
S_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	329.5	516.5	56.8	626.1	0.3	530.2	11:23	53
SH_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	742	819.5	10.4	1126.9	0.4	824.1	13:03	49
Rand_9	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	38	161	323.7	205	0.0	178.0	10:14	1:01
S_10	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	305	482	58.0	700.3	0.2	532.5	1:22:10	52
Rand_10	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	118	270	128.8	392.9	0.1	290.7	1:17:14	46
S_11	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	832.5	1350.5	62.2	1741.9	1.0	1434.4	10:12:43	1:26
Rand_11	-	$(\mathtt{SDP}_{\mathtt{trivial}})$	64.5	319.5	395.4	415.5	0.3	372.9	8:52:46	1:14

Table 14.9: Global bounds for (MRFLP) instances with 3 and 5 rows and up to 12 departments provided by the Single-Row bound in conjunction with a SDP-based rounding heuristic. The running times are given in sec or min:sec or in h:min:sec respectively.

1. ($\mathtt{SDP}_{\mathtt{trivial}}$) combines efficiency with effectiveness in a clearly better way and

2. ($\mathtt{SDP}_{\mathtt{trivial}}$) yields even better results (see again column "Average (SRFLP) gap") in the multi-row case because fewer departments are assigned to each row.

Note that the Single-Row bound differs only slightly between ($\mathtt{SDP}_{\mathtt{trivial}}$) and ($\mathtt{SDP}_{\mathtt{standard}}$) (never more than 1.4 %) even tough the average (SRFLP) gap is several times quite large for ($\mathtt{SDP}_{\mathtt{trivial}}$). The reasons is that the row assignments that determine the lower bound are balanced ones, i.e. about the same number of departments are assigned to each row, and such balanced row assignments result in smaller and hence easier corresponding (SRFLPs).

The gaps obtained for the (DRFLP) by the Single-Row bound are in the same range as the gaps of the (COP). This shows both the practical difficulty of the (COP) and the good quality of the global (DRFLP) bounds provided. For a larger number of rows the layout problem gets more difficult (no global bounds were known for instances with 6 or more departments) and accordingly the quality of our bounds decreases. But for a fixed number of rows the quality of our bounds tends to increase for an increasing number of departments. This is because on the one hand the fraction of the inner-row costs in the objective function increases and on the other hand the inner-row costs are more tightly approximated by the Single-row bound.

The upper bound heuristic proposed is quite fast. Additional experiments indicated that considering more row assignments, e.g. 50 instead of 10, and adding more spaces helps to further (slightly) improve the upper bounds obtained. Especially for more rows, adding spaces not only to the shortest row yields considerably stronger layouts for some instances but we refrain from stating detailed results to allow for a fair and clear comparison of the results in Tables 14.8 and 14.9. Comparing the columns "Average lower bound" and "Average upper bound" shows that in fact we identify promising row assignments because the average costs of our layouts over the 10 chosen row assignments are always smaller than the average lower bounds over all row assignments.

14.6 Conclusion

In this paper we proposed new combinatorial optimization problem and an exact algorithm for solving it. The NP-hard Checkpoint Ordering Problem (COP) is both of theoretical and practical interest and has several important relations to other well-studied combinatorial optimization problems. The (COP) can be modeled as a Quadratic Ordering Problem and is very hard to solve in practice. Additionally we suggested a new approach to obtain global bounds for the Multi-Row Facility Layout Problem and showed its merits by applying it to several well-known layout benchmark instances.

The most important step to further improve on the computational performance of the presented approach is to study the polyhedral structure of the (COP) and add promising facets or valid inequalities to our semidefinite relaxations. For obtaining reasonable global bounds for even larger multi-row instances the key step would be to avoid the enumeration over all row assignments by incorporating the Single-Row bound in an appropriate integer linear program.

Part III

Logistics

Chapter 15

A Semidefinite Optimization Approach to the Target Visitation Problem

Authors: Philipp Hungerländer

Abstract: We propose an exact algorithm for the Target Visitation Problem (TVP). The (TVP) is a composition of the Linear Ordering Problem and the Traveling Salesman Problem. It has several military and non-military applications, where two important, often competing factors are the overall distance traveled (e.g. by an unmanned aerial vehicle) and the visiting sequence of the various "targets" or "points of interest". Hence our algorithm can be used to find the optimal visiting sequence of various pre-determined targets.

First we show that the (TVP) is a special Quadratic Position Problem. Building on this finding we propose an exact semidefinite optimization approach to tackle the (TVP) and finally demonstrate its efficiency on a variety of benchmark instances with up to 50 targets.

Keywords: Target Visitation Problem; Linear Ordering Problem; Traveling Salesman Problem; Semidefinite Programming; Global Optimization

15.1 Introduction

The Target Visitation Problem (TVP) was suggested by Grundel and Jeffcoat [125] in 2004. It is a composition of the Linear Ordering Problem (LOP) and the Traveling Salesman Problem (TSP). Hence let us briefly review the most important theoretical and practical aspects of the (LOP), the (TSP) and the (TVP).

The Linear Ordering Problem (LOP). Ordering problems associate to each ordering (or permutation) of the set $[n] := \{1, 2, \dots, n\}$ a profit and the goal is to find an ordering of maximum profit. In the simplest case of the Linear Ordering Problem (LOP), this profit is determined by those pairs $(u, v) \in [n] \times [n]$, where u comes before v in the ordering. Thus in its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that

$$\sum_{\substack{i,j \in [n] \\ i<j}} w_{\pi(i),\pi(j)},$$

is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of nodes i and j either arc (i, j) or arc (j, i), but not both. Then the (LOP) consists of finding an acyclic tournament, i.e.

a tournament without directed cycles, of G of maximum total edge weight. We refer to the book by Martí and Reinelt [219] and the references therein for further material on the (LOP), its variants and various applications and details on many heuristic and exact methods.

The (LOP) is equivalent to the Acyclic Subdigraph Problem (ASP) and the Feedback Arc Set Problem (FASP). It is well known to be NP-hard [112] and it is even NP-hard to approximate (LOP) within the factor $\frac{65}{66}$ [238]. Surprisingly there is not much known about heuristics with approximation guarantees. If all entries of W are nonnegative, a $\frac{1}{2}$-approximation is trivial, but no better polynomial time approximation is known. To narrow this quite large gap $[\frac{1}{2}, \frac{65}{66}]$ is a challenging open problem. Some worthwhile steps have already been taken into that direction: Newman and Vempala [238] showed that widely-studied polyhedral relaxations for the (LOP) cannot be used to approximate the problem within a factor better than $\frac{1}{2}$. Furthermore Newman [237] analyzed a semidefinite programming (SDP) relaxation based on position variables that has an integrality gap of 1.64 (hence smaller than 2) on certain random graphs. Note that this SDP relaxation is in fact a rudimentary version (not suited for reasonable practical bound computations) of the SDP relaxations that we will propose to obtain optimal tours and strong upper bounds for large (TVP) instances in this paper.

SDP is the extension of linear programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. A (primal) SDP can be expressed as the following optimization problem

$$\begin{aligned} &\inf_X \{ \langle C, X \rangle \ : \ X \in \mathcal{P} \}, \\ &\mathcal{P} := \{ X \mid \langle A_i, X \rangle = b_i, \ i \in \{1, \dots, m\}, \ X \succcurlyeq 0 \}, \end{aligned} \tag{SDP}$$

where the data matrices $A_i, \ i \in \{1, \dots, m\}$ and C are symmetric. For further information on SDP we refer to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

The (LOP) arises in a large number of applications in such diverse fields as economy (ranking and voting problems [182, 286] and input-output analysis [60, 155, 205]), sociology (determination of ancestry relationships [119]), graph drawing (one sided crossing minimization [173]), archaeology, scheduling (with precedences [39]), assessment of corruption perception [1] and ranking in sports tournaments. Additionally problems in the context of mathematical psychology and the theory of social choice can be formulated as linear ordering problems, see [104] for a survey.

There are also several problems that are closely related to the (LOP) like the (LOP) with cumulative costs [37, 92] that has a very interesting application in the area of mobile phone telecommunication and the Coupled Task Problem [33] that is concerned with scheduling n jobs each of which consists of two subtasks with associated required delays.

The Coupled Task Problem is concerned with scheduling n jobs each of which consists of two subtasks. Furthermore there is a requirement that between the execution of these subtasks a delay is required. We refer to [33] for an optimization model for this problem that successfully uses linear ordering variables together with additional constraints for modelling the processing times and delays properly.

The current state-of-the-art exact algorithm for the (LOP) is a Integer Linear Programming (ILP) Branch-and-Cut approach that was developed by the working group of Reinelt in Heidelberg and is based on sophisticated cut generation procedures (for details see [219]). It can solve large instances from specific instance classes with up to 150 objects, while it fails on other much smaller instances with only 50 objects. Hungerländer and Rendl [168] proposed an SDP approach (based on products of ordering variables) that proved to be a valuable alternative to the ILP approach for larger and/or notoriously difficult instances. There exist also many heuristics and methaheuristics for the (LOP) and some of them are quite good in finding the optimal solution for large instances in reasonable time. For a recent survey and comparison see [220].

Traveling Salesman Problem (TSP). The (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible tour that visits each city exactly once and returns to the origin city? The NP-hard (TSP) is doubtless the most famous of all (combinatorial) optimization problems with high importance in both operations research and theoretical computer science. We refer to the books [75, 127, 260] and the references therein for extensive material on the (TSP), its variants and various applications, details on many heuristic and exact methods and relevant theoretical results.

The Christofides's algorithm approximates the cost of an optimal symmetric (TSP) tour within the factor 1.5 [69]. In the asymmetric case paths may not exist in both directions or the distances might be different, forming a directed graph. This may be e.g. due to traffic collisions, one-way streets and motorways. In the asymmetric, metric case, only logarithmic performance guarantees are known. The best current algorithm achieves performance ratio $0.814\log(n)$ [176]. It is an open question if a constant factor approximation exists. Results on the even more difficult non Euclidean (TSP) are e.g. discussed in [245].

The (TSP) has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, e.g. in DNA sequencing. In many further applications the (TSP) with additional constraints, such as limited resources or time windows, is of relevance.

Even though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved completely[1] and even problems with millions of cities can be approximated within a small fraction of 1%.

Target Visitation Problem. The (TVP) asks for a permutation $(p_1, p_2, \ldots, p_n)$ of n targets with given pairwise weights $w_{ij},\ i, j \in [n], i \neq j$, and pairwise distances $d_{ij},\ i, j \in [n], i \neq j$, maximizing the objective function

$$\sum_{\substack{i,j\in[n]\\ i<j}} w_{p_i,p_j} - \left(\sum_{i=1}^{n-1} d_{p_i,p_{i+1}} + d_{p_n,,p_1}\right).$$

As the NP-hard (LOP) and (TSP) are special cases of the (TVP), the (TVP) is also NP-hard.

The formulation of the (TVP) was inspired by the use of single unmanned aerial vehicles (UAVs) that have been used increasingly (especially for military purposes) over the last decades. Civilian applications of the (TVP) include environmental assessment, combat search and rescue and disaster relief [125]. In all military and non-military applications two important, often competing factors are the overall distance traveled by the UAV and the visiting sequence of the various "targets" or "points of interest". E.g. in military applications we aim to visit "high chance" waypoints quickly, such that the coalition force may act on the intelligence they receive. Furthermore optimal (TVP) solutions could be of use for cooperative systems [229], where multiple dynamic entities (e.g. UAVs) share information or tasks to accomplish a common, though perhaps not singular, objective.

Despite the manifold and relevant applications of the (TVP), no exact algorithms are available to date for solving it. There exist two heuristics for the (TVP): a very simple one proposed in [125] and a genetic algorithm by Arulselvan et al. [27] that was tested on (TVP) instances with up to 16 targets. Also note that Hildenbrandt et al. [152] are currently working on the first polyhedral study of the (TVP) polytope. Based on their findings they are also developing an exact ILP approach for the (TVP) that has a strong potential to solve large-scale (TVP) instances to optimality.

Now let us give an easy integer programming (IP) formulation of the (TVP) appropriately combining the

[1] The Branch-and-Cut algorithm by Applegate et al. [23] holds the current record, solving an instance with 85,900 cities.

standard IP formulations of the (LOP) and the (TSP): First we introduce the traveling salesman variables

$$x_{i,j} := \begin{cases} 1, & \text{if target } j \text{ is visited immediately after target } i, \\ 0, & \text{otherwise.} \end{cases}$$

and the ordering variables

$$y_{i,j} := \begin{cases} 1 & \text{if target } i \text{ is visited (for the first time)}^2 \text{ before target } j \text{ is visited for the first time,} \\ 0, & \text{otherwise.} \end{cases}$$

And now we can already formulate the (TVP) as the following IP:

$$\max \sum_{\substack{i,j\in[n] \\ i\neq j}} w_{ij}y_{ij} - \sum_{\substack{i,j\in[n] \\ i\neq j}} d_{ij}x_{i,j} \tag{15.1}$$

subject to:

$$\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{j,i} = 1, \qquad \sum_{\substack{j=1 \\ j\neq i}}^{n} x_{i,j} = 1, \quad i \in [n], \tag{15.2}$$

$$\sum_{\substack{i,j\in S \\ i\neq j}} x_{i,j} \leq |S| - 1, \quad \forall S \subset V,\ 2 \leq |S| \leq n, \tag{15.3}$$

$$y_{j,i} = 1 - y_{i,j}, \quad i,j \in [n],\ i < j \tag{15.4}$$

$$0 \leq y_{i,j} + y_{j,k} - y_{i,k} \leq 1,\ i,j,k \in [n],\ i < j < k, \tag{15.5}$$

$$x_{i,j} - y_{i,j} - \frac{1}{n-1}\sum_{\substack{k=1 \\ k\neq i}}^{n} y_{k,i} \leq 0, \quad i,j \in [n],\ i \neq j, \tag{15.6}$$

$$x_{i,j} \in \{0,1\}, \quad y_{i,j} \in \{0,1\}, \quad i,j \in [n],\ i \neq j.$$

Constraints (15.2) and (15.3) are the standard constraints for the asymmetric (TSP) and (15.4) and (15.5) are typically used to model the (LOP). Inequalities (15.6) connect the two problems, where the additional term $\frac{1}{n-1}\sum_{\substack{k=1 \\ k\neq i}}^{n} y_{k,i}$ makes sure that possible edges from location n to location 1 are allowed. Finally the integrality conditions ensure that the variables used are binary. Hence the above optimization problem is obviously an IP formulation of the (TVP). If we replace the integrality conditions by the bound constraints

$$0 \leq x_{i,j} \leq 1, \quad 0 \leq y_{i,j} \leq 1, \quad i,j \in [n],\ i \neq j, \tag{15.7}$$

we obtain a basic linear programming relaxation for the (TVP) that we denote by (LP_{TVP}).

Note that in the previous papers on the (TVP) [27, 125] the starting (and concurrently final) point of the tour was fixed and denoted as base. In this paper we suggest a more general version of the (TVP) where the starting point of the tour is not fixed in general. Nonetheless we can easily introduce a base by special preference settings, i.e. we can fix one target implicitly (through the input data) to be the first "target" (base) in the ordering by choosing the linear ordering weights in the row corresponding to the base high enough.

Toy Example. Next we want to further clarify the workings of the (TVP) with the help of a toy example. We consider 5 targets, where we set target 1 to be the base and hence the first in the ordering. We are

[2] The first target is visited twice such that the tour is closed.

given the following (LOP) weights W for the remaining 4 targets and the (TSP) distances D between all five targets:

$$W = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 6 & 0 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 3 & 5 & 5 & 3 \\ 3 & 0 & 3 & 5 & 5 \\ 5 & 3 & 0 & 3 & 5 \\ 5 & 5 & 3 & 0 & 3 \\ 3 & 5 & 5 & 3 & 0 \end{bmatrix} \tag{15.8}$$

Figure 15.1 illustrates the optimal (TSP) tours a.) and b.), the optimal (LOP) solution c.) and the optimal (TVP) tour d.) (the tours are displayed by grey edges) together with their corresponding (LOP) benefits (red edges and numbers) and (TSP) costs (grey numbers). Hence considering travel distances and target preferences simultaneously leads to optimal tours that can be quite different from the optimal (TSP) and (LOP) solutions.

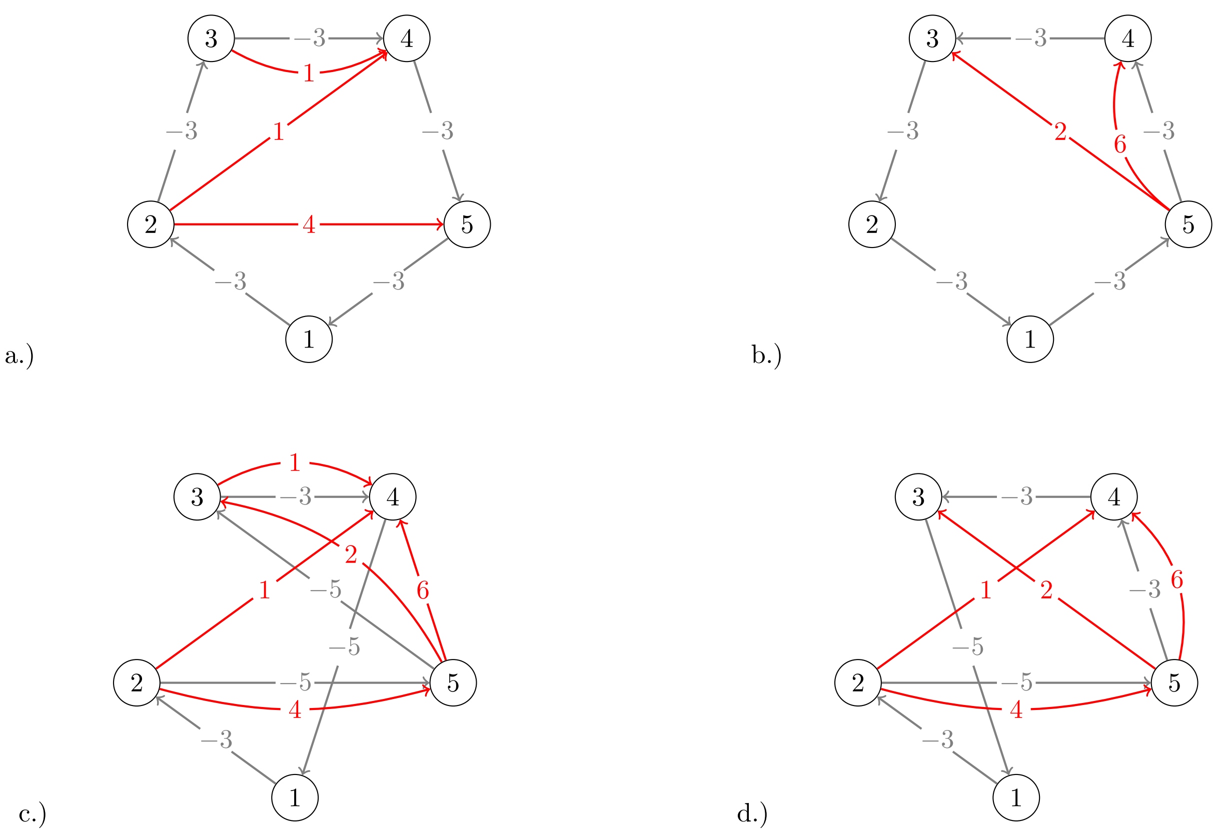


Figure 15.1: We are given 5 targets, where target 1 is the operation base. The input data is provided in (15.8). In a.) and b.) we display the optimal (TSP) tours with objective value -15 and corresponding (TVP) objective values -9 and -7 respectively. In c.) we depict the optimal (LOP) solution with objective value 14 and associated (TVP) objective value -7. Finally in d.) we display the optimal (TVP) tour with corresponding objective value -6.

The above toy example can also be enhanced by a motivational story that we partly adopt from [27]. Suppose that the targets represent a collection of villages in which a sought after terrorist is suspected of hiding out. The operation base is the location of the coalition force. Moreover assume the available intelligence data determining target preferences is summarized in W from (15.8). Such data can e.g. be based on hiding probabilities of the terrorist or the size of communication networks between the villages. In the application described the person of interest moves frequently and hence the intelligence date becomes less accurate quite fast. Now suppose the coalition force has the ability to launch a UAV that visits the targets in a pre-determined tour and returns to the base. During its flight, the UAV is capable of telemetering data back to the coalition force helping to establish the known location of the terrorist they seek. In summary already this simple motivational story demonstrates the importance of considering both distance and visitation sequence when looking for an optimal tour in various (military) applications.

Outline. The main contributions of this paper are the following:

- We show that the `(TVP)` that we define as a combination of the asymmetric, non-Euclidean `(TSP)` and the `(LOP)` is a special Quadratic Position Problem `(QPP)`.
- Building on this finding we propose the first exact (semidefinite) optimization approach to tackle the `(TVP)`.
- We showcase in a computational study that our algorithmic approach yields very promising results on a large variety of benchmark instances with up to 50 targets.

We refer to the companion paper [162] for a more detailed analysis of the `(QPP)`, its polytope and some theoretical properties of our semidefinite relaxations for the `(LOP)` and the `(TSP)` respectively.[3] Herein, we are mainly interested in applying our SDP approach to the `(TVP)` and its associated military applications.

The paper is structured as follows. In Section 15.2 we define and discuss the `(QPP)` and then propose a matrix-based formulation and several semidefinite relaxations for the `(TVP)`. In Section 15.3 we explain how to solve our SDP relaxations for `(TVP)` instances and describe a simple two-opt improvement heuristic for obtaining feasible tours with a high `(TVP)` objective value. In Section 15.4 we show first results for our SDP relaxations on small instances. Based on this comparison, we choose the `(SDP)` relaxation best suited for tackling large scale instances and present associated, promising results in Section 15.5. Section 15.6 concludes the paper.

15.2 A Matrix-Based Formulation and SDP Relaxations of the Target Visitation Problem

SDP methods proved to be a serious alternative to Branch & Cut approaches for ordering problems [62, 168, 169] and also yield interesting theoretical results for the `(TSP)` [77, 86]. As the `(TSP)` variables cannot be expressed as linear-quadratic terms in `(LOP)` variables and vice versa, we are looking for an alternative matrix-based formulation of the `(TVP)` to avoid lifting a vector with all `(LOP)` and `(TSP)` variables. Such a formulation can be obtained by encoding tours with the help of position variables. We will introduce these variables and the corresponding Quadratic Position Problem `(QPP)` in the following subsection. In Subsection 15.2.2 we propose several semidefinite relaxations of the `(QPP)`. Finally in Subsection 15.2.3 we show that the `(TVP)` is a special `(QPP)` and hence we can use the SDP relaxations of the `(QPP)` to obtain upper bounds to the optimal solution value of `(TVPs)`.

[3]We note that the content of the companion paper is fairly disjoint from this paper: it deals with the `(QPP)` from a polyhedral point of view and applies the resulting SDP relaxations to facet defining inequalities of small `(TSP)` and `(LOP)` polytopes to facilitate the theoretical analysis of the relaxations proposed. In the companion paper we do not consider the `(TVP)` nor conduct any large-scale computations. Due to space restrictions we omit the proofs concerning the `(QPP)` and refer to our companion paper for details.

15.2.1 The Quadratic Position Problem (QPP)

Let a vector v of size $n(n-1)$ that contains bivalent position variables be given. We can write the components $v_{(i-1)n+j},\ i \in [n],\ j \in [n-1]$ of v more compactly as

$$v_i^j \in \{-1, 1\}, \quad i \in [n],\ j \in [n-1]. \tag{15.9}$$

With the help of these variables we model the position of object i in an ordering of n objects:

$$v_i^j - v_i^{j-1} = \begin{cases} 2, & \text{if object } i \text{ is assigned to location } j, \\ 0, & \text{if object } i \text{ is not assigned to location } j, \end{cases} \tag{15.10}$$

where we additionally use the parameters

$$v_i^0 := -1, \quad v_i^n := 1, \quad i \in [n]. \tag{15.11}$$

To further clarify this definition, we encode the tours displayed in our toy example (see Figure 15.1) in v, where we separate variables associated to different objects by a vertical dash $|$:

$$\begin{aligned}
&a.)\quad v = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & -1 & 1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}^\top \\
&b.)\quad v = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & -1 & 1 \,|\, -1 & -1 & -1 & -1 & 1 & 1 \,|\, -1 & -1 & -1 & 1 & 1 & 1 \,|\, -1 & -1 & 1 & 1 & 1 & 1 \end{pmatrix}^\top \\
&c.)\quad v = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & -1 & 1 \,|\, -1 & -1 & -1 & 1 & 1 & 1 \end{pmatrix}^\top \\
&d.)\quad v = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & 1 & 1 & 1 & 1 \,|\, -1 & -1 & -1 & -1 & -1 & 1 \,|\, -1 & -1 & -1 & -1 & 1 & 1 \,|\, -1 & -1 & -1 & 1 & 1 & 1 \end{pmatrix}^\top
\end{aligned} \tag{15.12}$$

Let us now propose a (new) combinatorial optimization problem corresponding to the position variables above. An instance of the (QPP) consists of n objects, n consecutive locations and individual and pairwise integer benefits $b_i^k,\ i,k \in [n]$, and $b_{i,j}^{k,l},\ i,j,k,l \in [n],\ i < j,\ k \neq l$. The optimization problem can be written down as

$$\max_{\pi \in \Pi_n} \sum_{\substack{i,j,k,l \in [n] \\ i<j,\ k \neq l}} \left(b_i^k w_i^k(\pi) + b_{i,j}^{k,l} w_i^k(\pi) w_j^l(\pi) \right), \tag{15.13}$$

where Π_n is the set of permutations of the indices $[n]$, defining an assignment of the n objects to the n consecutive locations and $w_i^k(\pi),\ i,k \in [n]$ is 1, iff object i is assigned to location k in the particular assignment $\pi \in \Pi_n$.[4] Note that the Quadratic Assignment Problem (QAP) [186, 211] can be formulated as a (QPP) and vice versa as the quadratic assignment variables $q_i^j,\ i,j \in [n]$ can be written as a difference of position variables: $2q_i^j = v_i^j - v_i^{j-1}$.

To model the (QPP) with the help of the bivalent position variables v introduced above, we ask for monotonicity of variables belonging to the same object:

$$v_i^j \leq v_i^k, \quad i \in [n],\ j,k \in [n-1],\ j < k. \tag{15.14}$$

Additionally we have to assure that a switch from -1 to 1 occurs for different objects at different positions:

$$\sum_{i=1}^{n} v_i^j = 2j - n, \quad j \in [n-1]. \tag{15.15}$$

These two types of constraints already suffice to ensure that all objects are assigned to different locations:

[4] If the benefits are negative, the associated optimization problem in fact minimizes the total costs over all assignments.

Lemma 15.1 *The constraints* (15.14) *and* (15.15) *form, together with the integrality conditions* (15.9)*, a minimal constraint system for modeling the* (QPP) *using* $n(n-1)$ *position variables.*

In summary we are able to rewrite (15.13) as a a mathematical model in v with linear constraints and a quadratic objective function, where we summarize the individual benefits in a vector b and the pairwise benefits in a matrix B:

Theorem 15.2 *Maximizing* $v^\top Bv + v^\top b$ *over* $v \in \{-1,1\}^{n(n-1)}$, (15.14) *and* (15.15) *solves the* (QPP)*.*

By replacing the integrality conditions above with 0-1 bounds we obtain a quadratic programming relaxation of the (QPP). Next we rewrite the objective function in terms of matrices to obtain a matrix-based formulation of the (QPP)

$$\max \left\{ \langle C, Z \rangle \ : \ v \in \{-1,1\}^{n(n-1)},\ v \text{ satisfies (15.14) and (15.15)} \right\}, \tag{15.16}$$

where all position variables and their products are contained in the $(n^2-n+1) \times (n^2-n+1)$ variable matrix $Z := \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix}$ with $V = vv^\top$ and the cost matrix C is given by $C := \begin{pmatrix} 0 & b^\top \\ b & B \end{pmatrix}$.

Finally we can further rewrite the above matrix-based formulation as an SDP, where we denote by e the vector of all ones and by $\mathcal{E}$ the elliptope

$$\mathcal{E} := \{\ Z \ : \ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}.$$

Theorem 15.3 *The semidefinite optimization problem*

$$\max \left\{ \ \langle C, Z \rangle : Z \textit{ satisfies } (15.14) \textit{ and } (15.15)\ ,\ Z \in \mathcal{E},\ v \in \{-1,1\}^{n(n-1)} \right\}, \tag{SDP-QPP}$$

is equivalent to the (QPP)*.*

Next we show how to construct several SDP relaxations from the SDP formulation of the (QPP) proposed above.

15.2.2 Semidefinite Relaxations of the Quadratic Position Problem

The above formulation of the (QPP) contains $n-1$ equalities stated in (15.15). We can use these equations to eliminate $n-1$ position variables. Such a reduction is irrelevant for the formulation but could matter for the values of relaxations of the (QPP). Hence we want to clarify, if we maybe get stronger semidefinite relaxations by working with matrices of order $n(n-1)+1$ or $(n-1)^2+1$? To answer this question, let us recall a result from [168]:

Theorem 15.4 *[168, Remark 2] Let* m *linear equality constraints* $Ay = c$ *be given. If there exists some invertible* $m \times m$ *matrix* B*, we can partition the linear system in the following way*

$$Ay = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} t \\ u \end{bmatrix} = c. \tag{15.17}$$

Then we do not weaken the relaxation by first moving into the subspace given by the equations, and then lifting the problem to matrix space.

In other words, it is equivalent in terms of tightness to eliminate m variables or to lift the m equality constraints in all possible ways to quadratic space. Hence we decide to eliminate $n-1$ variables through (15.15) to reduce the number of variables to $(n-1)^2$ and to avoid additional constraint classes. Of course, this decision has also a disadvantage: We have to consider two versions for several constraint types, namely the cases where object n is considered and not considered.

Dropping the integrality condition on v in (SDP-QPP) and reducing the problem dimension with the help of (15.15), we obtain the following basic semidefinite relaxation of the (QPP)

$$\max \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (15.14)},\ Z_s \in \mathcal{E}\ \}, \tag{SDP$_0$}$$

where the cost and variable matrices with index s consist of the first $(n-1)^2+1$ rows and columns of their larger counterparts from the previous subsection: $C_s = C_{1:(n-1)^2+1}$, $Z_s = Z_{1:(n-1)^2+1}$. Note that the equality constraints (15.15) are implicitly assured by the above semidefinite relaxation. In general (SDP$_0$) gives quite weak upper bounds to the optimal solution value of the (QPP). Hence we will suggest several ways to improve on the tightness of (SDP$_0$).

First we propose $n(n-1)(n-2)$ valid equalities for the (QPP) polytope

$$\mathcal{P}_{\mathtt{QPP}} := \operatorname{conv}\left\{ Z_s \ :\ Z_s = \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix},\ v \in \{-1,1\}^{(n-1)^2} \text{ satisfies (15.14)},\ V = vv^\top \right\},$$

and show that their rank is $n(n-1)(n-2)-1$, i.e. their number minus 1. The following equalities, consisting of 5 different types, can be easily deduced by exploiting the structure of v induced by (15.14) and (15.15):

$$v_i^j - v_i^k - v_{i,i}^{j,k} = -1, \quad i,j,k \in [n-1],\ j<k, \tag{15.18}$$

$$m_k \sum_{i=1}^{n-1} v_i^j + m_j \sum_{i=1}^{n-1} v_i^k - \sum_{h=1}^{n-1}\sum_{i=1}^{n-1} v_{h,i}^{j,k} = m_k m_j, \qquad m_k = 2k-n-1,\ m_j = 2j-n+1,\ j,k \in [n-1],\ j<k, \tag{15.19}$$

$$v_i^1 + v_j^1 + v_{i,j}^{1,1} = -1, \quad i,j \in [n-1],\ i<j, \tag{15.20}$$

$$v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} = 1, \quad i,j \in [n-1],\ i<j, \tag{15.21}$$

$$v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k} = 0, \quad i,j \in [n-1],\ i<j,\ k \in [n-1],\ k \neq 1. \tag{15.22}$$

Lemma 15.5 *The $n(n-1)(n-2)$ equalities* (15.18) – (15.22) *are valid for $\mathcal{P}_{\mathtt{QPP}}$ and have rank $n(n-1)(n-2)-1$.*

We can also show that the the equalities (15.18) suffice together with the implicitly assured linear constraints (15.15) to ensure monotonicity in the variable vector for all objects.

Lemma 15.6 *The monotonicity constraints* (15.14) *are assured by the equalities* (15.18) *and* (15.15) *together with $Z_s \in \mathcal{E}$.*

In summary (SDP$_0$) can be tightened by adding the equalities analyzed above that implicitly ensure (15.14):

$$\max \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (15.18) -- (15.22)},\ Z_s \in \mathcal{E}\ \}, \tag{SDP$_1$}$$

Now we can further improve the relaxation strength of (SDP$_1$) by adding several types of inequalities valid for $\mathcal{P}_{\mathtt{QPP}}$. First we discuss inequalities obtained by exploiting the structure of v induced by (15.14) and (15.15). Secondly we suggest valid inequalities associated to the integrality conditions $v \in \{-1,1\}^{(n-1)^2}$.

Lemma 15.7 *The following inequality constraints hold for $Z_s \in \mathcal{P}_{\mathtt{QPP}}$:*

$$-v_{i,j}^{h,m} - v_{i,j}^{g,l} + v_{i,j}^{g,m} + v_{i,j}^{h,l} \le 0, \quad i,j \in [n-1],\ g,h,l,m \in [n],\ g<h,\ l<m, \tag{15.23}$$

$$2(m-l)\left(v_i^g - v_i^h\right) + \sum_{j=1}^{n-1}\left(v_{i,j}^{h,m} + v_{i,j}^{g,l} - v_{i,j}^{g,m} - v_{i,j}^{h,l}\right) \le 0, \quad i \in [n-1],\ g,h,l,m \in [n],\ g<h,\ l<m. \tag{15.24}$$

In fact the inequalities (15.23) and (15.24) are linear combinations of the following smaller set of constraints.

Lemma 15.8 *Inequalities* (15.23) *and* (15.24) *are assured by*

$$-v_{i,j}^{h,l} - v_{i,j}^{h-1,l-1} + v_{i,j}^{h-1,l} + v_{i,j}^{h,l-1} \leq 0, \quad i,j \in [n-1],\ i<j,\ h,l \in [n]. \tag{15.25}$$

$$-2v_i^h + 2v_i^{h-1} + \sum_{j=1}^{n-1} \left(v_{i,j}^{h,l} + v_{i,j}^{h-1,l-1} - v_{i,j}^{h-1,l} - v_{i,j}^{h,l-1} \right) \leq 0, \quad i \in [n-1],\ h,l \in [n]. \tag{15.26}$$

Adding the inequalities discussed above to $(\mathtt{SDP}_1)$ yields the following stronger semidefinite relaxation:

$$\max \{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (15.18) – (15.22), (15.25) and (15.26)},\ Z_s \in \mathcal{E} \}, \tag{$\mathtt{SDP}_2$}$$

There exists another obvious way to tighten our semidefinite relaxations. As $Z_s \in \mathcal{P}_{\mathtt{QPP}}$ is actually a matrix with $\{-1,1\}$ entries, it satisfies the triangle inequalities defining the metric polytope that are known to be facet-defining for the cut polytope, see e.g. [88]:

$$\mathcal{M} = \left\{ Z_s \ :\ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Z_{i,j} \\ Z_{j,k} \\ Z_{i,k} \end{pmatrix} \leq e, \quad 1 \leq i < j < k \leq (n-1)^2 + 1 \right\}. \tag{15.27}$$

Hence we can improve $(\mathtt{SDP}_1)$ and $(\mathtt{SDP}_2)$ by asking in addition that $Z_s \in \mathcal{M}$, which yields the following two tractable relaxation of $\mathcal{P}_{\mathtt{QPP}}$:

$$\max \{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (15.18) – (15.22)},\ Z_s \in (\mathcal{E} \cap \mathcal{M}) \}, \tag{$\mathtt{SDP}_3$}$$

$$\max \{ \langle C_s, Z_s \rangle : Z_s \text{ satisfies (15.18) – (15.22), (15.25) and (15.26)},\ Z_s \in (\mathcal{E} \cap \mathcal{M}) \}. \tag{$\mathtt{SDP}_4$}$$

Already the semidefinite relaxation $(\mathtt{SDP}_3)$ implicitly implies the monotonicity constraints lifted to quadratic space with the help of an approach suggested by Lovász and Schrijver [214]: Applying their approach to our problem suggests to multiply the monotonicity constraints (15.14) by the nonnegative expressions $(1 - v_h^l)$ and $(1 + v_h^l)$ which gives

$$v_i^j - v_{i,h}^{j,l} - v_i^k + v_{i,h}^{k,l} \leq 0, \qquad v_i^j + v_{i,h}^{j,l} - v_i^k - v_{i,h}^{k,l} \leq 0, \quad i,j,k,h,l \in [n-1],\ j<k, \tag{15.28}$$

and correspondent inequalities if object n is involved.

Lemma 15.9 *The lifted monotonicity constraints* (15.28) *are assured by the equalities* (15.18) *together with* $Z_s \in \mathcal{M}$.

15.2.3 The Target Visitation Problem as a Quadratic Position Problem

In the following we explain how to model the (TVP) as a special (QPP): First we show that the (LOP) and the (TSP) variables can be expressed as products of position variables and then we use this property to reformulate $(\mathtt{LP}_{\mathtt{TVP}})$ as a semidefinite optimization problem in position variables that yields provably stronger upper bounds than $(\mathtt{LP}_{\mathtt{TVP}})$.

Lemma 15.10 *We can express the traveling salesman variables and the linear ordering variables as linear-*

quadratic terms in position variables:

$$x_{i,j} = \frac{1}{4}\Bigg[1 - v_{i,j}^{1,1} + v_{i,j}^{1,2} + v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k} + v_{i,j}^{k-1,k}\right) - v_i^{n-2} + v_{i,j}^{n-2,n-1} - v_{i,j}^{n-1,n-1} - v_{i,j}^{n-1,1}\Bigg], \quad i,j \in [n-1],\ i \neq j, \tag{15.29}$$

$$x_{i,n} = \frac{1}{4}\Bigg[5 - n + \sum_{j=1}^{n-1}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2} - v_j^2\right) + \sum_{k=2}^{n-2}\sum_{j=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k} - v_{i,j}^{k,k+1} + v_{i,j}^{k,k}\right) + (n-1)v_i^{n-2} + \sum_{j=1}^{n-1}\left(-v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad i \in [n-1], \tag{15.30}$$

$$x_{n,j} = \frac{1}{4}\Bigg[5 - n + \sum_{i=1}^{n-1}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2}\right) - (n-1)v_j^2 + \sum_{k=2}^{n-2}\sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k} + v_{i,j}^{k,k}\right) + \sum_{i=1}^{n-1}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad j \in [n-1], \tag{15.31}$$

$$y_{i,j} = \frac{1}{4}\left[1 - v_j^1 + \sum_{k=2}^{n-1}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k-1}\right) + v_i^{n-1} - v_{i,j}^{n-1,n-1}\right], \quad i,j \in [n-1],\ i \neq j, \tag{15.32}$$

$$y_{i,n} = \frac{1}{4}\left[n - 1 + \sum_{j=1}^{n-1} v_j^1 + \sum_{k=2}^{n-1}\left(2v_i^{k-1} + \sum_{j=1}^{n-1}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\right) - (n-3)v_i^{n-1} + \sum_{j=1}^{n-1} v_{i,j}^{n-1,n-1}\right], i \in [n-1]. \tag{15.33}$$

$$y_{n,j} = \frac{1}{4}\left[n - 1 + (n-3)v_j^1 + \sum_{k=2}^{n-1}\left(-2v_j^k + \sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\right) - \sum_{i=1}^{n-1}\left(v_i^{n-1} - v_{i,j}^{n-1,n-1}\right)\right], \quad j \in [n-1]. \tag{15.34}$$

Proof. Using $V = vv^\top$ and (15.11) in (15.29) yields

$$x_{i,j} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) + \left(1 - v_i^{n-1}\right)\left(v_j^1 + 1\right)\right], \quad i,j \in [n-1],\ i \neq j.$$

Due to (15.14) $v_i^k - v_i^{k-1} \geq 0,\ i \in [n-1]$, and $v_j^{k+1} - v_j^k \geq 0,\ j \in [n-1]$, hold and $\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) = 1$ iff target i is assigned to the precedent location of target j or target i is assigned to location n and target j to location 1.

Using $V = vv^\top$, (15.11) and (15.15) in (15.30) and (15.31) respectively gives

$$x_{i,n} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_i^k - v_i^{k-1}\right)\left(v_n^{k+1} - v_n^k\right) + \left(1 - v_i^{n-1}\right)\left(v_n^1 + 1\right)\right], \quad i \in [n-1],$$

$$x_{n,i} = \frac{1}{4}\left[\sum_{k=1}^{n-1}\left(v_n^k - v_n^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) + \left(1 - v_n^{n-1}\right)\left(v_j^1 + 1\right)\right], \quad j \in [n-1].$$

Analogically to the reasoning above, (15.14) ensures that both $x_{i,n}$ and $x_{n,j}$ are equal to 1, iff target n is assigned to the successive (first type) or the precedent (second type) location of target i and target j respectively.

Finally applying $V = vv^\top$, (15.11) and (15.15) to (15.32) and (15.33) yields

$$y_{i,j} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad i,j \in [n-1],\ i \neq j$$

$$y_{i,n} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_i^{k-1}\left(v_n^k - v_n^{k-1}\right)\right], \quad i \in [n-1], \quad y_{n,j} = \frac{1}{4}\left[2 + \sum_{k=1}^{n} v_n^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad j \in [n-1].$$

Due to (15.14) $v_j^k - v_j^{k-1} \geq 0,\ j,k \in [n]$, holds and $v_j^k - v_j^{k-1} = 2$, iff target j is assigned to location k. Hence the term $v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j,k \in [n]$, is equal to 2, iff target j is assigned to location k and additionally target i is assigned to some location in front of location k. In summary $\sum_{k=1}^{n} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j \in [n],\ i < j$ is equal to 2, iff target j is assigned to location k and additionally target i is assigned to some location in front of location k, and equal to 0 otherwise. □

Corollary 15.11 *The* (TVP) *objective function* (15.1) *and the* (TVP) *constraints* (15.2) – (15.7) *can be reformulated as linear-quadratic expressions in position variables.*

Proof. On the one hand the (TVP) objective function and (TVP) constraints are linear expressions in the (TSP) variables $x_{i,j},\ i,j \in [n],\ i \neq j$ and the (LOP) variables $y_{i,j},\ i,j \in [n], i < j$, and on the other hand the (TSP) and the (LOP) variables can be expressed as linear-quadratic expressions in position variables due to Lemma 15.10. □

Remark 2 *We refrain from restating the objective function* (15.1) *and the constraints* (15.2) – (15.7) *as (partly quite long) linear-quadratic terms in position variables but of course we encoded all of them in our algorithm. But let us point out that the a bit complicated condition* (15.6) *can be reformulated in position variables in a very elegant way: For $i,j \in [n-1],\ i \neq j$, we just have to subtract the term $1 + v_j^1 - v_i^{n-1} - v_{i,j}^{n-1,1}$ from* (15.29) *to rule out the edges from location n to location 1. In this way we get rid of the term $\frac{1}{n-1}\sum_{k=1, k\neq i}^{n} y_{ki}$. In summary* (15.6) *can be reformulated as follows in position variables:*

$$v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k+1}\right) + v_{i,j}^{n-2,n-1} - v_i^{n-1} \leq 1, \quad i,j \in [n-1],\ i \neq j.$$

Note that these differences of (TSP) *and* (LOP) *variables result in easier expressions in position variables than the* (TSP) *variables. Similar simplifications also occur when rewriting* (15.6) *for $i = n$ and $j = n$:*

$$-\sum_{j=1}^{n-1} v_j^2 - 2\sum_{k=2}^{n-1} v_i^{k-1} + \sum_{k=2}^{n-2}\sum_{j=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + (n-1)v_i^{n-2} - \sum_{j=1}^{n-1} v_{i,j}^{n-2,n-1} \leq n-3, \quad i \in [n-1],$$

$$-(n-1)v_j^2 + 2\sum_{k=2}^{n-1} v_j^k + \sum_{k=2}^{n-2}\sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + \sum_{i=1}^{n-1}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1}\right) \leq n-3, \quad j \in [n-1].$$

Furthermore we can show that the equality constraints for both the (TSP) and the (LOP), reformulated as linear-quadratic expressions in position variables, are already satisfied by (SDP_1).

Lemma 15.12 *The equality constraints* (15.4) *for the* (LOP) *are assured by* (15.20) – (15.22).

Proof. For two fixed targets i and j with $i,j \in [n-1],\ i < j$, we combine (15.20) – (15.22) to obtain

$$2 - v_i^1 - v_j^1 - v_{i,j}^{1,1} + v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} + \sum_{k=2}^{n-1}\left(v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k}\right) = 4\left(y_{i,j} + y_{j,i}\right).$$

Hence the constraints (15.4) not considering target n are linear combinations of the constraints (15.20) – (15.22).

Next we analyze the equalities (15.4) considering target n, i.e. our fixed targets are $i,\ i \in [n-1]$, and n. Adding up 4 times (15.33) and 4 times (15.34) yields

$$4\left(y_{i,n}+y_{n,i}\right) = 2\left(n-1\right) + (n-1)v_i^1 + \sum_{j=1}^{n-1} v_j^1 + \sum_{j=1}^{n-1} v_{i,j}^{1,1} + \sum_{k=2}^{n-1}\sum_{j=1}^{n-1}\Big(v_{i,j}^{k,k} - v_{i,j}^{k-1,k} - v_{i,j}^{k,k-1} + v_{i,j}^{k-1,k-1}\Big) - (n-1)v_i^{n-1} - \sum_{j=1}^{n-1} v_j^{n-1} + \sum_{j=1}^{n-1} v_{i,j}^{n-1,n-1}, \quad i \in [n-1]. \tag{15.35}$$

But the above linear-quadratic expressions in position variables are again linear combinations of the equalities (15.20) – (15.22): Summing up (15.20) for $j \in [n-1],\ j \neq i$, (15.21) for $j \in [n-1],\ j \neq i$, and (15.22) for $j, k \in [n-1],\ j \neq i,\ k \neq 1$, gives (15.35). □

Lemma 15.13 *The equality constraints* (15.2) *for the* (TSP) *are assured by* (15.18) – (15.22).

Proof. The equalities (15.18) ensure that $x_{i,i} = 0,\ i \in [n-1]$, holds and the equalities (15.19) – (15.22) ensure $x_{n,n} = 0$. Now fixing target $i,\ i \in [n-1]$, and summing up (15.29) for $j \in [n-1]$, gives minus the right hand side of (15.30) plus 1: 1−(15.30). Hence finally adding (15.30) proves $\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{i,j} = 1,\ i \in [n-1]$.

The remaining (TSP) equalities, i.e. $\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{j,i} = 1,\ i \in [n-1]$, and $\sum_{\substack{j=1 \\ j\neq n}}^{n} x_{n,j} = \sum_{\substack{j=1 \\ j\neq n}}^{n} x_{j,n} = 1$, can be shown in an analogous fashion. □

Finally adding the (TVP) inequality constraints (15.3) and (15.5) – (15.7), reformulated as linear-quadratic terms in position variables, to ($\mathtt{SDP}_4$) gives our strongest, still tractable relaxation ($\mathtt{SDP}_5$):

$$\max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (15.3), (15.5) – (15.7), (15.18) – (15.22), (15.25) and (15.26), } Z_s \in (\mathcal{E}\cap\mathcal{M})\}. \tag{$\mathtt{SDP}_5$}$$

Corollary 15.14 *The semidefinite programming relaxation* ($\mathtt{SDP}_5$) *is as least as strong as the linear programming relaxation* ($\mathtt{LP}_{\mathtt{TVP}}$).

