A Semidefinite Optimization Approach to the Parallel Row Ordering Problem

P. Hungerländer
Alpen-Adria Universität Klagenfurt
Institut für Mathematik
9020 Klagenfurt, Austria
philipp.hungerlaender@uni-klu.ac.at

March 17, 2015

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 a good complement to a mixed integer programming (MIP) formulation that was proposed by Amaral [7] 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.

Key words. Facilities planning and design; Flexible manufacturing systems; Semidefinite Programming; Global Optimization

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 [42] 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 [29]. 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 [43]. In practice, two of the most frequently encountered layout types are the single-row layout (Figure 1) and multi-row layouts (Figure 2) [31].

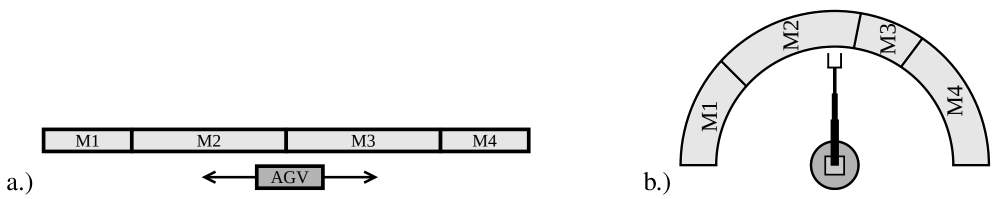

Figure 1: In a.) an AGV transports parts between the machines moving in both directions along a straight line. In b.) a material-handling industrial robot carries parts between the machines.

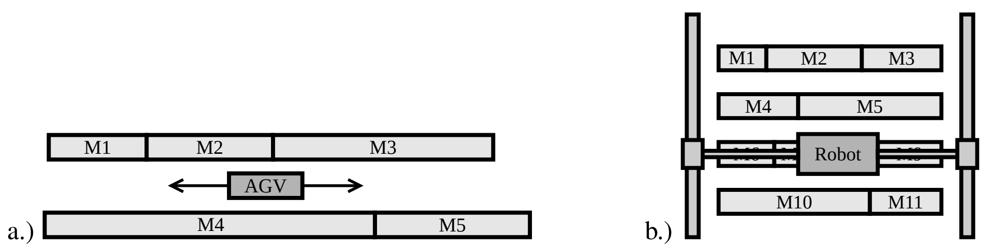

Figure 2: In a.) an AGV transports parts between the machines that are located on both sides of a linear path of travel. In b.) a gantry robot is used when the space is limited.

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 [32]. An instance of the `(SRFLP)` consists of n one-dimensional machines, with given positive lengths $l_1, \dots, l_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

$$\min_{\pi \in \Pi_n} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi}, \tag{1}$$

where Π_n is the set of permutations of the indices $[n] := \{1, 2, \dots, n\}$ 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 [49], the assignment of airplanes to gates in an airport terminal [52], the arrangement of machines in flexible manufacturing systems [32], the arrangement of books on a shelf and the assignment of disk cylinders to files [44]. Accordingly several heuristic algorithms have been suggested to tackle instances of interesting size of the (SRFLP), the best ones to date are [19, 37, 47].

The (SRFLP) is NP-hard, even if all department lengths are equal and the connectivities are binary [25] . 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 [4] that can solve instances with up to 35 departments within a few hours. The strongest SDP approach to date using products of ordering variables [36] 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 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:

$$\min_{\pi^1 \in \Pi^1, \ldots, \pi^k \in \Pi^k} \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} z_{ij}^{\pi^{r(i)}, \pi^{r(j)}}, \tag{2}$$

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 [7] 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 $r : \ [n] \rightarrow \mathcal{R}$ 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:

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

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 [53], 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. [38]. The best methods known to date for the (SF−DRFLP) are an algorithm based on

[1] We will discuss two different ways for defining the distance between pairs of departments in Subsection 3.1.

a MIP formulation proposed by Amaral [6] and an SDP approach suggest by Hungerländer and Anjos [34] 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 [6] also proposed two heuristics (based on 2-opt and 3-opt) and showed that these heuristics can handle larger instances with 30 departments. Hungerländer and Anjos [34] 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. Additionally very recently Ahonen et al. [1] suggested several heuristic methods to tackle the (SF−DRFLP).