Proof. As the SDP formulation ($\mathtt{SDP}_5$) ensures all ($\mathtt{LP}_{\mathtt{TVP}}$) constraints (reformulated as linear-quadratic expressions in position variables), the claim follows immediately. □

15.3 Obtaining Upper Bounds and Feasible Tours

The core of our SDP approach is to solve our SDP relaxations. The resulting fractional solutions constitute upper bounds for the exact SDP formulation of the (TVP). By the use of an easy two-opt improvement heuristic, which inter alia exploits the fractional SDP solutions, we obtain quite strong lower bounds, i.e., integer feasible solutions that describe feasible tours through the targets. Hence, in the end we have some feasible solution, together with a certificate of how far this solution could possibly be from the true optimum. Let us now give some details on our upper and lower bound computation.

Looking at the constraint classes and their sizes it should be clear that maintaining explicitly $O(n^2)$ or more constraints is not an attractive option, at least for large-scale instances with for $n \geq 15$ targets. We therefore consider an approach suggested in [103] that was applied to the max cut problem [262] and several ordering problems [168], and adapt it for our SDP relaxations of the (TVP). Initially, we only aim at explicitly ensuring $Z \in \mathcal{E}$, which can be achieved with standard interior point methods, see, e.g. [146].

All other constraints are handled through Lagrangian duality in the objective function f. Thus the objective function f becomes non-smooth. The bundle method [103] iteratively evaluates f at some trial points and uses subgradient information to obtain new iterates. Evaluating f amounts to solving an SDP with the constraint $Z \in \mathcal{E}$ that can be solved efficiently by using again interior point methods. Finally we obtain an approximate minimizer of f that is guaranteed to yield a upper bound to the optimal solution of the SDP relaxations. Since the bundle method has a rather weak local convergence behavior, we limit the number of function evaluations that are responsible for more than 95% of the required running time to control the overall computational effort. This limitation of the number of function evaluations leaves some room for further incremental improvement.

Let us further mention that we also tried to apply quadratic programming approaches to our (TVP) formulations, which however proved to be inapplicable already for small instances with 12 targets. Similar results were observed for quadratic ordering problems in the area of graph drawing [62].

To obtain feasible tours, we apply a simple two-opt improvement heuristic that yields quite good practical results. We call the heuristic every five bundle iterations and conduct each time 100 runs to generate 100 locally optimal tours until the duality gap is closed. If the current best tour is improved or another tour with the best known objective value is found, we update the current best tour. We use three different types of starting tours as input to our heuristic:

1. In the first run we generate an input tour from the current SDP solution vector v as follows: We compute the sums $\sum_{j=1}^{n-1} v_i^j$ for all $i \in [n]$ and sort them downwards to obtain a feasible tour. If the SDP upper bound is equal to the optimal objective value, this strategy yields the optimal tour.

2. If the current best tour is updated, we use the reversed best tour as input tour in the following run of our heuristic.

3. For the remaining runs we use randomly generated input tours.

To improve the input tours to locally optimal tours, we switch the targets i and j and adapt the current tour (Figure 15.2 a.)) as depicted in Figure 15.2 b.), which results in the following change of the objective function

$$\Delta_{\text{obj}} = w_{ji} - w_{ij} - d_{(i-1)j} - d_{j(i+1)} - d_{(j-1)i} - d_{i(j+1)} + d_{(i-1)i} + d_{i(i+1)} + d_{(j-1)j} + d_{j(j+1)}.$$

We update the tour if $\Delta_{\text{obj}} > 0$ and rerun the heuristic until no improvement of the tour is found. As we execute each improving switch, we consider the targets in ascending and descending order, each with 50 runs, to add further randomness to the heuristic procedure.

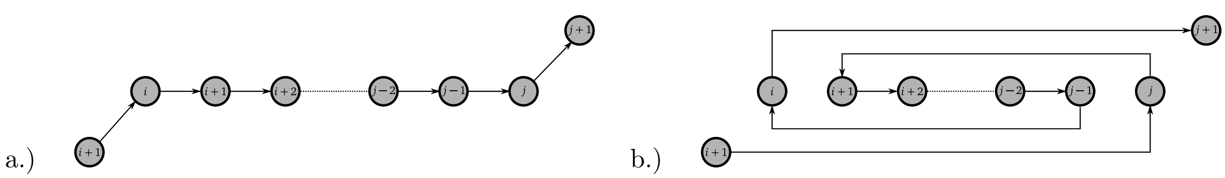


Figure 15.2: If we switch two targets i and j, we propose to adapt the original tour a.) as depicted in b.).

15.4 First Results and Comparisons of our SDP Relaxations on Small and Medium Instances

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7. We generate (TVP) benchmark instances in 2 different ways by

Instance		T	n	Best	$\texttt{SDP}_1$ (exact)				$\texttt{SDP}_2$ (exact)			
LOP	TSP			tour	ub	gap	# c	time	ub	gap	# c	time
Toy example			5	-6	-5.39	0.03	64	0	-6.0	0.00	88	0
PAL11	BR17	a	8	151	175.0	0.16	421	3	152.3	0.01	941	42
PAL11	BR17	a	9	211	247.0	0.17	598	5	211.0	0.00	1502	1:37
PAL11	BR17	a	10	253	313.9	0.24	1208	9	267.6	0.06	2496	26:36
PAL11	BR17	a	11	304	385.9	0.27	1208	31	328.7	0.08	3760	1:33:21
RAND8		r	8	76.4	80.3	0.05	376	12	76.4	0.00	820	11
RAND9		r	9	93.6	107.6	0.15	566	22	95.6	0.02	1511	5:01
RAND10		r	10	132.0	148.5	0.12	820	36	132.0	0.00	2422	36:28
RAND11		r	11	179.4	197.4	0.10	1112	1:20	180.4	0.01	3310	1:26:26

Instance		$\texttt{SDP}_3$ (exact)				$\texttt{SDP}_4$ (exact)				$\texttt{SDP}_5$ (exact)			
LOP	TSP	ub	gap	# c	time	ub	gap	# c	time	ub	gap	# c	time
Toy example		-6.0	0.00	107	0	-6.0	0.00	137	0	-6.0	0.00	242	0
PAL11	BR17	152.1	0.01	1734	6:17	151.0	0.00	1849	1:23	151.0	0.00	2238	2:17
PAL11	BR17	211.0	0.00	2097	5:15	211.0	0.00	2456	3:37	211.0	0.00	3094	5:28
PAL11	BR17	279.8	0.11	3338	59:18	254.2	0.00	4956	4:28:08	253.0	0.00	4615	1:52:40
PAL11	BR17	348.8	0.15	4191	2:29:59	312.1	0.03	6587	11:24:50	305.8	0.01	7327	21:14:05
RAND8		76.4	0.00	1494	1:02	76.4	0.00	1887	1:00	76.4	0.00	2418	4:16
RAND9		98.6	0.05	2323	28:09	93.6	0.00	2986	57:17	93.6	0.00	3571	1:11:14
RAND10		136.0	0.03	3459	1:39:37	132.0	0.00	2986	1:08:49	132.0	0.00	3914	1:50:12
RAND11		184.4	0.03	5501	10:49:29	179.4	0.00	4720	7:14:59	179.4	0.00	6823	17:54:13

Table 15.1: `(TVP)` results obtained by our semidefinite relaxations in conjunction with our 2-opt heuristic. The SDP relaxations are solved exactly by Sedumi. The running times are given in sec or min:sec or in h:min:sec respectively.

Instance (LOP)	Instance (TSP)	T	n	Best tour	$\texttt{SDP}_1$ (bundle) ub	gap	time	$\texttt{SDP}_2$ (bundle) ub	gap	time	$\texttt{SDP}_3$ (bundle) ub	gap	time	$\texttt{SDP}_4$ (bundle) ub	gap	time	$\texttt{SDP}_5$ (bundle) ub	gap	time
PAL19	BR17	a	10	253	332.5	0.31	11	309.5	0.22	12	304.4	0.20	13	294.9	0.17	12	337.3	0.33	13
PAL19	BR17	a	11	304	412.6	0.36	15	388.9	0.28	16	366.6	0.21	18	364.3	0.20	19	424.2	0.40	24
PAL19	BR17	a	12	426	506.6	0.19	23	478.4	0.12	24	474.4	0.11	25	457.3	0.07	28	553.3	0.30	25
PAL19	BR17	a	13	498	607.9	0.22	52	575.5	0.16	44	568.7	0.14	47	549.8	0.10	45	685.9	0.38	43
PAL19	BR17	a	14	555	736.0	0.33	1:21	678.6	0.22	1:04	712.1	0.28	1:08	667.4	0.20	1:06	854.2	0.72	1:03
PAL19	BR17	a	15	627	837.1	0.34	1:26	787.8	0.26	1:28	812.5	0.30	1:33	766.7	0.22	1:31	1144.9	0.83	1:28
PAL19	BR17	a	16	736	952.5	0.29	1:55	932.3	0.27	1:56	938.6	0.28	1:57	900.7	0.22	2:13	1395.5	0.90	2:01
PAL19	BR17	a	17	816	1124.7	0.38	2:40	1063.9	0.30	2:48	1109.5	0.36	2:49	1052.5	0.29	2:44	1301.3	0.59	2:49
PAL19	FTV33	a	18	8607	10132	0.18	3:33	9973	0.16	3:32	10006	0.16	3:43	9913	0.15	3:45	11688	0.36	3:46
PAL19	FTV33	a	19	9309	11368	0.22	4:52	11115	0.19	4:40	11264	0.21	5:17	11017	0.18	4:56	14629	0.57	4:56
RAND10		r	10	132.0	152.5	0.16	11	142.5	0.08	12	142.5	0.08	13	138.0	0.05	18	139.5	0.06	15
RAND11		r	11	179.4	203.8	0.14	16	191.4	0.07	18	195.4	0.09	19	188.4	0.05	26	192.4	0.07	21
RAND12		r	12	207.6	239.6	0.15	25	223.2	0.07	26	227.2	0.09	28	218.2	0.05	28	224.2	0.08	27
RAND13		r	13	251.4	296.8	0.18	45	273.8	0.09	46	280.4	0.12	50	286.4	0.07	48	278.4	0.11	49
RAND14		r	14	282.4	335.4	0.19	1:04	310.4	0.10	1:02	319.8	0.13	1:07	304.8	0.08	1:05	315.4	0.12	1:08
RAND15		r	15	350.2	414.6	0.18	1:32	385.6	0.10	1:26	401.2	0.15	1:35	380.6	0.09	1:32	392.6	0.12	1:33
RAND16		r	16	399.0	483.4	0.21	2:08	451.4	0.13	1:58	467.4	0.17	1:57	445.0	0.12	2:05	482.4	0.21	2:04
RAND17		r	17	448.0	535.0	0.19	2:47	505.6	0.13	2:46	520.0	0.16	2:46	499.0	0.11	2:57	548.4	0.22	2.52
RAND18		r	18	510.2	615.6	0.21	3:32	577.2	0.13	3:49	599.2	0.17	3:40	570.2	0.12	3:55	694.6	0.36	3:41
RAND19		r	19	582.8	709.8	0.22	4:36	667.2	0.15	4:53	592.2	0.19	4:56	659.2	0.13	5:14	840.2	0.44	5:02

Table 15.2: (TVP) results obtained by our semidefinite relaxations in conjunction with our 2-opt heuristic. The (SDP) relaxations are solved approximately by the bundle method, restricted to 250 function evaluations. The running times are given in sec or min:sec respectively.

- combining well-known (LOP) and asymmetric (TSP) instances from the literature,
- using random generated data, where all (LOP) weights and non (Euclidean distances) are uniformly distributed random integers between 1 and 10 and the matrix collecting the (LOP) weights has a density of 50 %.[5]

For variant 15.4 we use appropriate instances from the well-known benchmark libraries LOLIB and TSPLIB.[6] If no instance of appropriate size s is available, we take the first s rows and columns of the instance next in size.[7] For the random data we generate five instances for each number of targets and report the averages over our runs. Additionally we assign one of the following two types T to each instance: a(symmetric), r(andom).

In Table 15.1 we compare the exact solution values of the relaxations (SDP_1)– (SDP_5) for (TVP) instances with up to 17 targets. For reasons of efficiency we used 10 function evaluations of the bundle method to obtain an initial set of constraints violated for our relaxations. We then applied Sedumi [289]; added the 500 most violated constraints; solved again using Sedumi; and repeated this process until no more violations were found or the duality gap is closed down completely. We also tried to solve our relaxations directly but the running times were at least one order of magnitude slower. The best tours are provided by the heuristic described in Section 15.3. The total running time of our heuristic is always clearly less than one second for all instances with up to 20 targets. Additionally we state the average number of constraints $\#c$ that we considered when obtaining the optimal solution of our SDP relaxations.

The results in Table 15.1 show that

- all proposed constraint classes help to improve the relaxation strength considerably,
- all relaxations yield quite tight upper bounds and especially (SDP_5) nearly always closes the duality gap entirely,
- the proposed heuristic yields very strong (TVP) tours, at least for small instances,
- computing the SDP upper bounds becomes time consuming for $n \geq 10$ targets due to the larger number of constraints considered.

Hence for large instances with $n \geq 12$ targets we apply only the bundle method (without Sedumi) to our SDP relaxations and summarize the results obtained in Table 15.2. We restrict the bundle method to 250 function evaluations because its convergence process mostly slows down before that point. With this restriction, the running times of the bundle method are about the same for all SDP relaxations but there is a quite clear ranking of the quality of the upper bounds provided: 1.) (SDP_4) $\rightarrow$ 2.) (SDP_2) $\rightarrow$ 3.) (SDP_3) $\rightarrow$ 4.) (SDP_1) $\rightarrow$ 5.) (SDP_5). Additionally we observe that the random instances are easier to solve than the ones combining difficult (LOP) and (TSP) instances from the literature and that instances containing (a submatrix of) "PAL11" are particularly hard to solve.

Note that we also tried to solve (SDP_1) exactly as an alternative to solving (SDP_4) with the bundle method. This was only possible for instances with up to $n \leq 17$ within one day of computing time and for all instances with $n \geq 12$ we got worse gaps at considerably higher computational costs.

15.5 Computational Experience on Large Scale Instances

Based on the above comparisons, we decide to use (SDP_4) to tackle large scale instances with $n \geq 20$ targets. We double the number of function evaluations to 500 because the convergence process of the

[5] We do note count the diagonal entries as the have to be 0 by definition

[6] See https://www.iwr.uni-heidelberg.de/groups/comopt/software/index.html for details.

[7] When combining the "pal"-Instances with "br17" and "ftv33", we multiply the binary entries by 10 and 100 respectively to balance the (LOP) benefits with the (TSP) distances. Otherwise the optimal (TVP) tour would be very similar to the optimal (TSP) tour.

bundle method elongates for larger instances. We summarize the results obtained for instances with up to 50 targets in Table 15.3 and 15.4.

Instance		T	n	Best	$\mathtt{SDP}_4$ (bundle)		
LOP	TSP			tour	ub	gap	time
PAL23	FTV33	a	20	11649	12377	0.06	14:25
PAL27	FTV33	a	25	19585	22134	0.13	57:37
PAL31	FTV33	a	30	24524	30737	0.25	2:08:26
P40	FTV35	a	35	18433	25465	0.38	9:13:42
P40	FTV44	a	40	24861	35302	0.42	21:51:08
P50	FTV44	a	45	31201	46139	0.48	41:48:57
P50	FTV55	a	50	39289	59769	0.52	65:39:28

Table 15.3: (TVP) results for instances from the literature, obtained by ($\mathtt{SDP}_4$) in conjunction with our 2-opt heuristic. ($\mathtt{SDP}_4$) solved approximately by the bundle method, restricted to 500 function evaluations. The running times are given in min:sec or h:min:sec respectively.

Instance		T	n	Best	$\mathtt{SDP}_4$ (bundle)		
LOP	TSP			tour	ub	gap	time
RAND20		r	20	655	733	0.12	14:06
RAND25		r	25	990	1173	0.18	53:57
RAND30		r	30	1440	1760	0.22	2:08:01
RAND35		r	35	1985	2486	0.25	9:06:39
RAND40		r	40	2568	3317	0.29	21:43:24
RAND45		r	45	3216	4301	0.34	41:44:03
RAND50		r	50	4056	5608	0.38	65:35:46

Table 15.4: (TVP) results for random instances, obtained by ($\mathtt{SDP}_4$) in conjunction with our 2-opt heuristic. ($\mathtt{SDP}_4$) solved approximately by the bundle method, restricted to 500 function evaluations. The running times are given in sec or min:sec or h:min:sec respectively.

We obtain reasonable gaps $\leq 55\%$ for all instances with up 50 targets. Again the random instances are easier to solve and our 2-opt improvement heuristic is very fast as its running time is always less than 5 seconds for all instances with up to 50 targets. In summary the results from both computational sections indicate that the size of the gaps obtained is dependent on the SDP relaxation applied, the difficulty of the (LOP) and (TSP) instances involved and the size of the (TVP) instance considered.

15.6 Conclusion

In this paper we proposed the first exact algorithm for the Target Visitation Problem (TVP) that has several (military) applications. We showed that the (TVP) is a special Quadratic Position Problem (QPP) and exploited this connection to suggest a both theoretical sound and practicably competitive semidefinite optimization approach. Extending the proposed approach to further difficult combinatorial problems that are also special cases of the (QPP) (e.g. the Quadratic Assignment Problem or the Sequential Ordering Problem) is a worthwhile direction for future research.

APPENDIX

Supplementary Information on the SDP Objective Function

Let us also take a closer look at the (TVP) objective function $\langle C_s, Z_s \rangle$ by recalling our toy example from the introduction with the input data provided in (15.8). The optimal layout encoded in position variables, given in (15.12), leads to the following matrix Z_s of dimension $4^2 + 1$ that is optimal for the corresponding cost matrix C_s:

$$Z_s = \begin{pmatrix}
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
-1 & -1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & -1 \\
+1 & +1 & +1 & +1 & +1 & -1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & +1
\end{pmatrix},$$

$$C_s = \begin{pmatrix}
-7.500 & -0.500 & 1.875 & -1.875 & 0.000 & 0.500 & 4.125 & -2.125 & -0.875 & -1.000 & 2.625 & -3.625 & 0.625 & -2.250 & 0.375 & -3.375 & 1.500 \\
-0.500 & -1.500 & 1.500 & -0.750 & -0.750 & -0.750 & 1.250 & -0.625 & -0.625 & -1.000 & 0.750 & -0.375 & -0.375 & -1.000 & 0.250 & -0.125 & -0.125 \\
1.875 & 1.500 & -1.500 & 1.500 & -0.750 & 0.750 & -0.750 & 1.250 & -0.625 & 1.000 & -1.000 & 0.750 & -0.375 & 1.000 & -1.000 & 0.250 & -0.125 \\
-1.875 & -0.750 & 1.500 & -1.500 & 1.500 & -0.625 & 0.750 & -0.750 & 1.250 & -0.375 & 1.000 & -1.000 & 0.750 & -0.125 & 1.000 & -1.000 & 0.250 \\
0.000 & -0.750 & -0.750 & 1.500 & -1.500 & -0.625 & -0.625 & 0.750 & -0.750 & -0.375 & -0.375 & 1.000 & -1.000 & -0.125 & -0.125 & 1.000 & -1.000 \\
0.500 & -0.750 & 0.750 & -0.625 & -0.625 & -1.500 & 2.000 & -1.250 & -1.250 & -1.500 & 1.250 & -0.875 & -0.875 & -1.125 & 0.375 & -0.375 & -0.375 \\
4.125 & 1.250 & -0.750 & 0.750 & -0.625 & 2.000 & -1.500 & 2.000 & -1.250 & 2.000 & -1.500 & 1.250 & -0.875 & 1.500 & -1.125 & 0.375 & -0.375 \\
-2.125 & -0.625 & 1.250 & -0.750 & 0.750 & -1.250 & 2.000 & -1.500 & 2.000 & -0.875 & 2.000 & -1.500 & 1.250 & -0.375 & 1.500 & -1.125 & 0.375 \\
-0.875 & -0.625 & -0.625 & 1.250 & -0.750 & -1.250 & -1.250 & 2.000 & -1.500 & -0.875 & -0.875 & 2.000 & -1.500 & -0.375 & -0.375 & 1.500 & -1.125 \\
-1.000 & -1.000 & 1.000 & -0.375 & -0.375 & -1.500 & 2.000 & -0.875 & -0.875 & -3.000 & 2.750 & -1.250 & -1.250 & -2.375 & 1.625 & -0.625 & -0.625 \\
2.625 & 0.750 & -1.000 & 1.000 & -0.375 & 1.250 & -1.500 & 2.000 & -0.875 & 2.750 & -3.000 & 2.750 & -1.250 & 2.000 & -2.375 & 1.625 & -0.625 \\
-3.625 & -0.375 & 0.750 & -1.000 & 1.000 & -0.875 & 1.250 & -1.500 & 2.000 & -1.250 & 2.750 & -3.000 & 2.750 & -0.625 & 2.000 & -2.375 & 1.625 \\
0.625 & -0.375 & -0.375 & 0.750 & -1.000 & -0.875 & -0.875 & 1.250 & -1.500 & -1.250 & -1.250 & 2.750 & -3.000 & -0.625 & -0.625 & 2.000 & -2.375 \\
-2.250 & -1.000 & 1.000 & -0.125 & -0.125 & -1.125 & 1.500 & -0.375 & -0.375 & -2.375 & 2.000 & -0.625 & -0.625 & -3.000 & 2.250 & -0.750 & -0.750 \\
0.375 & 0.250 & -1.000 & 1.000 & -0.125 & 0.375 & -1.125 & 1.500 & -0.375 & 1.625 & -2.375 & 2.000 & -0.625 & 2.250 & -3.000 & 2.250 & -0.750 \\
-3.375 & -0.125 & 0.250 & -1.000 & 1.000 & -0.375 & 0.375 & -1.125 & 1.500 & -0.625 & 1.625 & -2.375 & 2.000 & -0.750 & 2.250 & -3.000 & 2.250 \\
1.500 & -0.125 & -0.125 & 0.250 & -1.000 & -0.375 & -0.375 & 0.375 & -1.125 & -0.625 & -0.625 & 1.625 & -2.375 & -0.750 & -0.750 & 2.250 & -3.000
\end{pmatrix}.$$

Now evaluating $\langle C_s, Z_s \rangle$ gives the objective value of the optimal (TVP) tour -6.

Supplementary Information on the Heuristic

We also tried to take always the best switch over all pairs of objects, which resulted in tours with clearly worse objective value in average compared to the variant described in the paper. Additionally we experimented with adapting the current tour (Figure 15.6 a.)) as depicted in Figure 15.6 b.). This adaption corresponds to the following change of the objective function

$$\Delta_{\text{obj}} = \sum_{\substack{l,m \in [n] \\ i \le l < m \le j}} (w_{ml} - w_{lm}) - d_{(i-1)j} - d_{i(j+1)} + d_{(i-1)i} + d_{j(j+1)} + \sum_{\substack{l \in [n] \\ i \le l < j}} d_{l(l+1)} - d_{(l+1)l}.$$

Also this strategy led to tours with clearly worse objective value in average, probably because the order of the targets is always reversed between i and j.

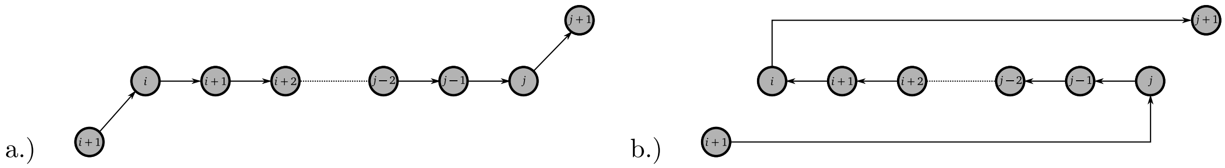


Figure 15.3: If we switch two targets i and j, we could also adapt the original tour a.) as depicted in b.).

Chapter 16

New Relaxations for the Linear Ordering and the Traveling Salesman Problem

Authors: Philipp Hungerländer

Abstract: In 2004 Newman [237] suggested a semidefinite programming relaxation for the Linear Ordering Problem (LOP) that is related to the semidefinite program used in the Goemans-Williamson algorithm to approximate the Max Cut problem [121]. Her model is based on the observation that linear orderings can be fully described by a series of cuts. Newman [237] shows that her relaxation seems better suited for designing polynomial-time approximation algorithms for the (LOP) than the widely-studied standard polyhedral linear relaxations.

In this paper we improve the relaxation proposed by Newman [237] and conduct a polyhedral study of the corresponding polytope. Furthermore we relate the relaxation to other linear and semidefinite relaxations for the (LOP) and for the Traveling Salesman Problem and elaborate on its connection to the Max Cut problem.

Keywords: Linear Ordering Problem; Max Cut Problem; Traveling Salesman Problem; Target Visitation Problem; Vertex Ordering Problems; Semidefinite Programming; Approximation Algorithms; Global Optimization

16.1 Introduction

For the past decades, combinatorial methods and linear programming (LP) techniques have partly failed to yield improved approximation guarantees for many well-studied vertex ordering problems such as the Linear Ordering Problem (LOP) or the Traveling Salesman Problem (TSP). Semidefinite programming (SDP) has proved to be a powerful tool for obtaining strong approximation results for a variety of cut problems, starting with the Max Cut problem [121].

SDP is the extension of LP to linear optimization over the cone of symmetric positive semidefinite matrices. This includes LP problems as a special case, namely when all the matrices involved are diagonal. For further information on SDP we refer to the handbooks [18, 303] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

SDP has been successfully applied to many other problems besides the Max Cut problem that can be considered as cut problems such as the dicut problem [97], coloring k-colorable graphs [177], maximum k-cut [107], maximum bisection and maximum uncut [129], to name a few. In contrast, there is no such comparably general approach for approximating vertex ordering problems.

In 2004 Newman [237] suggested an SDP relaxation for the (LOP) that is related to the semidefinite program used in the Goemans-Williamson algorithm to approximate the Max Cut problem [121]. She observes that linear orderings can be fully described by a series of cuts. Accordingly she suggests an SDP relaxation using cut variables to approximate the (LOP). Newman [237] shows that her relaxation seems well-suited for designing polynomial-time approximation algorithms for the (LOP). In detail she proves that for sufficiently large n her SDP relaxation has an integrality gap of no more than 1.64 for a class of random graphs on n vertices. These random graphs include the graphs used in [238] to demonstrate integrality gaps of 2 for widely-studied polyhedral linear relaxations,

Due to the interesting connection between orderings and cuts and the promising theoretical results, we decided to study the SDP relaxation suggested by Newman [237] in more detail. In this paper we propose a formulation of the (LOP) using cut variables and improve the SDP relaxation of Newman [237] by studying the corresponding polytope. Furthermore we extend the relaxation to other vertex ordering problems and elaborate on its connection to the Max Cut problem. As our relaxations provide new polynomial-time convex approximations of the (LOP) and (TSP) with a rich mathematical structure, we hope that they may be helpful to improve approximation results for vertex ordering problems.

We refer to our companion paper [161] for an application of the theoretical results in this paper, i.e. we use the new semidefinite relaxations suggested here to design an exact SDP approach for the Target Visitation Problem (TVP) [125, 153] that is as a combination of the (LOP) and the (TSP).[1]

The paper is structured as follows. In Section 16.2 we recall some basic facts about the (LOP) and consider different ways of modelling it. Section 16.3 is mainly devoted to the polyhedral study of the (LOP) model using cut variables. In Section 16.4 we show the exact relation of our model using cut variables to the Max Cut problem and show how to extend it to other vertex ordering problems. In Section 16.5 we use small but hard (LOP) and (TSP) instances to compare the strength of our relaxations proposed with other linear and semidefinite relaxations from the literature. Section 16.6 concludes the paper.

16.2 Linear and Quadratic Models for the (LOP)

In this section first we briefly review the basic properties, state-of-the art exact and heuristic approaches and main areas of application of the (LOP). Then we discuss linear and quadratic formulations of the (LOP) using ordering variables in Subsection 16.2.2. Finally in Subsections 16.2.3 we consider a quadratically constrained quadratic program using cut variables proposed by Newman [237] that gives an upper bound for the (LOP). We show how to adapt this program to obtain a quadratic formulation in cut variables for the (LOP) that forms the basis for the polyhedral study in Section 16.3.2.

16.2.1 A Brief Review on the Linear Ordering Problem

Ordering problems associate to each ordering (or permutation) of the set $[n] := \{1, 2, \ldots, n\}$ a profit and the goal is to find an ordering of maximum profit. In the simplest case of the (LOP), this profit is determined by those pairs $(u, v) \in [n] \times [n]$, where u comes before v in the ordering. Thus in its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that $\sum_{\substack{i,j \in [n] \\ i<j}} w_{\pi(i),\pi(j)}$, is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of vertices i and j either arc (i, j) or arc (j, i), but not both. Then the (LOP) consists of finding an acyclic tournament, i.e. a tournament without directed cycles, of G of maximum total edge weight. Let us further clarify this definition with the help of a toy example. We consider 4 vertices and the pairwise weights $w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. Figure 16.1 illustrates the optimal ordering of the vertices and the corresponding benefit.

The (LOP) is well known to be NP-hard [112] and it is even NP-hard to approximate the (LOP) within the factor $\frac{65}{66}$ [238]. Surprisingly there is not much known about heuristics with approximation guarantees.

[1] We note that the content of the companion paper is as disjoint from this paper as possible: it designs an exact SDP approach to the (TVP) and its associated (military) applications. In the companion paper we omitted all proofs concerning the polyhedral properties and referred to this paper for details.

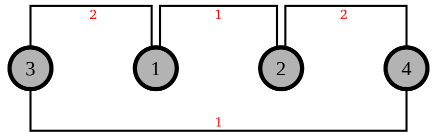


Figure 16.1: We are given 4 vertices and the pairwise weights $w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. We display the optimal linear ordering $(3, 1, 2, 4)$ with the corresponding benefit of $1 + 1 + 2 + 2 = 6$.

If all entries of W are nonnegative, a $\frac{1}{2}$-approximation is trivial: In any ordering of the vertices, either the set of forward edges or the set of backward edges accounts for at least half of the total edge weight. It is not known whether the maximum can be estimated to a better factor using a polynomial-time algorithm. Newman and Vempala [238] showed that widely-studied polyhedral linear relaxations for the (LOP) cannot be used to narrow the quite large gap $[\frac{1}{2}, \frac{65}{66}]$ (for more details see Subsection 16.2.2 below).

The (LOP) arises in a large number of applications in such diverse fields as economy (ranking and voting problems [286] and input-output analysis [205]), sociology (determination of ancestry relationships [119]), graph drawing (one sided crossing minimization [173]), archaeology, scheduling (with precedences [39]), assessment of corruption perception [1] and ranking in sports tournaments. Additionally problems in the context of mathematical psychology and the theory of social choice can be formulated as linear ordering problems, see [104] for a survey. For a recent survey and comparison of exact and heuristic methods for solving the (LOP) we refer to [219].

16.2.2 Formulations Using Ordering Variables

The (LOP) has a natural formulation as an integer linear program (ILP) in 0-1 variables. Let us introduce binary ordering variables x_{ij} with $x_{ij} = 1$ if vertex i comes before vertex j and $x_{ij} = 0$ otherwise. It is well-known [296, 306] that the following constraints describe linear orderings of n vertices:

$$x_{ij} + x_{ji} = 1, \quad i, j \in [n],\ i \neq j, \tag{16.1}$$
$$x_{ij} + x_{jk} + x_{ki} \in \{1, 2\}, \quad i, j, k \in [n], \tag{16.2}$$
$$x_{ij} \in \{0, 1\}, \quad i, j \in [n],\ i \neq j. \tag{16.3}$$

The first condition models the fact that either vertex i is before vertex j or vertex j is before vertex i. The second condition rules out the existence of directed 3-cycles and is sufficient to insure that there is no directed cycle. Hence the feasible solutions for these constraints describe acyclic tournaments of the complete directed graph G with vertex set $[n]$. Maximizing the objective function

$$\sum_{\substack{i,j\in[n],\\ i\neq j}} w_{ij} x_{ij} \tag{16.4}$$

over the constraints (16.1) – (16.3) therefore solves the (LOP). The equations (16.1) are used to eliminate x_{ji} for $j > i$. This leads to the following formulation of the (LOP) as an LP in binary variables (cf. [124])

$$\max \left\{ \sum_{1\leq i<j\leq n} (w_{ij} - w_{ji}) x_{ij} + w_{ji} :\ x \in \mathcal{P}_{LOP} \right\}, \tag{LOP}$$

where the linear ordering polytope is defined as

$$\mathcal{P}_{LOP} = \text{conv}\left\{ x : x \in \{0,1\}^{\binom{n}{2}},\ \ 0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1, \quad i, j, k \in [n],\ i < j < k \right\}.$$

The linear relaxation ($\mathtt{LP_{LOP}}$) is obtained by replacing the integrality conditions on the variables with the bound constraints $0 \le x_{ij} \le 1,\ i,j \in [n],\ i < j$.

Newman and Vempala [238] proved that $(\mathtt{LP_{LOP}})$ has an integrality gap of 2 The graphs used to demonstrate these integrality gaps are random graphs with uniform edge probability of approximately $\frac{2^{\log n}}{n}$, where n is the number of vertices. Newman and Vempala [238] further show that the integrality gap for these graphs remains 2 even if $(\mathtt{LP_{LOP}})$ is augmented with k-fence constraints for any k, and with k-Möbius ladder constraints for k up to 7. If $(\mathtt{LP_{LOP}})$ is augmented with k-Möbius ladder constraints for general k, the gap is at least $\frac{33}{17} \approx 1.94$.

The current state-of-the-art exact algorithm for the (LOP) is an ILP branch-and-cut approach that was developed by the working group of Reinelt in Heidelberg and is based on sophisticated cut generation procedures (for details see [219]). It can solve large instances from specific instance classes with up to 150 vertices, while it fails on other much smaller instances with only 50 vertices. Hungerländer and Rendl [168] proposed an SDP approach that proved to be a valuable alternative to the ILP approach for larger and/or notoriously difficult instances. In the following we briefly recall the underlying formulation of this approach.

For this purpose it is convenient to reformulate the (LOP) in variables taking the values -1 and $+1$. The variable transformation

$$y_{ij} = 2x_{ij} - 1, \quad i,j \in [n],\ i < j, \tag{16.5}$$

leads to the following equivalent formulation of the (LOP):

$$\begin{aligned} \max \sum_{\substack{i,j\in[n]\\ i<j}} (w_{ij} - w_{ji})\frac{y_{ij}+1}{2} + w_{ji}, & \\ \text{subject to:}\ -1 \le y_{ij} + y_{jk} - y_{ik} \le 1, \quad i,j,k \in [n],\ i<j<k, & \\ y_{ij} \in \{-1,1\}^{\binom{n}{2}}, \quad i,j \in [n],\ i<j. & \end{aligned} \tag{16.6}$$

In [138] it is shown that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties (like semidefiniteness constraints in SDPs) are preserved. Additionally we want to reformulate the 3-cycle inequalities (16.6) as quadratic conditions. A natural way to do this consists in squaring both sides of $|y_{ij} + y_{jk} - y_{ik}| = 1$ and using $y_{ij}^2 = 1$ to simplify the resulting expression. This finally leads to the simple 3-cycle equations

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k \in [n],\ i<j<k. \tag{16.7}$$

In summary we obtain the following alternative formulation of the (LOP):

$$\begin{aligned} \max \sum_{\substack{i,j\in[n]\\ i<j}} (w_{ij} - w_{ji})\frac{y_{ij}+1}{2} + w_{ji}, & \\ \text{subject to:}\ y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k \in [n],\ i<j<k, & \\ y_{ij} \in \{-1,1\}^{\binom{n}{2}}, \quad i,j \in [n],\ i<j. & \end{aligned} \tag{16.8}$$

16.2.3 A Formulation Using Cut Variables

Newman [237] suggested another integer quadratically constrained quadratic program for the (LOP) that can be seen as a generalization of the semidefinite programming relaxation of the Max Cut problem [121] (see also Subsection 16.4.1 for a more details on this connection).

The model by Newman [237] is based on the following observation that relates cuts and orderings: A linear ordering of n vertices can be fully described by a series of n cuts. Let a vector v of size $n(n-1)$ that contains bivalent cut variables be given. We can write the components $v_{(i-1)n+j},\ i \in [n],\ j \in [n-1]$, of v more compactly as

$$v_i^j \in \{-1, 1\}, \quad i \in [n],\ j \in [n-1]. \tag{16.9}$$

Additionally we will use the parameters

$$v_i^0 := -1, \quad v_i^n := 1, \quad i \in [n]. \tag{16.10}$$

Now we can describe a linear ordering of n vertices with help of v, where vertex $i \in [n]$ is associated to the $n-1$ variables $v_i^1, \dots, v_i^{n-1}$, as follows: We define the mapping φ between linear orderings π of the vertices $i \in [n]$ on the one hand and the cut variables $v_i^j,\ i \in [n],\ j \in [n-1]$ on the other hand:

$$\varphi(\pi) \to \begin{cases} v_i^j = -1, & \text{if vertex } i \text{ comes after position } j \text{ in the ordering,} \\ v_i^j = +1, & \text{if vertex } i \text{ is at position } j \text{ or before in the ordering.} \end{cases} \tag{16.11}$$

To further clarify this definition, we encode the optimal ordering (3,1,2,4) of the toy example depicted in Figure 16.1 in v, where we separate variables associated to different vertices by a vertical dash | :

$$v_{\text{toy}} = \begin{pmatrix} -1 & 1 & 1 \mid -1 & -1 & 1 \mid & 1 & 1 & 1 \mid -1 & -1 & -1 \end{pmatrix}^\top. \tag{16.12}$$

Now let us express the binary ordering variables (16.3) as linear-quadratic terms in the cut variables v, where we additionally use the parameters (16.10):

$$x_{ij} = \frac{1}{4} \sum_{\substack{k,l \in [n], \\ k<l}} \left(v_i^k - v_i^{k-1}\right)\left(v_j^l - v_j^{l-1}\right) = \frac{1}{4} \sum_{\substack{k,l \in [n], \\ k<l}} \left(v_i^k v_j^l + v_i^{k-1} v_j^{l-1} - v_i^k v_j^{l-1} - v_i^{k-1} v_j^l\right), \quad i, j \in [n],\ i \neq j.$$

Newman [237] proposed the following quadratically constrained quadratic program that gives an upper bound to the optimal solution of the (LOP), where she assumes w.l.o.g. n odd to simplify constraint (16.13f):

$$\max \frac{1}{4} \sum_{\substack{i,j \in [n], \\ i \neq j}} \sum_{\substack{k,l \in [n], \\ k<l}} w_{ij} \left(v_i^k v_j^l + v_i^{k-1} v_j^{l-1} - v_i^k v_j^{l-1} - v_i^{k-1} v_j^l\right) \tag{16.13a}$$

$$\text{subject to:} \quad v_i^k v_j^l + v_i^{k-1} v_j^{l-1} - v_i^k v_j^{l-1} - v_i^{k-1} v_j^l \geq 0, \quad i, j \in [n],\ k, l \in [n], \tag{16.13b}$$

$$v_i^k v_i^k = 1 \quad i, k \in [n], \tag{16.13c}$$

$$v_i^0 v_0 = -1, \quad i \in [n], \tag{16.13d}$$

$$v_i^n v_0 = 1, \quad i \in [n], \tag{16.13e}$$

$$\sum_{i,j \in [n]} v_i^{\frac{n}{2}} v_j^{\frac{n}{2}} = 0, \tag{16.13f}$$

$$v_i^k \in \{1, -1\}, \quad i, k \in [n]. \tag{16.13g}$$

In an integral solution constraint (16.13b) enforces that v_i^{h-1} and v_i^h differ for only one value of $h \in [n]$. This position h denotes the position of vertex i in the ordering.

Next Newman [237] obtains a semidefinite programming relaxation for the (LOP) by removing the integrality conditions (16.13g). She shows that for sufficiently large n this semidefinite programming relaxation has (with high probability) an integrality gap of no more than 1.64 (hence smaller than 2) on random graphs with uniform edge probability $p = \frac{d}{n}$ (i.e. every edge in the complete directed graph on n vertices is chosen with probability p). Note that in particular the graphs used in [238] to demonstrate integrality gaps of 2 for the widely-studied polyhedral linear relaxations fall into this category of random graphs.

Next we want to extend (and simplify) the quadratically constrained quadratic program (16.13) to obtain a formulation of the (LOP) in cut variables. First we can rewrite the objective function (16.13a) as

$$\sum_{\substack{i,j\in[n],\\ i\neq j}} \frac{1}{4} w_{ij} \left(2 + \sum_{k\in[n]} \left(v_i^{k-1} \left(v_j^k - v_j^{k-1} \right) \right) \right) = \sum_{\substack{i,j\in[n],\\ i\neq j}} \frac{w_{ij}}{4} \left(2 + \sum_{k\in[n]} \left(v_i^{k-1} v_j^k - v_i^{k-1} v_j^{k-1} \right) \right). \quad (16.14)$$

Next consider the quadratic constraint (16.13b) ensuring that for each vertex $i \in [n]$ two consecutive variables v_i^{h-1} and v_i^h differ for only one $h \in [n]$ in an integral solution. We can replace (16.13b) by the following simpler linear constraint that is equivalent for integral solutions:

$$v_i^j \leq v_i^k, \quad i \in [n],\ j,k \in [n-1],\ j < k. \quad (16.15)$$

Note that also for non-integral solutions (16.15) enforces monotonicity of variables belonging to the same vertex and ensures together with (16.10): $-1 \leq v_i^j \leq 1,\ i \in [n],\ j \in [n-1]$.

Furthermore we replace the quadratic equation (16.13f) by the following set of linear equations:

$$\sum_{i\in[n]} v_i^j = 2j - n, \quad j \in [n-1]. \quad (16.16)$$

Assuming w.l.o.g. n odd, equality (16.13f) only ensures that half of the variables $v_i^{\frac{n}{2}},\ i \in [n]$, are -1 in an integral solution. Hence it is equivalent to $\sum_{i\in[n]} v_i^{\frac{n}{2}} = 0$.

The following lemma indicates that the linear constraints proposed form a minimal constraint system for the (LOP).

Lemma 16.1 *The constraints* (16.15) *and* (16.16) *form, together with the integrality conditions* (16.9)*, a minimal constraint system for modeling the* (LOP) *with the help of* $n(n-1)$ *cut variables.*

Proof. Applying the mapping φ defined in (16.11) to a linear ordering of n vertices yields cut variables that fulfill the constraints (16.15), (16.16) and the integrality conditions (16.9). The inverse operation to (16.11) is given by

$$v_i^j - v_i^{j-1} = \begin{cases} 2, & \text{if vertex } i \text{ is at position } j \text{ in the linear ordering,} \\ 0, & \text{if vertex } i \text{ is not at position } j \text{ in the linear ordering.} \end{cases} \quad (16.17)$$

Hence applying (16.17) to cut variables that fulfill constraints (16.15), (16.16) and the integrality conditions (16.9) yields a linear ordering of n vertices. Omitting any of the constraints would allow for feasible cut variable configurations that cannot be associated through (16.17) to a valid linear ordering. □

Finally we are able to formulate the (LOP) as an integer quadratic program in v:

Theorem 16.1 *The integer quadratic program*

$$\begin{aligned} &\max \frac{1}{4} \sum_{\substack{i,j\in[n],\\ i\neq j}} w_{ij} \left(2 + \sum_{k\in[n]} \left(v_i^{k-1} v_j^k - v_i^{k-1} v_j^{k-1} \right) \right), \\ \textit{subject to: } & v_i^j \leq v_i^k, \quad i \in [n],\ j,k \in [n-1],\ j < k, \\ & \sum_{i\in[n]} v_i^j = 2j - n, \quad j \in [n-1]. \\ & v_i^j \in \{-1, 1\},\ v_i^0 = -1,\ v_i^n = 1, \quad i \in [n],\ j \in [n-1]. \end{aligned} \quad (16.18)$$

is a formulation of the (LOP).

Proof. The objective function (16.14) gives the correct objective value for any feasible linear ordering. Furthermore the constraints (16.15) model monotonicity on $v \in \{-1,1\}^{n(n-1)}$ and suffice together with the integrality conditions on v and (16.16) to induce all feasible linear orderings of n vertices (see also Lemma 16.1). □

The integer quadratic program (16.18) forms the basis for semidefinite relaxations using cut variables that we analyze in the following section.

16.3 Semidefinite Relaxations for the Linear Ordering Problem

In Subsection 16.3.1 we briefly recall the standard semidefinite relaxation for the `(LOP)` using ordering variables, for more details see [168]. In Subsection 16.3.2 we conduct a polyhedral study to obtain semidefinite relaxations for the `(LOP)` based on cut variables.

16.3.1 Relaxations Using Ordering Variables

In this subsection we briefly review how to obtain standard semidefinite relaxations from (16.8). We are interested in lifting the `(LOP)` into quadratic space and hence we take the vector y collecting the bivalent ordering variables and consider the matrix $Y = yy^\top$. The linear-quadratic ordering polytope can be defined as:

$$\mathcal{P}_{LQO} := \operatorname{conv}\left\{ \begin{pmatrix}1\\y\end{pmatrix}\begin{pmatrix}1\\y\end{pmatrix}^\top \;:\; y \in \{-1,1\}^{\binom{n}{2}},\ y \text{ satisfies } (16.6) \right\}.$$

Let us first slightly rewrite (16.7) using the entries of matrix Y:

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i,j,k \in [n],\ i<j<k. \tag{16.19}$$

In [48] it is shown that these equations formulated in the $\{0,1\}$-model describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. Now we are able to give a matrix-based formulation of the `(LOP)`.

Theorem 16.2 *[168] The* `(LOP)` *is equivalent to the following optimization problem*

$$\max\left\{ \langle C_y, Z_y\rangle : Z_y \textit{ satisfies } (16.7)\ ,\ y \in \{-1,1\}^{\binom{n}{2}} \right\}, \tag{16.20}$$

where all ordering variables and their products are contained in the variable matrix $Z_y := \begin{pmatrix}1 & y^\top\\ y & Y\end{pmatrix} \in \mathbb{R}^{\binom{n}{2}+1}$ *with* $Y = yy^\top$ *and the cost matrix* C_y *is given by* $C_y := \begin{pmatrix}K_y & c^\top\\ c & 0\end{pmatrix}$, $K_y = \sum_{i,j\in[n],\, i<j} \frac{w_{ij}+w_{ji}}{2}$, $c_{ij} = \frac{w_{ij}-w_{ji}}{2}$, $i,j \in [n],\ i<j$.

Finally we can further rewrite the above matrix-based formulation as an SDP, where we denote by e the vector of all ones and by $\mathcal{E}$ the elliptope $\mathcal{E} := \{\ Z\ :\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}$.

Theorem 16.3 *[168] The problem*

$$\max\left\{\langle C_y, Z_y\rangle : Z \textit{ satisfies } (16.7),\ Z \in \mathcal{E},\ y \in \{-1,1\}^{\binom{n}{2}}\right\},$$

is equivalent to the `(LOP)`*.*

We are now dropping the integrality condition on y and obtain the following basic semidefinite relaxation:

$$\max\ \{\ \langle C_y, Z_y\rangle :\ Z \text{ satisfies } (16.7),\ \ Z \in \mathcal{E}\ \}\,. \tag{$\mathtt{SDP}_1^{\mathtt{ord}}$}$$

It is easy to show that $(\mathtt{SDP}_1^{\mathtt{ord}})$ is at least as strong as the linear relaxation $(\mathtt{LP}_{\mathtt{LOP}})$ [168].

There are some obvious ways to tighten $(\mathtt{SDP}_1^{\mathtt{ord}})$. First of all we observe that Z_y is actually a matrix with $\{-1,1\}$ entries in (16.20). Hence it satisfies the triangle inequalities defining the metric polytope $\mathcal{M}$, see e.g. [88]:

$$\mathcal{M} = \left\{ Z_y \ : \ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Z_{i,j} \\ Z_{j,k} \\ Z_{i,k} \end{pmatrix} \leq e, \quad 1 \leq i < j < k \leq \binom{n}{2} + 1 \right\}. \tag{16.21}$$

By additionally asking for $Z_y \in \mathcal{M}$ we can improve the basic relaxation $(\mathtt{SDP}_1^{\mathtt{ord}})$:

$$\max \ \{ \ \langle C_y, Z_y \rangle : \ Z \text{ satisfies (16.7)}, \ \ Z \in (\mathcal{E} \cap \mathcal{M}) \ \} \,. \tag{$\mathtt{SDP}_2^{\mathtt{ord}}$}$$

Another generic improvement was suggested by Lovász and Schrijver in [214]. Applied to this model, their approach suggests to multiply the 3-cycle inequalities (16.6) by the nonnegative expressions $(1-y_{lm})$ and $(1+y_{lm})$. This results in the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \ \ i,j,k,l,m \in [n], \ i<j<k, \ l<m, \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \ \ i,j,k,l,m \in [n], \ i<j<k, \ l<m, \end{aligned} \tag{16.22}$$

defining the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{ \ Z_y \ : \ Z_y \text{ satisfies (16.22)} \ \}. \tag{16.23}$$

Now $(\mathtt{SDP}_1^{\mathtt{ord}})$ can also be improved by asking in addition that $Z \in \mathcal{LS}$:

$$\max \ \{ \ \langle C_y, Z_y \rangle : \ Z \text{ satisfies (16.7)}, \ \ Z \in (\mathcal{E} \cap \mathcal{LS}) \ \} \,. \tag{$\mathtt{SDP}_3^{\mathtt{ord}}$}$$

Combining $(\mathtt{SDP}_2^{\mathtt{ord}})$ and $(\mathtt{SDP}_3^{\mathtt{ord}})$ we obtain the following relaxation of $\mathcal{P}_{LQO}$:

$$\max \{ \langle C_y, Z_y \rangle \ : \ Z \text{ satisfies (16.7)}, \ Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \} \,. \tag{$\mathtt{SDP}_4^{\mathtt{ord}}$}$$

Note that $(\mathtt{SDP}_4^{\mathtt{ord}})$ forms the basis for the currently strongest exact approach to both the Single-Row Facility Layout Problem [169] and Multi-Level Verticality Optimization [62].

16.3.2 Relaxations Using Cut Variables

In this subsection we suggest several new semidefinite relaxations using cut variables by studying the polytope corresponding to the quadratic programming formulation (16.18). We start with rewriting (16.18) in terms of matrices to obtain another matrix-based formulation of the (LOP):

$$\max \left\{ \langle C_a, Z_v \rangle \ : \ v \in \{-1,1\}^{n(n-1)}, \ v \text{ satisfies (16.15) and (16.16)} \ \right\}, \tag{16.24}$$

where all cut variables and their products are contained in the $(n^2-n+1) \times (n^2-n+1)$ variable matrix $Z_v := \begin{pmatrix} 1 & v^\top \\ v & V \end{pmatrix}$ with $V = vv^\top$ and the cost matrix C_a is given by $C_a := \begin{pmatrix} K_a & a^\top \\ a & A \end{pmatrix}$, $K_a = \sum_{\substack{i,j \in n, \\ i \neq j}} \frac{w_{ij}}{4}$, and

$$a_i^k = \begin{cases} -\frac{1}{8} \sum_{\substack{j \in [n], \\ j \neq i}} w_{ij}, & \text{if } k = 1, \\ \frac{1}{8} \sum_{\substack{j \in [n], \\ j \neq i}} w_{ij}, & \text{if } k = n-1, \qquad i \in [n], \\ 0, & \text{otherwise,} \end{cases}$$

$$A_{i,j}^{k,l} = \begin{cases} -\frac{w_{ij}}{4}, & \text{if } k = l, \\ \frac{w_{ij}}{4}, & \text{if } k = l-1, \\ 0, & \text{otherwise}, \end{cases} \qquad i,j \in [n],\ i \neq j,\ k,l \in [n-1],\ k \leq l.$$

To further clarify this definition, let us again recall the toy example depicted in Figure 16.1 with the optimal ordering (3,1,2,4). The input data translates as follows into our matrix-based formulation (16.24), where v_{toy} has already been stated in (16.12):

$$K_a = 1.75, \qquad a = \begin{pmatrix} -\frac{3}{8} \\ 0 \\ \frac{1}{8} \\ -\frac{1}{8} \\ 0 \\ \frac{1}{4} \\ 0 \\ 0 \\ \frac{3}{8} \\ -\frac{3}{8} \\ 0 \\ \frac{1}{8} \end{pmatrix}, \qquad A = \begin{pmatrix} 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$v_{\text{toy}} = \begin{pmatrix} -1 \\ +1 \\ +1 \\ -1 \\ -1 \\ +1 \\ +1 \\ +1 \\ +1 \\ -1 \\ -1 \\ -1 \end{pmatrix}, \quad Z v_{\text{toy}} = \begin{pmatrix} +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ +1 & -1 & +1 & +1 & -1 & -1 & +1 & +1 & +1 & +1 & -1 & -1 & -1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ -1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \end{pmatrix}, \quad \langle C_a, Z_{v_{\text{toy}}} \rangle = 6.$$

Finally we can further rewrite the (LOP) as a semidefinite program in cut variables:

Theorem 16.4 *The problem*

$$\min \left\{ \langle C_a, Z_v \rangle : Z_v \text{ satisfies } (16.15) \text{ and } (16.16)\ ,\ Z_v \in \mathcal{E},\ v \in \{-1,1\}^{n(n-1)} \right\}, \tag{16.25}$$

is equivalent to the (LOP)*.*

Proof. The Schur complement lemma [43, Appendix A.5.5] implies $V - vv^\top \succcurlyeq 0 \Leftrightarrow Z_v \succcurlyeq 0$. Since $v_{i,i}^{j,j} = 1$, $i \in [n]$, $j \in [n-1]$, and $\operatorname{diag}(Z_v) = e$, we have $\operatorname{diag}(V - vv^\top) = 0$, which together with $V - vv^\top \succcurlyeq 0$ shows that in fact $V = vv^\top$ is integral. By Theorem 16.1, integrality on V and v together with (16.15) and (16.16) suffice to induce all feasible linear orderings of n vertices and finally the objective function $\langle C_a, Z_v \rangle$ gives the correct benefit for all linear orderings. □

The above formulation of the (LOP) contains the $n-1$ equalities stated in (16.16). We use the following theorem to eliminate $n-1$ variables through (16.16) and hence to reduce the number of variables to $(n-1)^2$.

Theorem 16.5 *[168] Let m linear equality constraints $Dy = d$ be given. If there exists some invertible $m \times m$ matrix F, we can partition the linear system in the following way*

$$Dy = \begin{bmatrix} F \; G \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} = d. \tag{16.26}$$

Then we do not weaken the relaxation by first moving into the subspace given by the equations, and then lifting the problem to matrix space.

In other words, it is equivalent in terms of tightness of the relaxation to eliminate m variables or to lift the m equality constraints in all possible ways to quadratic space, i.e. squaring them and multiplying them with all other linear equality and inequality constraints. Hence working with the smaller SDP reduces the computational effort for solving the (LOP) and also simplifies the model as we do not have to lift (16.16) in all possible ways and hence are able avoid additional constraint classes. But note that due to the variable elimination we have to examine two versions for several constraint types, namely the cases where vertex n is considered and not considered.

Dropping the integrality condition on v in (16.25) and reducing the problem dimension with the help of (16.16), we obtain the following basic semidefinite relaxation of the (LOP) in cut variables:

$$\max \{\langle C_s, Z_s \rangle : Z_s \text{ satisfies (16.15)}, \; Z_s \in \mathcal{E} \}. \tag{$\mathtt{SDP}_0^{\mathtt{cut}}$}$$

The cost and variable matrices with index s consist of the first $(n-1)^2 + 1$ rows and columns of their larger counterparts C_a and Z_v from above, where additionally the weights involving vertex n are added to C_s (by exploiting (16.16)).[2] Note that the equality constraints (16.16) are implicitly assured in the above semidefinite relaxation and hence the SDP relaxation analyzed by Newman [237] is rudimentary version of ($\mathtt{SDP}_0^{\mathtt{cut}}$). In general ($\mathtt{SDP}_0^{\mathtt{cut}}$) gives quite weak upper bounds to the optimal solution value of the (LOP) (for details see the numerical experiments in Subsection 16.5.1). Hence we will suggest several ways to improve the tightness of ($\mathtt{SDP}_0^{\mathtt{cut}}$).

First we propose $n(n-1)(n-2)$ valid equalities for the linear ordering polytope in cut variables

$$\mathcal{P}_{\mathtt{LOPCUT}} := \operatorname{conv} \left\{ \begin{pmatrix} 1 \\ v \end{pmatrix} \begin{pmatrix} 1 \\ v \end{pmatrix}^\top \; : \; v \in \{-1,1\}^{(n-1)^2}, \; v \text{ satisfies (16.15)} \right\},$$

and show that their rank is $n(n-1)(n-2) - 1$, i.e. their number minus 1: The following equalities, consisting of 5 different types, can be deduced by exploiting the structure of v induced by (16.15) and (16.16):

$$v_i^j - v_i^k - v_{i,i}^{j,k} = -1, \quad i,j,k \in [n-1], \; j < k, \tag{16.27}$$

$$m_k \sum_{i=1}^{n-1} v_i^j + m_j \sum_{i=1}^{n-1} v_i^k - \sum_{h=1}^{n-1} \sum_{i=1}^{n-1} v_{h,i}^{j,k} = m_k m_j, \quad m_k = 2k-n-1, \; m_j = 2j-n+1, \; j,k \in [n-1], \; j<k, \tag{16.28}$$

$$v_i^1 + v_j^1 + v_{i,j}^{1,1} = -1, \quad i,j \in [n-1], \; i<j, \tag{16.29}$$

$$v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} = 1, \quad i,j \in [n-1], \; i<j, \tag{16.30}$$

$$v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k} = 0, \quad i,j \in [n-1], \; i<j, \; k \in [n-1], \; k \neq 1. \tag{16.31}$$

Now let us prove that the above equalities are valid for $\mathcal{P}_{\mathtt{LOPCUT}}$ and have rank $n(n-1)(n-2)-1$:

Lemma 16.2 *The $n(n-1)(n-2)$ equalities (16.27) – (16.31) are valid for $\mathcal{P}_{\mathtt{LOPCUT}}$.*

[2] We refrain from writing down C_s explicitly as the technical expressions are not relevant for the following analysis.

Proof. Using $V = vv^\top$ and additionally (16.16) for (16.28), we can rewrite (16.27) – (16.31) as

$$v_i^j - v_i^k - v_i^j v_i^k + 1 = (1 - v_i^k)(1 + v_i^j) = 0, \quad i, j, k \in [n-1],\ j < k, \tag{16.32}$$

$$\begin{aligned} m_k \sum_{i=1}^{n-1} v_i^j + m_j \sum_{i=1}^{n-1} v_i^k - \sum_{h=1}^{n-1}\sum_{i=1}^{n-1} v_h^j v_i^k - m_j m_k = \\ v_n^j - v_n^k - v_n^j v_n^k + 1 = \left(1 - v_n^k\right)\left(1 + v_n^j\right) = 0, \quad j, k \in [n-1],\ j < k, \end{aligned} \tag{16.33}$$

$$v_i^1 + v_j^1 + v_i^1 v_j^1 + 1 = \left(1 + v_i^1\right)\left(1 + v_j^1\right) = 0, \quad i, j \in [n-1],\ i < j, \tag{16.34}$$

$$v_i^{n-1} + v_j^{n-1} - v_i^{n-1} v_j^{n-1} - 1 = \left(-1 + v_i^{n-1}\right)\left(-1 + v_j^{n-1}\right) = 0, \quad i, j \in [n-1],\ i < j, \tag{16.35}$$

$$v_i^k v_j^k + v_i^{k-1} v_j^{k-1} - v_i^k v_j^{k-1} - v_i^{k-1} v_j^k = \left(v_i^k - v_i^{k-1}\right)\left(v_j^k - v_j^{k-1}\right) = 0,\ i, j \in [n-1],\ i < j,\ k \in [n-1], k \neq 1. \tag{16.36}$$

The equalities (16.32) are valid for $\mathcal{P}_{\texttt{LOPCUT}}$ because the combination $v_i^k = -1$ and $v_i^j = 1$ is not feasible for $j < k < n$ due to monotonicity of the variables belonging to the same vertex, see (16.15). The same argument also holds true for vertex n in (16.33), where we additionally use (16.16) to rewrite the equation. The equalities (16.34) are valid for $\mathcal{P}_{\texttt{LOPCUT}}$ because there can only be one vertex at position 1 in the linear ordering and hence $v_i^1 = v_j^1 = 1$ is not allowed. The same argument holds true for (16.35): There can only be one vertex at position n in the linear ordering and hence $v_i^{k-1} = v_j^{k-1} = -1$ is not allowed. Finally the equalities (16.36) are valid for $\mathcal{P}_{\texttt{LOPCUT}}$ as there can be only one vertex at position $k \in [n]$ in the linear ordering: $v_i^k - v_i^{k-1} \geq 0$ holds for $i \in [n-1]$ because of (16.15) and $v_i^k - v_i^{k-1} > 0$ is only possible if vertex i is at position k in the linear ordering.

□

Lemma 16.3 *The equalities* (16.27) – (16.31) *have rank* $n(n-1)(n-2) - 1$.

Proof. First we show that the $\frac{(n+1)n(n-2)}{2}$ equalities $(16.27)_{k-j=1}$, $(16.28)_{k-j=1}$, (16.29), (16.30) and (16.31) are linear dependent and have rank $\frac{(n+1)n(n-2)}{2} - 1$. Note that in these equalities all products of cut variables $v_{i,j}^{k,l},\ i, j, k, l \in [n-1],\ i < j,\ |k - l| \leq 1$, occur twice, each time with multiplicity ± 1. Hence in a linear dependent linear combination of the equations considered, the terms $v_{i,j}^{k,k},\ i, j \in [n-1],\ i < j,\ k \in [n-1]$, in (16.29) – (16.31) have to cancel out. But this is only possible in one unique way:

$$\begin{aligned} &\sum_{\substack{i,j \in [n-1] \\ i<j}} \left[-\left(v_i^1 + v_j^1 + v_{i,j}^{1,1}\right) + (-1)^{n-1}\left(v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1}\right) + \sum_{k=2}^{n-1} (-1)^k \left(v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - \right.\right. \\ &\left.\left. v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k}\right) \right] = \sum_{\substack{i,j \in [n-1] \\ i<j}} \left[-\left(v_i^1 + v_j^1\right) + (-1)^{n-1}\left(v_i^{n-1} + v_j^{n-1}\right) + \sum_{k=2}^{n-1} (-1)^{k+1}\left(v_{i,j}^{k,k-1} + v_{i,j}^{k-1,k}\right) \right]. \end{aligned}$$

Now to obtain a linear dependent linear combination we have to add the equalities $(16.27)_{k-j=1}$ and $(16.28)_{k-j=1}$ to the above equation in the following, uniquely determined way:

$$\begin{aligned} &\sum_{\substack{i,j \in [n-1] \\ i<j}} \left[-\left(v_i^1 + v_j^1\right) + (-1)^{n-1}\left(v_i^{n-1} + v_j^{n-1}\right) + \sum_{k=2}^{n-1} (-1)^{k+1}\left(v_{i,j}^{k,k-1} + v_{i,j}^{k-1,k}\right) \right] + \\ &\sum_{k=2}^{n-1} \left[\sum_{i \in [n-1]} (-1)^k \left(v_i^{k-1} - v_i^k - v_{i,i}^{k-1,k}\right) + (-1)^{k+1} \left((2k - n - 1) \left(\sum_{i \in [n-1]} v_i^{k-1} + v_i^k \right) - \sum_{i,j \in [n-1]} v_{i,j}^{k-1,k} \right) \right] = \\ &\sum_{k=2}^{n} \left((-1)^{k+1} \sum_{\substack{i,j \in [n-1] \\ i<j}} \left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k}\right) \right) + (2 - n) \sum_{i \in [n-1]} v_i^1 + \sum_{i \in [n-1]} v_i^1 + (n-3) \sum_{i \in [n-1]} v_i^1 + \end{aligned}$$

$$(-1)^{n-1}(n-2)\sum_{i\in[n-1]} v_i^{n-1} - (-1)^{n-1}\sum_{i\in[n-1]} v_i^{n-1} + (-1)^{n-1}(3-n)\sum_{i\in[n-1]} v_i^1 = 0.$$

In summary the $\frac{(n+1)n(n-2)}{2}$ equalities $(16.27)_{k-j=1}$, $(16.28)_{k-j=1}$, (16.29), (16.30) and (16.31) are linear dependent and have rank $\frac{(n+1)n(n-2)}{2} - 1$ because all choices for obtaining a linear dependent linear combination were uniquely determined. The remaining $\frac{n(n-2)(n-3)}{2}$ equalities $(16.27)_{k-j\neq 1}$ and $(16.28)_{k-j\neq 1}$ are linear independent: The products of the cut variables $v_{i,i}^{j,k}$, $i,j,k \in [n-1]$, $k \geq j+2$ occur once in $(16.27)_{k-j\neq 1}$ and once in $(16.28)_{k-j\neq 1}$, each time with multiplicity ± 1. To cancel them out the two particular equalities have to be combined in a unique way, resulting in a linear term in cut variables unequal to zero. Summing up, we have shown that the equalities (16.27) – (16.31) have rank $\frac{(n+1)n(n-2)}{2} - 1 + \frac{n(n-2)(n-3)}{2} = n(n-1)(n-2) - 1$. □

We can also show that the the equalities (16.27) suffice together with the implicitly assured linear constraints (16.16) to ensure monotonicity in the variable vector for all vertices.

Lemma 16.4 *The monotonicity constraints* (16.15) *are assured by the equalities* (16.16) *and* (16.27) *together with* $Z_s \in \mathcal{E}$.

Proof. $Z_s \in \mathcal{E}$ guarantees $-1 \leq v_{i,i}^{j,k} \leq 1$, $i,j,k \in [n-1]$, $j < k$. Now applying the equalities (16.27) yields

$$v_i^k - v_i^j = 1 - v_{i,i}^{j,k} \geq 0, \quad i,j,k \in [n-1],\ j<k.$$

Finally (16.15) for all vertices $i \in [n-1]$ together with (16.16) also ensures (16.15) for vertex n. □

In summary $(\mathtt{SDP}_0^{\mathtt{cut}})$ can be tightened by adding the equalities analyzed above that implicitly ensure (16.15):

$$\max\{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (16.27) – (16.31)},\ Z_s \in \mathcal{E}\}, \qquad (\mathtt{SDP}_1^{\mathtt{cut}})$$

Now we can further improve the relaxation strength of $(\mathtt{SDP}_1^{\mathtt{cut}})$ by adding several types of inequalities valid for $\mathcal{P}_{\mathtt{LOPCUT}}$. First we discuss inequalities obtained by exploiting the structure of v induced by (16.15) and (16.16). Secondly we suggest valid inequalities associated to the integrality conditions $v \in \{-1,1\}^{(n-1)^2}$.

Lemma 16.5 *The following inequality constraints hold for* $Z_s \in \mathcal{P}_{\mathtt{LOPCUT}}$*:*

$$-v_{i,j}^{h,m} - v_{i,j}^{g,l} + v_{i,j}^{g,m} + v_{i,j}^{h,l} \leq 0, \quad i,j,g,h,l,m \in [n-1],\ g<h, l<m, \tag{16.37}$$

$$2(m-l)\left(v_i^g - v_i^h\right) + \sum_{j\in[n-1]}\left(v_{i,j}^{h,m} + v_{i,j}^{g,l} - v_{i,j}^{g,m} - v_{i,j}^{h,l}\right) \leq 0, \quad i,g,h,l,m \in [n-1],\ g<h,\ l<m. \tag{16.38}$$

Proof. Using $V = vv^\top$ and (16.16) we can rewrite (16.37) and (16.38) as

$$-v_{i,j}^{h,m} - v_{i,j}^{g,l} + v_{i,j}^{g,m} + v_{i,j}^{h,l} = \left(v_i^h - v_i^g\right)\left(v_j^l - v_j^m\right) \leq 0, \quad i,j,g,h,l,m \in [n-1],\ g<h,\ l<m, \tag{16.39}$$

$$2(l-m)v_i^h + 2(m-l)v_i^g + \sum_{j\in[n-1]}\left(v_{i,j}^{h,m} + v_{i,j}^{g,l} - v_{i,j}^{g,m} - v_{i,j}^{h,l}\right) = \left(v_i^h - v_i^g\right)\Big(2(l-m) +$$

$$\sum_{j\in[n-1]} v_j^l - \sum_{j\in[n-1]} v_j^m\Big) = \left(v_i^h - v_i^g\right)\left(v_n^l - v_n^m\right) \leq 0, \quad i,g,h,l,m \in [n-1],\ g<h,\ l<m. \tag{16.40}$$

Now (16.39) holds as (16.15) ensures $v_i^h - v_i^g \geq 0$ and $v_j^l - v_j^m \leq 0$. Furthermore the inequalities (16.40) are valid for $\mathcal{P}_{\mathtt{LOPCUT}}$ because (16.16) yields $2(l-m) + \sum_{j\in[n-1]} v_j^l - \sum_{j\in[n-1]} v_j^m = v_n^l - v_n^m$ and (16.15) guarantees $v_i^h - v_i^g \geq 0$, $v_n^l - v_k^m \leq 0$. □

Next we show that the inequalities (16.37) and (16.38) are linear combinations of the following smaller set of constraints.

Lemma 16.6 *Inequalities* (16.37) *and* (16.38) *are assured by*

$$-v_{i,j}^{h,l} - v_{i,j}^{h-1,l-1} + v_{i,j}^{h-1,l} + v_{i,j}^{h,l-1} \le 0, \quad i,j \in [n-1],\ i<j,\ h,l \in [n-1],\ h \ne 1,\ l \ne 1. \tag{16.41}$$

$$-2v_i^h + 2v_i^{h-1} + \sum_{j\in[n-1]} \left(v_{i,j}^{h,l} + v_{i,j}^{h-1,l-1} - v_{i,j}^{h-1,l} - v_{i,j}^{h,l-1} \right) \le 0, \quad i,h,l \in [n-1],\ h \ne 1,\ l \ne 1. \tag{16.42}$$

Proof. Adding $(16.41)_{h=s,l=t}$ and $(16.41)_{h=s-1,l=t}$ yields

$$-v_{i,j}^{s,t} - v_{i,j}^{s-2,t-1} + v_{i,j}^{s-2,t} + v_{i,j}^{s,t-1} \le 0, \quad i,j \in [n-1],\ i<j.$$

Additionally adding $(16.41)_{h=s,l=t-1}$ and $(16.41)_{h=s-1,l=t-1}$ to the above inequalities gives

$$-v_{i,j}^{s,t} - v_{i,j}^{s-2,t-2} + v_{i,j}^{s-2,t} + v_{i,j}^{s,t-2} \le 0, \quad i,j \in [n-1],\ i<j.$$

Thus any inequality of type (16.37) can be written as a (telescoping) sum of particular inequalities from (16.41). Analogically (16.38) is assured by (16.42). □

Adding the inequalities discussed above to $(\texttt{SDP}_1^{\texttt{cut}})$ yields the following stronger semidefinite relaxation:

$$\max \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies } (16.27) - (16.31),\ (16.41) \text{ and } (16.42),\ Z_s \in \mathcal{E}\ \}, \tag{$\texttt{SDP}_2^{\texttt{cut}}$}$$

As $Z_s \in \mathcal{P}_{\texttt{LOPCUT}}$ is actually a matrix with $\{-1,1\}$ entries, we can also tighten our semidefinite relaxations by asking for $Z_s \in \mathcal{M}$, see also (16.21). Furthermore also vertex n has to fulfill the triangle inequalities. The corresponding inequalities are obtained with the help of (16.16) and read as follows:

$$\begin{aligned}
-v_{i,j}^{k,l} + \sum_{h\in[n-1]} \left(v_{i,h}^{k,m} + v_{j,h}^{l,m} \right) - (2m-n)\left(v_i^k + v_j^l \right) &\le 1, \quad i,j,k,l,m \in [n-1],\\
-v_{i,j}^{k,l} - \sum_{h\in[n-1]} \left(v_{i,h}^{k,m} + v_{j,h}^{l,m} \right) + (2m-n)\left(v_i^k + v_j^l \right) &\le 1, \quad i,j,k,l,m \in [n-1],\\
v_{i,j}^{k,l} + \sum_{h\in[n-1]} \left(v_{i,h}^{k,m} - v_{j,h}^{l,m} \right) - (2m-n)\left(v_i^k - v_j^l \right) &\le 1, \quad i,j,k,l,m \in [n-1],\\
v_{i,j}^{k,l} + \sum_{h\in[n-1]} \left(-v_{i,h}^{k,m} + v_{j,h}^{l,m} \right) - (2m-n)\left(-v_i^k + v_j^l \right) &\le 1, \quad i,j,k,l,m \in [n-1],
\end{aligned} \tag{16.43}$$

$$\begin{aligned}
&\sum_{j\in[n-1]} \left(v_{i,j}^{k,l} + v_{i,j}^{k,m} + (2l-n)v_j^m + (2m-n)v_j^l \right) - \sum_{j,h\in[n-1]} v_{j,h}^{l,m}\\
&\qquad - (2m+2l-2n)v_i^k \le 1 + (2l-n)(2m-n), \quad i,k,l,m \in [n-1],\\
&\sum_{j\in[n-1]} \left(v_{i,j}^{k,l} - v_{i,j}^{k,m} - (2l-n)v_j^m - (2m-n)v_j^l \right) + \sum_{j,h\in[n-1]} v_{j,h}^{l,m}\\
&\qquad + (2m-2l)v_i^k \le 1 - (2l-n)(2m-n), \quad i,k,l,m \in [n-1],\\
&\sum_{j\in[n-1]} \left(-v_{i,j}^{k,l} + v_{i,j}^{k,m} - (2l-n)v_j^m - (2m-n)v_j^l \right) + \sum_{j,h\in[n-1]} v_{j,h}^{l,m}\\
&\qquad - (2m-2l)v_i^k \le 1 - (2l-n)(2m-n), \quad i,k,l,m \in [n-1],\\
&\sum_{j\in[n-1]} \left(-v_{i,j}^{k,l} - v_{i,j}^{k,m} + (2l-n)v_j^m + (2m-n)v_j^l \right) - \sum_{j,h\in[n-1]} v_{j,h}^{l,m}\\
&\qquad + (2m+2l-2n)v_i^k \le 1 + (2l-n)(2m-n), \quad i,k,l,m \in [n-1],
\end{aligned} \tag{16.44}$$

$$
\begin{aligned}
&\sum_{j\in[n-1]} \left((2k+2m-2n)v_j^l + (2k+2l-2n)v_j^m + (2m+2l-2n)v_j^k\right) - \sum_{j,h\in[n-1]} \left(v_{j,h}^{k,l} + v_{j,h}^{k,m} + v_{j,h}^{l,m}\right) \\
&\qquad\qquad \le 1 + (2l-n)(2m-n) + (2k-n)(2l-n) + (2k-n)(2m-n), \quad k,l,m\in[n-1], \\
&\sum_{j\in[n-1]} \left((2k-2m)v_j^l - (2k+2l-2n)v_j^m + (-2m+2l)v_j^k\right) - \sum_{j,h\in[n-1]} \left(v_{j,h}^{k,l} - v_{j,h}^{k,m} - v_{j,h}^{l,m}\right) \\
&\qquad\qquad \le 1 - (2l-n)(2m-n) + (2k-n)(2l-n) - (2k-n)(2m-n), \quad k,l,m\in[n-1], \\
&\sum_{j\in[n-1]} \left(-(2k+2m-2n)v_j^l + (2k-2l)v_j^m + (2m-2l)v_j^k\right) - \sum_{j,h\in[n-1]} \left(-v_{j,h}^{k,l} + v_{j,h}^{k,m} - v_{j,h}^{l,m}\right) \\
&\qquad\qquad \le 1 - (2l-n)(2m-n) - (2k-n)(2l-n) + (2k-n)(2m-n), \quad k,l,m\in[n-1], \\
&\sum_{j\in[n-1]} \left((-2k+2m)v_j^l + (-2k+2l)v_j^m - (2m+2l-2n)v_j^k\right) - \sum_{j,h\in[n-1]} \left(-v_{j,h}^{k,l} - v_{j,h}^{k,m} + v_{j,h}^{l,m}\right) \\
&\qquad\qquad \le 1 + (2l-n)(2m-n) - (2k-n)(2l-n) - (2k-n)(2m-n), \quad k,l,m\in[n-1],
\end{aligned} \tag{16.45}
$$

Adding $Z_s \in \mathcal{M}$ and (16.43)–(16.45) to $(\mathtt{SDP}_1^{\mathtt{cut}})$ and $(\mathtt{SDP}_2^{\mathtt{cut}})$ respectively yields the following two SDP relaxations:

$$\max\left\{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (16.27) – (16.31) and (16.43) – (16.45)},\ Z_s \in (\mathcal{E}\cap\mathcal{M})\right\}, \quad (\mathtt{SDP}_3^{\mathtt{cut}})$$

$$\max\ \left\{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (16.27) – (16.31), (16.41), (16.42) and (16.43) – (16.45)},\ Z_s \in (\mathcal{E}\cap\mathcal{M})\right\}. \quad (\mathtt{SDP}_4^{\mathtt{cut}})$$

We can also reformulate the monotonicity constraints on the cut variables as linear-quadratic inequalities (as we have done it above for the 3-cycle inequalities (16.6), see constraints (16.22)) with the help of the approach suggested by Lovász and Schrijver [214]. Hence multiplying the monotonicity constraints (16.15) by the nonnegative expressions $(1 - v_h^l)$ and $(1 + v_h^l)$ gives

$$v_i^j - v_{i,h}^{j,l} - v_i^k + v_{i,h}^{k,l} \le 0, \qquad v_i^j + v_{i,h}^{j,l} - v_i^k - v_{i,h}^{k,l} \le 0, \quad i,j,k,h,l\in[n-1],\ j<k, \tag{16.46}$$

and correspondent inequalities if vertex n is involved:

$$2(k-j)v_h^l + \sum_{i\in[n-1]} \left(v_i^k - v_i^j + v_{i,h}^{j,l} - v_{i,h}^{k,l}\right) \le 2(k-j), \quad j,k,h,l\in[n-1],\ j<k, \tag{16.47}$$

$$2(j-k)v_h^l + \sum_{i\in[n-1]} \left(v_i^k - v_i^j - v_{i,h}^{j,l} + v_{i,h}^{k,l}\right) \le 2(k-j), \quad j,k,h,l\in[n-1],\ j<k, \tag{16.48}$$

$$(2l-n)\left(v_i^k - v_i^j\right) + v_i^j + \sum_{h\in[n-1]} v_{i,h}^{j,l} - v_i^k - \sum_{h\in[n-1]} v_{i,h}^{k,l} \le 0, \quad i,j,k,l\in[n-1],\ j<k, \tag{16.49}$$

$$(2l-n)\left(v_i^j - v_i^k\right) + v_i^j - \sum_{h\in[n-1]} v_{i,h}^{j,l} - v_i^k + \sum_{h\in[n-1]} v_{i,h}^{k,l} \le 0, \quad i,j,k,l\in[n-1],\ j<k. \tag{16.50}$$

But the inequalities above are not helpful to further strengthen our relaxations as they are already implicitly contained in $(\mathtt{SDP}_3^{\mathtt{cut}})$.

Lemma 16.7 *The lifted monotonicity constraints* (16.46) *are assured by the equalities* (16.27) *together with* $Z_s \in \mathcal{M}$.

Proof. Applying the equalities (16.27) to the left hand side of (16.46) yields

$$-1 + v_{i,i}^{j,k} - v_{i,h}^{j,l} + v_{i,h}^{k,l}, \qquad -1 + v_{i,i}^{j,k} + v_{i,h}^{j,l} - v_{i,h}^{k,l}, \quad i,j,k,h,l\in[n-1],\ j<k.$$

But the two expressions above are ≤ 0 due to $Z_s \in \mathcal{M}$. □

Analogical (but more cumbersome) calculations show that also (16.47) – (16.50) are already contained in $(\mathtt{SDP}_3^{\mathtt{cut}})$.

Finally we state explicitly how to model the binary ordering variables via the entries of matrix Z_s and then use this transformations to add the constraints (16.1) – (16.3) to $(\mathtt{SDP}_4^{\mathtt{cut}})$.

Lemma 16.8 *We can express the linear ordering variables as:*

$$x_{ij} = \frac{1}{4}\left[1 - v_j^1 + \sum_{k=2}^{n-1}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k-1}\right) + v_i^{n-1} - v_{i,j}^{n-1,n-1}\right], \quad i,j \in [n-1],\ i \neq j, \tag{16.51}$$

$$\begin{aligned} x_{in} = \frac{1}{4}\Bigg[n - 1 + \sum_{j\in[n-1]} v_j^1 + \sum_{k=2}^{n-1}\Bigg(2v_i^{k-1} + \sum_{j\in[n-1]}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\Bigg) \\ -(n-3)v_i^{n-1} + \sum_{j\in[n-1]} v_{i,j}^{n-1,n-1}\Bigg], \quad i \in [n-1]. \end{aligned} \tag{16.52}$$

$$\begin{aligned} x_{n,j} = \frac{1}{4}\Bigg[n - 1 + (n-3)v_j^1 + \sum_{k=2}^{n-1}\Bigg(-2v_j^{k} + \sum_{i\in[n-1]}\left(v_{i,j}^{k-1,k-1} - v_{i,j}^{k-1,k}\right)\Bigg) \\ - \sum_{i\in[n-1]}\left(v_i^{n-1} - v_{i,j}^{n-1,n-1}\right)\Bigg], \quad j \in [n-1]. \end{aligned} \tag{16.53}$$

Proof. Applying $V = vv^\top$, (16.10) and (16.16) to (16.51) and (16.52) yields

$$x_{ij} = \frac{1}{4}\left[2 + \sum_{k\in[n]} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad i,j \in [n-1],\ i \neq j$$

$$x_{in} = \frac{1}{4}\left[2 + \sum_{k\in[n]} v_i^{k-1}\left(v_n^k - v_n^{k-1}\right)\right], \quad i \in [n-1], \quad x_{n,j} = \frac{1}{4}\left[2 + \sum_{k\in[n]} v_n^{k-1}\left(v_j^k - v_j^{k-1}\right)\right], \quad j \in [n-1].$$

Due to (16.15) $v_j^k - v_j^{k-1} \geq 0,\ j,k \in [n]$, holds and $v_j^k - v_j^{k-1} = 2$, if and only if vertex j is at position k in the linear ordering. Hence the term $v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j,k \in [n],\ i \neq j$, is equal to 2, if and only if vertex j is at position k and additionally vertex i comes before vertex j in the linear ordering. In summary $\sum_{k\in[n]} v_i^{k-1}\left(v_j^k - v_j^{k-1}\right),\ i,j \in [n],\ i < j$, is equal to 2, if vertex i comes before vertex j in the linear ordering, and equal to 0 otherwise. □

Note that equation (16.51) has already been implicitly used by the deduction of the cost matrix C_a in (16.24).

Corollary 16.9 *The* (LOP) *objective function* (16.4) *and the* (LOP) *constraints* (16.1) – (16.3) *can be reformulated as linear-quadratic expressions in cut variables.*

In particular the equality constraints (16.1) for the (LOP) are already assured by $(\mathtt{SDP}_1^{\mathtt{cut}})$.

Lemma 16.10 *The equality constraints* (16.1) *are linear combinations of the equalities* (16.29) – (16.31).

Proof. For two fixed vertices i and j with $i,j \in [n-1],\ i < j$, we combine (16.29) – (16.31) to obtain

$$2 - v_i^1 - v_j^1 - v_{i,j}^{1,1} + v_i^{n-1} + v_j^{n-1} - v_{i,j}^{n-1,n-1} + \sum_{k=2}^{n-1}\left(v_{i,j}^{k,k} + v_{i,j}^{k-1,k-1} - v_{i,j}^{k,k-1} - v_{i,j}^{k-1,k}\right) = 4\left(x_{ij} + x_{ji}\right).$$

Hence the constraints (16.1) not considering vertex n are linear combinations of the constraints (16.29) – (16.31).

Next we analyze the equalities (16.1) considering vertex n, i.e. our fixed vertices are $i,\ i \in [n-1]$, and n. Adding up 4 times (16.52) and 4 times (16.53) yields

$$\begin{aligned}4\,(x_{in}+x_{ni}) = 2\,(n-1) + (n-1)v_i^1 + \sum_{j\in[n-1]} v_j^1 + \sum_{k=2}^{n-1}\sum_{j\in[n-1]} \Big(v_{i,j}^{k,k} - v_{i,j}^{k-1,k} \\ -v_{i,j}^{k,k-1} + v_{i,j}^{k-1,k-1}\Big) - (n-1)v_i^{n-1} - \sum_{j\in[n-1]} v_j^{n-1} + \sum_{j=1}^{n-1} v_{i,j}^{n-1,n-1}, \quad i \in [n-1].\end{aligned} \tag{16.54}$$

But the above linear-quadratic expression is again a linear combination of the equalities (16.29) – (16.31): Summing up (16.29) for $j \in [n-1],\ j \neq i$, (16.30) for $j \in [n-1],\ j \neq i$ and (16.31) for $j,k \in [n-1],\ j \neq i,\ k \neq 1$ gives (16.54). □

Finally adding the (LOP) inequality constraints (16.2) and (16.3), reformulated via the entries of matrix Z_s, to ($\mathtt{SDP}_4^{\mathtt{cut}}$) gives the last relaxation of this section:

$$\max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (16.2), (16.3), (16.27) – (16.31), (16.41) and (16.42)},\ Z_s \in (\mathcal{E}\cap\mathcal{M})\ \}. \qquad (\mathtt{SDP}_5^{\mathtt{cut}})$$

Corollary 16.11 *The semidefinite programming relaxation* ($\mathtt{SDP}_5^{\mathtt{cut}}$) *is as least as strong as the linear programming relaxation* ($\mathtt{LP}_{\mathtt{LOP}}$).

Proof. As ($\mathtt{SDP}_5^{\mathtt{cut}}$) ensures all constraints of ($\mathtt{LP}_{\mathtt{LOP}}$), the claim follows immediately. □

16.4 Extensions to Other Combinatorial Optimization Problems

In this section we first elaborate on the connection of our model using cut variables to the Max Cut problem in Subsection 16.4.1. In Subsections 16.4.2 and 16.4.3 we show that our model can be extended to two further vertex ordering problems, namely the Traveling Salesman and the Target Visitation Problem.

16.4.1 Max Cut With Constraints on the Partition

Let us start with reconsidering the formulation of the (LOP) from (16.24), where we now allow for an arbitrary cost matrix $C_v : \begin{pmatrix} K_v & b^\top \\ b & B\end{pmatrix}$:

$$\max\Big\{\langle C_v, Z_v\rangle\ :\ v \in \{-1,1\}^{n(n-1)},\ v \text{ satisfies (16.15) and (16.16)}\ \Big\}. \tag{16.55}$$

Problem (16.55) is most naturally interpreted as the following combinatorial optimization problem that we introduced as Quadratic Position Problem (QPP) in our companion paper [161]. The input of the (QPP) consists of n vertices and individual and pairwise benefits $b_i^k,\ i \in [n],\ k \in [n-1]$, and $b_{i,j}^{k,l},\ i,j \in [n],\ i<j,\ k,l \in [n-1],\ k \neq l$. The optimization problem can be written down as

$$\max_{\pi\in\Pi_n} \sum_{\substack{i,j\in[n],\ i<j,\\ k,l\in[n-1],\ k\neq l}} \Big(b_i^k w_i^k(\pi) + b_{i,j}^{k,l} w_i^k(\pi) w_j^l(\pi)\Big), \tag{16.56}$$

where Π_n is the set of permutations of the vertices $[n]$, and $w_i^k(\pi),\ i \in [n],\ k \in [n-1]$, is 1, if vertex i is at position h with $h \leq k$ in the particular ordering $\pi \in \Pi_n$. Otherwise $w_i^k(\pi)$ is set to -1. Hence the individual benefits b_i^k are obtained, if vertex i is at location k or before and the pairwise benefits $b_{i,j}^{k,l}$ are collected, if

- vertex i is at position k or before and vertex j is at position l or before,
- vertex i comes after position k and vertex j comes after position l.

Next we suggest another possible interpretations of (16.55) as a special Max Cut problem on $n(n+1)$ vertices. We consider a weighted graph $G = (V, E)$ with $|V| = n(n+1)$ and weights w_e for $e \in E$ that we summarize in the weighted adjacency matrix W. Now let us define the following optimization problem that asks for a maximum cut in G, where the partition (S, T) has to fulfill some additional properties:

$$\max \sum_{i \in S,\ j \in T,\ ij \in E} w_{ij}, \tag{16.57a}$$

$$\text{subject to: } S = \bigcup_{i \in [n]} S_i, \quad T = \bigcup_{i \in [n]} T_i, \quad S_i, T_i \subsetneq [(i-1)(n+1)+1, i(n+1)], \quad i \in [n], \tag{16.57b}$$

$$\max S_i < \min T_i, \quad i \in [n], \tag{16.57c}$$

$$|S_i| \neq |S_j|, \quad i, j \in [n],\ i < j. \tag{16.57d}$$

Condition (16.57b) models that the partition (S, T) consists of n smaller partitions on $n+1$ consecutive vertices. Condition (16.57c) enforces that all disjoint sets S_i and T_i, $i \in [n]$, contain vertices with consecutive numbers. Finally (16.57d) ensures that all disjoint sets S_i, $i \in [n]$, (and hence also T_i) have different cardinality.

Now we consider the Laplacian matrix $L_G = \mathrm{Diag}(We) - W$ of G, where e is the vector of all ones. We partition the vertex set V into two disjoint subsets A and B with

$$A := A_1 \cup A_2, \quad A_1 = \bigcup_{i \in [n]} (i-1)(n+1)+1, \quad A_2 = \bigcup_{i \in [n]} i(n+1), \qquad B := [n(n+1)] \setminus A. \tag{16.58}$$

Next we set

$$K_v = \sum_{\substack{i \in A_1,\ j \in A_2, \\ ij \in E}} w_{ij} + \frac{1}{2} \sum_{\substack{i \in A,\ j \in B, \\ ij \in E}} w_{ij}, \qquad b = \frac{1}{4} \left(\sum_{i \in A_2} L_{B,i} - \sum_{i \in A_1} L_{B,i} \right), \qquad B = \frac{1}{4} L_{B,B}. \tag{16.59}$$

Now we can establish a connection between the combinatorial optimization problems (16.55) and (16.57):

Theorem 16.6 *Problem* (16.55) *with* C_v *defined in* (16.59) *is equivalent to problem* (16.57).

Proof. Let us interpret the cut variables (16.9) together with the parameters (16.10) as follows:

$$v_i^{j-1} = \begin{cases} -1, & \text{if vertex } (i-1)(n+1)+j \text{ belongs to set } S, \\ +1, & \text{if vertex } (i-1)(n+1)+j \text{ belongs to set } T, \end{cases} \qquad i \in [n],\ j = [n+1]. \tag{16.60}$$

Hence (16.60) connects the cut variables v to the partition (S, T) of the $n(n+1)$ vertices of G. Clearly the constraints (16.15), (16.16) and (16.60) impose the same restrictions on (S, T) as the constraints (16.57b) – (16.57d). Finally the objective function $\langle C_v, Z_v \rangle$ with C_v defined in (16.59) gives the sum of the weights of the edges in the cut (S, T), where the constant term K_v and the linear terms in b account for the weights of edges incident to one or two vertices from the set A that are fixed by (16.10). □

16.4.2 Traveling Salesman Problem

Let us now consider the probably most famous of all (combinatorial) optimization problems. The Traveling Salesman Problem (`TSP`) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible tour that visits each city exactly once and returns to the origin city? Even though the (TSP) is NP-hard [178], a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved completely[3] and even problems with millions of cities can be approximated within a small fraction of 1%. Nonetheless the (TSP) continues to pose grand challenges. The Christofides's algorithm approximates the cost of an optimal symmetric (TSP) tour within the factor 1.5 [69]. Lampis [196] proved that no $\frac{185}{184}$-approximation algorithm exists for the symmetric (TSP) unless $P = NP$. Is it possible to further narrow this quite large gap? In the asymmetric (TSP) paths may not exist in both directions or the distances might be different, forming a directed graph. This may be e.g. due to traffic collisions, one-way streets and motorways. Papadimitriou and Vempala [247] proved that no $\frac{118}{117}$-approximation algorithm exists for the asymmetric (TSP) unless $P = NP$. It is an open question if a constant factor approximation exists. Results on the even more difficult non Euclidean (TSP) are e.g. discussed in [245]. We will see below that our semidefinite relaxations can be used to model all these variants of the (TSP). Hence they provide new polynomial-time convex approximations of (TSP) with a rich mathematical structure. Our hope is that they may help to improve approximation results for the (TSP) and maybe even can be used to answer one of the open question mentioned above.

We refer to the books [75, 127, 260] and the references therein for extensive material on the (TSP), its variants and various applications, details on many heuristic and exact methods and relevant theoretical results.

Now let us state one of the best-known integer programming formulation of the (TSP). First we introduce the traveling salesman variables

$$s_{ij} := \begin{cases} 1, & \text{if city } j \text{ is visited immediately after city } i, \\ 0, & \text{otherwise.} \end{cases} \tag{16.61}$$

Now for a given distance matrix $D = (d_{ij})$ the (TSP) can be formulated as:

$$\min \sum_{\substack{i,j \in [n] \\ i \neq j}} d_{ij} s_{ij} \tag{16.62}$$

$$\text{subject to: } \sum_{\substack{j \in [n], \\ j \neq i}} s_{ij} = 1, \qquad \sum_{\substack{j \in [n], \\ j \neq i}} s_{ji} = 1, \quad i \in [n], \tag{16.63}$$

$$\sum_{\substack{i,j \in S \\ i \neq j}} s_{ij} \leq |S| - 1, \quad \forall S \subset [n],\ 2 \leq |S| \leq n, \tag{16.64}$$

$$s_{ij} \in \{0, 1\}, \quad i, j \in [n],\ i \neq j.$$

Constraints (16.63) and (16.64) are the standard constraints for the asymmetric (TSP). If we replace the integrality conditions by the bound constraints

$$0 \leq s_{ij} \leq 1, \quad i, j \in [n],\ i \neq j, \tag{16.65}$$

we obtain the standard relaxation ($\mathtt{LP}_{\mathtt{TSP}}$) that was suggested by Dantzig et al. [78] in 1954. Its optimal value coincides with the LP bound of Held and Karp [137]. The integrality gap of ($\mathtt{LP}_{\mathtt{TSP}}$) is conjectured to be $\frac{4}{3}$ for the metric symmetric (TSP).

Wolsey [304] showed that Christofides' algorithm [69] computes a tour of length at most $\frac{3}{2}$ times the optimal value of ($\mathtt{LP}_{\mathtt{TSP}}$) (see also [282]). An interesting question is whether a similar result may be proved for one of our semidefinite relaxations.

In the following we explain how to model the (TSP) as a special (QPP): First we show that the (TSP)

[3] The branch-and-cut algorithm by Applegate et al. [23] holds the current record, solving an instance with 85,900 cities.

variables can be expressed via the entries of matrix Z_s and then we use this property to reformulate the (TSP) as a semidefinite optimization problem that yields provably stronger lower bounds than $(\mathtt{LP_{TSP}})$.