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). 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 multiple 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 [50], campus planning [21], scheduling [27], typewriter keyboard design [45], hospital layout [22], the layout of machines in an automated manufacturing system [33], balancing hydraulic turbine runners [39], numerical analysis [14], optimal digital signal processors memory layout generation [55]. Different extensions of the (MRFLP) like considering a clearance between any two adjacent machines given as a fuzzy set [26] or the design of a FMS in one or multiple rows [23] 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 [32] proposed a non-linear programming model and obtained locally optimal solutions to the (SRFLP) and the (DRFLP). Recently Chung and Tanchoco [18] (see also [58]) 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 [5] 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 $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4$. Additionally we are given the pairwise connectivities $c_{2,3} = 0,\ c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$. Figure 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 a good complement to a MIP formulation that was proposed by Amaral [7] 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 2 we recall different formulations and relaxations for the (SRFLP). In Section 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

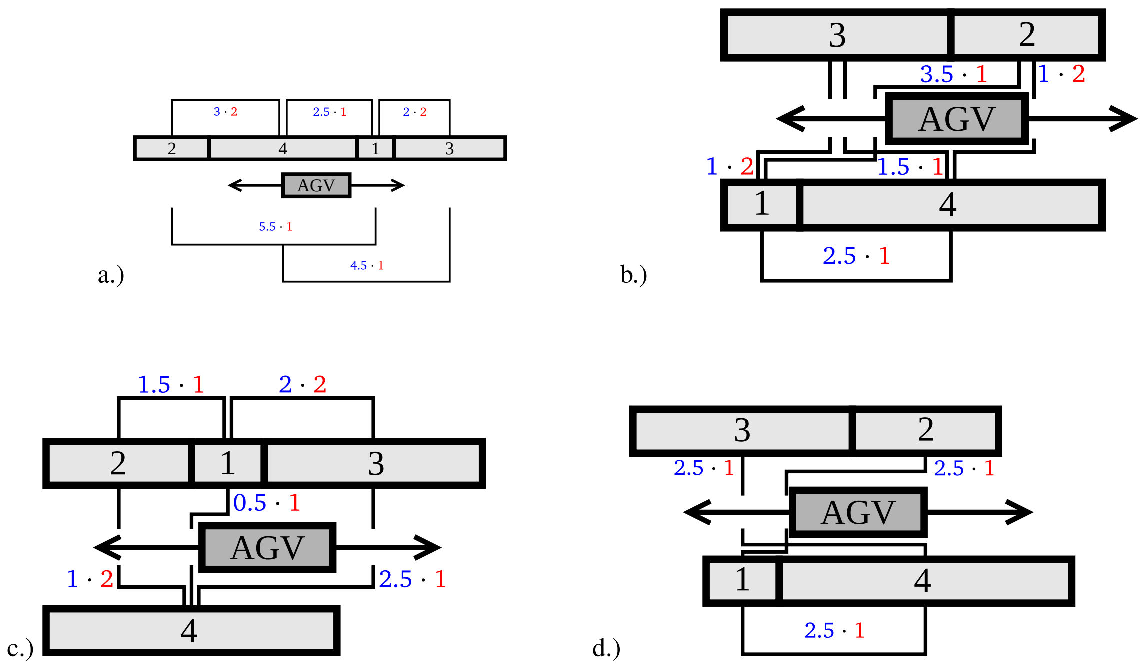

Figure 3: We are given the following data: $l_1 = 1,\ l_2 = 2,\ l_3 = 3,\ l_4 = 4,\ c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$. In a.) we display the optimal layout for the (SRFLP) with associated costs of $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$. In b.) we depict the optimal layout for the (PROP) with machines 3 and 2 assigned to row 1 and machines 1 and 4 assigned 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 $2.5 \cdot 1 + 2.5 \cdot 1 + 2.5 \cdot 1 = 7.5$.

(SRFLP) plus two further constraint classes. In Section 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 5. Section 6 concludes the paper.

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 [41] and Amaral [2]. Both models suffer from weak lower bounds and hence have high computation times and memory requirements. Recently Amaral and Letchford [8] 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).