Lemma 16.12 *We can express the traveling salesman variables* (16.61) *via the entries of matrix* Z_s*:*

$$\begin{aligned} s_{ij} = \frac{1}{4}\Bigg[1 - v_{i,j}^{1,1} + v_{i,j}^{1,2} + v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k} + v_{i,j}^{k-1,k}\right) \\ -v_i^{n-2} + v_{i,j}^{n-2,n-1} - v_{i,j}^{n-1,n-1} - v_{i,j}^{n-1,1}\Bigg], \quad i,j \in [n-1],\ i \neq j, \qquad (16.66) \end{aligned}$$

$$\begin{aligned} s_{in} = \frac{1}{4}\Bigg[5 - n + \sum_{j\in[n-1]}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2} - v_j^2\right) + \sum_{k=2}^{n-2}\sum_{j\in[n-1]}\Big(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k} - v_{i,j}^{k,k+1} \\ +v_{i,j}^{k,k}\Big) + (n-1)v_i^{n-2} + \sum_{j\in[n-1]}\left(-v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad i \in [n-1], \qquad (16.67) \end{aligned}$$

$$\begin{aligned} s_{nj} = \frac{1}{4}\Bigg[5 - n + \sum_{i\in[n-1]}\left(v_{i,j}^{1,1} - v_{i,j}^{1,2}\right) - (n-1)v_j^2 + \sum_{k=2}^{n-2}\sum_{i\in[n-1]}\Big(v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k} \\ +v_{i,j}^{k,k}\Big) + \sum_{i\in[n-1]}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1} + v_{i,j}^{n-1,n-1} + v_{i,j}^{n-1,1}\right)\Bigg], \quad j \in [n-1], \qquad (16.68) \end{aligned}$$

Proof. Using $V = vv^\top$ and (16.10) in (16.66) yields

$$s_{ij} = \frac{1}{4}\left[\sum_{k\in[n-1]}\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) + \left(1 - v_i^{n-1}\right)\left(v_j^1 + 1\right)\right], \quad i,j \in [n-1],\ i \neq j.$$

Due to (16.15), $v_i^k - v_i^{k-1} \geq 0,\ i \in [n-1]$, and $v_j^{k+1} - v_j^k \geq 0,\ j \in [n-1]$, hold and $\left(v_i^k - v_i^{k-1}\right)\left(v_j^{k+1} - v_j^k\right) = 1$, if city i is visited directly before city j or if city i is the last city of the tour (i.e. the n^{th} city) and city j is the first city of the tour.

Using $V = vv^\top$, (16.10) and (16.16) in (16.67) and (16.68) respectively gives

$$s_{in} = \frac{1}{4}\left[\sum_{k\in[n-1]}\left(v_i^k - v_i^{k-1}\right)\left(v_n^{k+1} - v_n^k\right) + \left(1 - v_i^{n-1}\right)\left(v_n^1 + 1\right)\right], \quad i \in [n-1],$$

$$s_{ni} = \frac{1}{4}\left[\sum_{k\in[n-1]}\left(v_n^k - v_n^{k-1}\right)\left(v_i^{k+1} - v_i^k\right) + \left(1 - v_n^{n-1}\right)\left(v_i^1 + 1\right)\right], \quad i \in [n-1].$$

Analogically to the reasoning above, (16.15) ensures that both s_{in} and s_{ni} are equal to 1, if city n is visited directly after (first type) or directly before (second type) city i. □

Corollary 16.13 *The* (TSP) *objective function* (16.62) *and the* (TSP) *constraints* (16.63) – (16.65) *can be reformulated via the entries of matrix* Z_s.

In particular the equality constraints (16.63) for the (TSP) are already assured by $(\mathtt{SDP}_1^{\mathtt{cut}})$.

Lemma 16.14 *The equality constraints* (16.63) *for the* (TSP) *are assured by* (16.27) – (16.31).

Proof. Rewriting (16.66) for $i=j$ gives:

$$s_{ii} = \frac{1}{4}\left[v_{i,i}^{1,2} + v_i^2 + \sum_{k=2}^{n-2}\left(v_{i,i}^{k,k+1} - v_{i,i}^{k-1,k+1} - 1 + v_{i,i}^{k-1,k}\right) - v_i^{n-2} + v_{i,i}^{n-2,n-1} - 1 - v_{i,i}^{n-1,1}\right]$$

$$= \frac{1}{4}\left[\left(-v_i^1 + v_i^2 + v_{i,i}^{1,2} - 1\right) + \sum_{k=2}^{n-2}\left[\left(-v_i^k + v_i^{k+1} + v_{i,i}^{k,k+1} - 1\right) + \left(v_i^{k-1} - v_i^k - v_{i,i}^{k-1,k+1} + 1\right) + \right.\right.$$
$$\left.\left.\left(-v_i^{k-1} + v_i^k + v_{i,i}^{k-1,k} - 1\right)\right] + \left(-v_i^{n-2} + v_i^{n-1} + v_{i,i}^{n-2,n-1} - 1\right) + \left(v_i^1 - v_i^{n-1} - v_{i,i}^{n-1,1} + 1\right)\right],\ i \in [n-1].$$

Now applying (16.27) yields $s_{ii} = 0,\ i \in [n-1]$. In the same way we can show that the equalities (16.28)–(16.31) ensure $s_{nn} = 0$.

Now fixing a city $i,\ i \in [n-1]$, and summing up (16.66) for $j \in [n-1]$, gives

$$\frac{1}{4}\left[n-1-\sum_{j\in[n-1]} v_{i,j}^{1,1} + \sum_{j\in[n-1]} v_{i,j}^{1,2} + \sum_{j\in[n-1]} v_j^2 + \sum_{k=2}^{n-2}\sum_{j\in[n-1]}\left(v_{i,j}^{k,k+1} - v_{i,j}^{k-1,k+1} - v_{i,j}^{k,k} + v_{i,j}^{k-1,k}\right)\right.$$
$$\left. -(n-1)v_i^{n-2} + \sum_{j\in[n-1]} v_{i,j}^{n-2,n-1} - \sum_{j\in[n-1]} v_{i,j}^{n-1,n-1} - \sum_{j\in[n-1]} v_{i,j}^{n-1,1}\right],$$

Finally adding (16.67) yields 1 and hence we proved $\sum_{j\in[n],\, j\neq i} s_{ij} = 1,\ i \in [n-1]$. The remaining (TSP) equalities, i.e. $\sum_{j\in[n],\, j\neq i} s_{ji} = 1,\ i \in [n-1]$, and $\sum_{j\in[n],\, j\neq n} s_{nj} = \sum_{j\in[n],\, j\neq n} s_{jn} = 1$, can be shown in an analogous fashion. □

Finally adding the (TSP) inequalities (16.64) and (16.65), reformulated as linear-quadratic terms in cut variables, to $(\mathtt{SDP}_4^{\mathtt{cut}})$ gives the following relaxation:

$$\max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies (16.27) – (16.31), (16.41), (16.42), (16.64) and (16.65)},\ Z_s \in (\mathcal{E}\cap\mathcal{M})\}. \quad (\mathtt{SDP}_6^{\mathtt{cut}})$$

Corollary 16.15 *The semidefinite programming relaxation* $(\mathtt{SDP}_6^{\mathtt{cut}})$ *is as least as strong as the linear programming relaxation* $(\mathtt{LP}_{\mathtt{TSP}})$*.*

Proof. As $(\mathtt{SDP}_6^{\mathtt{cut}})$ ensures all constraints of $(\mathtt{LP}_{\mathtt{TSP}})$, the claim follows immediately. □

Note that the exact formulation from Theorem 16.4 is in fact (with an appropriately defined cost matrix) also an exact formulation of the (TSP) with only polynomially many constraints.

Let us now propose another formulation of the (TSP) with polynomially many constraints. To do so we first suggest an additional constraint type that connects the (LOP) with the (TSP) variables:

$$s_{ij} - x_{ij} - \frac{1}{n-1}\sum_{\substack{k\in[n],\\ k\neq i}} x_{ki} \le 0, \quad i,j \in [n],\ i \neq j. \qquad (16.69)$$

The term $\frac{1}{n-1}\sum_{k\in[n],\, k\neq i} x_{ki}$ ensures that we are allowed to go from the last city visited back to the starting city and hence to close the tour. Now we exploit this link between (LOP) and (TSP) variables in the following lemma:

Lemma 16.16 *For* $x_{ij} \in \{0,1\},\ i,j \in [n],\ i \neq j$, *and* $s_{ij} \in \{0,1\},\ i,j \in [n],\ i \neq j$, *the subtour elimination constraints* (16.64) *are implicitly satisfied by the* (LOP) *constraints* (16.1) *and* (16.2) *together with the* (TSP) *equalities* (16.63) *and the linking constraints* (16.69).

Proof. The (LOP) constraints ensure that the ordering variables $x_{ij},\ i,j \in [n],\ i \neq j$, describe a feasible ordering of n vertices. Now the linking constraints (16.69) together with the (TSP) equalities further guarantee that the (TSP) variables describe the same ordering as a tour. □

But let us point out that the subtour elimination constraints (16.64) may still help to tighten relaxations where the linking constraints and all other (LOP) and (TSP) constraints are satisfied. Finally we discuss how the linking constraints (16.69) can be reformulated via the entries of matrix Z_s: For $i,j \in [n-1],\ i \neq j$, we just have to subtract the term $\frac{1}{4}\left(1 + v_j^1 - v_i^{n-1} - v_{i,j}^{n-1,1}\right)$ from (16.66) to rule out the edges from the last city visited to the starting city. Hence in this way we account for the term $-\frac{1}{n-1}\sum_{k\in[n],\, k\neq i} x_{ki}$ in (16.69). Now subtracting (16.51) from the adapted version of (16.66) gives:

$$v_j^2 + \sum_{k=2}^{n-2}\left(v_{i,j}^{k-1,k} - v_{i,j}^{k-1,k+1}\right) + v_{i,j}^{n-2,n-1} - v_i^{n-1} \leq 1, \quad i,j \in [n-1],\ i \neq j.$$

Note that these differences of (TSP) and (LOP) variables result in easier expressions in cut variables than the (TSP) variables. Similar simplifications also occur when rewriting (16.69) for $i = n$ and $j = n$:

$$-\sum_{j\in[n-1]} v_j^2 - 2\sum_{k=2}^{n-1} v_i^{k-1} + \sum_{k=2}^{n-2}\sum_{j\in[n-1]}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + (n-1)v_i^{n-2} - \sum_{j\in[n-1]} v_{i,j}^{n-2,n-1} \leq n-3, \quad i \in [n-1],$$

$$-(n-1)v_j^2 + 2\sum_{k=2}^{n-1} v_j^k + \sum_{k=2}^{n-2}\sum_{i=1}^{n-1}\left(v_{i,j}^{k-1,k+1} - v_{i,j}^{k-1,k}\right) + \sum_{i=1}^{n-1}\left(v_i^{n-2} - v_{i,j}^{n-2,n-1}\right) \leq n-3, \quad j \in [n-1].$$

As all constraints valid for the (TSP) also hold for the (LOP) and vice versa, we combine all constraint types discussed in this paper in our strongest relaxation ($\mathtt{SDP}_7^{\mathtt{cut}}$):

$$\begin{aligned} \max\ \{\langle C_s, Z_s\rangle : Z_s \text{ satisfies } &(16.2),\ (16.3),\ (16.27)-(16.31),\ (16.41),\\ &(16.42),(16.64),\ (16.65) \text{ and } (16.69),\ \ Z_s \in (\mathcal{E}\cap\mathcal{M})\ \}. \end{aligned} \qquad (\mathtt{SDP}_7^{\mathtt{cut}})$$

Finally note that all constraint types in ($\mathtt{SDP}_7^{\mathtt{cut}}$) may help to improve the bounds obtained.

Lemma 16.17 *No constraint types in* ($\mathtt{SDP}_7^{\mathtt{cut}}$) *are dominated.*

Proof. We can give an instance for each constraint type, where this particular constraint type helps to further improve the value of the relaxation, even if all other constraints are already satisfied. Most conveniently the respective instances are generated by replicating the particular constraints in the cost function. □

16.4.3 Target Visitation Problem

Another vertex ordering problem is the Target Visitation Problem (TVP) that has been suggested in [125]. It is a composition of the (LOP) and the (TSP), i.e. we are looking for a permutation $(p_1, p_2, \ldots, p_n)$ of n targets with given pairwise weights $w_{ij},\ i,j \in [n]$, and pairwise distances $d_{ij},\ i,j \in [n]$, maximizing the objective function

$$\sum_{\substack{i,j\in[n]\\ i<j}} w_{p_ip_j} - \left(\sum_{i\in[n-1]} d_{p_ip_{i+1}} + d_{p_np_1}\right).$$

As the (LOP) and the (TSP) are special cases of the (TVP), the (TVP) is also NP-hard.

The formulation of the (TVP) was inspired by the use of single unmanned aerial vehicles that have been used increasingly over the last decades. Applications of the (TVP) include environmental assessment, combat search and rescue and disaster relief [125].

Clearly our relaxations $(\mathtt{SDP}_1^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ can also be used to obtain upper bounds for the (TVP), if we define the C_s appropriately. For more information on the (TVP) we refer to our companion paper [161], where we apply the semidefinite relaxations using cut variables from this paper to design an exact semidefinite optimization approach that is able to obtain reasonable upper and lower bounds on a variety of benchmark instances with up to 50 targets. Also note that Hildenbrandt et al. [153] recently conducted the first polyhedral study of the (TVP) polytope. They present several possible IP-models for the (TVP) and compare them to their usability for branch-and-cut approaches. Based on their findings Hildenbrandt et al. [152] are currently developing an exact IP approach for the (TVP) that has strong potential to solve large-scale (TVP) instances to optimality.

16.5 Numerical Examples

In this section we aim to compare the relaxation strength of our new relaxations $(\mathtt{SDP}_1^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ with other relaxations from the literature for the (LOP) (Subsection 16.5.1) and the (TSP) (Subsection 16.5.2) respectively. It is not difficult to create small but hard instances by just considering the facets of the (LOP) and the (TSP) polytopes for small dimensions. We will see below that our new relaxations are worse than the best relaxations for the respective problem type. However notice that $(\mathtt{SDP}_1^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ are the only relaxations considered that can be applied to both the (LOP) and the (TSP). We exploited exactly this property in our companion paper [161].

16.5.1 Comparison of Relaxations for the Linear Ordering Problem on Small Facets

In this subsection we compare the $(\mathtt{LP}_{\mathtt{LOP}})$ to the semidefinite relaxations from Section 16.3. We consider the full description of the linear ordering polytope in small dimensions, and try to recover the correct right hand side of the classes of facets for $n \in \{6, 7\}$. The members of each class are equal modulo a permutation of the vertices, and we need therefore only consider one representative per class. The facets are collected under `http://comopt.ifi.uni-heidelberg.de/software/SMAPO/lop/lop.html`. As usual, n denotes the number of vertices and *opt* gives the optimal solution. All relaxations are solved to optimality using the standard settings of SEDUMI [289]. Table 16.1 summarizes our results.

We observe that the semidefinite relaxations provide a substantial improvement over $(\mathtt{LP}_{\mathtt{LOP}})$ in the approximation of the linear ordering polytope of small dimensions. Furthermore the relaxations on ordering variables seem in general tighter than the relaxations on cut variables. Nonetheless the strongest relaxation using ordering variables $(\mathtt{SDP}_4^{\mathtt{ord}})$ does not dominate relaxations $(\mathtt{SDP}_5^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ (on these instances), see facet number 14.

16.5.2 Comparison of Relations for the Traveling Salesman Problem on Small Facets

For the (TSP) we consider instances corresponding to classes of facets of the symmetric (TSP) on 8 vertices [67]. The facets are collected under `http://comopt.ifi.uni-heidelberg.de/software/SMAPO/tsp/tsp.html`. These benchmark instances were also used in [77, 84, 86, 102]. Cvetković et al. [77] suggested a semidefinite relaxation that is however dominated by the Held-Karp bound [122]. De Klerk et al. [86] and de Klerk and Sotirov [84] proposed SDP relaxations for the symmetric (TSP) that can be obtained via an SDP relaxation of the more general quadratic assignment problem and are motivated by the theory of association schemes. Finally Fischer [102] tested her polyhedral linear relaxation for the

Linear Ordering Problem Instances Constructed from Facet Defining Inequalities													
Facet	n	Opt	(LP)	$\mathtt{SDP}_1^{\mathtt{ord}}$	$\mathtt{SDP}_2^{\mathtt{ord}}$	$\mathtt{SDP}_3^{\mathtt{ord}}$	$\mathtt{SDP}_4^{\mathtt{ord}}$	$\mathtt{SDP}_1^{\mathtt{cut}}$	$\mathtt{SDP}_2^{\mathtt{cut}}$	$\mathtt{SDP}_3^{\mathtt{cut}}$	$\mathtt{SDP}_4^{\mathtt{cut}}$	$\mathtt{SDP}_5^{\mathtt{cut}}$	$\mathtt{SDP}_7^{\mathtt{cut}}$
FC3	6	7	7.5	7.35	7	7	7	7.32	7	7	7	7	7
FC 4	6	8	8.5	8.35	8	8	8	8.42	8.02	8	8	8	8
FC5	6	8	8.5	8.35	8	8	8	8.42	8.02	8	8	8	8
FC3	7	7	7.5	7.35	7	7	7	7.57	7.35	7	7	7	7
FC4, 20	7	8	8.5	8.35	8	8	8	8.76	8.35	8.20	8.03	8.03	8.03
FC5	7	9	9.5	9.37	9	9.09	9	9.88	9.37	9.11	9.01	9.01	9.01
FC6	7	9	9.5	9.37	9	9	9	9.84	9.30	9.09	9.02	9.02	9.02
FC7	7	10	10.5	10.37	10.11	10	10	10.99	10.46	10.23	10.07	10.07	10.07
FC8	7	10	10.5	10.37	10.19	10	10	10.92	10.42	10.28	10.08	10.08	10.08
FC9	7	10	10.5	10.37	10.11	10	10	10.93	10.43	10.26	10.06	10.06	10.06
FC10, 25	7	9	9.5	9.37	9.06	9	9	9.59	9.05	9.13	9	9	9
FC11	7	10	10.5	10.37	10	10	10	11.18	10.36	10.24	10.03	10.02	10.02
FC12	7	10	10.5	10.37	10	10.03	10	10.96	10.42	10.28	10.08	10.08	10.08
FC13	7	10	10.5	10.37	10.19	10	10	10.99	10.43	10.29	10.08	10.08	10.08
FC14	7	10	10.5	10.35	10.35	10.24	10.22	10.96	10.50	10.51	10.21	10.21	10.21
FC15	7	11	11.5	11.37	11.22	11	11	12.11	11.52	11.38	11.16	11.15	11.15
FC16	7	11	11.5	11.37	11.22	11	11	12.50	11.58	11.46	11.16	11.15	11.15
FC17	7	13	13.5	13.40	13	13	13	13.71	13.26	13.05	13	13	13
FC18	7	14	14.5	14.40	14.17	14.04	14	14.88	14.35	14.10	14	14	14
FC19	7	14	14.5	14.40	14.10	14.01	14	15.00	14.33	14.19	14.01	14.01	14.01
FC21	7	9	9.5	9.37	9	9.01	9	9.88	9.08	9.11	9.01	9.01	9.01
FC22	7	10	10.5	10.37	10.11	10	10	10.99	10.13	10.23	10.07	10.07	10.07
FC23	7	10	10.5	10.37	10.19	10	10	10.92	10.19	10.28	10.08	10.08	10.08
FC24	7	10	10.5	10.37	10.11	10	10	10.93	10.15	10.26	10.06	10.06	10.06
FC26	7	11	11.5	11.37	11.23	11	11	12.11	11.24	11.40	11.16	11.15	11.15
FC27	7	13	13.5	13.40	13	13	13	13.71	13.03	13.05	13	13	13

Table 16.1: Marginal improvement of various semidefinite relaxations as compared to the linear relaxation on facets of the linear ordering polytope for $n = 6$ and $n = 7$.

quadratic (TSP) on the facets of the symmetric (TSP). In Table 16.2 we compare $(\mathtt{SDP}_1^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ with the Held-Karp bound [137] and the other relaxations described above.

We already know from Corollary 16.15 that $(\mathtt{SDP}_6^{\mathtt{cut}})$ dominates the Held-Karp bound [137] and hence also the SDP relaxation from [77]. Additionally we observe that on the one hand $(\mathtt{SDP}_4^{\mathtt{cut}})$ dominates the semidefinite relaxation from [86] on all instances considered and on the other hand our strongest relaxation $(\mathtt{SDP}_7^{\mathtt{cut}})$ is dominated by the relaxations from [84] and [102]. It would be interesting to investigate, if general results between the strength of the different relaxations could be established. Finally note that $(\mathtt{SDP}_1^{\mathtt{cut}})$– $(\mathtt{SDP}_7^{\mathtt{cut}})$ are the only relaxations from this study that can also be applied to the asymmetric and even to the non-Euclidean (TSP)

16.6 Conclusion

In this paper we suggested a formulation of the Linear Ordering Problem with the help of cut variables and conducted a polyhedral study of the corresponding polytope. In this way we improved the semidefinite relaxation proposed by Newman [237] that proved well-suited for obtaining strong approximation results for the Linear Ordering Problem. We elaborated on the connection of this model using cut variables to the Max Cut problem and related our new semidefinite relaxations to other linear and semidefinite relaxations for the Linear Ordering Problem and for the Traveling Salesman Problem.

In the companion paper [161] we show that semidefinite relaxations using cut variables can be applied to obtain reasonable bounds for difficult, large-scale combinatorial optimization problems. As our relaxations provide new polynomial-time convex approximations of the (LOP) and (TSP) with a rich mathematical structure, we hope that they may be helpful to improve approximation results for vertex ordering problems. Their close relation to the Max Cut problem could be a useful property in that direction and therefore a possible subject of future research.

Travelling Salesman Problem Instances Constructed from Facet Defining Inequalities for the TSP Polytope on 8 Vertices													
Facet ($n = 8$)	Opt	Held-Karp [137]	SDP [77]	SDP [86]	SDP [84]	LP [102]	$\mathtt{SDP}_1^{\mathtt{cut}}$	$\mathtt{SDP}_2^{\mathtt{cut}}$	$\mathtt{SDP}_3^{\mathtt{cut}}$	$\mathtt{SDP}_4^{\mathtt{cut}}$	$\mathtt{SDP}_6^{\mathtt{cut}}$	$\mathtt{SDP}_7^{\mathtt{cut}}$
FC3	2	2	2	2	2	2	1.586	1.634	2	2	2	2
FC4	2	2	1.098	1.628	2	2	1.098	1.300	1.540	1.629	2	2
FC5	2	2	1.172	1.172	1.893	2	1.172	1.172	1.172	1.172	2	2
FC6	10	9	8.507	8.671	10	10	7.324	8.090	8.241	8.671	9	9
FC7	10	9	9	9	10	10	7.890	8.419	8.623	9	9	9
FC8	10	9	8.566	8.926	10	10	7.529	8.227	8.443	8.926	9	9
FC9	10	9	8.586	8.586	10	9	6.929	7.741	8.053	8.586	9	9
FC10	10	9	8.570	8.926	10	9.500	7.440	8.174	8.485	8.926	9	9
FC11	10	9	9	9	10	9.600	7.883	8.511	8.683	9	9	9
FC12	10	9	8.411	8.902	10	10	6.970	7.960	8.331	8.902	9	9
FC13	10	9	8.422	8.899	10	9	6.968	7.926	8.359	8.899	9	9
FC14	0	0	0	0	0	0	-0.707	-0.290	-0.282	0	0	0
FC15	12	11	10.586	10.667	11.777	11	9.757	10.206	10.351	10.667	11	11
FC16	13	12	12	12	12.777	12	10.962	11.244	11.505	12	12	12
FC17	14	$12\frac{2}{3}$	12.408	12.444	13.663	13	11.063	11.574	11.892	12.444	12.667	12.667
FC18	16	14	14	14.078	15.651	16	12.096	13.258	13.434	14.115	14.126	14.126
FC19	18	16	16	16	17.824	18	13.573	14.780	15.219	16	16	16
FC20	18	16	16	16	17.698	18	13.852	14.966	15.332	16	16	16
FC21	18	16	16	16	18	17.500	14.273	15.194	15.513	16	16	16
FC22	18	16	15.185	15.926	18	17	13.207	14.607	15.211	15.937	16.023	16.023
FC23	20	18	18	18.025	19.568	19	16.356	17.124	17.561	18.034	18.044	18.044
FC24	22	20	20	20	21.287	20	19.515	19.715	20	20	20	20
FC25	26	23	23	23.033	25.460	25	20.249	21.785	22.355	23.042	23.055	23.055
FC26	38	35	34.586	34.739	37.141	35	32.082	33.052	33.759	34.739	35	35

Table 16.2: Comparison of the relaxation strength of various linear and semidefinite relaxations on facets of the traveling salesman polytope for $n = 8$.

Chapter 17

An Exact Approach for the Combined Cell Layout Problem

Authors: Philipp Hungerländer and Miguel F. Anjos

Abstract: We propose an exact solution method based on semidefinite optimization for simultaneously optimizing the layout of two or more cells in a cellular manufacturing system in the presence of parts that require processing in more than one cell. To the best of our knowledge, this is the first exact method proposed for this problem. We consider single-row and directed circular (or cyclic) cell layouts but the method can in principle be extended to other layout types. Preliminary computational results suggest that optimal solutions can be obtained for instances with 2 cells and up to 60 machines.

Keywords: Facilities planning and design; Flexible manufacturing systems; Cell layout; Cyclic layout; Semidefinite Programming; Global Optimization

17.1 Introduction

In a cellular manufacturing system, the parts that are similar in their processing requirements are grouped into part families, and the machines needed to process the parts are grouped into machine cells. Ideally, cells should be designed in such a way that the part families are fully processed in a single machine-cell so that the machine-cells are mutually independent with no inter-cell movement. In a real-world situation however it may be impractical and/or uneconomical to require mutually independent cells. The consequence is that some parts will require processing in more than one cell.

This paper is concerned with finding the optimal layout of each cell in the presence of parts that require processing in more than one cell. Cell layout usually takes place after the machine cells are determined, see e.g. [71]. While much research has been done on the cell formation problem, the layout of machines within the cells has received less attention. We propose an exact solution method that simultaneously minimizes the inter-cell and intra-cell material handling costs for a given machine-cell assignment. To the best of our knowledge, this is the first exact method proposed for this problem. The method is based on the application of semidefinite optimization models and algorithms. The machines of each cell can be arranged in a row or on a circle. We denote this problem as Combined Cell Layout Problem (`CCLP`). The two types of cells that we consider have been previously studied in the literature but independently rather than combined.

The first type of cell layout is single-row layout. The Single-Row Facility Layout Problem (`SRFLP`), sometimes called the one-dimensional space allocation problem [252], consists of finding a permutation of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines is minimized. This problem arises for example as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle (AGV) travelling in both directions on a straight-line path [150]. Other applications are the arrangement of rooms along a corridor in hospitals, supermarkets, or offices [283] and the assignment of airplanes to gates in an airport terminal [291].

The second type is circular layout. The Directed Circular Facility Layout Problem (DCFLP) seeks to arrange the machines on a circle so as to minimize the total weighted sum of the center-to-center distances measured in the clockwise direction. Notice that three facility layout problems that are extensively discussed in the literature, namely the Equidistant Unidirectional Cyclic Layout Problem, the Balanced Unidirectional Cyclic Layout Problem and the Directed Circular Arrangement Problem are special cases of the (DCFLP) Problem (for details see [157]).

17.2 Problem Description and Matrix-Based Formulation

We assume that only one part is produced at a time and that we want to produce more than one part using the same layout. Hence we want to find the arrangement of the machines that minimizes the overall material flow. Let $\mathcal{P} = \{1, \dots, P\}$ be the set of part types. Each part type is associated to a different process plan S_p that gives the sequence in which part type p visits the machines. (For simplicity we will assume that the process plan of each part type is unique although in general this may not be the case.) So for each part, the cost function is different and dependent on n_{ij}^p, the number of moves part type p makes from machine i to machine j per period of time. n_{ij}^p is easily determined from S_p. Furthermore n_p denotes the number of parts of type p that are processed per period of time. Hence we assume that each part forms a percentage of the total production, this gives us weights to aggregate the various cost functions into a single cost function that optimizes the layout for all the parts at once. Now the total number of parts of type $p \in \mathcal{P}$ that flow from machine i to machine j per period of time is obtained as $f_{ij}^p = n_p n_{ij}^p, i, j \in \mathcal{M},\ i \neq j,\ p \in \mathcal{P}$, where $\mathcal{M} = \{1, \dots, M\}$ is the set of machines. The total part flow from machine i to machine j per time period can be expressed as $f_{ij} = \sum_{p \in \mathcal{P}} f_{ij}^p,\ i, j \in \mathcal{M},\ i \neq j$.

Each machine has a given integer length and is pre-assigned by the function $c:\ \mathcal{M} \to \mathcal{C}$ to one of the cells in the set $\mathcal{C} := \{1, \dots, C\}$. Furthermore each cell is associated to one of two different layout types: $\ell:\ \mathcal{C} \to \{\ \texttt{(SRFLP)}, \texttt{(DCFLP)}\}$. The parts enter and exit each cell at a pre-specified machine of that cell, specified by $s:\ \mathcal{C} \to \mathcal{M}$, where we assume w.l.o.g. that $s(i) < j,\ i \in \mathcal{C},\ c(s(i)) = c(j),\ s(i) \neq j$. Finally we have given an integer distance $e_{ij},\ i, j \in \mathcal{C},\ i < j$ for each pair of cells.

To model the arrangement of the machines in a cell we introduce ordering variables $y_{ij},\ i, j \in [n],\ i < j,\ c(i) = c(j)$,

$$y_{ij} = \begin{cases} 1, & \text{if machine } i \text{ lies left of machine } j, \\ -1, & \text{otherwise.} \end{cases} \tag{17.1}$$

Any feasible ordering of the machines has to fulfill the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1, \quad i, j, k \in [n],\ i < j < k,\ c(i) = c(j) = c(k), \tag{17.2}$$

It is well-known that the 3-cycle inequalities together with integrality conditions on the ordering variables suffice to describe feasible orderings, see e.g. [296, 306].

For the (SRFLP) the center-to-center distances between machines d_{ij} can be encoded using products of ordering variables [22]:

$$d_{ij} = \frac{1}{2}(l_i + l_j) - \sum_{\substack{k \in [n] \\ k<i,\ c(k)=c(i)}} l_k y_{ki} y_{kj} + \sum_{\substack{k \in [n] \\ i<k<j,\ c(k)=c(i)}} l_k y_{ik} y_{kj} - \sum_{\substack{k \in [n] \\ k>j,\ c(k)=c(i)}} l_k y_{ik} y_{jk},$$
$$i, j \in [n],\ i < j,\ c(i) = c(j),\ \ell(c(i)) = \text{"(SRFLP)"}.$$

As one machine is fixed to be "first" in any cell (the one where the parts enter and exit the cell), we set $y_{ij} = 1,\ j \in [n],\ s(c(j)) = i,\ i < j$.

Similar reasonings lead to a linear expression in terms of the ordering variables for the center-to-center distances between machines in a circular layout and between machines in different cells. The details are omitted due to space limitations and will be provided in a forthcoming paper.

Hence we can model all distances between machines in the (CCLP) as linear-quadratic expressions in ordering variables. To reformulate the (CCLP) we define an appropriate cost matrix C, collect the ordering variables in a vector y and introduce a matrix $Z := \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top$ containing all products of ordering variables. Letting t denote the total number of ordering variables, we can give a matrix-based formulation of the (CCLP).

Theorem 17.1 *Minimizing $\langle C, Z \rangle$ over $y \in \{-1, 1\}^t$ fulfilling* (17.2) *solves the* (CCLP).

Proof. The inequalities (17.2) together with the integrality conditions on y suffice to induce a feasible layout for both single-row and circular layout and the definition of C ensures that the distances between machines are computed correctly. □ □

17.3 Semidefinite Relaxations and Computational Experience

We apply standard techniques to the matrix-based formulation of the (CCLP) proposed in Theorem 17.1 to construct (SDP) relaxations over the multi-level quadratic ordering polytope

$$\mathcal{P}_{MQO} := \operatorname{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top \ :\ y \in \{-1, 1\}^t,\ y \text{ satisfies } (17.2) \right\}.$$

Similar relaxations have already been successfully applied to combinatorial optimization problems arising in the area of graph drawing [48, 61, 62, 63, 156].

The core of our semidefinite approach is to solve our semidefinite relaxation ($\mathtt{SDP}_4$) by using the bundle method in conjunction with interior point methods. The resulting fractional solutions constitute lower bounds. By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e. integer solutions that describe a feasible layout of the machines. Hence, in the end we have some feasible solution, together with a proof how far this solution could possibly be from the true optimum. We will discuss these two steps in more detail in a forthcoming paper.

We report the results for different computational experiments with our semidefinite relaxation ($\mathtt{SDP}_4$). All benchmark instances used can be downloaded together with the best layouts found from `http://anjos.mgi.polymtl.ca/flplib`. The (SDP) computations were conducted on an Intel Xeon E5160 (Dual-Core) with 2 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7.

We set the number of parts P to 50. For each part type we generated the process plans S_p as follows: We took a random integer number r_p between 1 and M from a uniform distribution to determine the number of machines to be visited. Then we computed r_p additional integer numbers from the uniform distribution $U(1, M)$ that represented the handling order of type p. Notice that each machine is allowed to occur more than once but not consecutively. Finally n_P is chosen from uniform distributions $U(1, 10)$,

$U(1, 50)$ and $U(1, 100)$ respectively to model low, medium and high variations in the part flows (for details see Tansel and Bilen [293]). We marked the instances with the letters L, M and H to identify their type of variation.

We generated 10 instances for each $\mathcal{M} \in \{20, 30, 40, 50, 60\}$ and for each variation type. The integer length of the machines are taken from $U(1, 10)$. We haven given two cells, one has a single-row and the other one a circular layout. The machines are randomly assigned to the cells. We consider here only the case that both cells have the same number of machines but the method can handle any number of machines per cell. We randomly pick the machine at which the parts enter and exit the cell (for the (SRFLP) this machine is fixed to be the first in the ordering). Finally the integer distance between the two cells e_{12} is chosen from $U(20, 40)$.

We summarize our computational results in Table 17.1. For the instances under consideration, we obtained globally optimal solutions in reasonable time.

17.4 Conclusion

We proposed an exact solution method based on semidefinite optimization for optimizing the layout of multiple cells in a cellular manufacturing system in the presence of parts that require processing in more than one cell. Our preliminary computational results suggest that optimal solutions can be obtained for instances with 2 cells and up to 60 machines. While we only considered single-row and directed circular cell layouts, the method can in principle be extended to other layout types and to a larger number of cells.

Instance	Time	Instance	Time	Instance	Time
CR20_L1	4.7	CR20_M1	4.2	CR20_H1	13.3
CR20_L2	3.7	CR20_M2	4.9	CR20_H2	3.2
CR20_L3	3.7	CR20_M3	6.6	CR20_H3	42.1
CR20_L4	3.6	CR20_M4	3.5	CR20_H4	3.4
CR20_L5	4.0	CR20_M5	27.0	CR20_H5	44.5
CR20_L6	3.4	CR20_M6	4.5	CR20_H6	3.1
CR20_L7	3.6	CR20_M7	9.8	CR20_H7	3.4
CR20_L8	3.3	CR20_M8	2.8	CR20_H8	2.4
CR20_L9	6.9	CR20_M9	1.9	CR20_H9	4.2
CR20_L10	4.3	CR20_M10	3.5	CR20_H10	4.2
CR30_L1	28.3	CR30_M1	1:09	CR30_H1	3:03
CR30_L2	59.8	CR30_M2	7:12	CR30_H2	16.8
CR30_L3	55.0	CR30_M3	2:55	CR30_H3	27.7
CR30_L4	22.3	CR30_M4	17.6	CR30_H4	44.9
CR30_L5	14.8	CR30_M5	26.1	CR30_H5	20.8
CR30_L6	19.2	CR30_M6	27.9	CR30_H6	29.5
CR30_L7	19.1	CR30_M7	54.7	CR30_H7	43.3
CR30_L8	41.0	CR30_M8	1:45	CR30_H8	23.4
CR30_L9	20.8	CR30_M9	24.0	CR30_H9	56.6
CR30_L10	20.4	CR30_M10	1:08	CR30_H10	51.7
CR40_L1	1:37	CR40_M1	3:43	CR40_H1	3:13
CR40_L2	3:01	CR40_M2	10:38	CR40_H2	5:38
CR40_L3	3:03	CR40_M3	4:40	CR40_H3	3:40
CR40_L4	4:39	CR40_M4	7:00	CR40_H4	4:34
CR40_L5	7:29	CR40_M5	1:18:45	CR40_H5	4:58
CR40_L6	13:00	CR40_M6	10:44	CR40_H6	2:56
CR40_L7	3:29	CR40_M7	3:00	CR40_H7	5:09
CR40_L8	2:06	CR40_M8	4:30	CR40_H8	4:22
CR40_L9	8:06	CR40_M9	5:54	CR40_H9	6:29
CR40_L10	5:16	CR40_M10	2:38	CR40_H10	2:42
CR50_L1	19:12	CR50_M1	11:49	CR50_H1	14:00
CR50_L2	9:06	CR50_M2	18:02	CR50_H2	22:53
CR50_L3	16:08	CR50_M3	12:26	CR50_H3	30:53
CR50_L4	14:52	CR50_M4	34:32	CR50_H4	17:28
CR50_L5	13:20	CR50_M5	14:50	CR50_H5	14:14
CR50_L6	27:08	CR50_M6	11:50	CR50_H6	33:06
CR50_L7	11:48	CR50_M7	19:02	CR50_H7	17:11
CR50_L8	31:42	CR50_M8	24:53	CR50_H8	24:09
CR50_L9	12:20	CR50_M9	25:14	CR50_H9	15:52
CR50_L10	27:39	CR50_M10	16:38	CR50_H10	37:56
CR60_L1	2:52:26	CR60_M1	55:56	CR60_H1	1:08:02
CR60_L2	1:20:11	CR60_M2	2:01:31	CR60_H2	1:42:34
CR60_L3	2:28:11	CR60_M3	1:10:07	CR60_H3	1:46:35
CR60_L4	1:08:07	CR60_M4	45:24	CR60_H4	49:55
CR60_L5	1:55:20	CR60_M5	1:04:58	CR60_H5	59:58
CR60_L6	1:31:03	CR60_M6	1:31:18	CR60_H6	1:10:23
CR60_L7	1:19:30	CR60_M7	46:23	CR60_H7	1:12:44
CR60_L8	1:24:40	CR60_M8	1:02:08	CR60_H8	2:38:21
CR60_L9	49:01	CR60_M9	1:14:03	CR60_H9	1:26:53
CR60_L10	1:12:44	CR60_M10	1:42:33	CR60_H10	2:21:55

Table 17.1: Results for instances with up to 60 machines that are assigned to 2 cells, each containing 30 machines. One cell has a single-row layout and the other one a circular layout. All instances could be solved to global optimality. The running times are given in sec or min:sec or in h:min:sec, respectively.

Part IV

Extensions, Conclusion and Outlook

Chapter 18

Extensions

This chapter contains findings that I could not yet convert into published or submitted research papers during the time of my PhD. Nonetheless these findings fit very well to the topic of the thesis and in my opinion have the potential for future publications in high-quality journals. The first two sections are concerned with new models and results in the area of facility layout planning. In Section 18.1 we define the weighted Linear Ordering Problem that considers individual node weights additional to pairwise weights. We show that the weighted Linear Ordering Problem generalizes row layout problems by allowing for asymmetric cost structures. Additionally we argue that the optimal ordering obtained is a worthwhile alternative to the optimal solution of the Linear Ordering Problem in many applications. In Section 18.2 we prove that for equidistant row layout instances the objective value of the optimal solution with m rows is less or equal to the objective value of the optimal single-row layout divided by m. Furthermore we show that for non-equidistant row layout instances no such relation between the objective values of single- and multi-row layouts exists. Finally Section 18.3 deals with an application in logistics. The Traveling Salesman Problem with forbidden neighbourhoods (TSPN) asks for a tour of minimal length in which points traversed successively have a given minimal distance from each other. This new variant of the TSP is motivated by an application in beam melting, where the points to be fused are arranged in a regular way and can be represented as grid graphs. We examine the length and structure of the optimal tours for different types of forbidden neighborhoods that are the most interesting ones with respect to the application mentioned above. Note that the results from Sections 18.2 and 18.3 originate from joint works with Anja and Frank Fischer and Anja Fischer respectively.

18.1 The Weighted Linear Ordering Problem

In this section we introduce, motivate and analyze an extension of the Linear Ordering Problem `(LOP)`. We will argue that the weighted Linear Ordering Problem `(wLOP)` is preferable compared to the `(LOP)` from an application point of view. An instance of the `(wLOP)` consists of n objects (or nodes) with given individual object (or node) weights $w_1, \ldots, w_n$ and pairwise object (or edge) weights $w_{ij},\ i, j \in [n],\ [n] := \{1, 2, \ldots, n\}$. The associated optimization problem can be written down as

$$\max_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ \pi(i) < \pi(j)}} w_{ij} d_{ij}^{\pi}, \tag{18.1}$$

where Π_n is the set of permutations of the objects $[n]$ and d_{ij}^{π} is the sum of the weights of the objects located between object i and object j in the permutation Π_n (plus halves of the weights of i and j):

$$d_{ij}^{\pi} = \begin{cases} \frac{w_i+w_j}{2} + \sum\limits_{\substack{k\in[n] \\ \pi(i)<\pi(k)<\pi(j)}} w_k, & \pi(i) < \pi(j), \\ 0, & \text{otherwise.} \end{cases}$$

The (LOP) can be expressed as the corresponding problem where d_{ij}^{π} is set to

$$d_{ij}^{\pi} = \begin{cases} 1, & \pi(i) < \pi(j), \\ 0, & \text{otherwise.} \end{cases}$$

In the following we will elaborate on the connections of the (wLOP) and the (LOP) and additionally relate the (wLOP) to graph layout and facility layout planning problems, namely Linear Arrangement (LA) and the Single-Row Facility Layout Problem (SRFLP). We will also briefly review the main applications areas of and the best algorithmic approaches to these three combinatorial optimization problems.

The Linear Ordering Problem (LOP). Ordering problems associate to each ordering (or permutation) of the set $[n]$ a profit and the goal is to find an ordering of maximum profit. In the simplest case of the Linear Ordering Problem (LOP), this profit is determined by those pairs $(u, v) \in [n] \times [n]$, where u comes before v in the ordering. Thus in its matrix version the (LOP) can be defined as follows. Given an $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that $\sum\limits_{\substack{i,j\in[n] \\ i<j}} w_{\pi(i),\pi(j)}$, is maximized. Equivalently, we can interpret w_{ij} as weights of a complete directed graph G with vertex set $V = [n]$. A tournament consists of a subset of the arcs of G containing for every pair of nodes i and j either arc (i, j) or arc (j, i), but not both. Then the (LOP) consists of finding an acyclic tournament, i.e. a tournament without directed cycles, of G of maximum total edge weight.

For finding the optimal triangulation of W the diagonal entries are irrelevant. The orderings compare also in the same way if we add some constant terms to w_{ij} and w_{ji}. However in both cases the quality of the bound does improve for larger entries. Hence to allow a fair comparison of the bounds obtained all (LOP) instances are transformed to their unique normal form where

1. all entries are integral and nonnegative

2. $w_{ii} = 0,\ i \in [n]$ and

3. $\min\{w_{ij}, w_{ji}\} = 0,\ i, j \in [n], i < j$.

Note that for the (wLOP) the above transformation may change the ranking of the orderings and hence may lead to a different optimal solution. Thus there exists no normal form for the (wLOP).

The (LOP) is equivalent to the Acyclic Subdigraph Problem (ASP) and the Feedback Arc Set Problem (FASP). It is well known to be NP-hard [112] and it is even NP-hard to approximate (LOP) within the factor $\frac{65}{66}$ [238]. Surprisingly there is not much known about heuristics with approximation guarantees. If all entries of W are nonnegative, a $\frac{1}{2}$-approximation is trivial, but no better polynomial time approximation is known. To narrow this quite large gap $[\frac{1}{2}, \frac{65}{66}]$ is a challenging open problem. For further details on approximation results for the (LOP) we refer to Chapter 16. Furthermore we refer to Chapter 5 for details on exact (and heuristic) methods for solving the (LOP).

The (LOP) arises in a large number of applications in such diverse fields as economy, sociology (determination of ancestry relationships [119]), graph drawing (one sided crossing minimization [173]), archaeology, scheduling (with precedences [39]), assessment of corruption perception [1] and ranking in sports tournaments. Additionally problems in the context of mathematical psychology and the theory of social choice can be formulated as linear ordering problems, see [104] for a survey.

In 1959 Kemeny [182] posed the first application of the (LOP) (Kemeny's problem) concerning the aggregation of individual orderings to a common one in the best possible way. Note that (LOPs) occur whenever rankings of some objects are to be determined, e.g. finding a common ranking based on $m\binom{n}{2}$ pairwise comparison experiments conducted by m persons for n objects. The optimal (LOP) solution then provides the ranking of the objects with the fewest contradictions to the individual rankings. In Kemeny's problem the relative ranking of the objects by each person is additionally consistent (which is not assumed in the example above). Similar applications of the (LOP) occur in the context of voting (see e.g. [286]) and have already been studied in the 18th century by Condorcet [73].

The probably most established application of the (LOP) is the triangulation of input-output matrices of an economy. Leontief [205, 206] was awarded the Nobel Prize in 1973 for his research on input-output analysis. Input-output analysis is a field of practical importance in economics as it is used for forecasting the development of industries, for structural planning and for structural comparisons between different countries [60, 155]. Its central component is the input-output table or matrix which contains the transactions between the different branches or sectors of an economy in a certain time period. Now triangulation of an input-output matrix allows a descriptive analysis of the transactions between the sectors as it establishes a hierarchy of all sectors such that the amount of flow incompatible with this hierarchy is as small as possible.

There are also several further problems that are closely related to the (LOP). The (LOP) with cumulative costs considers, like the (wLOP), arc weights $w_{ij},\ i,j \in [n]$, and node weights $w_i,\ i \in [n]$, but minimizes a different objective function $\sum_{i=1}^{n} \alpha_i$ with

$$\alpha_i = w_i + \sum_{\substack{j \in [n] \\ \pi(i) < \pi(j)}} w_{ij}\alpha_j.$$

This problem has an application in the area of mobile phone telecommunication and the best algorithm to date is in fact an enumerative scheme that solves instances with up to 16 objects and outperforms a corresponding mixed-integer programming approach, see [37] for more details. This figures already showcase that the additional consideration of node weights makes the problem much harder in general. Heuristic algorithms for the (LOP) with cumulative costs are presented in [92]. The Coupled Task Problem is concerned with scheduling n jobs each of which consists of two subtasks. Furthermore between the execution of these subtasks a delay is required. We refer to [33] for an optimization model for this problem that successfully uses linear ordering variables together with additional constraints for modelling the processing times and delays properly. Finally the Target Visitation Problem that was suggested in [125] is a composition of the (LOP) and the Traveling Salesman Problem, i.e. we are looking for a permutation $(p_1, p_2, \ldots, p_n)$ of n targets with given pairwise weights $w_{ij},\ i,j \in [n]$, and pairwise distances $z_{ij},\ i,j \in [n]$, maximizing the objective function

$$\sum_{\substack{i,j \in [n] \\ i<j}} w_{p_i p_j} - \left(z_{0p_1} + \sum_{i=1}^{n-1} z_{p_i p_{i+1}} + z_{p_n 0} \right).$$

For further details on the (TVP) we refer to Chapters 15 and 16.

Evidently the (wLOP) has the same applications as the (LOP). It remains to argue why the (wLOP) is preferable to the (LOP): even for a small example such as the one depicted in Figure 18.1 the (LOP) results in multiple solutions that are equivalent w.r.t. their objective values, even though the bottom right one is clearly preferable from the application point of view as it represents the collective preferences most accurately. Intuitively, we prefer that objects with high pairwise weights are positioned far away from each other and objects with no or few weights are placed in the middle part of the ordering. Such orderings can be obtained by taking the pairwise (LOP) preferences, setting the weights of all nodes to 1 and solving the (wLOP) as this procedure allows to consider the distances between objects in the objective function. We also compare the optimal (LOP) and (wLOP) orderings on the following well-known data set containing pairwise preferences between 10 different brands of beer [259] in Figure 18.2.

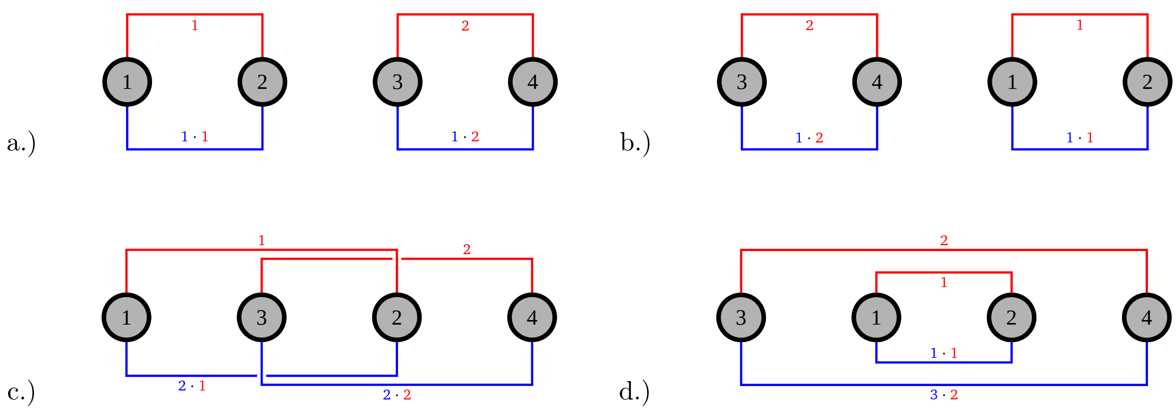


Figure 18.1: We are given 4 objects and the edge weights $w_{12} = 1,\ w_{34} = 2$. We display four optimal (LOP) solutions with benefit of $1+2=3$. Additionally we state the corresponding (wLOP) costs, where we set the weights of all nodes to 1. The ordering in d.) is the optimal (wLOP) solution and at the same time the preferable ranking from a practical point of view. Hence already this toy example illustrates that the (wLOP) is especially favorable to the (LOP) when considering sparse instances with few pairwise weights.

Additionally the (wLOP) is of theoretical interest as it extends the (ASP) and the (FASP) such that not only arc weights but also node weights can be considered. In the following paragraph we show that the (wLOP) can also be interpreted as a generalization of two well-known layout problems.

Linear Arrangement and Single-Row Layouts. Minimum Linear Arrangement (LA) belongs to the class of graph layout problems and can be defined as follows: Given an undirected graph $G(V, E)$, find a permutation $\pi : V \to [n]$ minimizing $\sum_{i,j \in E} |\pi(i) - \pi(j)|$. (LA) can be modeled as (quadratic) ordering problem

$$\min \sum_{(i,j)\in E} \sum_{k\in[n]} x_{ik}x_{kj},$$

where the ordering variables $x_{ij},\ i,j \in [n]$, satisfy the following constraints:

$$x_{ij} + x_{ji} = 1,\ \ i \neq j \in [n], \qquad x_{ij} + x_{jk} + x_{ki} \in \{1,2\},\ \ i,j,k \in [n], \qquad x_{ij} \in \{0,1\},\ \ i \neq j \in [n].$$

The term $\sum\limits_{k} x_{ik}x_{kj}$ 'counts', how many nodes k lie between i and j in the ordering π defined by the vector x collecting all ordering variables.

(LA) is NP-hard [112], even if the underlying graph is bipartite [111] and was originally proposed by Harper [131, 132] to develop error-correcting codes with minimal average absolute errors. If we allow pairwise weights (or connectivities) $w_{ij},\ i,j \in [n],\ i < j$, weighting the relation of nodes, (LA) generalizes

$$W = \begin{bmatrix} 0, 2, 2, 3, 3, 5, 5, 5, 4, 4 \\ 4, 0, 3, 3, 4, 3, 2, 3, 2, 2 \\ 4, 3, 0, 3, 5, 4, 3, 2, 4, 4 \\ 3, 3, 3, 0, 5, 6, 3, 4, 4, 3 \\ 3, 2, 1, 1, 0, 1, 4, 4, 5, 3 \\ 1, 3, 2, 0, 5, 0, 5, 4, 1, 4 \\ 1, 4, 3, 3, 2, 1, 0, 2, 1, 3 \\ 1, 3, 4, 2, 2, 2, 4, 0, 4, 2 \\ 2, 4, 2, 2, 1, 5, 5, 2, 0, 4 \\ 2, 4, 2, 3, 3, 2, 3, 4, 2, 0 \end{bmatrix}$$

Orderings	(LOP)	(wLOP)
(7, 8, 5, 2, 10, 6, 9, 4, 1, 3)	168	607
(7, 8, 5, 2, 10, 6, 9, 1, 4, 3)	168	608
(7, 8, 5, 2, 10, 6, 9, 1, 3, 4)	168	610
(2, 7, 8, 5, 10, 6, 9, 4, 1, 3)	168	597
(2, 7, 8, 5, 10, 6, 9, 1, 4, 3)	168	598
(2, 7, 8, 5, 10, 6, 9, 1, 3, 4)	168	600
(2, 7, 8, 10, 6, 9, 5, 4, 1, 3)	168	592
(2, 7, 8, 10, 6, 9, 5, 1, 4, 3)	168	593
(2, 7, 8, 10, 6, 9, 5, 1, 3, 4)	168	595
(2, 7, 8, 10, 6, 9, 1, 5, 3, 4)	168	586
(2, 7, 8, 10, 6, 9, 1, 5, 4, 3)	168	584
(2, 7, 8, 10, 5, 6, 9, 4, 1, 3)	168	596
(2, 7, 8, 10, 5, 6, 9, 1, 4, 3)	168	597
(2, 7, 8, 10, 5, 6, 9, 1, 3, 4)	168	599
(2, 7, 8, 10, 9, 5, 6, 4, 1, 3)	168	591
(2, 7, 8, 10, 9, 5, 6, 1, 4, 3)	168	592
(2, 7, 8, 10, 9, 5, 6, 1, 3, 4)	168	594
(2, 7, 9, 8, 5, 10, 6, 4, 1, 3)	168	587
(2, 7, 9, 8, 5, 10, 6, 1, 4, 3)	168	588
(2, 7, 9, 8, 5, 10, 6, 1, 3, 4)	168	590
(2, 7, 9, 8, 10, 5, 6, 4, 1, 3)	168	586
(2, 7, 9, 8, 10, 5, 6, 1, 4, 3)	168	587
(2, 7, 9, 8, 10, 5, 6, 1, 3, 4)	168	589
(2, 7, 10, 9, 8, 5, 6, 4, 1, 3)	168	587
(2, 7, 10, 9, 8, 5, 6, 1, 4, 3)	168	588
(2, 7, 10, 9, 8, 5, 6, 1, 3, 4)	168	590
(2, 10, 7, 9, 8, 5, 6, 4, 1, 3)	168	582
(2, 10, 7, 9, 8, 5, 6, 1, 4, 3)	168	583
(2, 10, 7, 9, 8, 5, 6, 1, 3, 4)	168	585
(7, 8, 5, 10, 6, 2, 9, 3, 1, 4)	164	613
(7, 8, 5, 6, 10, 2, 9, 3, 1, 4)	162	613
(7, 5, 8, 10, 6, 2, 9, 3, 1, 4)	162	613
(7, 5, 8, 6, 10, 2, 9, 3, 1, 4)	160	613

Figure 18.2: For the well-known data set from [259] containing pairwise preferences between 10 different brands of beer there exist 29 optimal `(LOP)` solutions with objective value of 168. For the `(wLOP)` we set all object weights to 1. There exist 4 optimal `(wLOP)` solutions with benefit 613. In average the optimal `(wLOP)` solutions have a higher `(LOP)` value than vice versa. Again the optimal `(wLOP)` orderings seem to reflect the overall preferences in a more accurate way than the optimal `(LOP)` solutions, i.e. the group of people providing the data would probably consider the `(wLOP)` orderings as a better representation of their preferences.

to weighted (`LA`):

$$\min_{\pi \in \Pi} \sum_{\substack{i,j \in [n] \\ i<j}} w_{ij} |\pi(i) - \pi(j)|,$$

$$\text{where } w_{ij} \text{ is } \begin{cases} \text{set to } 0, & \text{if } (i,j) \notin E, \\ \text{an element} \in \mathbb{N}, & \text{otherwise.} \end{cases}$$

Weighted (`LA`) can be viewed as a special (`wLOP`). If we set all node weights to 1, use minimization instead of maximization in (18.1) and additionally request that $w_{ij} = w_{ji},\ i,j \in [n], i < j$, holds, the (`wLOP`) reduces to weighted (`LA`). We refer to Chapter 5 for details on applications and exact (and heuristic) methods for solving the weighted (`LA`).

When considering weighted (`LA`) in the manufacturing context, it is denoted as Single-Row Equidistant Facility Layout Problem (`SREFLP`). The (`SREFLP`) is a special case of the Quadratic Assignment Problem and the Single-Row Facility Layout Problem (`SRFLP`). The (`wLOP`) can also be interpreted as a generalization of the (`SRFLP`) that allows for asymmetric costs. We refer to Chapters 6 and 7 for details on both the (`SREFLP`) and the (`SRFLP`).

Toy Example. Finally we want to clarify the workings of the different combinatorial optimization problems discussed above with the help of a toy example. We consider 4 objects and the pairwise weights $w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. Figure 18.3 illustrates the optimal orderings of the objects and the corresponding benefits for the (`LOP`), weighted (`LA`) and the (`wLOP`) with all object weights set to one. If we allow arbitrary object weights we can compare the optimal (`SRFLP`) and (`wLOP`) solutions, for an example see Figure 18.4.

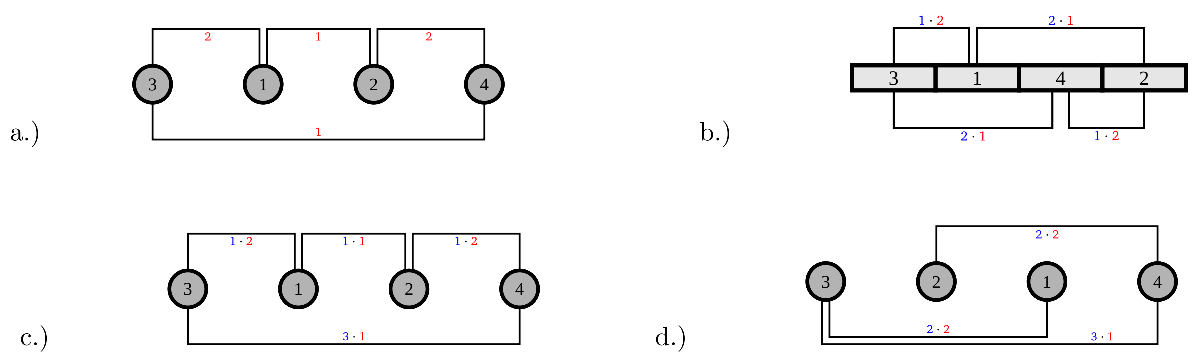
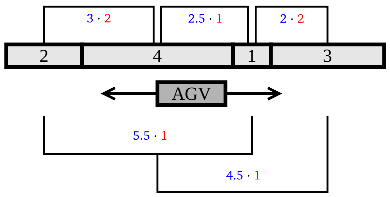


Figure 18.3: We are given 4 objects and the pairwise weights $w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. In a.) we display the optimal (`LOP`) solution with the corresponding benefit of $1+1+2+2 = 6$. To consider weighted (`LA`) we set all department lengths to 1. In b.) we show the optimal weighted (`LA`) layout with associated costs of $1 \cdot 2 + 2 \cdot 1 + 2 \cdot 1 + 1 \cdot 2 = 8$. In c.) we display the (`wLOP`) costs for the optimal (`LOP`) ordering from a.). If we set all object weights to 1, the corresponding benefit is $1 \cdot 2 + 1 \cdot 1 + 1 \cdot 2 + 3 \cdot 1 = 8$. Finally in d.) we depict the optimal (`wLOP`) layout with associated benefit of $2 \cdot 2 + 2 \cdot 2 + 3 \cdot 1 = 11$.

Outline. The main contributions we are aiming for in our research project on the (`wLOP`) are the following:

- Propose a new combinatorial optimization problem that is both of theoretical and practical interest.
- Present the (`wLOP`) is an extension/generalization of several well-known and well-studied combinatorial optimization problems like the (`LOP`), (`LA`) and the (`SRFLP`).

a.) 

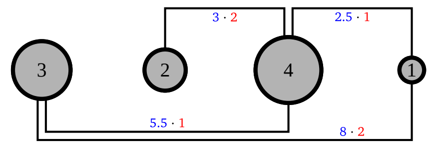
b.) 

Figure 18.4: We are given 4 objects and the weights $w_1 = 1,\ w_2 = 2,\ w_3 = 3,\ w_4 = 4,\ w_{12} = w_{41} = w_{34} = 1,\ w_{31} = w_{24} = 2$. For the (SRFLP) we interpret the objects weights as department or object lengths. In a.) we display the optimal (SRFLP) layout with corresponding costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we depict the optimal (wLOP) solution with associated benefit of $3 \cdot 2 + 2.5 \cdot 1 + 5.5 \cdot 1 + 8 \cdot 2 = 30$.

- Design a model of the (wLOP) as a Quadratic Ordering Problem (QOP). Building on this model we could apply an exact semidefinite optimization approach, which is highly competitive for several other (QOPs), to tackle the (wLOP).

- Show in a computational study that our suggested algorithmic approach yields highly competitive results on a large variety of problem classes and benchmark instances and that it clearly outperforms possible extensions of available exact integer linear programming approaches for layout problems. Note that preliminary computational results on the benchmark instances used in Chapters 5 and 6 suggest that the quality of the SDP bounds is the same for the (SRFLP) and the (wLOP) and hence instances with up to 40 objects can be solved to global optimality with the algorithmic approach used throughout this thesis.

18.2 Relations Between Optimal Single- and Multi-Row Layouts (joint work with Anja and Frank Fischer)

In this section we first show that for equidistant row layout instances the objective value of the optimal solution with m rows is less or equal to the objective value of the optimal single-row layout divided by m. As the corresponding proof is of constructive nature, we plan to use its ideas to design fast heuristics for the Multi-Row Equidistant Facility Layout Problem (MREFLP). For details on the (MREFLP) we refer to Chapter 13.

In a second step we showcase that for non-equidistant row layout instances no such relation between the objective values of single- and multi-row layouts exists. Nonetheless first preliminary computational results suggest that heuristics building on the ideas presented in this section work very well for both the equidistant and the non-equidistant case. For details on the Multi-Row Facility Layout Problem (MRFLP) we refer to Chapter 12. We also incorporate space-free layouts in our analysis below, for an exact definition and details on applications and algorithmic approaches we refer to Chapter 11. Note that we denote the objective value of the optimal layout using i rows by v_i^* throughout this section.

18.2.1 The Equidistant Case

We proved in Chapter 13 that there is always an optimal solution to the (MREFLP) on the grid. Furthermore we showed that this theorem has a significant impact for the computational perspective of solving the (MREFLP) to optimality because it allows to formulate the (MREFLP) as a discrete optimization problem. In the following analysis we again consider layouts lying on the grid. According to the notation in Chapter 13 we say for layouts fulfilling this grid property that department i lies in column j if the center of i is located at the j^{th} grid point .

First we consider special instances with $c_{ij} = 1,\ i,j \in [n],\ i < j$, i.e. the objective value of a given layout is just sum of all pairwise distances.

Lemma 18.1 *Let n be the number of departments. Then the following properties hold*

1. *The sum of the distances between all pairs of departments in single-row layouts is (independent of the order of the departments)* $\sum_{\substack{i,j\in[n],\\ i<j}} z_{ij} = \frac{(n+1)n(n-1)}{6}$.
2. *The sum of the distances between all pairs of departments in space-free double-row layouts is (independent of the order of the departments):*

$$\sum_{\substack{i,j\in[n],\\ i<j}} z_{ij} = \begin{cases} \frac{(n+1)n(n-1)}{12}, & \text{for } n \text{ odd,} \\ \frac{(n+2)n(n-2)}{12}, & \text{for } n \text{ even.} \end{cases}$$

Proof. 1. For single-row layouts we have:

$$\sum_{\substack{i,j\in[n],\\ i<j}} z_{ij} = \sum_{i=1}^{n-1}\sum_{j=1}^{n-i} j = \sum_{i=1}^{n-1}(n-i)i = n\sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} i^2 =$$
$$= \frac{n^2(n-1)}{2} - \frac{n(n-1)(2n-1)}{6} = \frac{(n+1)n(n-1)}{6}.$$

2. For space-free double-row layouts with n even we have:

$$\sum_{\substack{i,j\in[n],\\ i<j}} z_{ij} = \sum_{i=1}^{n-1}\sum_{j=1}^{n-i} \left\lfloor \frac{j}{2} \right\rfloor = \sum_{i=1}^{\frac{n-2}{2}} 4\left(\frac{n}{2} - i\right) i = 2n\sum_{i=1}^{\frac{n-2}{2}} i - 4\sum_{i=1}^{\frac{n-2}{2}} i^2 =$$
$$= \frac{2n^2(n-2)}{8} - \frac{4n(n-2)(n-1)}{24} = \frac{(n+2)n(n-2)}{12}.$$

For space-free double-row layouts with n odd we have:

$$\sum_{\substack{i,j\in[n],\\ i<j}} z_{ij} = \sum_{i=1}^{n-1}\sum_{j=1}^{n-i} \left\lfloor \frac{j}{2} \right\rfloor = 4\sum_{i=1}^{\frac{n-3}{2}} \left(\frac{n-1}{2} - i\right) i + 2\sum_{i=1}^{\frac{n-1}{2}} \left(\frac{n+1}{2} - i\right) =$$
$$= \frac{2(n-1)^2(n-3)}{8} - \frac{4(n-1)(n-3)(n-2)}{24} + \frac{(n+1)(n-1)}{2} - \frac{(n+1)(n-1)}{4} =$$
$$= \frac{(n+1)(n-1)(n-3)}{12} + \frac{(n+1)(n-1)}{4} = \frac{(n+1)n(n-1)}{12}.$$

□

Corollary 18.2 *The sum of the distances between all pairs of departments in space-free double-row layouts is at most half of the sum of the distances between all pairs of departments in the corresponding single-row layouts.*

Remark 3 *Notice that we cannot hope to reduce the optimal objective value by more than one half when going from single- to double-row layouts even for the equidistant case because for n odd the sum of the pairwise distances is reduced exactly by one half in the special case above.*

Now let us assume that the optimal (SREFLP) layout S is given, where w.l.o.g. we number the departments from left to right consecutively, see Figure 18.5.

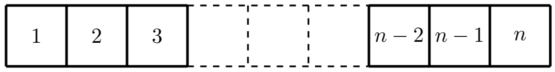


Figure 18.5: Optimal single-row layout, where the departments are numbered from left to right consecutively in increasing order.

Additionally we consider the following m (MREFLP) layouts $L_1, L_2, \ldots, L_m$ using m rows with objective values $v_{L_1}, v_{L_2}, \ldots, v_{L_m}$: In the i^{th} layout L_i with $i \in \{1, \ldots, m\}$, we assign the first i departments of S to the first column and totally fill up all other columns with the remaining departments in ascending order. We refer to Figures 18.6 and 18.7 for an illustration of the special case $m = 2$. In Figure 18.8 we depict a diagram of L_m and Figure 18.9 gives an illustration of the general layout $L_i,\ i \in [n]$.

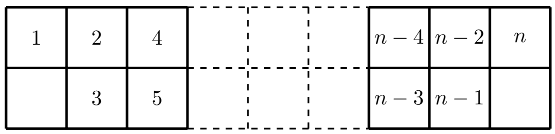


Figure 18.6: Double-row layout L_1 deduced from the optimal single-row layout. Note that in this drawing we again assume w.l.o.g. that n is even.

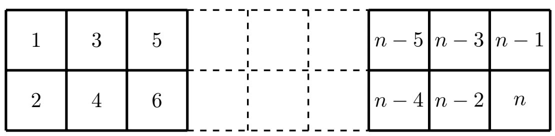


Figure 18.7: Double-row layout L_2 deduced from the optimal single-row layout. Note that in this drawing we assume w.l.o.g. that n is even.

Theorem 18.1 *The following equality holds:* $\sum_{i=1}^{m} v_{L_i} = v_1^*$.

Proof. Let the layouts $S, L_1, L_2, \ldots, L_m$ be given. We consider an arbitrary but fixed pair of departments i and j with $i < j$. The distance between i and j in S is $j - i$ and hence there arise costs of $(j-i)c_{ij}$. Now let us analyze the sum of the costs resulting from departments i and j in the layouts $L_1, L_2, \ldots, L_m$: The length between i and j is $(j-i) \div m$ in $(i-j) \bmod m$ layouts and $((j-i) \div m) + 1$ in $(j-i) \bmod m$ layouts. Hence it total the costs are:

$$[(j-i) \div m]\,[(i-j) \bmod m]\,c_{ij} + [((j-i) \div m) + 1]\,[(j-i) \bmod m]\,c_{ij} = (j-i)c_{ij}.$$

□

Corollary 18.3 *The following inequality holds:* $v_m^* \leq \frac{v_1^*}{m}$.

Note that $\{L_1, L_2, \ldots, L_m\}$ have a special structure, as they contain possible spaces only in the first and/or last column and hence in the double-row case they can always be interpreted as space-free layouts.

For the (SF-MRFLP), where the departments can have arbitrary lengths, the property discussed above does not hold even for two rows and two departments, i.e. consider the instance $l_1 = 4,\ l_2 = 1,\ c_{12} = 1$ with $v_1^* = 2.5$ and optimal space-free double-row layout with objective value $v_{2,SF}^* = 1.5$. Also note that the ratio $\frac{v_{2,SF}^*}{v_1^*}$ converges to 1 for $l_1 \to \infty$, $l_2 = 1$ and c_{12} arbitrary.

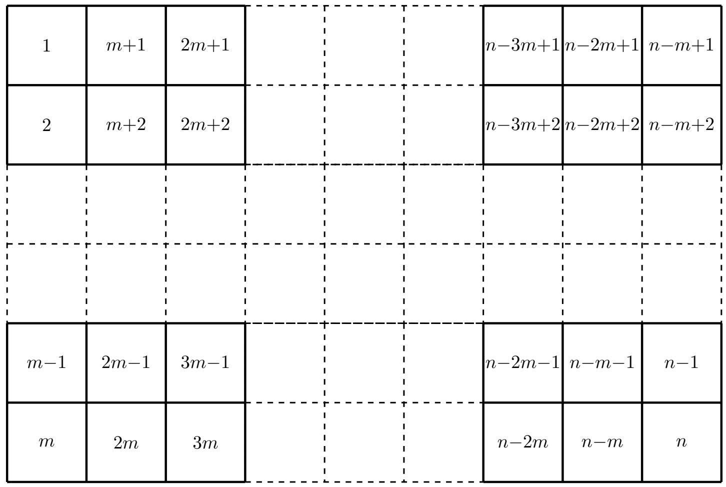


Figure 18.8: Depiction of a multi-row layout with m rows deduced from the optimal single-row layout, where all columns are completely filled up with departments in ascending order. Note that in this drawing we assume w.l.o.g. $n \bmod m = 0$.

18.2.2 The General Case

Next let us consider general multi-row layouts. Already the following small instance shows that the approach from above for the equidistant case does not work here: $\ell_1 = \ell_3 = 1,\ \ell_2 = \ell_4 = 2,\ c_{12} = c_{23} = c_{24} = c_{34} = 1$. The optimal single-row layout has objective value 7.5, while the two double-row layouts, which are constructed in the same way as suggested above for the equidistant case, have objective values 4 and 4.5 respectively, see also Figure 18.10. But also note that the optimal double-row layout has objective value 3 and hence $\frac{v_1^*}{v_2^*} < \frac{1}{2}$ holds for this instance.

For the case $n = 3$ the inequality $2v_2^* \leq v_1^*$ holds: Just consider the layouts suggested in Figure 18.11, then at least one of the two layouts must have the claimed objective value as the sum of the distances over each pair of departments is less or equal to the length between the respective pairs of departments in the optimal single-row layout:

$$
\begin{aligned}
c_{12}: &\qquad 0 + \frac{l_1 + \min(l_2, l_3)}{2} \leq \frac{l_1 + l_2}{2} \\
c_{13}: &\qquad \frac{\min(l_1, l_2) + l_3}{2} + \frac{l_1 + \min(l_2, l_3)}{2} \leq \frac{l_1 + l_3}{2} + l_2 \\
c_{23}: &\qquad \frac{\min(l_1, l_2) + l_3}{2} + 0 \leq \frac{l_2 + l_3}{2}
\end{aligned}
$$

For $n = 4$, we can give a sufficient condition for the inequality $2v_2^* \leq v_1^*$ to hold. Note that we assume $\max(l_1, l_2) + \min(l_3, l_4) \geq \max(l_3, l_4) + \min(l_1, l_2)$. This inequality holds, if we either number the departments in the optimal single-row layout either from left to right or from right to left.

Lemma 18.4 *Let us assume w.l.o.g.* $\max(l_1, l_2) + \min(l_3, l_4) \geq \max(l_3, l_4) + \min(l_1, l_2)$. *Then* $2v_2^* \leq v_1^*$ *holds, if* $l_1 \leq l_2$.

Proof. We can again compare the sum of the distances of the two double-row layouts suggested in Figure

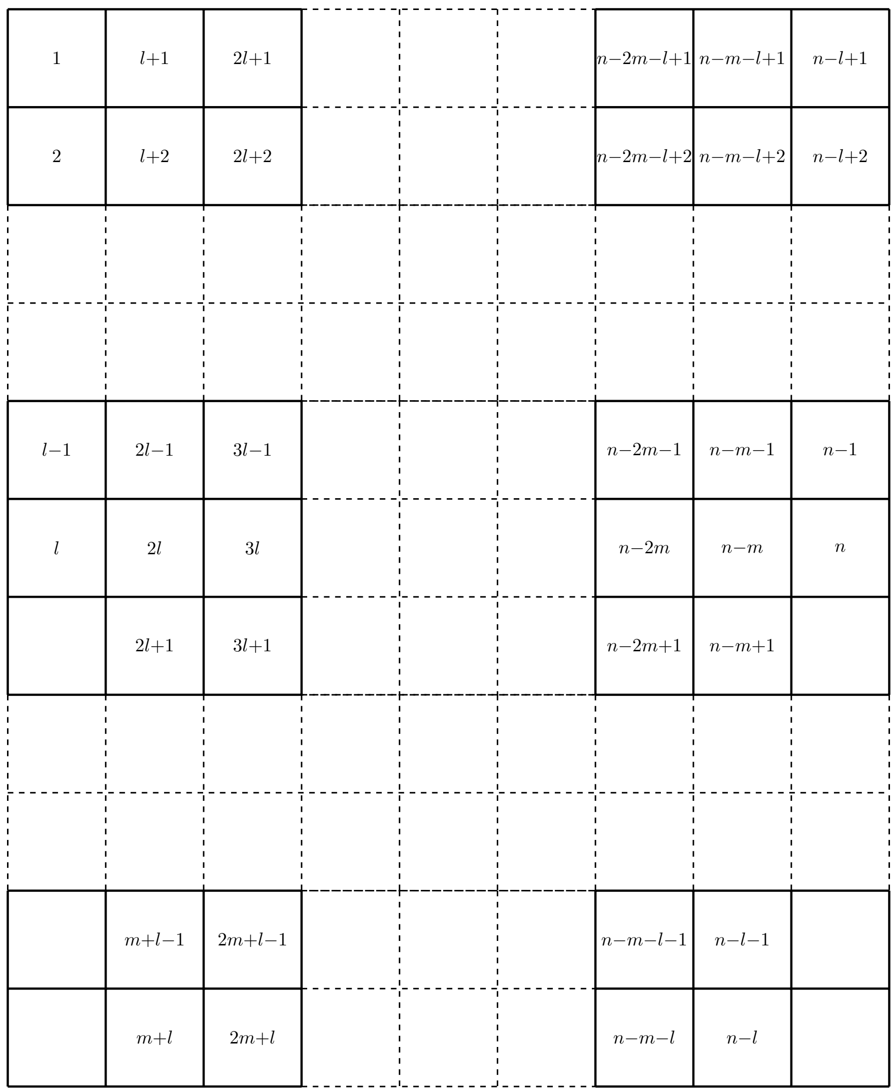


Figure 18.9: Illustration of a multi-row layout with m rows deduced from the optimal single-row layout, where the first $l \leq m$ departments are assigned to column 1 and the other columns are completely filled up with the remaining departments in ascending order. Note that in this drawing we assume w.l.o.g. $2l = m$ and $n \bmod m = 0$.

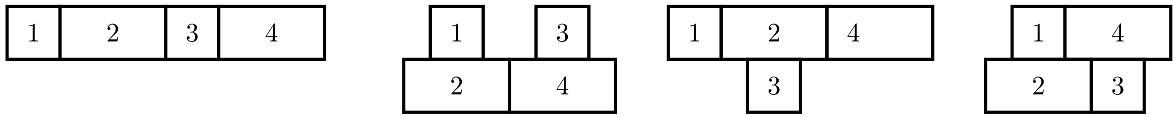


Figure 18.10: We are given the following instance: $\ell_1 = \ell_3 = 1,\ \ell_2 = \ell_4 = 2,\ c_{12} = c_{23} = c_{24} = c_{34} = 1$. The optimal single-row layout has objective value 7.5, the double-row layouts, which are constructed as suggested above for the equidistant case, have objective values 4 and 4.5 respectively and the optimal double-row layout has objective value 3.

18.12 with the distances in the optimal single-row layout for each pair of departments:

$$\begin{aligned}
c_{12}: &\quad 0 + \frac{l_1 + \min(l_2, l_3)}{2} \le \frac{l_1 + l_2}{2}, \\
c_{13}: &\quad \frac{\max(l_1, l_2) + \min(l_3, l_4)}{2} + \frac{l_1 + \min(l_2, l_3)}{2} \le \frac{l_1 + l_3}{2} + l_2, \\
c_{23}: &\quad \frac{\max(l_1, l_2) + \min(l_3, l_4)}{2} + 0 \le \frac{l_2 + l_3}{2}, \\
c_{14}: &\quad \frac{\max(l_1, l_2) + \min(l_3, l_4)}{2} + \frac{l_1 + l_4}{2} + \min(l_2, l_3) \le \frac{l_1 + l_4}{2} + l_2 + l_3, \\
c_{24}: &\quad \frac{\max(l_1, l_2) + \min(l_3, l_4)}{2} + \frac{\min(l_2, l_3) + l_4}{2} \le \frac{l_2 + l_4}{2} + l_3, \\
c_{34}: &\quad 0 + \frac{\min(l_2, l_3) + l_4}{2} \le \frac{l_3 + l_4}{2}.
\end{aligned}$$

Now all inequalities hold, if we additionally use the assumption $l_1 \le l_2$. □

Notice that for the example given in Figure 18.10 the condition from Lemma 18.4 ($l_1 \le l_2$) holds. This result can be generalized for $n > 4$ (and also for $m > 2$), but the larger n (and m), the more assumptions on the relation of the department lengths have to be fulfilled.

Next we show that for $n = 4$ departments the inequality $v_2^* \le \frac{2v_1^*}{3}$ always holds.

Lemma 18.5 *For $n = 4$ the following inequality holds: $3v_2^* \le 2v_1^*$.*

Proof. Consider the three layouts depicted in Figure 18.13. Let us again compare the sum of the distances of these three double-row layouts with twice the distances in the optimal single-row layout for each pair of departments:

$$\begin{aligned}
c_{12}: &\quad 0 + \frac{l_1 + \min(l_2, l_3)}{2} + \frac{l_1 + l_2}{2} \le l_1 + l_2, \\
c_{13}: &\quad \frac{\min(l_1, l_2) + l_3}{2} + \frac{l_1 + \min(l_2, l_3)}{2} + \frac{l_1 + \min(l_3, l_4)}{2} + l_2 \le l_1 + 2l_2 + l_3, \\
c_{23}: &\quad \frac{\min(l_1, l_2) + l_3}{2} + 0 + \frac{l_2 + \min(l_3, l_4)}{2} \le l_2 + l_3, \\
c_{14}: &\quad \frac{\min(l_1, l_2) + l_4}{2} + l_3 + \frac{l_1 + l_4}{2} + \min(l_2, l_3) + \frac{l_1 + \min(l_3, l_4)}{2} + l_2 \le l_1 + 2l_2 + 2l_3 + l_4, \\
c_{24}: &\quad \frac{\min(l_1, l_2) + l_4}{2} + l_3 + \frac{\min(l_2, l_3) + l_4}{2} + \frac{l_2 + \min(l_3, l_4)}{2} \le l_2 + 2l_3 + l_4, \\
c_{34}: &\quad \frac{l_3 + l_4}{2} + \frac{\min(l_2, l_3) + l_4}{2} + 0 \le l_3 + l_4.
\end{aligned}$$

As all of the inequalities hold, at least one of the three layouts suggested has objective value $\le \frac{2}{3}v_1^*$. □

Next we show that the inequality $3v_2^* \leq 2v_1^*$ is tight for $n = 4$, i.e. that for each $\delta > 0$ there exists an instance for which the inequality $\frac{v_2^*}{v_1^*} > \frac{2}{3} - \delta$ holds.

$\exists$ We have given 4 departments with length $l_1 = l_2 = l_3 = \varepsilon,\ l_4 = 2 - \varepsilon$, and connectivities $c_{12} = c_{23} = 1,\ c_{34} = \varepsilon$. Clearly $v_1^* = 3\varepsilon$ and the optimal double-row layout for $\varepsilon \leq 1$ is depicted in Figure 18.14 and has objective value $v_2^*(\varepsilon) = \varepsilon + \left(1 - \frac{\varepsilon}{2}\right)\varepsilon$. Hence we have $\frac{v_2^*}{v_1^*} = \frac{\varepsilon(2-\varepsilon)}{3\varepsilon} = \frac{2-\varepsilon}{3}$ and $\frac{v_2^*}{v_1^*} > \frac{2}{3} - \delta$ holds for $\varepsilon < 3\delta$.

Finally we indicate how to generalize the construction above for an arbitrary number of departments.

Corollary 18.6 *The following inequality holds for arbitrary $n \geq 2$: $(n-1)v_2^* \leq (n-2)v_1^*$. Furthermore for each $\delta > 0$ and $n \geq 3$ there exists an instance that fulfills the inequality $\frac{v_2^*}{v_1^*} > \frac{n-2}{n-1} - \delta$.*

Proof. First we show that $(n-1)v_2^* \leq (n-2)v_1^*$ holds for $n \geq 2$ by simply generalizing the proof idea used in Lemma 18.5. We consider $n-1$ double-row layouts, where the i^{th} layout is obtained as follows: All departments are assigned in the order of the optimal single-row layout to row 2 except for the departments i and $i+1$: The shorter of the two departments is also assigned to row 2, the longer is assigned to row 1 and the centers of the departments i and $i+1$ have the same x-coordinate. Additionally there is no space between the departments in row 2. Then at least one of the layouts suggested has objective value less or equal to $\frac{n-2}{n-1}v_1^*$, as the sum over all $n-1$ layouts has objective value less or equal to $(n-2)v_1^*$: Just notice that in each layout department i or its longer neighbor is not contained in the second row. Hence the sum of the distances over all layouts is less or equal to $(n-2)\left(\frac{l_i+l_j}{2} + \sum_{\substack{k\in[n],\\ i<k<j}} l_k\right)$ for each pair of departments i and j.

Next we explain how to construct instances that fulfill the inequality $\frac{v_2^*}{v_1^*} > \frac{n-2}{n-1} - \delta$ for any given $\delta > 0$ and $n \geq 3$: We define such instances recursively. They consist of a corresponding instance of size $n-1$, one additional sufficiently large department, and one additional connectivity between this department and the second largest department. Let us start with $n = 3$, where we can give an equidistant instance with $v_2^* = \frac{v_1^*}{2}$: $l_1 = l_2 = l_3 = \varepsilon,\ c_{12} = c_{23} = 1$. Next we add a fourth department of length $2 - \varepsilon$ and one additional connectivity $c_{34} = \varepsilon$ to obtain the instance proposed in Example 18.2.2. We choose $\varepsilon = \delta$, hence the inequality $\frac{v_2^*}{v_1^*} > \frac{2}{3} - \delta$ is fulfilled. Next we add a fifth department with length $2k - 2 + \varepsilon > 2$ and one additional connectivity $c_{45} = \frac{\varepsilon}{k}$. The optimal double-row layout can be obtained by arranging the first 4 departments as good as possible (i.e. use the optimal layout for four departments) and then additionally arrange department 5 such that the centers of departments 4 and 5 are as close as possible, see Figure 18.15. The objective value of this optimal double-row layout is $v_2^* = \varepsilon + \left(1 - \frac{\varepsilon}{2}\right)\varepsilon + (k - 2 + \varepsilon)\frac{\varepsilon}{k}$ and thus $\frac{v_2^*}{v_1^*} = \frac{\varepsilon\left(2-\varepsilon+\frac{(k-2+2\varepsilon)}{k}\right)}{4\varepsilon} = \frac{3-\varepsilon-\frac{(2-2\varepsilon)}{k}}{4} > \frac{3}{4} - \delta$ holds for $\varepsilon = \delta$ and k large enough, e.g. $k = \frac{1}{\delta}$. In general we take an instance with $n-1$ departments that is constructed like the ones above. Hence this instance fulfills the inequality $(n-2)v_2^* \leq (n-3)v_1^*$. Now we add an additional department and the connectivity $c_{(n-1)n} = \varepsilon\delta^{n-4}$. Next we choose the length of the department such that $v_1^* = (n-1)\varepsilon$ holds. Then we have $\frac{v_2^*}{v_1^*} > \frac{n-2}{n-1} - \delta$. $\square$

Finally note that the result from Lemma 18.1 cannot be generalized for arbitrary department lengths, see e.g. the different optimal double-row layouts depicted in Figure 18.16 with ratio $\frac{v_2^*}{v_1^*} > 0.5$. Clearly the objective value depends on the order of the departments for the general (MRFLP): In the single-row case the optimal layout is obtained by putting the shortest departments in the middle and the longest on the border of the layout as the objective value is

$$\sum_{i,j\in[n],\ i<j} \left(\frac{l_i + l_j}{2} + \sum_{\substack{k\in[n],\\ i<k<j}} l_k\right)$$

Figure 18.11: Two double-row layouts with 3 departments, both constructed from the optimal single-row layout. At least one of them has objective value $\leq \frac{v_1^*}{2}$.

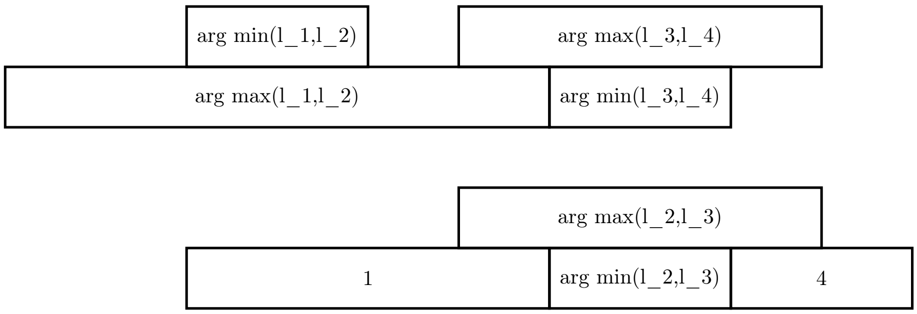


Figure 18.12: Two double-row layouts with 4 departments, both constructed from the optimal single-row layout. At least one of them has objective value $\leq \frac{v_1^*}{2}$, if $l_1 \leq l_2$ holds.

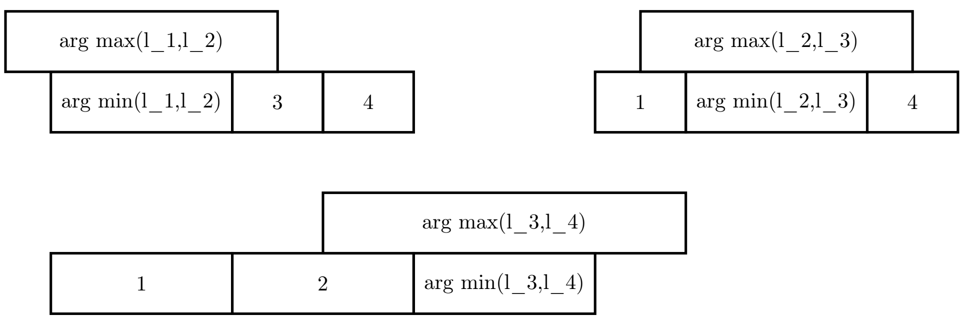


Figure 18.13: Three double-row layouts with 4 departments constructed from the optimal single-row layout. At least one of them has objective value $\leq \frac{2v_1^*}{3}$.

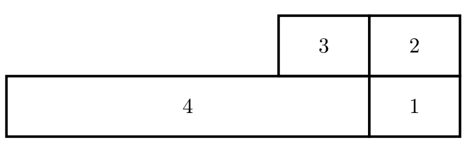


Figure 18.14: We are given the instance $l_1 = l_2 = l_3 = \varepsilon,\ l_4 = 2 - \varepsilon,\ c_{12} = c_{23} = 1,\ c_{34} = \varepsilon$. The optimal single-row layout is just $(1, 2, 3, 4)$ and has objective value $v_1^* = 3\varepsilon$. The optimal double-row layout for $\varepsilon \leq 1$ is depicted above and has objective value $v_2^* = \varepsilon + (1 - \varepsilon)\,\varepsilon$.

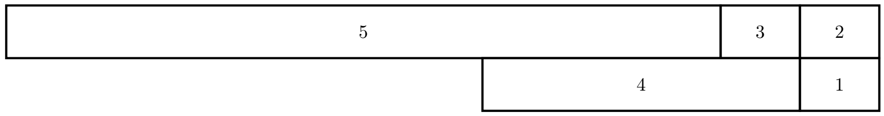


Figure 18.15: We are given the instance $l_1 = l_2 = l_3 = \varepsilon,\ l_4 = 2 - \varepsilon,\ l_5 = 2k - 2 + \varepsilon,\ c_{12} = c_{23} = 1,\ c_{34} = \varepsilon,\ c_{45} = \frac{\varepsilon}{k}$. The optimal single-row layout is just $(1, 2, 3, 4, 5)$ and has objective value $v_1^* = 4\varepsilon$. The optimal double-row layout for $\varepsilon \leq 1,\ 2k - 2 + \varepsilon > 2$ is depicted above and has objective value $v_2^* = \varepsilon + (1 - \varepsilon)\,\varepsilon + (k - 2 + 2\varepsilon)\frac{\varepsilon}{k}$.

and hence the lengths of the departments in the middle are counted more often. In the double-row case the optimal layout cannot be obtained by such a simple rule.

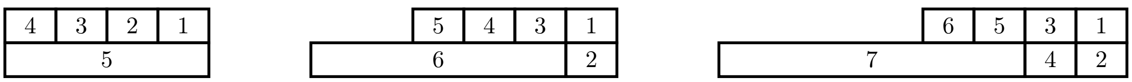


Figure 18.16: We consider the following instance for $n \in \{5, 6, 7\}$: $l_1 = \ldots = l_{n-1} = 1,\ l_n = n - 1,\ c_{ij} = 1,\ i, j \in [n],\ i < j$. In the optimal single-row layout the departments are simply arranged from left to right in increasing order. The corresponding objective value is $v_1^* = \frac{n(n-1)(n-2)}{6} + (n - 1)^2$. The optimal double-row layouts for $n \in \{5, 6, 7\}$ are depicted above and they have objective values 14, 25 and 41 respectively. Hence the ratios $\frac{v_1^*}{v_2^*}$ are $\frac{14}{26} \approx 0.54$, $\frac{25}{45} \approx 0.56$ and $\frac{41}{71} \approx 0.58$ respectively.

18.3 The TSP on Grids with Forbidden Neighborhoods (joint work with Anja Fischer)

In this section we consider the Traveling Salesman Problem (TSP) with forbidden neighbourhoods (TSPN): Given n points in the Euclidean plane and a radius $r \in \mathbb{R}$, the task is to determine a shortest Hamiltonian cycle $(t_0, \ldots, t_{n-1}, t_0)$ over the n points such that $\|t_i - t_{i+1}\| > r$ holds for $i \in \{0, \ldots, n-1\}$. These inequalities ensure that points traversed successively have a distance larger than r in the plane. We will mainly consider the Euclidean norm but we also extend all results to the 1-norm, where they either stay the same or simplify.

The TSPN is motivated by an application in beam melting. With an optimal solution of an appropriately defined TSPN one hopes to reduce the internal stress of the workpiece. Indeed during the handling process small parts of the workpiece that can be represented as points are heated up and if neighboring parts are close in the tour the workpiece might have a larger temperature than expected at a point. The points in the application have a regular structure and can be represented as grids.

Motivated by the application above we will concentrate on the TSPN on grids: Given nm points $V = \{p_{i_1}, \ldots, p_{i_{mn}}\}$ $(m \leq n)$ on the $m \times n$ grid in the Euclidean plane and a radius $r \in \mathbb{R}$, the task is to determine a shortest Hamiltonian cycle $(t_0, \ldots, t_{mn-1}, t_0)$ over the nm points such that the following inequalities hold:

$$\|t_i - t_{i+1}\| > r, \qquad i \in \{0, \ldots, mn - 1\}. \tag{18.2}$$

We will associate the nm points with nm equidistant vertices on the $m \times n$ grid. All vertices are connected by edges, where the edge weights give the Euclidean distances between the vertices. Furthermore we associate each vertex with a correspondent quadratic cell, i.e. each $m \times n$ grid can also be viewed as an $m \times n$ chessboard. In the following we will examine the length and structure of the optimal tours for the smallest reasonable neighborhoods on the grid, namely $r \in \{1, \sqrt{2}, 2\}$, see Figure 18.17 for an illustration.

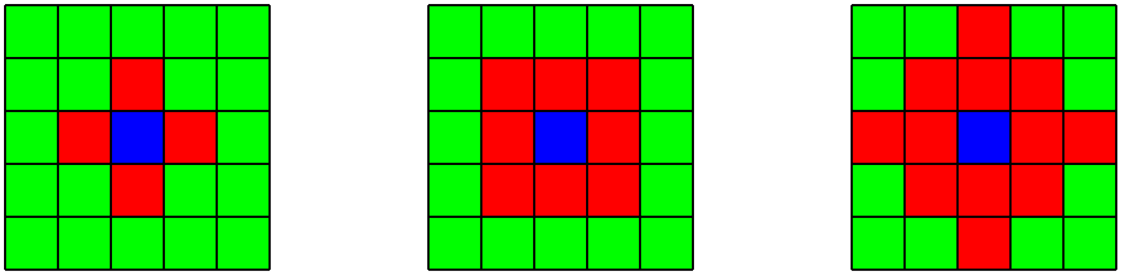


Figure 18.17: Illustration of forbidden neighbourhoods for different values of the radii: We consider $r = 1$, $r = \sqrt{2}$ and $r = 2$ from left to right. The current cell is blue, the forbidden cells are red and the allowed cells are green.

18.3.1 Results for $r = 0$

To begin with we consider optimal tours on $m \times n$ grids without forbidden neighbourhood, hence simply the TSP on rectangular grids. We can assign a pair of coordinates to each point, where we set the left upper point of the grid to $(1, 1)$. Hence the right lower point of the grid has the coordinates (m, n). Let us denote vertices as odd (even) vertices, if their coordinates sum up to an odd (even) number.

Lemma 18.7 *The optimal TSP tour on an $m \times n$ grid has the following length:*

a) $2(n-1)$ for $m = 1$,

b) nm for m or n even, see Figure 18.18 a) and b) for illustrations of optimal tours,

c) $nm - 1 + \sqrt{2}$ for m and n odd, see Figure 18.18/c) for an illustration of an optimal tour.

Proof. a) The vertices $(1, 1)$ and $(1, n)$ are $n - 1$ away from each other and must both be visited.

b) There are mn vertices on the $m \times n$ grid, hence a tour contains mn edges. The shortest edges of the graph have length one. Hence mn is a lower bound for the length of the optimal tour. But a tour using only edges of length one is possible because m or n is even and the drawing patterns illustrated in Figure 18.18 a) and b) can be applied.

c) Edges of length one always link an odd and an even vertex but if m and n are both odd the number of odd vertices exceeds the number of even vertices by one. Hence each tour must contain one edge connecting two odd vertices. Such an edge has length $\geq \sqrt{2}$. Hence $nm - 1 + \sqrt{2}$ is a lower bound for the length of the optimal tour. But a tour with this length always exists because the drawing pattern depicted in Figure 18.18/c) can be applied. □

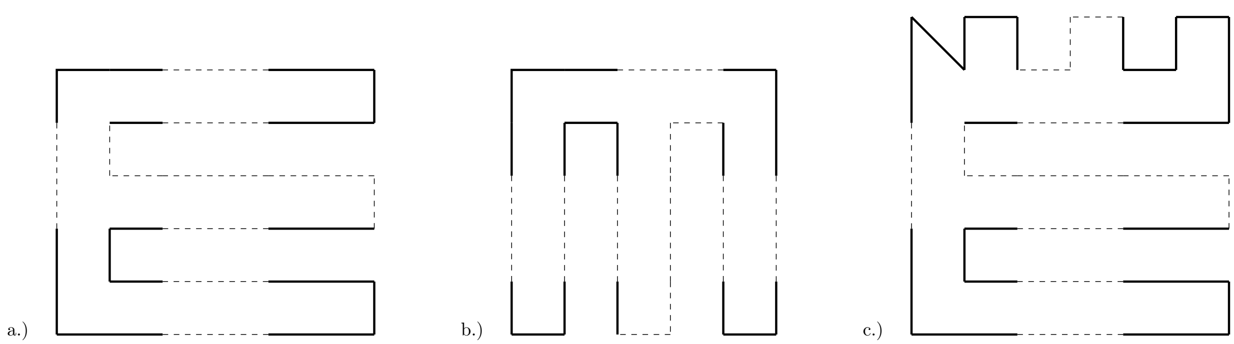


Figure 18.18: Optimal TSP tours on $m \times n$ grids for m even, n even and both m and n odd, respectively.

We call a tour a rook tour (in the style of the term knight's tour, see e.g. [276]), iff all of its edges connect odd and even vertices from adjacent cells. In the lemma above we implicitly proved that an $m \times n$ grid allows for a rook tour, iff m or n is even. The results of Lemma 18.7 stay basically the same, if we use Manhattan distance instead of the Euclidean norm. We only have to replace $nm - 2 + \sqrt{2}$ by $nm + 1$ in Lemma 18.7/c). The shortest Hamiltonian path on $m \times n$ grids that can also be denoted as an open tour has length $nm - 1$ for all variants.

The situation becomes more involved if we allow varying widths of the rows and columns, i.e. we have given two sets M and N containing m respectively n (not necessarily consecutive) integers. The coordinates of the mn vertices are obtained by the Cartesian product of M and N. In this case there can also be edges between pairs of odd or pairs of even vertices in the optimal tour of an $m \times n$ grid with m or n even, for a small example we refer to Figure 18.19. Additionally the optimal tour may also contain edges between vertices of non-adjacent cells, see Figure 18.20 for a small example. Note that edges between vertices of non-adjacent cells that share a common row, column or diagonal cannot be contained in the optimal tour due to a simple geometric argument.

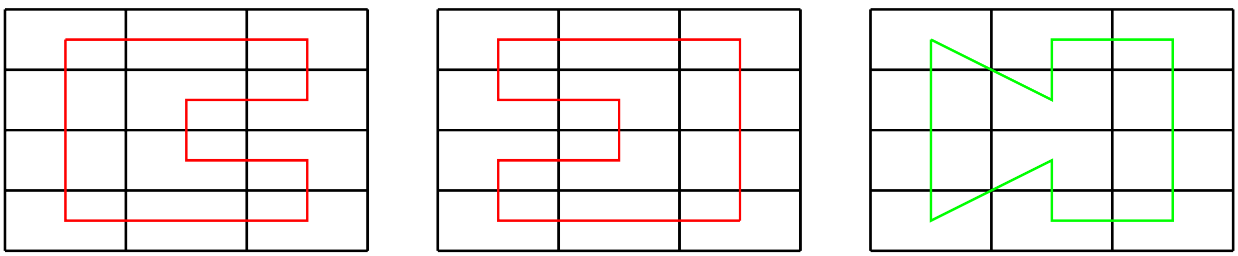


Figure 18.19: Possible rook tours and the optimal TSPN tour on a 4×3 grid generated as Cartesian product of the sets $M = \{1, 2, 3, 4\}$ and $N = \{1, 3, 5\}$. While the two rook tours have length 18, the optimal tour has length $12 + 2\sqrt{5} \approx 16.47$.

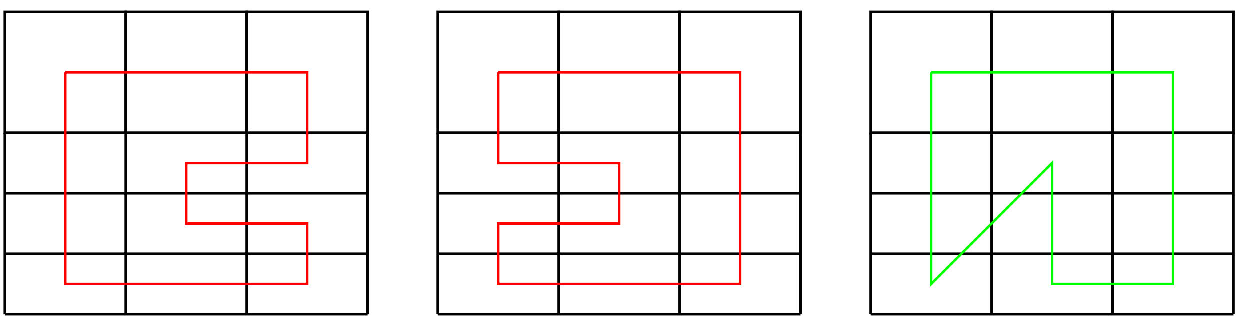


Figure 18.20: Possible rook tours and the optimal TSPN tour on a 4×3 grid generated as Cartesian product of the set $M = \{1, 3, 4, 5\}$ and $N = \{1, 3, 5\}$. While the two rook tours have length 20, the optimal tour has length $16 + \sqrt{8} \approx 18.83$.

The two examples above show that we cannot expect to find an easy construction rule for the optimal tours on grids with varying row and column widths, even if there is no forbidden neighborhood.

Even the complexity of the problem is unclear: Itai et al. [170] showed that the Hamilton cycle problem is NP-complete on planar grid graphs. A solid planar grid graph is a planar grid graph without holes. Umans and Lenhart [298] give a polynomial time algorithm for solid planar grid graphs. Recently Arkin et al. [25] gave a systematic study of Hamiltonicity of grids. The complexity of solving the TSP on a solid planar grid graph is still open, see e.g. Arkin et al. [24] or Problem 54 on the well-known list [87], known as "The Open Problems Project". In summary introducing a forbidden neighborhood makes the TSP more difficult in general, hence we will only consider grids with equidistant rows and columns in the remainder of this section.

18.3.2 Results for $r = 1$

In this subsection we examine optimal TSPN tours on $m \times n$ grids with $r = 1$. We will show that for $m \geq 5$ the optimal tours can be obtained as a combination of shortest Hamiltonian paths on the odd and even vertices respectively. To do so we first establish a lower bound on the length of Hamiltonian paths on the odd and even vertices for different parities of m and n. The shortest possible steps have step length $\sqrt{2}$. In the following lemma we examine how many steps of length $> \sqrt{2}$ are needed for different dimensions of the grid. To simplify the corresponding counting argument we define the following sets: First let us divide the vertices on the grid into outer and inner vertices, where the outer vertices lie on the boarder of the grids with dimensions $O := \{(m-4i) \times (n-4i)\},\ i \in \{0, 1, \ldots, \lfloor \frac{m}{4} \rfloor\}$ and the remaining vertices $I := V \setminus O$ are the inner vertices, see also Figure 18.21 for clarifying examples.

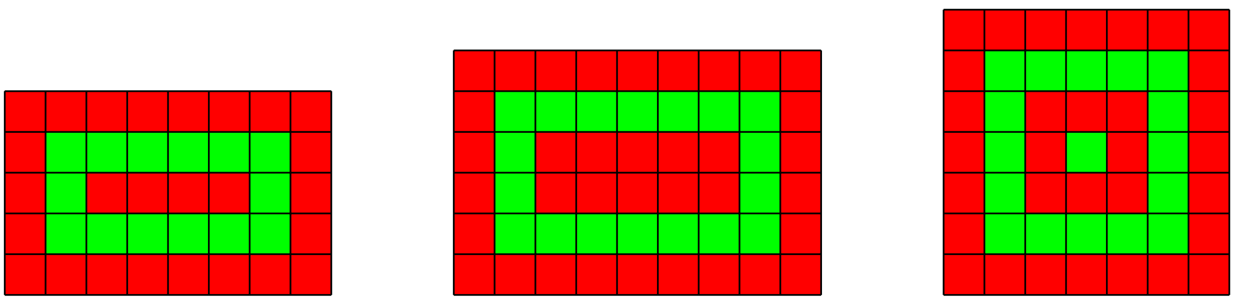


Figure 18.21: Division of the vertices into outer vertices (red cells) and inner vertices (green cells) on the 5×8, 6×9 and 7×7 grid.

Now we can speak about outer even and odd and inner even and odd vertices, for an illustration we refer to Figure 18.22.

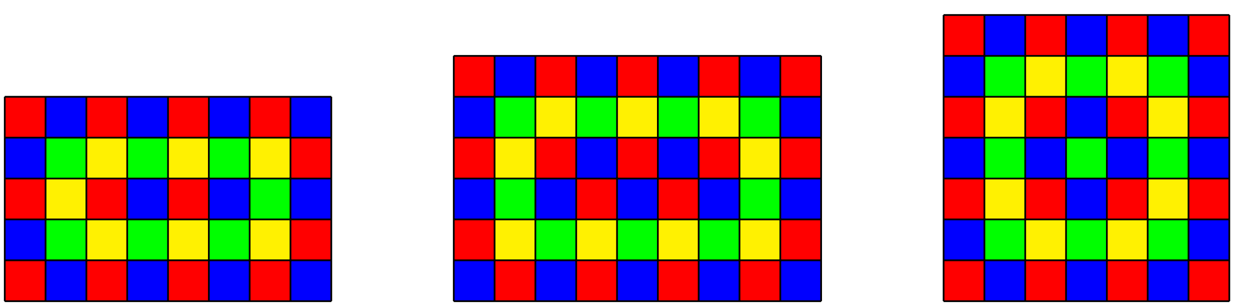


Figure 18.22: Division of the vertices into outer even vertices (red cells), outer odd vertices (blue cells), inner even vertices (green cells) and inner odd vertices (yellow cells) on the 5×8, 6×9 and 7×7 grid.

Finally we define a corner vertex as an outer vertex that lies on at least two different borders of one of the grids with dimensions O and friendly vertices as outer vertices that have an edge of length $\sqrt{2}$ to another outer vertex, see Figure 18.23 for an illustration of both definitions.

Lemma 18.8 *For a given $m \times n$ grid the following numbers are lower bounds on the number of edges with length ≥ 2 in any Hamiltonian path on the odd and even vertices:*

a) $\frac{m}{2} - 1$ for odd and even vertices with m even,

b) $\frac{n}{2} - 1$ for odd and even vertices with m odd and n even,

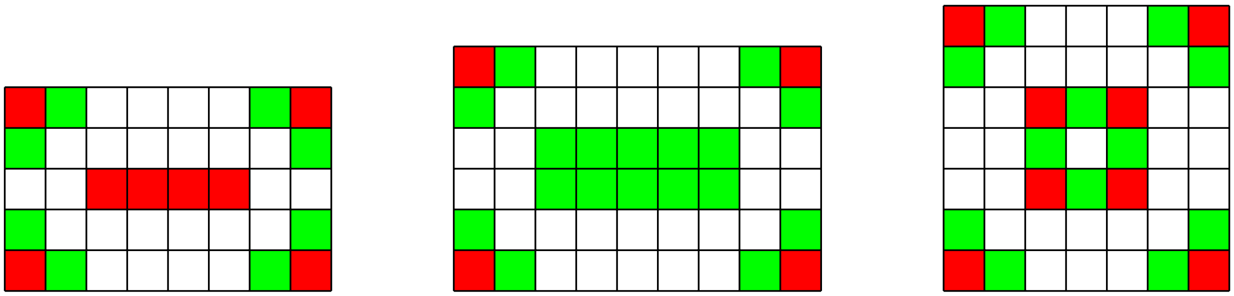


Figure 18.23: We highlighted the corner vertices (red cells) and the friendly vertices (green cells) on the 5×8, 6×9 and 7×7 grid.

c) $\frac{n+m}{2} - 1$ *for even vertices with* m *and* n *odd,*

d) $\max\{0, \frac{n-m}{2} - 1\}$ *for odd vertices with* m *and* n *odd.*

Proof. We use the following idea to establish lower bounds: Outer vertices (except for friendly vertices) can only be reached and left over edges with length $\sqrt{2}$ from inner vertices. The same holds true for friendly vertices if we combine friendly vertices connected by an edge of length $\sqrt{2}$ to one outer vertex. Each vertex has to be visited exactly once. Hence if the number of outer vertices exceeds the number of inner vertices by k, we need at least $k-1$ vertices of length ≥ 2, if we additionally take into account that we can start and finish the Hamiltonian path at an outer vertex. Now we are going to count the number of outer and inner vertices for the different parities of the grid to obtain the lower bounds stated above.

a) For m even we have a symmetric situation for odd and even vertices (simply rotate the grid 180 degrees), hence we consider w.l.o.g. the even vertices. Let us start with the outer vertices, where we count connected twin vertices as one vertex:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n + m + 4 - 8i) + \frac{m \bmod 4}{2}.$$

The number of inner even vertices is given by:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n + m + 2 - 8i).$$

Subtracting the number of inner even vertices from the number of outer even vertices gives $2\left\lfloor \frac{m}{4} \right\rfloor + \frac{m \bmod 4}{2} = \frac{m}{2}$. Finally subtracting 1 for the free choice of start and final vertex gives the lower bound $\frac{m}{2} - 1$ on the number of edges with length ≥ 2 in the any Hamiltonian path on the even (and odd) vertices.

b) For m odd and n even, the situation for odd and even vertices is again symmetric because of the even number of columns. Hence we again consider w.l.o.g. the even vertices. We start again with the outer vertices, count connected twin vertices as one vertex and distinguish the cases $m \bmod 4 = 1$ and $m \bmod 4 = 3$:

$$\begin{aligned} &\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n + m + 4 - 8i) + \frac{n+1-m}{2}, \quad \text{if } m \bmod 4 = 1, \\ &\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n + m + 4 - 8i) + n - m + 2, \quad \text{if } m \bmod 4 = 3. \end{aligned}$$

Next let us count the number of inner even vertices for both cases:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i), \quad \text{if } m \bmod 4 = 1,$$
$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i) + \frac{n-m+1}{2}, \quad \text{if } m \bmod 4 = 3.$$

Subtracting the number of inner even vertices from the number of outer even vertices gives:

$$2\left\lfloor \frac{m}{4} \right\rfloor + \frac{n}{2} - \frac{m}{2} + \frac{1}{2} = \frac{n}{2}, \quad \text{if } m \bmod 4 = 1,$$
$$2\left\lfloor \frac{m}{4} \right\rfloor + n - m + 2 - \frac{n}{2} + \frac{m}{2} - \frac{1}{2} = \frac{m}{2} - \frac{3}{2} + 2 + \frac{n}{2} - \frac{m}{2} - \frac{1}{2} = \frac{n}{2}, \quad \text{if } m \bmod 4 = 3.$$

Finally subtracting 1 for the free choice of start and final vertex gives the lower bound $\frac{n}{2} - 1$ on number of edges with length ≥ 2 in both cases.

c) For m and n odd, the situation for odd and even vertices is not symmetric. Hence we first consider the even vertices. We start with the outer vertices[1] and distinguish the cases $m \bmod 4 = 1$ and $m \bmod 4 = 3$:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+6-8i) + \frac{n-m}{2} + 1, \quad \text{if } m \bmod 4 = 1,$$
$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+6-8i) + n - m + 2, \quad \text{if } m \bmod 4 = 3.$$

The number of inner even vertices is:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i), \quad \text{if } m \bmod 4 = 1,$$
$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i) + \frac{n-m}{2} + 1, \quad \text{if } m \bmod 4 = 3.$$

Subtracting the number of inner even vertices from the number of outer even vertices gives:

$$4\left\lfloor \frac{m}{4} \right\rfloor + \frac{n}{2} - \frac{m}{2} + 1 = m - 1 + \frac{n}{2} - \frac{m}{2} + 1 = \frac{n+m}{2}, \quad \text{if } m \bmod 4 = 1,$$
$$4\left\lfloor \frac{m}{4} \right\rfloor + n - m + 2 - \frac{n-m}{2} - 1 = m - 1 + \frac{n-m}{2} + 1 = \frac{n+m}{2}, \quad \text{if } m \bmod 4 = 3.$$

Subtracting 1 for the free choice of start and final vertex gives the lower bound $\frac{n+m}{2} - 1$ on number of edges with length ≥ 2 in both cases.

d) Finally we consider the odd vertices for m and n odd. We start with the outer vertices, count

[1] Note that there are no twin vertices in this case

connected twin vertices as one vertex and distinguish the cases $m \bmod 4 = 1$ and $m \bmod 4 = 3$:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i) + \frac{n-m}{2}, \quad \text{if } m \bmod 4 = 1,$$
$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i) + n - m + 1, \quad \text{if } m \bmod 4 = 3.$$

Counting the number of inner even vertices gives:

$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i), \quad \text{if } m \bmod 4 = 1,$$
$$\sum_{i=1}^{\lfloor \frac{m}{4} \rfloor} (n+m+2-8i) + \frac{n-m}{2} + 1, \quad \text{if } m \bmod 4 = 3.$$

Subtracting the number of inner even vertices from the number of outer even vertices yields $\frac{n-m}{2}$ for both cases. Finally subtracting 1 for the free choice of start and final vertex gives the lower bound $\max\{0, \frac{n-m}{2} - 1\}$ on the number of edges with length ≥ 2 in any Hamiltonian path on the odd vertices. □

Corollary 18.9 *For a given $m \times n$ grid the following numbers are lower bounds on the length of shortest TSPN tour with $r = 1$:*

a) $m(n-1)\sqrt{2} + 2(m-2) + 2\sqrt{5}$ for m even,

b) $n(m-1)\sqrt{2} + 2(n-2) + 2\sqrt{5}$ for m odd and $m \neq n$,

c) $(m^2 - m - 1)\sqrt{2} + 2(m-1) + 2\sqrt{5}$ for $m = n$ odd.

Proof. A trivial lower bound on the length of any tour would be $(mn-2)\sqrt{2} + 2\sqrt{5}$ because we have to switch between odd and even vertices at least twice in the tour and the shortest edges to do so have length $\sqrt{5}$. But we can improve on this bound using Lemma 18.8.

a) Using the result from Lemma 18.8/a) on both the odd and even vertices we know that a TSPN tour with $r = 1$ must contain at least $m - 2$ edges of length at least 2 between pairs of odd or pairs of even vertices. For alternating between odd and even vertices we need even longer edges. Hence in summary we obtain $(nm - m)\sqrt{2} + 2(m-2) + 2\sqrt{5}$ as a lower bound on the length of any tour.

b) Now we use the results from Lemma 18.8 b)–d) and additionally assume $m \neq n$. For both n even and n odd a TSPN tour with $r = 1$ must contain at least $n - 2$ edges of length at least 2 between pairs of odd or pairs of even vertices as $2\left(\frac{n}{2} - 1\right) = \frac{n+m}{2} - 1 + \frac{n-m}{2} - 1 = n - 2$. Hence in summary we obtain $(nm - m)\sqrt{2} + 2(n-2) + 2\sqrt{5}$ as a lower bound on the length of any tour.

c) For $m = n$ odd we can again use Lemma 18.8 c) and d) to see that a TSPN tour with $r = 1$ must contain at least $m - 1$ edges of length at least 2. Thus we obtain $(m^2 - m - 1)\sqrt{2} + 2(m-1) + 2\sqrt{5}$ as a lower bound on the length of any tour. □

Theorem 18.2 *The optimal TSPN tour with $r = 1$ on an $m \times n$ grid has the following length:*

a) $m(n-1)\sqrt{2} + 2(m-2) + 2\sqrt{5}$ for $m \geq 6$ even.

b) $n(m-1)\sqrt{2} + 2(n-2) + 2\sqrt{5}$ for $m \geq 3$ odd and $m \neq n$,

c) $(m^2 - m - 1)\sqrt{2} + 2(m-1) + 2\sqrt{5}$ for $m \geq 3$ odd and $m = n$.

Proof. Due to Corollary 18.9 the numbers above are lower bounds on the length of the optimal TSPN tour with $r = 1$, hence giving an instruction for constructing tours of this particular lengths is sufficient.

a) $\wedge$ b) In the following we assume w.l.o.g. that m is even. The same construction is applicable for m odd but n even, resulting in a tour of length $n - 2$. A shortest Hamiltonian path on the odd vertices is simply constructed by starting at vertex $(1, 2)$, then visiting in counter-clockwise direction all outer vertices on the boarder of the $m \times n$ grid (this can be done using only edges of length $\sqrt{2}$ except for two edges of length 2 for leaving corner vertices) and then visiting the remaining vertices, i.e. the vertices on the $m - 4 \times n - 4$ grid as follows: Visit consecutively all vertices in the columns $(3, 4)$,$(5, 6)$,...,$(m - 3, m - 2)$. Vertices located in the same pairs of columns can be visited using only edges of length $\sqrt{2}$, changing the column pairs requires edges of length 2. Hence we need $\frac{m}{2} - 3$ edges of length 2 for changing between the pairs of columns and in the entire Hamilton path we need $\frac{m}{2} - 1$ edges of length 2 and otherwise only edges of length $\sqrt{2}$. Hence the total length of the suggested Hamilton path corresponds with the lower bound established in Lemma 18.8/a). For the even vertices an equivalent Hamiltonian path can be obtained by mirroring the Hamilton path for the odd vertices on the line $y = \frac{m}{2}$ if m mod $4 = 0$ and by rotating the Hamilton path for the odd vertices by 180 degrees if m mod $4 = 2$. The two Hamilton paths can be connected by edges of length $\sqrt{5}$, thus we obtain a tour of total length $(nm-m)\sqrt{2}+2(m-2)+2\sqrt{5}$ meeting the lower bound from Corollary 18.9 a) $\wedge$ b). Note that the construction only works if m respectively n is greater of equal to 6.

c) A shortest Hamiltonian path on the odd vertices can be constructed by starting at vertex $(1, 2)$, then visiting in counter-clockwise direction all outer vertices on the boarder of the $m \times n$ grid and then doing the same on the grid of dimension $m - 2 \times n - 2$ and so on until reaching the vertex $(\frac{m+1}{2}, n - \frac{m+1}{2})$. All edges that do not run within row $\frac{m+1}{2}$ have length $\sqrt{2}$, the last $\max\{0, \frac{n-m}{2} - 1\}$ edges have length 2. Hence the total length of the suggested Hamilton path corresponds with the lower bound established in Lemma 18.8/d). Our Hamiltonian path on the even vertices starts at vertex $(3, 1)$, then visits in counter-clockwise direction all outer vertices on the boarder of the grid (this can be done using only edges of length $\sqrt{2}$ except for four edges of length 2 for reaching respectively leaving corner vertices) and then does the same on the grid of dimension $m - 2 \times n - 2$ and so on until it reaches vertex $(\frac{m+1}{2} + 1, n - \frac{m+1}{2})$, i.e. the vertex above the final vertex of the Hamiltonian path on the odd vertices. Next it visits vertex $(\frac{m}{2}, n - \frac{m+1}{2} + 1)$ and then traverses the remaining vertices in row $\frac{m}{2}$ (using edges of length 2). Finally we visit the remaining vertices in rows $\frac{m}{2} - 1$ and $\frac{m}{2} - 2$ (using edges of length $\sqrt{2}$) such that the path finishes at vertex $(\frac{m+1}{2} + 2, n - \frac{m+1}{2} - 1)$. The Hamiltonian path on the even vertices uses in total $\frac{n+m}{2} - 1$ edges of length 2 and all other edges have length $\sqrt{2}$. Thus its length is equal to the lower bound established in Lemma 18.8/c). The two Hamilton paths can be connected by edges of length $\sqrt{5}$, thus we obtain a tour of total length $(nm - n)\sqrt{2} + 2(n - 2) + 2\sqrt{5}$ for $n > m$ and $(m^2 - m - 1)\sqrt{2} + 2(m - 1) + 2\sqrt{5}$ otherwise, both meeting the lower bounds from Corollary 18.9 b) and c). Note that the construction only works if m is greater or equal to 3. □

The length of the shortest open tours is an immediate consequence of the theorem above.

Corollary 18.10 *The optimal open TSPN tour with $r = 1$ on an $m \times n$ grid has the following length:*

a) $(nm - m)\sqrt{2} + 2(m - 2) + \sqrt{5}$ *for* $m \geq 6$ *even,*

b) $(nm - n)\sqrt{2} + 2(n - 2) + \sqrt{5}$ *for* $m \geq 3$ *odd and* $m \neq n$*,*

c) $(nm - m - 1)\sqrt{2} + 2(m - 1) + \sqrt{5}$ *for* $m \geq 3$ *odd and* $m = n$*.*

A Hamilton path visiting all odd (or even) vertices is a bishop's walk (in the style of the term knight's tour), iff all vertices are connected by edges of length $\sqrt{2}$. The following result on the existence of bishop's walks is another direct consequence of Theorem 18.2.

Corollary 18.11 *An $m \times n$ grid allows for a bishop's walk on the odd vertices, iff one of the following conditions holds:*

a) m *odd and* $m = n$,

b) $m \geq 3$ *odd and* $m = n - 2$,

c) $m = 2$.

A bishop's walk on the even vertices exists, iff $m = 2$.

To complete our analysis of the optimal TSPN tour with $r = 1$ we finally consider the grid sizes that have not been covered in Theorem 18.2 as their lengths are not equal to the lower bounds established in Corollary 18.9. In particular we consider the $1 \times n$ grids for $n \geq 5$ (for $n \leq 4$ there exists no feasible tour), the $2 \times n$ grids for $n \geq 3$ (for $n = 2$ there exists no feasible tour) and the $4 \times n$ grids.

Lemma 18.12 *The optimal TSPN tour with* $r = 1$ *on an* $m \times n$ *grid has the following length:*

a) $2(n - 4) + 14$ *for* $m = 1$ *and* $n \geq 10$,

b) $(2n - 6)\sqrt{2} + 8 + 2\sqrt{5}$ *for* $m = 2$ *and* $n \geq 6$,

c) $(4n - 5)\sqrt{2} + 6 + 2\sqrt{5}$ *for* $m = 4$.

Proof. a) To establish a lower bound for $m = 1$ and $n \geq 10$ we first note that we have to use at least two edges of length ≥ 3 connecting odd and even vertices. Furthermore let us consider the vertices $(1, 1)$ and $(n, 1)$ that are equivalent with respect to the following argument: If they are visited in the tour using edges with length ≥ 4 then the lower bound is already assured. Hence let us assume that these vertices are visited using edges of length 2 and length 3. But then the vertices $(2, 1)$ and $(n - 1, 1)$ both have to be visited using one edge of length ≥ 4 and thus also in this case the lower bound $2(n-4)+14$ is assured. Now let us suggest a TSPN tour attaining this bound: All vertices are connected using edges of length 2 expect for the upper six and lower six vertices on the line that are connected as depicted in Figure 18.24 using two edges of length 3 and two edges of length 4.

b) A TSPN tour with $r = 1$ on the $2 \times n$ grid with $n \geq 6$ has to contain 2 edges of length $\geq \sqrt{5}$. Next let us consider the four corner vertices $(1, 1)$, $(1, 2)$, $(n, 1)$ and $(n, 2)$: If all four corner vertices are visited by their two shortest incident edges of length $\sqrt{2}$ and 2 then lower bound of $(2n - 6)\sqrt{2} + 8 + 2\sqrt{5}$ is assured. If all four corner vertices are visited by edges of length $\sqrt{2}$ and $\sqrt{5}$ then the lower bound of $(2n - 6)\sqrt{2} + 8 + 2\sqrt{5}$ is also assured because using two edges of length $\sqrt{2}$ either in $(2, 1)$ and $(2, 2)$ or in $(n - 1, 1)$ and $(n - 2, 2)$ would yield a subtour either in the first three rows or in the last rows. The third possibility is to use edges of length $\sqrt{2}$ and $\sqrt{5}$ for vertex $(1, 1)$ and edges of length $\sqrt{2}$ and 2 for vertex $(1, 2)$. But now we need another edge of length ≥ 2 in vertex $(2, 1)$. As the same argument applies to the vertices $(n - 1, 1)$ and $(n - 2, 2)$ the lower bound is assured. The last possibility is to use edges of length $\sqrt{2}$ and $\sqrt{5}$ for vertex $(1, 1)$ and edges of length $\sqrt{2}$ and $\sqrt{10}$ for vertex $(1, 2)$. But also in this case the lower bound is assured as $\sqrt{2} + \sqrt{10} > 4$ holds. A TSPN tour with $r = 1$ attaining the lower bound $(2n - 6)\sqrt{2} + 8 + 2\sqrt{5}$ simply applies the third pattern proposed above on both ends of the grid.

c) Using Lemma 18.8/a) ensures that the Hamiltonian paths on the odd and even vertices of the $4 \times n$ grid have to contain at least 1 step of length $> \sqrt{2}$. Such walks can also easily be constructed, but they have to start and end at a corner vertex and an adjacent outer vertex with distance two to the corner vertex because traversing a corner vertex in the Hamilton walk needs one edge of length ≥ 2. Connecting the start and end vertices of the two Hamiltonian paths of minimal length is not possible with edges of length $\sqrt{5}$.

Hence $(4n - 5)\sqrt{2} + 6 + 2\sqrt{5}$ is a lower bound for the length of any TSPN tour with $r = 1$ on the $4 \times n$ grid. But a tour of this length can be constructed as follows: On the even vertices we construct a Hamiltonian path of length $(2n - 2)\sqrt{2} + 2$ starting at vertex $(1, 1)$ and finishing at vertex $(3, 1)$. On the odd vertices we construct a Hamiltonian path of length $(2n - 3)\sqrt{2} + 4$ starting at vertex $(1, 2)$ and finishing at vertex $(2, 3)$. Now we can connect these two paths on the odd and even vertices using two edges of length $\sqrt{5}$. □

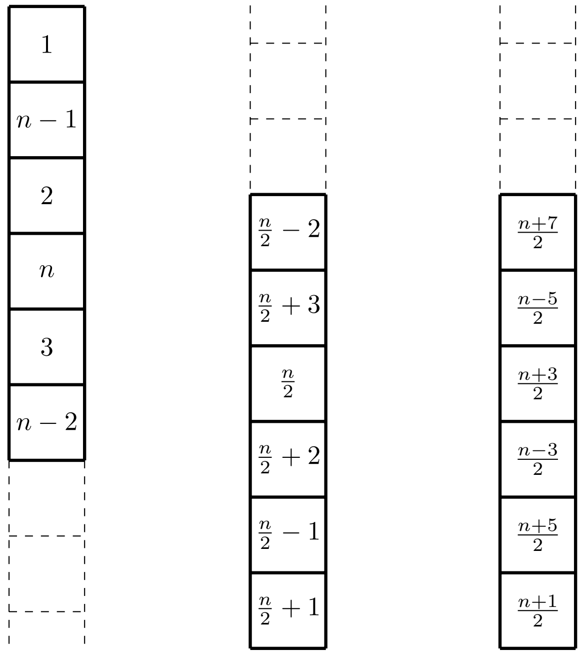


Figure 18.24: The numbers in the cells indicate the order in which the upper six and lower six vertices on the line can be visited to attain a TSPN tour of length $2(n-4)+14$. The line in the center deals with the last six vertices for n even and the right line deals with the case n odd.

The results presented in this subsection can be considerably simplified if we consider the Manhattan distance.

Corollary 18.13 *For the $1 \times n$ grid with $n \geq 5$ the optimal TSPN tour with $r = 1$ minimizing Manhattan distance has exactly the length given in Lemma 18.12 a) and b). For the $m \times n$ grid with $m \geq 2$ and $n \geq 3$ the length of the TSPN tour with $r = 1$ and Manhattan distance is always $2(nm-2)+6$.*

Proof. The shortest possible edges have length 2 and the shortest edges for connecting odd and even vertices have length 3. Hence $2(nm-2)+6$ is a lower bound on the length of the optimal tour. But the constructions of optimal tours for the various grid sizes in Theorem 18.2 and Lemma 18.12 all attain this lower bound. □

18.3.3 Results for $r = \sqrt{2}$

In this subsection we examine optimal TSPN tours on $m \times n$ grids with $r = \sqrt{2}$. First let us extend our definition of odd and even vertices: We now denote vertices according to the parity of their coordinates, e.g. (o, e)-vertices, if their first coordinate is odd and their second coordinate is even. We will show that for $m \geq 4$ the optimal tours can be obtained as a combination of shortest Hamiltonian paths on the (o, e)-vertices, (o, o)-vertices, (e, o)-vertices and (e, e)-vertices.

Lemma 18.14 *For a given $m \times n$ grid with $m \geq 2$, the number $2(nm-4)+4\sqrt{5}$ is a lower bound on the length of any TSPN tour with $r = \sqrt{2}$.*

Proof. All feasible edges have length ≥ 2. To switch between (o, e)-vertices, (o, o)-vertices, (e, o)-vertices and (e, e)-vertices we have to use edges of length $\geq \sqrt{5}$. Hence in total the lower bound $2(nm-4)+4\sqrt{5}$ is valid. □

Theorem 18.3 *The optimal TSPN tour with $r = \sqrt{2}$ on an $m \times n$ grid with $m \geq 4$ has length $2(nm-4)+4\sqrt{5}$.*

Proof. The claimed length is a lower bound to any tour due to Lemma 18.14. Now let us suggest different drawing patterns for shortest Hamiltonian paths on the (o,o)-vertices, (o,e)-vertices, (e,o)-vertices and (e,e)-vertices. We subdivide the drawing patterns with respect to their end vertices, where we have to take into account the different parities of the rows and columns of the grid. The first and last two drawing patterns start at the upper right vertex, i.e. $(1,1)$, $(1,2)$, $(2,1)$ and $(2,2)$ respectively. For the other drawing patterns the start vertex is shifted one to the right or below.

1. Drawing pattern 1 finishes at the next vertex right of the start vertex, i.e. $(3,1)$, $(3,2)$, $(4,1)$ and $(4,2)$ respectively. It can be realized, if m or n is even, for an illustration see Figure 18.25.

2. Drawing pattern 2 finishes at the next vertex below of the start vertex, i.e. $(1,3)$, $(1,4)$, $(2,3)$ and $(2,4)$ respectively. It can be realized, if m or n is even, for an illustration see Figure 18.26.

3. Drawing patterns 3 and 4 finish at the upper left vertex of the grid. They can be realized, if m or n is even, for an illustration see Figures 18.27 and 18.28.

4. Drawing pattern 5 finishes at the vertices $(5,1)$, $(5,2)$, $(6,1)$ and $(6,2)$ respectively. It can be realized, if m or n is even, for an illustration see Figure 18.29.

5. Drawing pattern 6 finishes at the vertices $(1,5)$, $(2,5)$, $(1,6)$ and $(2,6)$ respectively. It can be realized, if m or n is even, for an illustration see Figure 18.30.

6. Drawing pattern 7 finishes next to the upper right vertex of the grid. It can be realized, if m is odd or n is even, for an illustration see Figure 18.31. We apply this pattern if the grids of the (o,o)-vertices, (o,e)-vertices, (e,o)-vertices and (e,e)-vertices have all the same dimension. In this case we can also rotate the pattern by 90, 180 and 270 degrees and then apply it.

7. Drawing pattern 8 finishes at the vertices $(3,3)$, $(3,4)$, $(4,3)$ and $(4,4)$ respectively. It can be realized, if m and n is odd. We can also switch the start and finish vertex if convenient. The corresponding Hamiltonian path is denoted as drawing pattern 9, for an illustration see Figure 18.32.

Table 18.1 lists all possible combinations of the parities of the dimension of the four grids corresponding to the (o,o)-, (o,e)-, (e,o)- and (e,e)-vertices and gives tours of length $2(nm-4)+4\sqrt{5}$ for each of these combinations. Hence we give a construction of an optimal TSPN tour with $r=\sqrt{2}$ for all $m \times n$ grids with $m \geq 4$.[2] Table 18.2 additionally visualizes the possible transitions from one of the four Hamiltonian paths on the (o,o)-, (o,e)-, (e,o)- and (e,e)-vertices to another one using edges of length $\sqrt{5}$. □

The length of the shortest open tour is an immediate consequence of the theorem above.

Corollary 18.15 *The optimal open TSPN tour with $r=\sqrt{2}$ on an $m \times n$ grid with $m \geq 4$ has length $2(nm-4)+3\sqrt{5}$.*

To complete our analysis of the optimal TSPN tour with $r=\sqrt{2}$ we finally consider the grid sizes that have not been covered in Theorem 18.3 as their lengths are not equal to the lower bound established in Lemma 18.14. Clearly the neighborhoods $r=1$ and $r=\sqrt{2}$ are equivalent on $1 \times n$ grids for $n \geq 5$ (for $n \leq 4$ there exists no feasible tour). Hence for this case we refer to Lemma 18.12 a) and b). Next we consider $3 \times n$ grids with $n \geq 4$. Note that there exists no feasible tour on the 3×3 grid.

Lemma 18.16 *On a $3 \times n$ grid with $n \geq 4$ the lower bound for the length of any TSPN tour with $r=\sqrt{2}$ is $2(nm-6)+6\sqrt{5}$.*

[2]Note that drawing patterns 1–4 are applicable for grids with dimension $m \geq 2$. For the other patterns m respectively n greater or equal 3 is required.

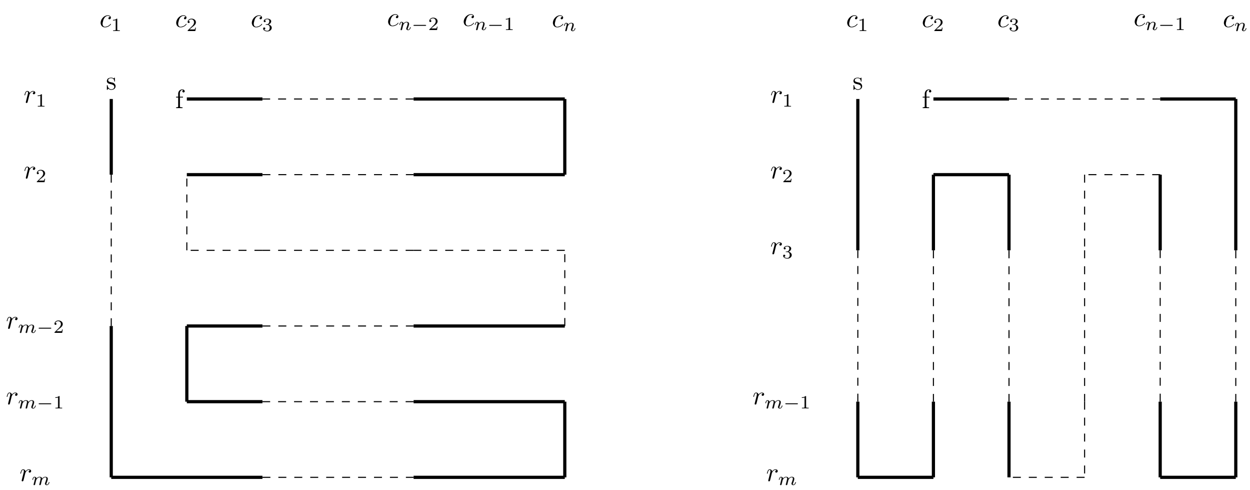


Figure 18.25: Drawing pattern 1 can be applied if m or n is even. We start from the left upper vertex and finish at the vertex right of the starting vertex.

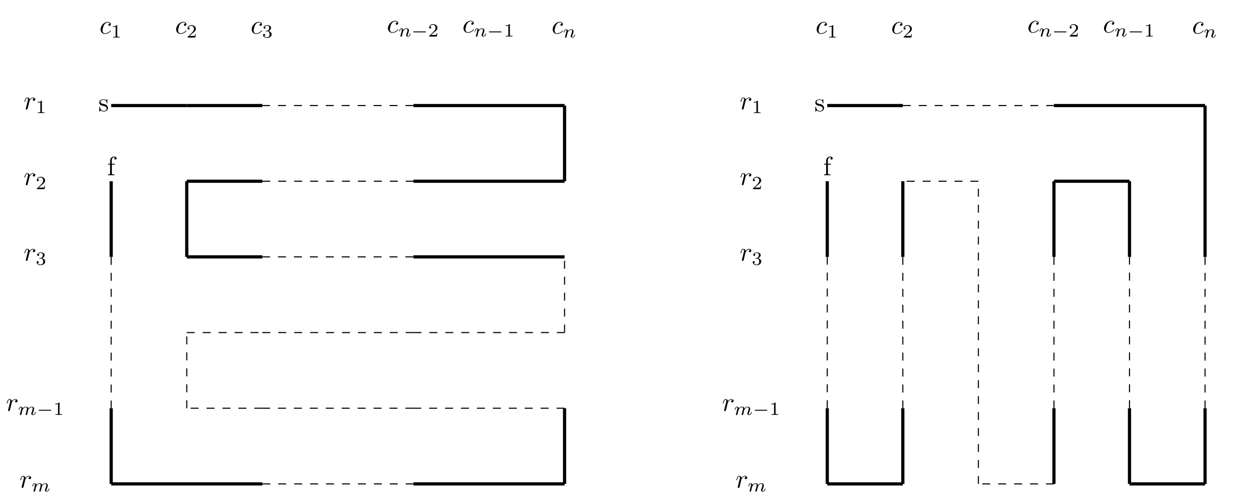


Figure 18.26: Drawing pattern 2 can be applied if m or n is even. We start from the left upper vertex and finish at the vertex below of the starting vertex.

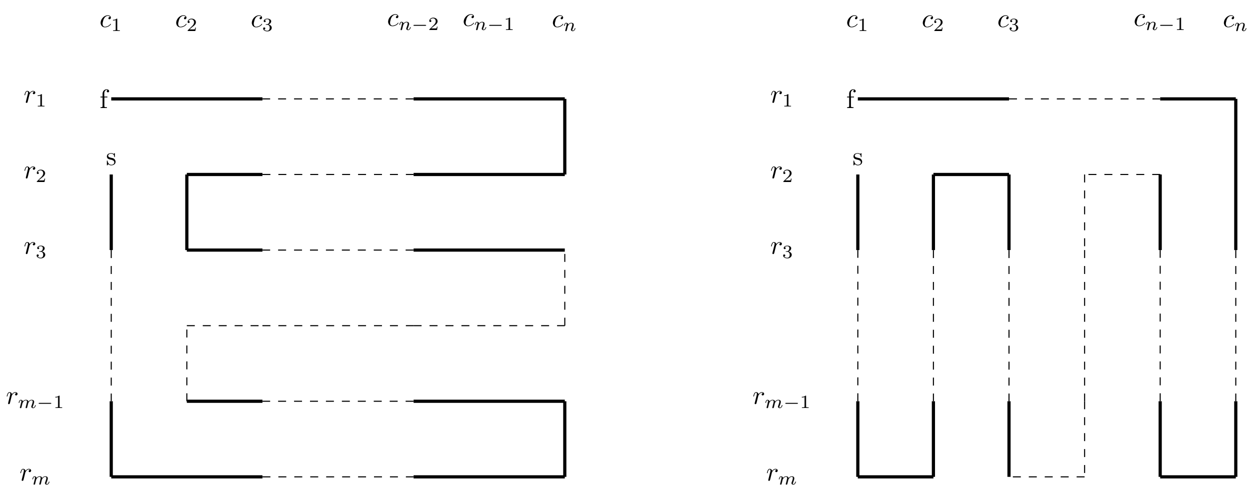


Figure 18.27: Drawing pattern 3 can be applied if m or n is even. We finish at the left upper vertex and start from the vertex right of it.

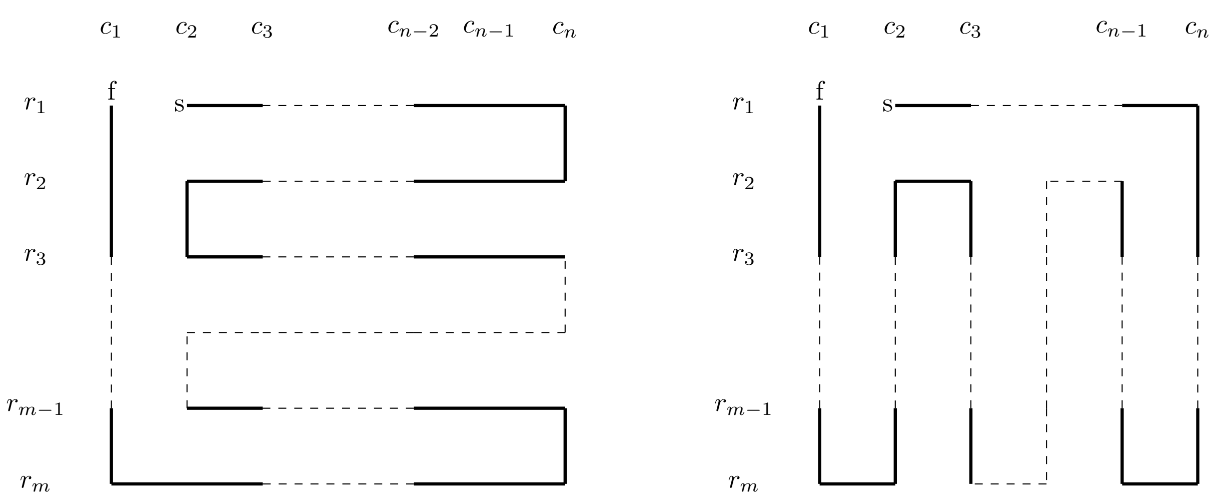


Figure 18.28: Drawing pattern 4 can be applied if m or n is odd. We finish at the vertex left and above of the lower right vertex.

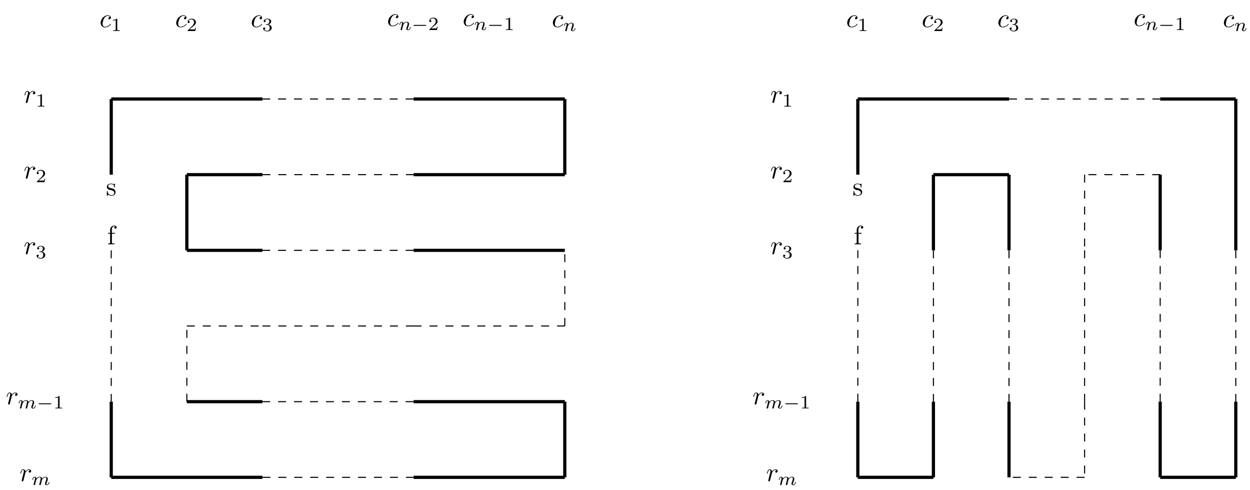


Figure 18.29: Drawing pattern 5 can be applied if m or n is even. We start from the vertex below the starting vertex and finish below of it.

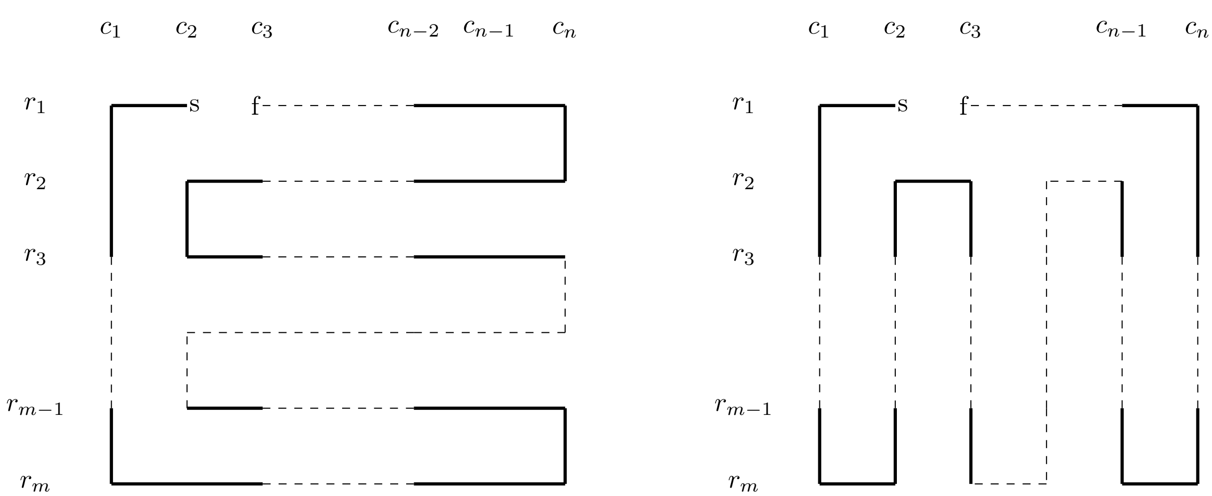


Figure 18.30: Drawing pattern 6 can be applied if m or n is even. We start from the vertex right the starting vertex and finish right of it.

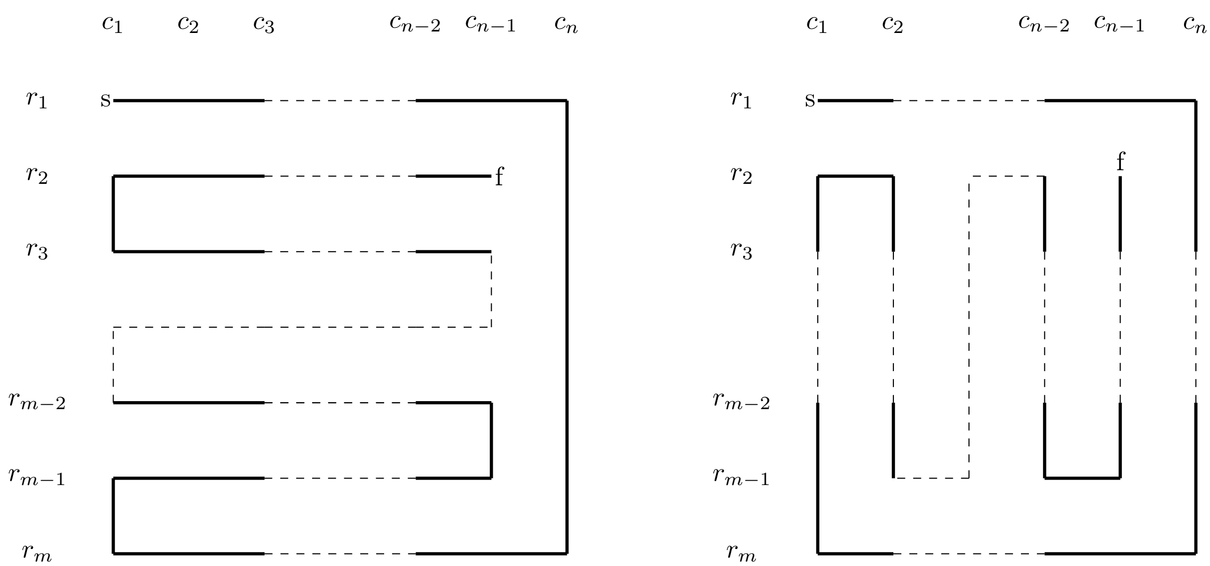


Figure 18.31: Drawing pattern 7 can be applied if m is odd or n is even. We start from the left upper vertex and finish at the vertex left and below of the upper right vertex.

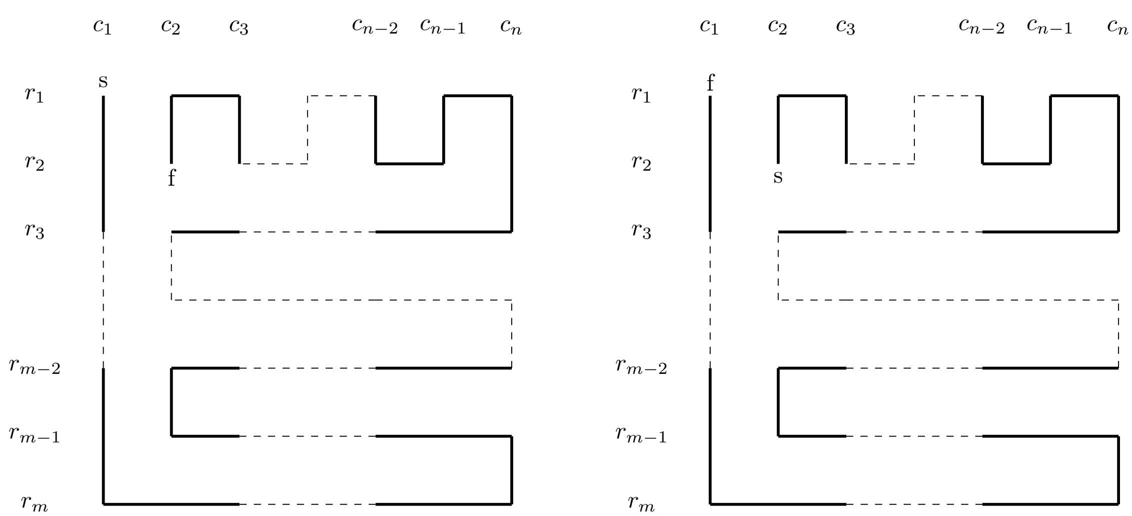


Figure 18.32: Drawing patterns 8 (left) and 9 (right) can be applied if m and n are odd. We start from the left upper vertex and finish at the vertex one right and below of it or we use the Hamiltonian path in the opposite direction.

r \ c		Parities of the column dimensions							
		e	e	e	o	o	e	o	o
Parities of the row dimensions	e	ee	ee	ee	eo	eo	ee	eo	eo
	e	ee	ee	ee	eo	eo	ee	eo	eo
	e	ee	ee	ee	eo	eo	ee	eo	eo
	o	oe	oe	oe	oo	oo	oe	oo	oo
	o	oe	oe	oe	oo	oo	oe	oo	oo
	e	ee	ee	ee	eo	eo	ee	eo	eo
	o	oe	oe	oe	oo	oo	oe	oo	oo
	o	oe	oe	oe	oo	oo	oe	oo	oo

Table 18.1: This table lists all possible combinations of dimensions of the four grids corresponding to the (o,o)-, (o,e)-, (e,o)- and (e,e)-vertices. Below and right of this table we give the construction of optimal TSPN tours with length $2(nm-4)+4\sqrt{5}$ that are possible for the squares with the respective background color.

Green squares:
1. Start with vertex $(2,2)$
2. Visit all (e,e)-vertices using drawing pattern 1
3. Use an edge of length $\sqrt{5}$ to reach vertex $(1,2)$
4. Visit all (o,e)-vertices using drawing pattern 2
5. Use an edge of length $\sqrt{5}$ to reach vertex $(1,1)$
6. Visit all (o,o)-vertices using drawing pattern 1
7. Use an edge of length $\sqrt{5}$ to reach vertex $(2,1)$
8. Visit all (e,o)-vertices using drawing pattern 2
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

Orange square:
1. Start with vertex $(2,2)$
2. Visit all (e,e)-vertices using drawing pattern 7
3. Use an edge of length $\sqrt{5}$ to reach vertex $(2,n-1)$
4. Visit all (e,o)-vertices using drawing pattern 7
5. Use an edge of length $\sqrt{5}$ to reach vertex $(n-1,n-1)$
6. Visit all (o,o)-vertices using drawing pattern 7
7. Use an edge of length $\sqrt{5}$ to reach vertex $(n-1,2)$
8. Visit all (e,o)-vertices using drawing pattern 7
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

Violet squares:
1. Start with a vertex having odd columns and rows, either vertex $(3,4)$ or $(4,3)$
2. Visit all vertices of that type using drawing pattern 9
3. Use an edge of length $\sqrt{5}$ to reach a vertex with even columns or even rows
4. Visit all vertices of that type using drawing pattern 3 or 4
5. Use an edge of length $\sqrt{5}$ to reach an (o,o)-vertex
6. Visit all (o,o)-vertices using drawing pattern 5 or 6
7. Use an edge of length $\sqrt{5}$ to each an (e,e)-vertex
8. Visit all (e,e)-vertices using drawing pattern 9
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

Red squares:
1. Start with vertex $(3,3)$
2. Visit all (o,o)-vertices using drawing pattern 9
3. Use an edge of length $\sqrt{5}$ to reach a vertex with even columns or even rows
4. Visit all vertices of that type using drawing pattern 3 or 4
5. Use an edge of length $\sqrt{5}$ to reach an (e,e)-vertex
6. Visit all (e,e)-vertices using drawing pattern 3 or 4
7. Use an edge of length $\sqrt{5}$ to each a vertex with odd columns and odd rows
8. Visit all vertices of that type using drawing pattern 9
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

Blue squares:
1. Start with vertices having odd rows and columns
2. Visit all vertices of that type using drawing pattern 9
3. Use an edge of length $\sqrt{5}$ to reach a vertex with either even columns or even rows
4. Visit all vertices of that type using drawing pattern 3
5. Use an edge of length $\sqrt{5}$ to reach a vertex with even columns and even rows
6. Visit all vertices of that type using drawing pattern 4
7. Use an edge of length $\sqrt{5}$ to reach a vertex with either even columns or even rows that has not been visited yet
8. Visit all vertices of that type using drawing pattern 5
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

Yellow squares:
1. Start with vertices having odd rows and columns
2. Visit all vertices of that type using drawing pattern 9
3. Use an edge of length $\sqrt{5}$ to reach a vertex with either even columns or even rows
4. Visit all vertices of that type using drawing pattern 3
5. Use an edge of length $\sqrt{5}$ to reach a vertex with even columns and even rows
6. Visit all vertices of that type using drawing pattern 4
7. Use an edge of length $\sqrt{5}$ to reach a vertex with either even columns or even rows that has not been visited yet
8. Visit all vertices of that type using drawing pattern 3
9. Use an edge of length $\sqrt{5}$ to reach the start vertex

(o,o)	(o,e)	(o,o)	(o,e)
(e,o)	(e,e)	(e,o)	(e,e)
(o,o)	(o,e)		
(e,o)	(e,e)		

(o,o)	(o,e)			(o,o)	(o,e)	(o,o)	(o,e)	(o,o)	(o,e)	(o,o)	(o,e)
(e,o)	(e,e)			(e,o)	(e,e)	(e,o)	(e,e)	(e,o)	(e,e)	(e,o)	(e,e)
		(o,o)	(o,e)			(o,o)	(o,e)	(o,o)	(o,e)		
		(e,o)	(e,e)			(e,o)	(e,e)	(e,o)	(e,e)		
(o,o)	(o,e)			(o,o)	(o,e)	(o,o)	(o,e)				
(e,o)	(e,e)			(e,o)	(e,e)	(e,o)	(e,e)				

Table 18.2: Transitions between the different Hamilton paths. The left figure deals with the green squares in Table 18.1. The central figure is concerned with the orange squares and the right one deals with all other squares. Boxes with gray background color contain the last vertices of a Hamiltonian paths of the applied drawing patterns. Boxes with yellow background color contain the first vertices of Hamiltonian paths of the applied drawing patterns. To link two Hamiltonian paths, we are looking for edges of length $\sqrt{5}$ connecting vertices from boxes with different background color. Pairs of vertices, for which such an edge exists, have the same text color.

Proof. a.) There does not exist a tour using four edges of length $\sqrt{5}$ and otherwise only edges of length 2.:
Contrary to grids with $m \geq 4$ there are no edges of length $\sqrt{5}$ connecting an (o,e)-vertex with an (e,e)-vertex. Hence overall there are no edges of length $\sqrt{5}$ connecting (e,e)-vertices with (o,o)-vertices, (o,e)-vertices with (e,o)-vertices and (o,e)-vertices with (e,e)-vertices. Hence from the (e,e)-vertices we can only reach the (e,o)-vertices by edges of length $\sqrt{5}$ and therefore there can be no closed tour using only four edges of length $\geq \sqrt{5}$.

b.) $2(nm-6)+6\sqrt{5}$ is a lower bound for the length of any TSPN tour with $r=\sqrt{2}$:
There are now two options to connect the different vertex types. Either we use one edge of length ≥ 3 to reach an (o,e)-vertex from an (e,e)-vertex (or the other way around) or we do not visit all vertices of one type at once and hence we use at least two additional edges of length $\sqrt{5}$ to realize for example the following tour: Visit all (o,e)-vertices $\rightarrow$ visit part of the (o,o)-vertices $\rightarrow$ visit part of the (e,o)-vertices $\rightarrow$ visit all (e,e)-vertices $\rightarrow$ visit the rest of (e,o) vertices $\rightarrow$ visit the rest of (o,o) vertices. As $3+2=5>4.47\approx 2\sqrt{5}$ the second option is the shorter one.[3] □

Theorem 18.4 *The optimal TSPN tour with $r=\sqrt{2}$ on an $3\times n$ grid with $n\geq 4$ has length $2(nm-6)+6\sqrt{5}$.*

Proof. The claimed length is a lower bound for the length of any TSPN tour with $r=\sqrt{2}$ due to Lemma 18.16. A TSPN tour with $r=\sqrt{2}$ of the claimed length is given for $4\leq n\leq 6$ in Figure 18.33.

Furthermore it can be constructed on all $3\times n$ grids with $n\geq 7$ as follows, where the colors of the text correspond with the colored numbers in Figure 18.34 below:

1. Start with the upper (o,e)-vertex $(1,2)$.
2. Visit all (o,e)-vertices.
3. Use an edge of length $\sqrt{5}$ to go to the right.
4. Visit the (o,o)-vertices going in clockwise direction until you reach the upper left corner point.
5. Use an edge of length $\sqrt{5}$ to reach the upper right (o,e)-vertex.
6. Visit the (o,e)-vertices in clockwise direction until you reach the (o,e)-vertex one above the lower left (o,e)-vertex.
7. Use an edge of length $\sqrt{5}$ to reach the lower (e,e)-vertex.

[3]Note that the first option is the shorter one if we consider the Manhattan distance instead of the Euclidean distance.

4	1	3
6	8	5
12	2	11
10	7	9

4	1	3
8	11	9
5	2	6
7	12	10

6	1	5
8	10	7
15	2	4
12	9	11
14	3	13

14	1	13
7	4	6
15	2	12
8	5	9
10	3	11

6	1	5
8	12	7
18	2	4
9	11	13
17	3	16
15	10	14

13	1	14
11	4	12
18	2	15
10	5	7
17	3	16
9	6	8

Figure 18.33: Optimal TSPN tours with $r = \sqrt{2}$ on $3 \times n$ grids with $4 \leq n \leq 6$ minimizing the Euclidean distance on the left and Manhattan distance on the right.

8. Visit all (e, e)-vertices.
9. Use an edge of length $\sqrt{5}$ to reach the (o, e)-vertex one below the upper right (o, e)-vertex.
10. Visit the remaining (o, e)-vertices in counter-clockwise direction.
11. Use an edge of length $\sqrt{5}$ to reach the right lower (o, o)-vertex.
12. Visit the remaining (o, o)-vertices in counter-clockwise direction.
13. Use an edge of length $\sqrt{5}$ to reach the start vertex.

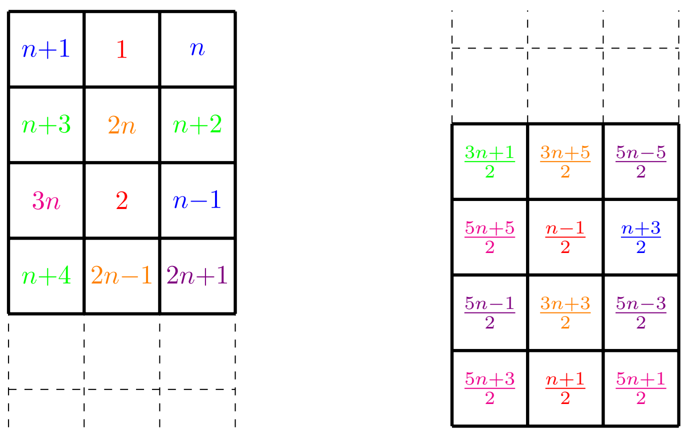


Figure 18.34: Construction of an optimal TSPN tour with $r = \sqrt{2}$ on the $3 \times n$ grid with $n \geq 7$ having length $2(nm - 4) + 6\sqrt{5}$. In the illustration we assume w.l.o.g. that n is odd.

□

Corollary 18.17 *The optimal open TSPN tour with $r = \sqrt{2}$ on the $3 \times n$ grid with $n \geq 4$ has length $2(nm - 4) + 3\sqrt{5}$.*

Proof. The given length is a lower bound because we always need an edge of length $\sqrt{5}$ to connect Hamiltonian paths on the (o, o)-vertices, (e, e)-vertices, (e, o)-vertices or (o, e)-vertices. An open open TSPN tour of the claimed length can be constructed on all $3 \times n$ grids with $n \geq 4$ as follows, where the colors of the text correspond with the colored numbers in Figure 18.35 below:

1. Start with the upper (o, e)-vertex.
2. Visit all (o, e)-vertices.
3. Use an edge of length $\sqrt{5}$ to go to the right.
4. Visit all (o, o)-points going in clockwise direction.
5. Use an edge of length $\sqrt{5}$ to go to the left corner vertex, if possible, otherwise there is only edge of length $\sqrt{5}$.
6. Visit all (o, e)-vertices in clockwise direction.
7. Use an edge of length $\sqrt{5}$ to reach the lowest (e, e)-vertex.
8. Visit all (e, e)-vertices.

□

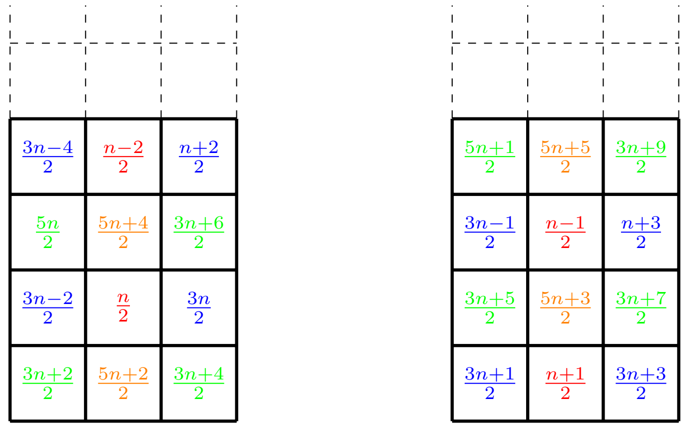


Figure 18.35: Construction of an optimal open TSPN tour with $r = \sqrt{2}$ on the $3 \times n$ grid with $n \geq 4$ having length $2(nm - 4) + 3\sqrt{5}$. On the left we depict the case n even and on the right the case n odd is illustrated.

Let us extend the above results for Manhattan distance. We have seen in Figure 18.33 that the optimal tours minimizing Euclidean and Manhattan distance are different for $3 \times n$ grids with $4 \leq n \leq 6$. In the following corollary we argue that these are the only cases in which the optimal tours differ.

Corollary 18.18 *The optimal TSPN tour with $r = \sqrt{2}$ minimizing Manhattan distance has length $nm+8$ on an $3 \times n$ grid with $n \geq 7$. The optimal open TSPN tour with $r = \sqrt{2}$ minimizing Manhattan distance has length $nm + 1$ on the $3 \times n$ grid with $n \geq 4$.*

Proof. The statement on optimal open TSPN tours follows immediately from Corollary 18.17 as the shortest edges to connect Hamiltonian paths on the (o, o)-vertices, (e, e)-vertices, (e, o)-vertices or (o, e)-vertices have Manhattan distance 3.

The Hamiltonian paths on the (o, o)-vertices, (e, e)-vertices, (e, o)-vertices or (o, e)-vertices can only be connected by edges having an odd Manhattan distance. Hence the suggested optimal tour length $nm+8$ is a valid lower bound for tours, where we use one edge with Manhattan distance ≥ 5. The proposed length $nm + 8$ is also a valid lower bound for tours, where we use 6 edges of Manhattan distance 3 (see Theorem 18.4 for details on the construction of such tours). Finally if we construct a tour using only 4 edges of length 3 to connect the four Hamiltonian paths then we have to use one edge of length 4 and $nm + 8$ is also a valid lower bound for the tour length. Without using an edge of length 4 it is not possible to visit all (o, e)-vertices and afterwards all (e, e)-vertices (or the other way around), see also the proof of Lemma 18.16. A tour using exactly one edge of length 4, four edges of length 3 and otherwise only edges of length 2 can be obtained by switching the last two vertices in the open tour suggested in Corollary 18.17 and then connecting start and finish vertex. □

Finally note that the optimal tours on $2 \times n$ grids have a more involved structure than the optimal tours on all other grid sizes. Hence we exclude them from our analysis here and leave them as an interesting topic for future research.

18.3.4 Results for $r = 2$

In this subsection we examine optimal TSPN tours on $m \times n$ grids with $r = 2$. The shortest available edges have length $\sqrt{5}$. Hence a tour using only such edges is for sure optimal and exists for the majority of grids.

Note that for $r = 2$ the optimal TSPN tours on grids are knight's tours, if they exist. See Figure 18.36 for an illustration of a knight's tour on the 5×6 grid that was discovered by Leonhard Euler, who conducted the first thorough scientific study of knight's tours in 1759 [96]. Schwenk [276] provided a comprehensive result on the existence of knight's tours:

Theorem 18.5 *[276] An $m \times n$ grid with $m \leq n$ has a knight's tour unless one or more of the following three conditions are met:*

a) m and n are both odd;

b) $m = 1, 2,$ or 4; or

c) $m = 3$ and $n = 4, 6,$ or 8.

1	14	25	6	29	16
24	5	30	15	20	7
13	2	11	26	17	28
10	23	4	19	8	21
3	12	9	22	27	18

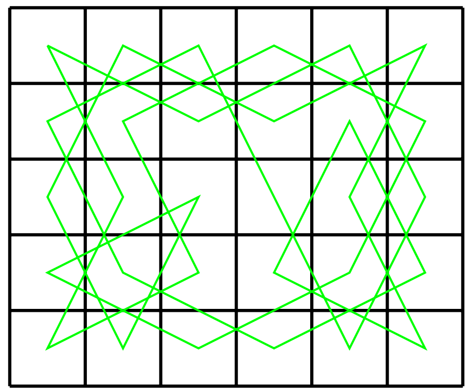


Figure 18.36: Knight's Tour on the 5×6 grid, first discovered by Leonard Euler.

A corresponding result also exists for open knight's tours.

Theorem 18.6 *[74, 76] Every $n \times m$ grid with $m \geq 5$ has an open knight's tour.*

Corollary 18.19 *The optimal TSPN tour with $r = 2$ on an $m \times n$ grid, on which a knight's tour exists, has length $nm\sqrt{5}$ (and $3nm$ when minimizing the Manhattan distance). The optimal open TSPN tour with $r = 2$ on an $m \times n$ grid with $m \geq 5$ has the following has length $(nm - 1)\sqrt{5}$ (and $3(nm - 1)$ when minimizing the Manhattan distance).*

Parberry [248] suggested a divide-and-conquer algorithm that can generate knight's tours on $n \times n$ and $n \times (n + 2)$ grids with $n \geq 10$ even, running in linear time (i.e. O(n^2)) His algorithm can also construct knight's tours on $n \times (n + 1)$ grids with $n \geq 6$. Later on Lin and Wei [210] proposed another approach that can construct knight's tour and open knight's tours on arbitrary $m \times n$ grids for which a tour exists. Their algorithms also run in linear time (i.e. O(nm)).

Finding an optimal TSPN tour with $r = 2$ on $m \times n$ grids, where no knight's tour exists would be an interesting research question that we plan to tackle in the future. Also the study of larger neighborhoods could be worthwhile.

Chapter 19

Conclusion

In this thesis we presented the comparison of existing and the design of new modelling approaches for several well-known problems in facility layout and logistics. We demonstrated that semidefinite relaxations provide theoretically and practically substantially tighter bounds than the corresponding linear programming relaxations. Although computing SDP relaxations is more time consuming, our experiments demonstrate that using them often pays off in practice. Hence we were able to compute optimal solutions for many layout problems, which have been considered in the literature for years, for the first time. The approaches presented in this thesis are the strongest exact methods to date for most row layout problems including the Single-Row (Equidistant) Facility Layout Problem, the Multi-Row Equidistant Facility Layout Problem and several variants of the general Multi-Row Facility Layout Problem.

This strong practical results were possible due to the following reasons:

- The derivation of new theoretical results that facilitated the handling of spaces: In Theorem 12.2 we showed that although the lengths of the spaces are in general continuous quantities, every multi-row problem has an optimal solution on the grid. In Theorem 13.2 we proved exact expressions for the minimum number of spaces that need to be added so as to preserve at least one optimal solution of multi-row equidistant layout problems. We demonstrated that both theoretical results have a significant impact for a computational perspective.

- The execution of polyhedral studies for the different layout problems yielding tight semidefinite relaxations containing the most important constraint classes.

- The usage of the appropriate algorithmic approach, namely bundle methods, for solving these semidefinite relaxations with a large number of constraints.

While there exist quite diverse exact integer linear programming approaches to the various layout problems discussed (due to the different structures of the underlying polytopes and the different cost functions), the presented semidefinite model based on products of ordering variables is uniformly applicable to all considered row and circular layout problems as it builds on the more general quadratic ordering polytope. This generality distinguishes the semidefinite approach and allowed us to tackle an arbitrary combination of row and circular layout problems that we denoted as Combined Cell Layout Problem. Considering multiple machine cells we were able to solve instances with up to 200 departments to optimality which is quite remarkable for facility layout problems that are known to be notoriously difficult.

Additionally we designed new facility layout problems and appropriate algorithmic approaches thereto. On the one hand we introduced the Directed Circular Facility Layout Problem. We showed that it contains several other relevant layout problems as special cases and can be solved by both heuristic and exact methods as a Linear Ordering Problem. On the other hand we proposed the Checkpoint Ordering

Problem and the weighted Linear Ordering Problem that have several interesting connections to well-known combinatorial optimization problems.

In the logistic part of the thesis we tackled the Target Visitation Problem, which is a variant of the famous Traveling Salesman Problem, via a semidefinite formulation and demonstrated the efficiency of our approach on a variety of benchmark instances with up to 50 targets. We also conducted a polyhedral study of the corresponding polytope, improving a semidefinite relaxation proposed by Newman [237]. We related our relaxation to other linear and semidefinite relaxations for the Linear Ordering Problem and for the Traveling Salesman Problem and elaborated on its connection to the Max Cut problem. While the Target Visitation Problem has applications in environmental assessment, combat search and rescue and disaster relief, we suggested another variant of the Traveling Salesman Problem with applications in beam melting. We examined the length and structure of the optimal traveling salesman tours considering different types of forbidden neighborhoods on grid graphs.

In a nutshell we extended the application area of semidefinite optimization and combinatorial optimization in facility layout and logistics through the design of new optimization problems and the development of efficient algorithmic frameworks based on semidefinite (and linear) programming.

Chapter 20

Outlook

We want to conclude by pointing out several research directions that emerged during the work on the projects connected to this thesis. Proposing semidefinite models to (combinatorial) optimization problems in facility layout and logistics is a very new and fruitful area of research, hence there are plenty of unexploited ideas building on the findings and results presented here. First we want to refer to the concluding sections of the papers collected in this thesis, where several possible extensions and generalizations of the current models and approaches are pointed out. Furthermore Chapter 18 contains three new current projects. At the moment we are also working on incorporating the SDP based bounds for the various row layout problems in a Branch-and-Bound framework. In the following few paragraphs we would like to give an outlook on some additional plans.

There exist various integer linear programming and mixed integer programming models for single- and multi-row layout problems that have been reviewed in Chapters 6 and 12 respectively. In the future we aim to develop alternative SDP models for various layout problems that are not only based on products of ordering variables. A first step into that direction has already been taken by the semidefinite model for the Multi-Row Equidistant Facility Layout Problem that allows to optimize simultaneously over the row assignments and the positions of the departments within each row. Extending this idea to the general multi-row case is planned.

Concerning circular layouts we think that the papers included in this thesis open up several further worthwhile fields of research:

- We would like to extend the Directed Circular Facility Layout Problem to the case, where more than one circle is present and hence the assignment of departments to the different circles and the arrangement of the departments within each circle are part of the optimization problem.
- A first step into that direction could be the investigation of the equidistant special case, i.e. the extension of the analysis from Chapter 13 to circular layouts.
- Furthermore circular layouts also allow for assembling more complex structures like e.g. overlapping circles. Designing appropriate models for such layouts is for sure another challenging task.

The paper dealing with the Combined Cell Layout Problem in Chapter 17 can be viewed as a first step towards approximating general two-dimensional layouts (that are extremely hard to solve, see e.g. [171]) through a combination of row and circular layouts. The next step into this direction would be the definition of an appropriate transformation routine that converts two dimensional departments from a general two dimensional facility into an appropriate number of machine cells with row or circular layouts, i.e. to an input of the Combined Cell Layout Problem.

Finally we plan to analyze so-called comb structures, i.e. the arrangement of departments along horizontal and vertical rows that are allowed to overlap. A related layout problem that has been studied in the literature is the Multi-Bay Manufacturing Facility Layout Problem [56, 223]. Designing optimization problems with an underlying combinatorial structure (and of course also the corresponding exact optimization approaches) for this problem type is another possibility to tackle more and more general layout structures in an efficient way, which is obviously the ultimate goal of our research.

Bibliography

[1] H. Achatz, P. Kleinschmidt, and J. Lambsdorff. Der corruption perceptions index und das linear ordering problem. *ORNews*, 26:10–12, 2006.

[2] D. Adolphson. Singlemachine job sequencing with precedence constraints. *SIAM Journal on Computing*, 6:40–54, 1977.

[3] P. Afentakis. A loop layout design problem for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 1:175–196, 1989.

[4] F. Alizadeh, J.-P. Haeberly, and M. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming*, 77:111–128, 1997. ISSN 0025-5610. URL `http://dx.doi.org/10.1007/BF02614432`. 10.1007/BF02614432.

[5] I. K. Altınel and T. Öncan. Design of unidirectional cyclic layouts. *International Journal of Production Research*, 43(19):3983–4008, 2005.

[6] A. R. S. Amaral. On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508–518, 2006.

[7] A. R. S. Amaral. An exact approach to the one-dimensional facility layout problem. *Operations Research*, 56(4):1026–1033, 2008.

[8] A. R. S. Amaral. A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157(1):183–190, 2009.

[9] A. R. S. Amaral. On duplex arrangement of vertices. Technical report, Departamento de Informática, Universidade Federal do Espírito Santo (UFES), Brazil, 2011.

[10] A. R. S. Amaral. The corridor allocation problem. *Computers & Operations Research*, 39(12): 3325–3330, 2012.

[11] A. R. S. Amaral. Optimal solutions for the double row layout problem. *Optimization Letters*, 7(2): 407–413, 2013.

[12] A. R. S. Amaral. A parallel ordering problem in facilities layout. *Computers & Operations Research*, 40(12):2930–2939, 2013.

[13] A. R. S. Amaral and A. N. Letchford. A polyhedral approach to the single row facility layout problem. *Mathematical Programming*, pages 1–25, 2012.

[14] M. Andersen, J. Dahl, and L. Vandenberghe. Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones. *Mathematical Programming Computation*, 2: 167–201, 2010. ISSN 1867-2949.

[15] N. A. G. Aneke and A. S. Carrie. A design technique for the layout of multi-product flowlines. *International Journal of Production Research*, 24(3):471–481, 1986.

[16] M. Anjos, A. Fischer, and P. Hungerländer. Solution approaches for equidistant double- and multi-row facility layout problems. Cahiers du GERAD G-2014-45, GERAD, Montreal, QC, Canada, 2015.

[17] M. Anjos, A. Fischer, and P. Hungerländer. An exact approach for the combined cell layout problem. In *Operations Research Proceedings 2014, accepted*, 2015.

[18] M. F. Anjos and J. B. Lasserre, editors. *Handbook on Semidefinite, Conic and Polynomial Optimization Theory, Algorithms, Software and Applications*. International Series in Operations Research &

Management Science. Springer-Verlag, New York, 2012.

[19] M. F. Anjos and F. Liers. Global approaches for facility layout and VLSI floorplanning. In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications*, International Series in Operations Research and Management Science. Springer, New York, 2011.

[20] M. F. Anjos and A. Vannelli. Computing Globally Optimal Solutions for Single-Row Layout Problems Using Semidefinite Programming and Cutting Planes. *INFORMS Journal On Computing*, 20(4): 611–617, 2008.

[21] M. F. Anjos and G. Yen. Provably near-optimal solutions for very large single-row facility layout problems. *Optimization Methods and Software*, 24(4):805–817, 2009.

[22] M. F. Anjos, A. Kennings, and A. Vannelli. A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optimization*, 2(2):113 – 122, 2005.

[23] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

[24] E. Arkin, M. Bender, E. Demaine, S. Fekete, J. Mitchell, and S. Sethia. Optimal covering tours with turn costs. *SIAM Journal on Computing*, 35(3):531–566, 2005.

[25] E. M. Arkin, S. P. Fekete, K. Islam, H. Meijer, J. S. Mitchell, Y. Núñez-Rodríguez, V. Polishchuk, D. Rappaport, and H. Xiao. Not being (super)thin or solid is hard: A study of grid hamiltonicity. *Computational Geometry*, 42(6–7):582 – 605, 2009.

[26] M. Armbruster, M. Fügenschuh, C. Helmberg, and A. Martin. LP and SDP branch-and-cut algorithms for the minimum graph bisection problem: a computational comparison. *Mathematical Programming Computation*, 4(3):275–306, 2012.

[27] A. Arulselvan, C. Commander, and P. Pardalos. A random keys based genetic algorithm for the target visitation problem. In *Advances in Cooperative Control and Optimization*, pages 389–397. Springer Berlin / Heidelberg, 2007.

[28] M. Assem, B. Ouda, and M. Wahed. Improving operating theatre design using facilities layout planning. In *Biomedical Engineering Conference (CIBEC), 2012 Cairo International*, pages 109–113, 2012.

[29] H. Aujac. La hiérarchie des industries dans un tableau des échanges interindustriels. *Revue économique*, 11(2):169–238, 1960.

[30] V. Balakrishnan and F. Wang. Sdp in systems and control theory. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of semidefinite programming*, volume 27 of *International Series in Operations Research & Management Science*, pages 421–442. Springer US, 2000.

[31] A. Bar-Noy, J. S. Naor, and B. Schieber. Pushing dependent data in clients-providers-servers systems. *Wireless Networks*, 9(5):421–430, 2003.

[32] F. Barahona and A. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[33] J. Békési, G. Galambos, M. Oswald, and G. Reinelt. Comparison of approaches for solving coupled task problems. Technical report, Ruprecht-Karls-Universität Heidelberg, 2008.

[34] R. Bellman and K. Fan. On systems of linear inequalities in hermitian matrix variables. In V. L. Klee, editor, *Convexity*, Proceedings of Symposia in Pure Mathematics, pages 1–11. American Mathematical Society, 1963.

[35] A. Belloni and C. Sagastizabal. Dynamic bundle methods. *Mathematical Programming, Series A*, 120(2):289–311, 2009.

[36] A. Ben-Tal and A. Nemirovski. Structural design. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming*, volume 27 of *International Series in Operations Research & Management Science*, pages 443–467. Springer US, 2000.

[37] L. Bertacco, L. Brunetta, and M. Fischetti. The linear ordering problem with cumulative costs. *European Journal of Operational Research*, 189(3):1345–1357, 2008.

[38] J. Bhasker and S. Sahni. Optimal linear arrangement of circuit components. *Proceedings of the 20th*

Annual Hawaii International Conference on System Sciences, 2:99–111, 1987.

[39] K. Boenchendorf. *Reihenfolgenprobleme/Mean-flow-time sequencing*. Mathematical Systems in Economics 74. Verlagsgruppe AthenÄaum, Hain, Scriptor, 1982.

[40] B. Borchers. *CSDP 5.0 User's Guide*. URL `http://www.optimization-online.org/DB_HTML/2002/10/551.html`, 2005.

[41] C. F. Bornstein and S. Vempala. Flow metrics. *Theoretical Computer Science*, 321:13–24, 2004.

[42] N. Bova, O. Ibanez, and O. Cordon. Image segmentation using extended topological active nets optimized by scatter search. *Computational Intelligence Magazine, IEEE*, 8(1):16–32, 2013.

[43] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[44] S. E. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, USA, 1994. ISBN 0-89871-334-X.

[45] Y. Bozer and S. Rim. Exact solution procedures for the circular layout problem. Technical report, University of Michigan, USA, Report No. 89-33, 1989.

[46] M. Brusco and S. Stahl. Using quadratic assignment methods to generate initial permutations for least-squares unidimensional scaling of symmetric proximity matrices. *Journal of Classification*, 17 (2):197–223, 2000.

[47] C. Buchheim, F. Liers, and M. Oswald. Speeding up ip-based algorithms for constrained quadratic 0–1 optimization. *Mathematical Programming*, 124:513–535, 2010. ISSN 0025-5610.

[48] C. Buchheim, A. Wiegele, and L. Zheng. Exact Algorithms for the Quadratic Linear Ordering Problem. *INFORMS Journal on Computing*, 22(1):168–177, 2010.

[49] S. Burer and R. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (B)*, 95:329–357, 2003.

[50] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, 2009.

[51] K. c. Toh, M. j. Todd, and R. H. Tütüncü. Sdpt3 - a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

[52] V. Campos, F. Glover, M. Laguna, and R. Martí. An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, 21(4):397–414, 2001.

[53] A. Caprara and J.-J. Salazar-González. Laying out sparse graphs with provably minimum bandwidth. *INFORMS Journal on Computing*, 17:356–373, 2005.

[54] A. Caprara, M. Jung, M. Oswald, G. Reinelt, and E. Traversi. Optimal linear arrangements using betweenness variables. *Mathematical Programming Computation*, 3(3):261–280, 2011.

[55] A. Caprara, A. N. Letchford, and J.-J. Salazar-González. Decorous lower bounds for minimum linear arrangement. *INFORMS Journal on Computing*, 23(1):26–40, 2011.

[56] I. Castillo and B. A. Peters. Integrating design and production planning considerations in multi-bay manufacturing facility layout. *European Journal of Operational Research*, 157(3):671 – 687, 2004.

[57] S. Chanas and P. Kobylanski. A new heuristic algorithm solving the linear ordering problem. *Computational Optimization and Applications*, 6(2):191–205, 1996.

[58] M. Charikar, M. T. Hajiaghayi, H. Karloff, and S. Rao. l_2^2 spreading metrics for vertex ordering problems. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, SODA '06, pages 1018–1027, New York, USA, 2006.

[59] P. P.-S. Chen. The entity-relationship model–toward a unified view of data. *ACM Transactions on Database Systems*, 1:9–36, 1976.

[60] H. Chenery and T. Watanabe. International comparisons of the structure of production. *Econometrica*, 26:487–521, 1958.

[61] M. Chimani and P. Hungerländer. Multi-level verticality optimization: Concept, strategies, and drawing scheme. *Journal of Graph Algorithms and Applications*, 2012. accepted, preprint available

at www.ae.uni-jena.de/Research_Pubs/MLVO.html.

[62] M. Chimani and P. Hungerländer. Exact approaches to multilevel vertical orderings. *INFORMS Journal on Computing*, 25(4):611–624, 2013.

[63] M. Chimani, P. Hungerländer, M. Jünger, and P. Mutzel. An SDP approach to multi-level crossing minimization. In *Proceedings of Algorithm Engineering & Experiments [ALENEX'2011]*, pages 116–126, 2011.

[64] M. Chimani, P. Hungerländer, M. Jünger, and P. Mutzel. An SDP approach to multi-level crossing minimization. *Journal of Experimental Algorithmics*, 17:3.3:3.1–3.3:3.26, 2012.

[65] W.-M. Chow. An analysis of automated storage and retrieval systems in manufacturing assembly lines. *IIE Transactions*, 18(2):204–214, 1986.

[66] T. Christof. *Low-dimensional 0/1-polytopes and branch-and-cut in combinatorial optimization*. PhD thesis, Ruprecht-Karls-Universität Heidelberg, Aachen, 1997.

[67] T. Christof, M. Jünger, and G. Reinelt. A complete description of the traveling salesman polytope on 8 nodes. *Operation Research Letters*, 10(9):497–500, 1991.

[68] T. Christof, M. Oswald, and G. Reinelt. Consecutive ones and a betweenness problem in computational biology. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO 1998)*, Lecture Notes in Computer Science 1412, pages 213–228. Springer, 1998.

[69] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical report, GSIA, Cranegie-Mellon University, 1976.

[70] N. Christofides and E. Benavent. An exact algorithm for the quadratic assignment problem on a tree. *Operations Research*, 37(5):760–768, 1989.

[71] C.-H. Chu. Recent advances in mathematical programming for cell formation. In *Planning, Design, and Analysis of Cellular Manufacturing Systems*, number 24 in Manufacturing Research and Technology, pages 3–46. Elsevier Science B.V., 1995.

[72] J. Chung and J. Tanchoco. The double row layout problem. *International Journal of Production Research*, 48(3):709–727, 2010.

[73] M. Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues a la pluralité des voix*. Paris, 1785.

[74] A. Conrad, T. Hindrichs, H. Morsy, and I. Wegener. Solution of the knight's hamiltonian path problem on chessboards. *Discrete Applied Mathematics*, 50(2):125–134, 1994.

[75] W. J. Cook. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. Princeton University Press, 2011.

[76] P. Cull and J. De Curtlns. Knight's tour revisited. *Fibonacci Quarterly*, 16:276–285, 1978.

[77] D. Cvetković, M. Cangalović, and V. Kovačevič-Vujčić. Semidefinite programming methods for the symmetric traveling salesman problem. In *Proceedings of the 7th International IPCO Conferenceo on Integer Programming and Combinatorial Optimization*, pages 126–136. Springer, 1999.

[78] G. Dantzig, D. Fulkerson, and S. Johnson. Solution of a large scale traveling salesman problem. *Journal of the Operations Research Society of America*, 2:393–410, 1954.

[79] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[80] D. Datta, A. R. S. Amaral, and J. R. Figueira. Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2):388–394, 2011.

[81] T. Davi and F. Jarre. High-accuracy solution of large-scale semidefinite programs. *Optimization Methods Software*, 27(4-5):655–666, 2012.

[82] T. Davi and F. Jarre. On the stable solution of large scale problems over the doubly nonnegative cone. *Mathematical Programming*, pages 1–25, 2013.

[83] E. de Klerk. *Aspects of semidefinite programming: interior point algorithms and selected applications*. Kluwer Academic Publishers, 2002.

[84] E. de Klerk and R. Sotirov. Improved semidefinite programming bounds for quadratic assignment

problems with suitable symmetry. *Mathematical Programming*, 133(1-2):75–91, 2012.

[85] E. de Klerk, J. Peng, C. Roos, and T. Terlaky. A scaled gauss-newton primal-dual search direction for semidefinite optimization. *SIAM Journal on Optimization*, 11:870–888, 2001.

[86] E. de Klerk, D. V. Pasechnik, and R. Sotirov. On semidefinite programming relaxations of the traveling salesman problem. *SIAM Journal on Optimization*, 19(4):1559–1573, 2008.

[87] E. D. Demaine, J. S. B. Mitchell, and J. O'Rourke. The open problems project. Technical report, `http://maven.smith.edu/~orourke/TOPP/`.

[88] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer Verlag, Berlin, 1997.

[89] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34: 313–356, 2002.

[90] J. Dickey and J. Hopkins. Campus building arrangement using topaz. *Transportation Research*, 6 (1):59–68, 1972.

[91] Z. Drezner. Finding a cluster of points and the grey pattern quadratic assignment problem. *OR Spectrum*, 28:417–436, 2006.

[92] A. Duarte, R. Martí, A. Álvarez, and F. Ángel-Bello. Metaheuristics for the linear ordering problem with cumulative costs. *European Journal of Operational Research*, 216(2):270–277, 2012.

[93] I. S. Duff, R. G. Grimes, and J. G. Lewis. Users' guide for the harwell-boeing sparse matrix collection. Technical report, CERFACS, Toulouse, France, 1992.

[94] R. J. Duffin. Infinite programs. In *Linear inequalities and related systems*, number 38 in Annals of Mathematics Studies, pages 157–170. Princeton University Press, 1956.

[95] A. N. Elshafei. Hospital layout as a quadratic assignment problem. *Operational Research Quarterly*, 28(1):167–179, 1977.

[96] L. Euler. Solution d'une question curieuse que ne paroit soumise à aucune analyse. *Mémoires de l'académie des sciences de Berlin*, 15:310–337, 1759.

[97] U. Feige and M. X. Goemans. Approximating the value of two power proof systems, with applications to MAX 2SAT and MAX DICUT. In *Proceedings of the Third Israel Symposium on Theory of Computing and Systems*, pages 182–189, 1995.

[98] U. Feige and J. Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57:187–199, 1998. ISSN 0022-0000.

[99] U. Feige and J. R. Lee. An improved approximation ratio for the minimum linear arrangement problem. *Information Processing Letters*, 101:26–29, 2007.

[100] T. A. Feo and M. G. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[101] M. Ficko, M. Brezocnik, and J. Balic. Designing the layout of single- and multiple-rows flexible manufacturing system by genetic algorithms. *Journal of Materials Processing Technology*, 157–158: 150–158, 2004.

[102] A. Fischer. *A Polyhedral Study of Quadratic Traveling Salesman Problems*. PhD thesis, Chemnitz University of Technology, Germany, 2013.

[103] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle method for semidefinite cutten plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105:451–469, 2006.

[104] P. C. Fishburn. Induced binary probabilities and the linear ordering polytope: a status report. *Mathematical Social Sciences*, 23(1):67–80, 1992.

[105] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13:117–156, 2002.

[106] J. Frenzel. Genetic algorithms. *Potentials, IEEE*, 12(3):21–24, 1993.

[107] A. Frieze and M. Jerrum. Improved approximation algorithms for MAXk-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997.

[108] K. Fujisawa, M. Kojima, and K. Nakata. Exploiting sparsity in primal-dual interior-point methods

for semidefinite programming. *Mathematical Programming*, 79:235–253, 1997.

[109] C. P. Gane and T. Sarson. *Structured Systems Analysis: Tools and Techniques.* Prentice Hall Professional Technical Reference, 1st edition, 1979.

[110] C. G. Garcia, D. Pérez-Brito, V. Campos, and R. Martí. Variable neighborhood search for the linear ordering problem. *Computer & Operations Research*, 33(12):3549–3565, 2006.

[111] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

[112] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, New York, 1974.

[113] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976.

[114] M. Gen, K. Ida, and C. Cheng. Multirow machine layout problem in fuzzy environment using genetic algorithms. *Computers & Industrial Engineering*, 29(1–4):519–523, 1995.

[115] A. Geoffrion and G. Graves. Scheduling parallel production lines with changeover costs : practical applications of a quadratic assignment/lp approach. *Operations Research*, 24:595–610, 1976.

[116] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1): 156–166, 1977.

[117] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[118] F. Glover and M. Laguna. *Tabu search.* Kluwer Academic Publishers, 1997.

[119] F. Glover, T. Klastorin, and D. Klingman. Optimal weighted ancestry relationships. *Management Science*, 20:1190–1193, 1974.

[120] M. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79:143–161, 1997. ISSN 0025-5610.

[121] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[122] M. X. Goemans and F. Rendl. Combinatorial optimization. In H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming.* Kluwer Academic Publishers, 2000.

[123] A. Gomes de Alvarenga, F. J. Negreiros-Gomes, and M. Mestria. Metaheuristic methods for a class of the facility layout problem. *Journal of Intelligent Manufacturing*, 11:421–430, 2000.

[124] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Reasearch*, 32(6):1195–1220, 1984.

[125] D. Grundel and D. Jeffcoat. Formulation and solution of the target visitation problem. *Proceedings of the AIAA 1st Intelligent Systems Technical Conference*, 2004.

[126] *Gurobi optimizer reference manual.* Gurobi Optimization Inc., 2014. URL `http://www.gurobi.com`.

[127] G. Gutin and A. Punnen. *The Traveling Salesman Problem and Its Variations.* Springer, 2002.

[128] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.

[129] E. Halperin and U. Zwick. A unified framework for obtaining improved approximation algorithms for maximum graph bisection problems. *Random Structures and Algorithms*, 20(3):382–402, 2002.

[130] P. Hammer. Some network flow problems solved with pseudo-Boolean programming. *Operations Research*, 13:388–399, 1965.

[131] L. H. Harper. Optimal assignments of numbers to vertices. *SIAM Journal on Applied Mathematics*, 12:131–135, 1964.

[132] L. H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1:385–393, 1966.

[133] M. M. D. Hassan. Machine layout problem in modern manufacturing facilities. *International Journal*

of Production Research, 32(11):2559–2584, 1994.

[134] J. Håstad. Some optimal inapproximability results. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 1–10, New York, NY, USA, 1997. ACM.

[135] J. Håstad. Clique is hard to approximate within $n^{1\text{-epsilon}}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(38), 1997.

[136] P. Healy and A. Kuusik. The vertex-exchange graph: A new concept for multilevel crossing minimisation. In *Proceedings of the Symposium on Graph Drawing [GD'99]*, pages 205–216. Springer, 1999.

[137] M. Held and R. Karp. The traveling salesman problem and minimum spanning trees. *Operations Research*, 18:1138–1162, 1970.

[138] C. Helmberg. Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3):952–969, 2000.

[139] C. Helmberg. Semidefinite programming. *European Journal of Operational Research*, 137:461–482, 2002.

[140] C. Helmberg. Numerical validation of SBmethod. *Mathematical Programming*, 95:381–406, 2003.

[141] C. Helmberg. *A Cutting Plane Algorithm for Large Scale Semidefinite Relaxations*, chapter 15, pages 233–256. 2004. doi: 10.1137/1.9780898718805.ch15. URL `http://epubs.siam.org/doi/abs/10.1137/1.9780898718805.ch15`.

[142] C. Helmberg. *ConicBundle 0.3.11.* Fakultät für Mathematik, Technische Universität Chemnitz, 2012. URL `http://www.tu-chemnitz.de/~helmberg/ConicBundle`.

[143] C. Helmberg and F. Oustry. Bundle methods to minimize the maximum eigenvalue function. In R. S. H. Wolkowicz and L. Vandenberghe, editors, *Handbook of semidefinite programming: theory, algorithms and applications*, volume 27 of *International Series in Operations Research & Management Science*, pages 307–337. Springer US, 2000.

[144] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 1999.

[145] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl. A spectral approach to bandwidth and separator problems in graphs. *Linear and Multilinear Algebra*, 39:73–90, 1995.

[146] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[147] C. Helmberg, K. Kiwiel, and F. Rendl. Incorporating inequality constraints in the spectral bundle method. In E. B. R.E. Bixby and R. Rios-Mercado, editors, *Integer Programming and combinatorial optimization*, pages 423–435. Springer Lecture Notes 1412, 1998.

[148] S. S. Heragu. *Facilities Design.* iUniverse, 2nd edition, 2006.

[149] S. S. Heragu and A. S. Alfa. Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research*, 57(2):190–202, 1992.

[150] S. S. Heragu and A. Kusiak. Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268, 1988.

[151] S. S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1):1–13, 1991.

[152] A. Hildenbrandt, O. Heismann, and G. Reinelt. The target visitation problem. Presentation held at the 18th Combinatorial Optimization Workshop in Aussois on January 09, 2014.

[153] A. Hildenbrandt, G. Reinelt, and O. Heismann. Integer programming models for the target visitation problem. In *Proceedings of the 16th International Multiconference of the Information Society*, volume A, pages 569–572, 2013.

[154] J.-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and minimization algorithms (vol. 1 and 2).* Springer, 1993.

[155] H.-W. Holub and H. Schnabl. *Input-Output-Rechnung: Input-Output-Tabellen.* Oldenbourg Wissenschaftsverlag, 1982.

[156] P. Hungerländer. *Semidefinite Approaches to Ordering Problems.* PhD thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2012.

[157] P. Hungerländer. A semidefinite optimization approach to the directed circular facility layout problem. In *7th IFAC Conference on Manufacturing Modelling, Management, and Control*, pages 2033–2038, 2013.

[158] P. Hungerländer. A new modelling approach for cyclic layouts and its practical advantages. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-01, 2014.

[159] P. Hungerländer. A semidefinite optimization approach to the parallel row ordering problem. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-05, 2014.

[160] P. Hungerländer. The checkpoint ordering problem. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-02, 2014.

[161] P. Hungerländer. A semidefinite optimization approach to the target visitation problem. *Optimization Letters*, 2014. doi: 10.1007/s11590-014-0824-9.

[162] P. Hungerländer. New semidefinite programming relaxations for the linear ordering and the traveling salesman problem. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-03, 2014.

[163] P. Hungerländer. Single-row equidistant facility layout as a special case of single-row facility layout. *International Journal of Production Research*, 52(5):1257–1268, 2014.

[164] P. Hungerländer and M. Anjos. A semidefinite optimization approach to space-free multi-row facility layout. Cahiers du GERAD G-2012-03, GERAD, Montreal, QC, Canada, 2012.

[165] P. Hungerländer and M. F. Anjos. An exact approach for the combined cell layout problem. In *Operations Research Proceedings 2012*, pages 275–281, 2013.

[166] P. Hungerländer and M. F. Anjos. An exact approach for the combined cell layout problem. Technical report, Alpen-Adria Universität Klagenfurt, Mathematics, Optimization Group, TR-ARUK-M-O-14-04, 2014.

[167] P. Hungerländer and M. F. Anjos. A semidefinite optimization-based approach for global optimization of multi-row facility layout. *European Journal of Operational Research*, 2015. accepted.

[168] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming*, 140(1):77–97, 2013.

[169] P. Hungerländer and F. Rendl. A computational study and survey of methods for the single-row facility layout problem. *Computational Optimization and Applications*, 55(1):1–20, 2013.

[170] A. Itai, C. Papadimitriou, and J. Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982.

[171] I. Jankovits, C. Luo, M. F. Anjos, and A. Vannelli. A convex optimisation framework for the unequal-areas facility layout problem. *European Journal of Operational Research*, 214(2):199 – 215, 2011.

[172] F. Jarre and F. Rendl. An augmented primal-dual method for linear conic problems. *SIAM Journal on Optimization*, 20:808–823, 2009.

[173] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1:1–25, 1997.

[174] M. Juvan and B. Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics*, 36(2):153–168, 1992.

[175] R. Kaas. A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, 8(4):355–362, 1981.

[176] H. Kaplan, M. Lewenstein, N. Shafrir, and M. Sviridenko. Approximation algorithms for asymmetric TSP by decomposing directed regular multigraphs. *Journal of the ACM*, 52(4):602–626, 2005.

[177] D. Karger, R. Motwani, and M. Sudan. Approximate graph coloring by semidefinite programming.

Journal of the ACM, 45(2):246–265, 1998.

[178] R. Karp. Reducibility among combinatorial problems. In R. Miller and J.W.Thather, editors, *Complexity of Computer Computation*, pages 85–103. Plenum Press, 1972.

[179] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[180] R. M. Karp. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 278–285, New York, USA, 1993. ACM.

[181] R. M. Karp and M. Held. Finite-state processes and dynamic programming. *SIAM Journal on Applied Mathematics*, 15(3):693–718, 1967.

[182] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[183] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[184] D. E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing*. ACM, New York, NY, USA, 1993. ISBN 0-201-54275-7.

[185] M. Kojima, S. Shindoh, and S. Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM Journal on Optimization*, 7(1):86–125, 1997.

[186] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.

[187] Y. Koren and D. Harel. A multi-scale algorithm for the linear arrangement problem. In *Revised Papers from the 28th International Workshop on Graph-Theoretic Concepts in Computer Science*, WG '02, pages 296–309, London, UK, 2002. Springer-Verlag.

[188] R. Kothari and D. Ghosh. Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods. *European Journal of Operational Research*, 224(1):93–100, 2013.

[189] P. Kouvelis and M. W. Kim. Unidirectional loop network layout problem in automated manufacturing systems. *Operations Research*, 40:533–550, 1992.

[190] P. Kouvelis, W.-C. Chiang, and A. Kiran. A survey of layout issues in flexible manufacturing systems. *Omega*, 20(3):375–390, 1992.

[191] P. Kouvelis, W.-C. Chiang, and G. Yu. Optimal algorithms for row layout problems in automated manufacturing systems. *IIE Transactions*, 27(1):99–104, 1995.

[192] S. Kruk, M. Muramatsu, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. The gauss-newton direction in semidefinite programming. *Optimization Methods and Software*, 15(1):1–28, 2001.

[193] K. R. Kumar, G. C. Hadjinicola, and T. li Lin. A heuristic procedure for the single-row facility layout problem. *European Journal of Operational Research*, 87(1):65 – 73, 1995.

[194] M. Laguna and R. Martí. *Scatter search: methodology and implementations in C*. Kluwer Academic Publishers, 2003.

[195] M. Laguna, R. Martí, and V. Campos. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computer & Operations Research*, 26(12):1217–1230, 1999.

[196] M. Lampis. Improved inapproximability for tsp. In A. Gupta, K. Jansen, J. Rolim, and R. Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 7408 of *Lecture Notes in Computer Science*, pages 243–253. Springer Berlin Heidelberg, 2012.

[197] A. Langevin, B. Montreuil, and D. Riopel. Spine layout design. *International Journal of Production Research*, 32(2):429–442, 1994.

[198] G. Laporte and H. Mercure. Balancing hydraulic turbine runners: A quadratic assignment problem. *European Journal of Operational Research*, 35(3):378–381, 1988.

[199] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224:439–461, 1995.

[200] M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17:530–547, 1996.

[201] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Discrete Optimization*, pages 393–514. Elsevier, 2005.

[202] C. Lemarechal. An extension of davidon methods to nondifferentiable problems. *Mathematical Programming Study*, 3:95–109, 1975.

[203] C. Lemarechal. Nonsmooth optimization and descent methods. Technical report, International Institute for Applied Systems Analysis, 1978.

[204] C. Lemarechal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–147, 1995.

[205] W. Leontief. Quantitative input-output relations in the economic system of the united states. *The Review of Economics and Statistics*, 18(3):105–125, 1936.

[206] W. Leontief. *Input-output economics*. Oxford University Press, 1966.

[207] J. Leung. Polyhedral structure and properties of a model for layout design. *European Journal of Operational Research*, 77(2):195–207, 1994.

[208] L. Li and K.-C. Toh. An inexact interior point method for l1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2:291–315, 2010. ISSN 1867-2949.

[209] V. Liberatore. Circular arrangements. In P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy, editors, *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 782–783. Springer Berlin / Heidelberg, 2002.

[210] S.-S. Lin and C.-L. Wei. Optimal algorithms for constructing knight's tours on arbitrary chessboards. *Discrete Applied Mathematics*, 146(3):219 – 232, 2005.

[211] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.

[212] R. Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47:57–66, 2003.

[213] L. Lovász. On the shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[214] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[215] R. F. Love and J. Y. Wong. On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14:139–143, 1967.

[216] B. Malakooti. Unidirectional loop network layout by a LP heuristic and design of telecommunications networks. *Journal of Intelligent Manufacturing*, 15(1):117–125, 2004.

[217] J. Malick, J. Povh, F. Rendl, and A. Wiegele. Regularization methods for semidefinite programming. *SIAM Journal on Optimization*, 20:336–356, 2009.

[218] R. Martí and M. Laguna. Heuristics and meta-heuristics for 2-layer straight line crossing minimization. *Discrete Applied Mathematics*, 127(3):665–678, 2003. ISSN 0166-218X. doi: http://dx.doi.org/10.1016/S0166-218X(02)00397-9.

[219] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*. Applied Mathematical Sciences. Springer, 2011.

[220] R. Martí, G. Reinelt, and A. Duarte. A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, pages 1–21, 2011.

[221] A. J. McAllister. A new heuristic algorithm for the linear arrangement problem. Technical report, University of New Brunswick, Faculty of Computer Science, TR-99-126a, 1999.

[222] R. Meller and K.-Y. Gau. The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 5(5):351–366, 1996.

[223] R. D. Meller. The multi-bay manufacturing facility layout problem. *International Journal of Production Research*, 35(5):1229–1237, 1997.

[224] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, 1994.

[225] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In T. T. H. Frenk, K. Roos and S. Zhang, editors, *High Performance Optimization*, pages 349–366. Kluwer Academic Publishers, 2000.

[226] G. Mitchison and R. Durbin. Optimal numberings of an N N array. *SIAM Journal on Algebraic and Discrete Methods*, 7:571–582, 1986.

[227] H. Mittelmann. Benchmarks for optimization software. `http://plato.asu.edu/bench.html`.

[228] R. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM Journal on Optmization*, 7:663–678, 1997.

[229] R. Murphey and P. Pardalos, editors. *Cooperative Control and Optimization.* Springer, 2002.

[230] C. C. Murray, A. E. Smith, and Z. Zhang. An efficient local search heuristic for the double row layout problem with asymmetric material flow. *International Journal of Production Research*, 51 (20):6129–6139, 2013.

[231] J. S. Naor and R. Schwartz. The directed circular arrangement problem. *ACM Transactions on Algorithms*, 6:47:1–47:22, 2010.

[232] A. C. Nearchou. Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, 101(2):312–328, 2006.

[233] A. S. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17: 191–234, 2008.

[234] Y. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. Discussion paper 9719, CORE, Catholic University of Louvain, Belgium, 1997.

[235] Y. Nesterov and A. Nemirovski. *Interior Point Polynomial Algorithms in Convex Programming.* SIAM Publications. SIAM, Philadelphia, USA, 1994.

[236] Y. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22:1–42, 1997.

[237] A. Newman. Cuts and orderings: On semidefinite relaxations for the linear ordering problem. In K. Jansen, S. Khanna, J. Rolim, and D. Ron, editors, *Lecture Notes in Computer Science*, volume 3122, pages 195–206. Springer, 2004.

[238] A. Newman and S. Vempala. Fences are futile: On relaxations for the linear ordering problem. In *Proceedings of the 8th International IPCO Conferenceo on Integer Programming and Combinatorial Optimization*, pages 333–347. Springer, 2001.

[239] V. S. Nori and B. R. Sarker. Designing multi-product lines: job routing in cellular manufacturing systems. *Journal of the Operational Research Society*, 48(4):412–422, 1997.

[240] C. E. Nugent, T. E. Vollmann, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16(1):150–173, 1968.

[241] T. Obata. *Quadratic assignment problem: evaluation of exact and heuristic algorithms.* PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, 1979.

[242] T. Öncan and I. K. Altınel. Exact solution procedures for the balanced unidirectional cyclic layout problem. *European Journal of Operational Research*, 189(3):609–623, 2008.

[243] J. Ostrowski, M. F. Anjos, and A. Vannelli. Tight mixed integer linear programming formulations for the unit commitment problem. *Power Systems, IEEE Transactions on*, 27(1):39 –46, feb. 2012. ISSN 0885-8950. doi: 10.1109/TPWRS.2011.2162008.

[244] F. Ozcelik and A. Islier. Unidirectional loop layout problem with balanced flow. In M. Ali and R. Dapoigny, editors, *Advances in Applied Artificial Intelligence*, volume 4031 of *Lecture Notes in Computer Science*, pages 741–749. Springer Berlin / Heidelberg, 2006.

[245] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[246] G. Palubeckis. A branch-and-bound algorithm for the single-row equidistant facility layout problem. *OR Spectrum*, 34:1–21, 2012.

[247] C. H. Papadimitriou and S. Vempala. On the approximability of the traveling salesman problem. *Combinatorica*, 26(1):101–120, 2006.

[248] I. Parberry. An efficient algorithm for the knight's tour problem. *Discrete Applied Mathematics*, 73 (3):251 – 260, 1997.

[249] P. A. Parillo. *Structured Semidefinite Programs and Semi-algebraic Geometry Methods in Robustness and Optimization.* PhD thesis, California Institute of Technology, Pasadena, California, USA, 2000.

[250] J. Petit. Combining spectral sequencing and parallel simulated annealing for the MinLA problem. *Parallel Processing Letters*, 13(1):77–91, 2003.

[251] J. Petit. Experiments on the minimum linear arrangement problem. *ACM Journal of Experimental Algorithmics*, 8, 2003.

[252] J.-C. Picard and M. Queyranne. On the one-dimensional space allocation problem. *Operations Research*, 29(2):371–391, 1981.

[253] M. Pollatschek, N. Gershoni, and Y. Radday. Optimization of the typewriter keyboard by computer simulation. *Angewandte Informatik*, 10:438–439, 1976.

[254] F. A. Potra and R. Sheng. A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming. *SIAM Journal on Optimization*, 8:1007–1028, 1998.

[255] J. Povh, F. Rendl, and A. Wiegele. A boundary point method to solve semidefinite programs. *Computing*, 78:277–286, 2006.

[256] M. V. Ramana and P. M. Pardalos. Semidefinite programming. In T. Terlaky, editor, *Interior point methods of mathematical programming*, pages 369–398. Kluwer, Dordrecht, The Netherlands, 1996.

[257] S. Rao and A. W. Richa. New approximation techniques for some linear ordering problems. *SIAM Journal on Computing*, 34:388–404, 2005.

[258] R. Ravi, A. Agrawal, and P. Klein. Ordering problems approximated: Single-processor scheduling and interval graphs connection. In J. L. Albert, B. R. Artalejo, and B. Monien, editors, *18th International Colloquium on Automata, Languages and Programming*, volume 150 of *Lecture Notes in Computer Science*, pages 751–762. Springer-Verlag New York, 1991.

[259] G. Reinelt. *The Linear Ordering Problem: Algorithms and Applications.* Heldermann, 1985.

[260] G. Reinelt. *The traveling salesman: computational solutions for TSP applications.* Springer, 1994.

[261] F. Rendl. Semidefinite relaxations for integer programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 687–726. Springer Berlin Heidelberg, 2010. ISBN 978-3-540-68279-0.

[262] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.

[263] C. C. Ribeiro, E. Uchoa, and R. F. Werneck. A hybrid grasp with perturbations for the steiner problem in graphs. *INFORMS Journal on Computing*, 14(3):228–246, 2002.

[264] R. T. Rockafellar. *Convex analysis.* Princeton Mathematical Series, No. 28. Princeton University Press, 1970.

[265] E. Rodriguez-Tello, J.-K. Hao, and J. Torres-Jimenez. An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. *Computers & Operations Research*, 35: 3331–3346, 2008.

[266] D. Romero and A. Sánchez-Flores. Methods for the one-dimensional space allocation problem. *Computers & Operations Research*, 17(5):465–473, 1990.

[267] H. Samarghandi and K. Eshghi. An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*, 205(1):98 – 105, 2010.

[268] S. Sanjeevi and K. Kianfar. A polyhedral study of triplet formulation for single row facility layout problem. *Discrete Applied Mathematics*, 158:1861–1867, 2010.

[269] B. Sarker, W. Wilhelm, and G. Hogg. One-dimensional machine location problems in a multi-product flowline with equidistant locations. *European Journal of Operational Research*, 105(3):401–426, 1998.

[270] B. R. Sarker. *The amoebic matrix and one-dimensional machine location problems*. PhD thesis, Department of Industrial Engineering, Texas A&M University, College Station, TX, 1989.

[271] B. R. Sarker and Y. Xu. Designing multi-product lines: job routing in cellular manufacturing systems. *IIE Transactions*, 32:219–235, 2000.

[272] B. R. Sarker and J. Yu. A two-phase procedure for duplicating bottleneck machines in a linear layout, cellular manufacturing system. *International Journal of Production Research*, 32(9):2049–2066, 1994.

[273] T. Schiavinotto and T. Stützle. The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 3(4):367–402, 2005.

[274] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM J. Optimization*, 2:121–152, 1992.

[275] R. Schwarz. *A branch-and-cut algorithm with betweenness variables for the linear arrangement problems*. Diploma Thesis, Heidelberg, 2010.

[276] A. J. Schwenk. Which rectangular chessboards have a knight's tour? *Mathematics Magazine*, 64(5): 325–332, 1991.

[277] H. Seitz. *Contributions to the Minimum Linear Arrangement Problem*. PhD thesis, University of Heidelberg, Germany, 2010.

[278] H. Sherali and W. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3): 411–430, 1990.

[279] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3): 411–430, 1990.

[280] H. D. Sherali and W. P. Adams. *Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Springer, 1998.

[281] M. Shida, S. Shindoh, and M. Kojima. Existence of search directions in interior-point algorithms for the sdp and the monotone sdlcp. *SIAM Journal on Optimization*, 8(2):387–396, 1998.

[282] D. B. Shmoys and D. P. Williamson. Analyzing the held-karp tsp bound: A monotonicity property with application. *Information Processing Letters*, 35(6):281–285, 1990.

[283] D. M. Simmons. One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17:812–826, 1969.

[284] D. M. Simmons. A further note on one-dimensional space allocation. *Operations Research*, 19:249, 1971.

[285] C. D. Simone. The cut polytope and the Boolean quadric polytope. *Discrete Mathematics*, 79(1): 71–75, 1990.

[286] P. Slater. Inconsistencies in a schedule of paired comparisons. *Biometrika*, 48(3-4):303–312, 1961.

[287] W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3 (1-2):59–66, 1956.

[288] L. Steinberg. The backboard wiring problem: A placement algorithm. *SIAM Review*, 3(1):37–50, 1961.

[289] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

[290] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.

[291] J. Suryanarayanan, B. Golden, and Q. Wang. A new heuristic for the linear placement problem. *Computers & Operations Research*, 18(3):255–262, 1991.

[292] T. G. Szymanski. Assembling code for machines with span-dependent instructions. *Communications of the ACM*, 21:300–308, 1978.

[293] B. C. Tansel and C. Bilen. Move based heuristics for the unidirectional loop network layout problem. *European Journal of Operational Research*, 108(1):36–48, 1998.

[294] M. Todd. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optimization Methods and Software*, 11:1–46, 1999.

[295] J. A. Tompkins. Modularity and flexibility: dealing with future shock in facilities design. *Industrial Engineering*, pages 78–81, 1980.

[296] A. W. Tucker. On directed graphs and integer programs. Technical report, IBM Mathematical Research Project, 1960.

[297] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95:189–217, 2003.

[298] C. Umans and W. Lenhart. Hamiltonian cycles in solid grid graphs. In *38th Annual Symposium on Foundations of Computer Science*, pages 496–505, 1997.

[299] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38:49–95, 1996.

[300] A. Vanelli and G. S. Rowan. An eigenvector based approach for multistack vlsi layout. In *Proceedings of the Midwest Symposium on Circuits and Systems*, volume 29, pages 135–139, 1986.

[301] S. Wang and B. R. Sarker. Locating cells with bottleneck machines in cellular manufacturing systems. *International Journal of Production Research*, 40(2):403–424, 2002.

[302] B. Wess and T. Zeitlhofer. On the phase coupling problem between data memory layout generation and address pointer assignment. In H. Schepers, editor, *Software and Compilers for Embedded Systems*, volume 3199 of *Lecture Notes in Computer Science*, pages 152–166. Springer Berlin Heidelberg, 2004.

[303] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.

[304] L. A. Wolsey. Heuristic analysis, linear programming and branch and bound. *Mathematical Programming Studies*, 13:121–134, 1980.

[305] S. Wright. *Primal-dual interior point methods*. SIAM, Philadelphia, 1997.

[306] D. H. Younger. Minimum feedback arc sets for a directed graph. *IEEE Transactions on Circuit Theory*, 10(2):238–245, 1963.

[307] J. Yu. *Machine-cell location problems for multi-product flowlines*. PhD thesis, Department of Industrial and Manufacturing Systems Engineering, Louisiana State University, Baton Rouge, 1999.

[308] J. Yu and B. R. Sarker. Directional decomposition heuristic for a linear machine-cell location problem. *European Journal of Operational Research*, 149(1):142–184, 2003.

[309] Y. Zhang. On extending some primal–dual interior-point algorithms from linear programming to semidefinite programming. *SIAM Journal on Optimization*, 8:365–386, 1998.

[310] Z. Zhang and C. Murray. A corrected formulation for the double row layout problem. *International Journal of Production Research*, 2011.

[311] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.

Philipp Hungerländer

"Mathematics has beauty and romance. It's not a boring place to be, the mathematical world. It's an extraordinary place; it's worth spending time there." - Marcus du Sautoy

Contact

current affiliation	**Alpen-Adria-Universität Klagenfurt, Department of Mathematics** Universitätsstraße 65-67, 9020 Klagenfurt, Austria
phone	**+43 463 2700 3117**
email	**philipp.hungerlaender@aau.at**
homepage	**http://philipphungerlaender.jimdo.com/**

Education

Mar 2012 – Nov 2014	**PhD in Economics**, *Alpen-Adria-Universität Klagenfurt*.
title of thesis	**Semidefinite Optimization Approaches to Applications in Facility Layout and Logistics**, *Supervisor: Miguel Anjos*.
Feb 2009 – Feb 2012	**PhD in Combinatorics and Optimization**, *Alpen-Adria-Universität Klagenfurt*.
title of thesis	**Semidefinite Approaches to Ordering Problems**, *Supervisor: Franz Rendl*.
Oct 2003 – Oct 2012	**(Under)Graduate in Business and Law**, *Alpen-Adria-Universität Klagenfurt*.
title of thesis	**The Prices of Anarchy, Information and Cooperation**, *Supervisor: Reinhard Neck*.
Oct 2003 – Feb 2009	**(Under)Graduate in Mathematics**, *Alpen-Adria-Universität Klagenfurt*, Specializations: Optimization, Game Theory.
title of thesis	**Algorithms for Convex Quadratic Programming**, *Supervisor: Franz Rendl*.
Oct 2003 – May 2008	**(Under)Graduate in Business Administration**, *Alpen-Adria-Universität Klagenfurt*, Specializations: Finance, Logistics, Economics.

title of thesis	**Discrete-Time Dynamic Noncooperative Game Theory**, *Supervisor: Reinhard Neck*.
Sep 1995 – June 2003	**Secondary School**, *BRG Spittal*.
Sep 1991 – June 1995	**Elementary School**, *Weißenstein*.

Academic Employment

since Sep 2012	**Senior assistant professor (Assistenzprofessor)**, Alpen-Adria-Universität Klagenfurt, Departement of Mathematics.
Oct 2014 - Feb 2015	**Senior researcher**, Technische Universität Dortmund, Department of Mathematics.
Sep 2008 – Sep 2012	**Research and teaching assistant (Universitätsassistent)**, Alpen-Adria-Universität Klagenfurt, Departement of Mathematics.
Aug 2011 - Oct 2011	**Researcher**, École Polytechnique de Montréal, Departement of Mathematics.
Feb 2007 - Aug 2008	**Project Assistent (Projektassistent)**, Alpen-Adria-Universität Klagenfurt, Departement of Econnomics.
Oct 2006 - Jan 2007	**Internship**, *Robert-Holzmann-Institut, Vienna*, Topic: Dynamic game theory.
Mar 2006 - June 2006	**Graduate Assistent**, Alpen-Adria-Universität Klagenfurt, Departement of Finance.

Other Employment

Mar 2007 – June 2007	**Project Assistent**, *"Pewag" company*, Topic: Development of a consolidation software.
Summer 03, Summer 02	**Tennis Instructor**.
Oct 2001 – Sep 2002	**Student representative**, BRG Spittal.
Aug 2001	**Internship**.

Professional Awards and Recognition

2015 **Doctoral degree in Economics under the auspices of the Federal President**, Highest possible distinction for excellent performance in academic studies in Austria.

2014 **Heidelberg Laureate Forum**, Selected participant.

2014 **"Audience Award" of the "Long Night of Research"**, Most popular inactive presentation at the Alpen-Adria Universität Klagenfurt.

2012 **Best Dissertation Award**, Austrian Society for Operations Research.

2012 **Best Dissertation Award**, Austrian Mathematical Society.

2012 **Doctoral degree in Optimization under the auspices of the Federal President**, Highest possible distinction for excellent performance in academic studies in Austria.

2010 **Award for Excellent Supervision of Talented Pupils**, Austrian Federal Ministry for Transport, Innovation and Technology.

2009 **Kurier High Potential '09**, Finalist.

2009 **WiWi Talents Program/Book 2008/2009**, Admission.

2008 **Lupe 2008**, Award for excellent science communication.

2008 **Award for excellent achievements in the university studies**, Austrian Federal Ministry of Research and Science.

Jan 2007 – Aug 2008 **Scholarship of the Austrian National Bank**, Research project: "The Macroeconomics of EMU Enlargement".

2008 **Award for the best master thesis in Economics**, Alpen-Adria Universität Klagenfurt.

2006 **European Forum Alpbach**, Scholarship holder.

2003–2009 **Merit scholarship**, Academic years 2003/04, 2004/05, 2005/06, 2006/07, 2007/08 and 2008/09.

Research Publications

Research Articles in International Journals

[1] H. and F. Rendl. **A Feasible Active Set Method for Strictly Convex Problems with Simple Bounds.** *SIAM Journal on Optimization*, 2015, accepted.

[2] H. and M. F. Anjos. **Semidefinite Optimization Approaches to Multi-Row Facility Layout.** *European Journal of Operational Research*, doi:10.1016/j.ejor.2015.02.049, 2015

[3] H. . **A Semidefinite Optimization Approach to the Target Visitation Problem.** *Optimization Letters*, doi:10.1007/s11590-014-0824-9, 2014.

[4] H. . **Single-Row Equidistant Facility Layout as a Special Case of Single-Row Facility Layout.** *International Journal of Production Research*, Vol. 52(5), pp. 1257-1268, 2014.

[5] M. Chimani and H. . **Multi-Level Verticality Optimization: Concept, Strategies, and Drawing Scheme.** *Journal of Graph Algorithms and Applications*, Vol. 17(3), pp. 329-362, 2013.

[6] M. Chimani and H. . **Exact Approaches to Multi-Level Vertical Orderings.** *INFORMS Journal on Computing*, Vol. 25(4), pp. 611-624, 2013.

[7] H. and F. Rendl. **A Computational Study and Survey of Methods for the Single-Row Facility Layout Problem.** *Computational Optimization and Applications*, Vol. 55(1), pp 1-20, 2013.

[8] H. and F. Rendl. **Semidenite Relaxations of Ordering Problems.** *Mathematical Programming*, Vol. 140(1), pp 77-97, 2013.

[9] M. Chimani, H. , M. Jünger and P. Mutzel. **An SDP approach to multi-level crossing minimization.** *Journal of Experimental Algorithmics*, 2012, Vol. 17(3), Article 3.3.

Research Articles in Refereed Conference Proceedings

[1] M.F. Anjos, A. Fischer and H. **Solution Approaches for the Double-Row Equidistant Facility Layout Problem.** *In Operations Research Proceedings 2014*, to appear.

[2] H. and M. F. Anjos. **An Exact Approach for the Combined Cell Layout Problem.** *In Operations Research Proceedings 2012*, pp. 275-281, 2013.

[3] H. . **A Semidefinite Optimization Approach to the Directed Circular Facility Layout Problem.** *In Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control*, pp. 2033-2038, 2013.

[4] M. Chimani, H. , M. Jünger and P. Mutzel. **An SDP approach to multi-level crossing minimization.** *In Proceedings of Algorithm Engineering & Experiments [ALENEX'2011]*, 2011.

Book Chapters and Lecture Notes

[1] H. and R. Neck. **An algorithmic equilibrium solution for n-person dynamic Stackelberg difference games with open-loop information pattern.** *In: H. Dawid et al.: Computational Methods in Economic Dynamics*, Springer Publishers, pp 197-214, 2010.

Preprints and Work in Progress

[1] H. and M. F. Anjos. **A Semidefinite Optimization Approach to Space-Free Multi-Row Facility Layout.**

[2] H. and M. F. Anjos. **An Exact Approach for the Combined Cell Layout Problem.**

[3] H. **A New Modelling Approach for Cyclic Layouts and its Practical Advantages.**

[4] H. **The Checkpoint Ordering Problem.**

[5] H. and F. Rendl. **An Infeasible Active Set Method with Step Size Control for Bound Constrained Convex Problems.**

[6] M. F. Anjos, A. Fischer and H. . **Solution Approaches for Equidistant Double- and Multi-Row Facility Layout Problems**

[7] H. **New Semidefinite Programming Relaxations for the Linear Ordering and the Traveling Salesman Problem.**

Technical Reports

[1] H. **A Semidefinite Optimization Approach for the Parallel Ordering Problem.**

[2] H. **Differential Games: Egoism, Cooperation and Altrusim.**

PhD and Master Theses

[1] **Semidefinite Optimization Approaches to Applications in Facility Layout and Logistics.** *PhD Thesis Economics*, 2014.

[2] **Semidefinite Approaches to Ordering Problems.** *PhD Thesis Mathematics*, 2012.

[3] **The Prices of Anarchy, Information and Cooperation.** *Master Thesis Business and Law*, 2012.

[4] **Algorithms for Convex Quadratic Programming.** *Master Thesis Mathematics*, 2009.

[5] **Discrete-Time Dynamic Noncooperative Game Theory.** *Master Thesis Economics*, 2008.

Grants

Jun 2015 - Jun 2017 **Austrian Federal Ministry of Research and Science**,
Exact Approaches to Combinatorial Problems in Facility Layout and Logistics,
Research and travel grants for outstanding young researchers € 11500.

May 2014 **Alpen-Adria Universität Klagenfurt, Lakeside Science & Technology Park and Kleine Zeitung**,
Publicity poly for recruiting mathematics students € 11800.

Feb 2014 **Alpen-Adria Universität Klagenfurt, Department of Mathematics**,
Graphics and computer programs to popularize row-layout problems,
Project financing a master student (Benjamin Hackl) for one month € 1100.

Nov 2013 **Program committee of the Long Night of Research**,
The committee consisted of representatives of the Alpen-Adria-Universität Klagenfurt, the Lakeside Science & Technology Park and media partners,
Financial support to realize our project "Who can quiet the animal shelter?" € 800.

Oct 2012 - Oct 2014 **Austrian Federal Ministry of Research and Science**,
Exact Approaches to Ordering Problems,
Research and travel grants for outstanding young researchers € 11500.

Research Supervision

Bachelor Students

2014 **Game Theory - Braess's Paradox and Selfish Routing.**
Mr. Christian Truden, Alpen-Adria Universität Klagenfurt.

2014 **The Taxi Routing Problem.**
Mrs. Stefanie Kokarnig, Alpen-Adria Universität Klagenfurt.

2012 **The Double Row Facility Layout Problem.**
Mr. Lucas Gregori, Alpen-Adria Universität Klagenfurt.

Summer Interns

2012 **Facility Layout Problems.**
Mr. Benjamin Hackl, Ms. Johanna Mlekusch and Mr. Peter Wiltsche,
Alpen-Adria Universität Klagenfurt.

2011 **Graph Drawing and Quadratic Assignment Problem.**
Mr. Benjamin Hackl and Ms. Miriam Smolnik, Alpen-Adria Universität Klagenfurt.

2010 **Traveling Salesman and Target Visitation Problem.**
Mr. Benjamin Hackl and Ms. Marlene Radl, Alpen-Adria Universität Klagenfurt.

Research Presentation

Invited Seminar Presentations

Jan 2015 **Exact Approaches to Row Layout Problems.**
Oberseminar on Discrete Optimization, Dortmund, Germany.

Jun 2012 **An SDP Approach to Circular and Multi-Row Facility Layout.**
Chemnitz, Germany.

Sep 2011 **Semidefinite Approaches to Ordering Problems.**
GERAD-Mprime Seminar in Optimization, Montreal, Canada.

Mar 2010 **A feasible active set method for convex problems with simple bounds.**
Institute of Mathematics and Scientific Computing, University of Graz, Austria.

Invited Conference Presentations (Minisymposia)

May 2014 **A feasible active set method for box-constrained convex problems.**
SIAM Conference on Optimization, San Diego, USA.

Sep 2013 **The Price of Cooperation in Differential Games**,
IFIP 2013, System Modelling and Optimization, Klagenfurt, Austria.

Aug 2012 **SDP Approaches to some Facility Layout Problems**,
21st International Symposium on Mathematical Programming, Berlin, Germany.

May 2011 **Semidefinite Relaxations of Ordering Problems**,
SIAM Conference on Optimization, Darmstadt, Germany.

Contributed Conference Presentations

Jul 2014 **An Infeasible Active Set Method with Step Size Control.**
Optimization 2014, Guimarães, Portugal.

Jul 2014 **An SDP Approach to the Parallel Row Ordering Problem.**
20^{th} Conference of the International Federation of OR Societies, Barcelona, Spain.

Jul 2014 **The Checkpoint Ordering Problem.**
12^{th} EUROPT Workshop on Advances in Continuous Optimization, Perpignan, France.

May 2014 **A Feasible Active Set Method for Box-Constrained Convex Problems.**
SIAM Conference on Optimization, San Diego, USA.

May 2014 **SDP Relaxations for the Linear Ordering and Traveling Salesman Problem.**
27^{th} ECCO Conference, Munich, Germany.

Jun 2013 **An SDP Approach to the Directed Circular Facility Layout Problem.**
7^{th} IFAC Conference on Manufacturing Modelling, Management, and Control, Sankt Petersburg, Russia.

Sep 2012 **An SDP Approach for Complex Facility Layout Structures.**
International Conference of the German OR Society 2012, Hannover, Germany.

Aug 2012 **A Comparison of Approaches for Ordering Problems.**
Future Research in Combinatorial Optimization 2012 (FRICO), Berlin, Germany.

Jul 2012 **A Semidefinite Optimization Approach to Multi-Row Facility Layout.**
25th European Conference on Operational Research (EURO), Vilnius, Lithuania.

Apr 2012 **An SDP Approach to Single-Row and Space-Free Multi-Row Facility Layout.**
25th ECCO Conference, Antalya, Turkey.

May 2011 **Exact Approaches to Mulit-level Vertical Orderings.**
2nd Alpen-Adria Workshop on Optimization 2011, Klagenfurt, Austria.

Jan 2011 **An SDP Approach to Multi-level Crossing Minimization.**
Algorithm Engineering & Experiments 2011, San Francisco, California, USA.

Jan 2011 **An SDP Approach to Multi-level Crossing Minimization.**
15th Combinatorial Optimization Workshop, Aussois, France.

Sep 2010 **An SDP Approach to Quadratic Ordering Problems.**
Future Research in Combinatorial Optimization 2010 (FRICO), Graz, Austria.

Jun 2010 **Nonstandard active set methods for convex problems with simple bounds.**
1st Alpen-Adria Workshop on Optimization, Klagenfurt, Austria.

Dec 2008 **N-Person Dynamic Stackelberg Affine-Quadratic Difference Games.**
35th Macromodels International Conference, Warsaw, Poland

Jul 2008 **N-Person Dynamic Stackelberg Affine-Quadratic Difference Games.**
13th International Symposium on Dynamic Games and Applications, Wroclaw, Poland.

Jun 2008 **N-Person Dynamic Stackelberg Open-Loop Difference Games.**
14th International Conference on Computing in Economics and Finance, Paris, France.

Poster Presentations

Sep 2014 **New Active Set Methods for Nonlinear Convex Optimization Problems.**
2nd Postgraduate Student & Young Faculty Conference, Klagenfurt, Austria.

Popularization of Mathematics

Dec 2014, Apr 2013 **Who can build the best city?**
Quadratic Assignment Problem, Workshop for several school classes at the Alpen-Adria Universität Klagenfurt.

Sep 2014 **Semidefinite Programming and its Famous Applications in Mathematics and Computer Science.**
Workshop at the Heidelberg Laureate Forum.

Apr 2014 **Who can quiet the animal shelter?**
Single-Row Facility Layout Problem, Interactive Presentation at the Long Night of Research.

Feb 2014 **How to behave in an interrogation?**
Game Theory, Presentation at the University for Kids at the Alpen-Adria Universität Klagenfurt.

Dec 2013, Nov 2009 **How to behave in an interrogation and a test of courage?**
Game Theory, Presentation for pupils at the Alpen-Adria Universität Klagenfurt.

Feb 2013 **Who can quiet the school class?**
Row Facility Layout Problems, Presentation at the University for Kids at the Alpen-Adria Universität Klagenfurt.

Apr 2012 **Who can build the best city?**
Quadratic Assignment Problem, Interactive Presentation at the Long Night of Research.

Feb 2011 **A Journey through the Alpen-Adria-Area.**
Traveling Salesman Problem, Presentation at the University for Kids at the Alpen-Adria Universität Klagenfurt.

Nov 2010 **Do you find the best itinerary?**
Target Visitation Problem, Interactive Presentation at the Long Night of Research at the Alpen-Adria Universität Klagenfurt.

Nov 2008 **Do you find the shortest path?**
Chinese Postman Problem, Interactive Presentation at the Long Night of Research at the Alpen-Adria Universität Klagenfurt.

Referee

I act as a referee for international journals, conferences and books including:

2×since 2014 **SIAM Journal on Optimization**.

6×since 2010 **Mathematical Programming A and B**.

6×since 2011 **European Journal on Operational Research**.

2015 **Optimization Letters**.

2015 **Computers & Operations Research**.

2×since 2014 **International Journal of Production Research**.

2×since 2010 **Discrete Applied Mathematics**.

2013 **Optimization Methods and Software**.

2015 **Applied Mathematics and Computation**.

2014 **Logistics Research**.

2013 **IPCO**.

2014 **ISCO**.

2010 **Handbook of Semidefinite, Cone and Polynomial Optimization**.

Research Visits

Oct 2015 – Dec 2016 **Planned: Massachusetts Institute of Technology**, Financed by an Erwin Schrödinger Fellowship.

2014 **Universidade de Coimbra**, *Department of Mathematics*, one week.

2012 **Technische Universität Chemnitz**, *Department of Mathematics*, one week.

2011 **Univerza v Ljubljani**, *Department of Mathematics*, three days.

2011 **Universität zu Köln**, *Department of Computer Science*, three days.

2010 **Ruprecht-Karls-Universität Heidelberg**, *Department of Computer Science*, two weeks.

Teaching Experience

My teaching includes lectures and exercise sessions for the study programs mathematics, computer science, information technology, business administration, and for teacher training.

Alpen-Adria-Universität Klagenfurt, Departement of Mathematics

Summer 15, Winter 12/13 **Lecture "Algorithmic Graph Theory" (45h)**, Undergraduate course in the area of discrete mathematics.

Summer 15 **Proseminar "Discrete Mathematics" (30h)**, Undergraduate course in the area of discrete mathematics.

Summer 14, Summer 13 **Lecture "Linear Optimization" (30h)**, Undergraduate course in the area of operations research.

Summer 14, 13 and 10 **2 $\times$ Exercises "Linear Optimization" (15h)**, Exercise part of the above lecture.

Winter 13/14 **Lecture "Combinatorial Structures" (45h)**, Undergraduate course in the area of discrete mathematics.

Winter 13/14 **Exercises "Combinatorial Structures" (15h)**, Exercise part of the above lecture.

Winter 12/13 **2 $\times$ Exercises "Mathematics for Economic Sciences" (15h)**, Exercise part of an undergraduate course in Economics.

Summer 12 **Lecture "Game Theory" (45h)**, Undergraduate course in the area of operations research.

Summer 11 **Lecture "Mathematical Models in Economics" (45h)**, Undergraduate course in the area of operations research.

Winter 11/12, Winter 10/11 **Lecture "Nonlinear Optimization" (45h)**, Undergraduate course in the area of operations research.

Winter 09/10 **2 $\times$ Exercises "Analysis I" (30h)**, Exercise part of an introductory undergraduate course.

Summer 09 **Exercises "Analysis II" (30h)**, Exercise part of an undergraduate course in the area of analysis.

Winter 08/09 **Exercises "Discrete Mathematics" (30h)**,
Exercise part of an undergraduate course in the area of discrete mathematics.

Summer 09 **Tutorial "Mathematics in Business Administration" (30h)**,
A supplementary course for business students.

Summer 09 **Tutorial "Analysis I" (30h)**,
A supplementary course in the area of analysis.

Winter 06/07 **Tutorial "Discrete Mathematics" (30h)**,
Winter 05/06 A supplementary course in the area of discrete mathematics.

Technische Univeristät Dortmund, Departement of Mathematics

Winter 06/07 **Lecture "Combinatorial Optimization" (45h)**,
Winter 05/06 A graduate course in the area of operations research.

Administrative activities

since 2011 Member of the **Curricularkommission Computer Science**.

since 2008 Member of the **department panel** of the department of Mathematics in Klagenfurt.

since 2010 **Co-organization** of 4 conferences at the Alpen-Adria-Universität Klagenfurt.

Public Relations

3×since 2011 **University for Kids**.

4×since 2008 **Long Night of Research**.

since 2009 Presentations and workshops in **high schools**.

2010–2012 Supervision of pupils under the project **Forschung macht Schule** in July each year.

Languages

Fluent **German, English**

Intermediate **French**

Basic **Spanish, Russian**

Further Qualification

I attended workshops and trainings for proposal writing, presentation skills, communication and leadership. Additionally I attended several scientific workshops and seminars including:

Sep 2014	**Heidelberg Laureate Forum 2014**, Germany.
Aug 2014	**EURO Summer Institute (ESI) 2010**, *Nonlinear Methods in Combinatorial Optimization*, Klagenfurt, Austria.
May 2010	**Oberwolfach–Seminar**, *Semidefinite Optimization: Theory, Algorithms and Applications*, Germany.
Jul 2007 – Jul 2009	**COMISEF-Research- and Trainingsnetwork**, Attendance of several workshops in Europe.