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 (1) in terms of ζ (for details see [4, Proposition 1 and 2]):

$$\min_{\zeta\in\mathcal{P}^n_{Btw}} \sum_{\substack{i,j,k\in[n],\\ i<j,k<j}} (c_{ij}l_k - c_{ik}l_j)\,\zeta_{ijk} + \sum_{\substack{i,j\in[n]\\ i<j}} \left(\frac{c_{ij}}{2}(l_i + l_j) + \sum_{\substack{k\in[n],\\ k>j}} c_{ij}l_k \right). \tag{3}$$

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 $\mathcal{P}^n_{Btw}$

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i < j < k \in [n]. \tag{4}$$

In [48] 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 [4] 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.

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 [9, 56] 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 $y_{ij},\ i, j \in [n],\ i < j$,

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise,} \end{cases} \tag{5}$$

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

$$\zeta_{ijk} = \frac{1 + y_{ik}y_{kj}}{2},\ i < k < j, \qquad \zeta_{ijk} = \frac{1 - y_{ki}y_{kj}}{2},\ k < i < j, \qquad \zeta_{ijk} = \frac{1 - y_{ik}y_{jk}}{2},\ i < j < k, \tag{6}$$

for $i, j, k \in [n]$. Using (6) we can easily rewrite the objective function (3) and equalities (4) in terms of ordering variables

$$K - \sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \left(\sum_{\substack{k \in [n] \\ k<i}} l_k y_{ki} y_{kj} - \sum_{\substack{k \in [n] \\ i<k<j}} l_k y_{ik} y_{kj} + \sum_{\substack{k \in [n] \\ k>j}} l_k y_{ik} y_{jk} \right), \tag{7}$$

$$y_{ij} y_{jk} - y_{ij} y_{ik} - y_{ik} y_{jk} = -1, \quad i < j < k \in [n], \tag{8}$$

with

$$K = \left(\sum_{\substack{i,j \in [n] \\ i<j}} \frac{c_{ij}}{2} \right) \left(\sum_{k \in [n]} l_k \right). \tag{9}$$

In [15] it is shown that the equations (8) formulated in a $\{0, 1\}$ model describe the smallest linear subspace that contains the quadratic ordering polytope

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

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 the (SRFLP) as a semidefinite program, first proposed in [13]

$$\min \left\{ \langle C, Y \rangle + K : \ Y \text{ satisfies (8)}, \ \ \text{diag}(Y) = e, \ \text{rank}(Y) = 1, \ Y \succcurlyeq 0 \right\}, \tag{SRFLP}$$

where the cost matrix C is deduced from (7). Dropping the rank constraint yields the basic semidefinite relaxation of the (SRFLP)

$$\min \left\{ \langle C, Y \rangle + K : \ Y \text{ satisfies (8)}, \ \ \text{diag}(Y) = e, \ Y \succcurlyeq 0 \right\}, \tag{$\text{SDP}_{\text{trivial}}$}$$

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 [11] proposed to further tighten ($\text{SDP}_{\text{trivial}}$) by adding the triangle inequalities, defining the metric polytope $\mathcal{M}$ and known to be facet-defining for the cut polytope, see e.g. [20]

$$\mathcal{M} = \left\{ Y : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{i,j} \\ Y_{j,k} \\ Y_{i,k} \end{pmatrix} \leq e, \ i < j < k \in \binom{n}{2} \right\}. \tag{10}$$

Adding the triangle inequalities to ($\text{SDP}_{\text{trivial}}$), we obtain the following relaxation of the (SRFLP)

$$\min \left\{ \langle C, Y \rangle + K : \ Y \text{ satisfies (8)}, \ Y \in \mathcal{M}, \ \text{diag}(Y) = e, \ Y \succcurlyeq 0 \right\}. \tag{$\text{SDP}_{\text{basic}}$}$$

As solving ($\text{SDP}_{\text{basic}}$) directly with an interior-point solver like CSDP gets far too expensive, Anjos and Vannelli [11] suggest to use the $\approx \frac{1}{12} n^6$ triangle inequalities as cutting planes in their algorithmic framework.

Recently Hungerländer and Rendl [36] suggested a further strengthening of ($\text{SDP}_{\text{basic}}$) and an alternative algorithmic approach to solve such large SDP relaxations. To this end we introduce the matrix

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{11}$$

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 [40] to further improve on the strength of the semidefinite relaxation. This yields the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \qquad & i < j < k \in [n],\ l < m \in [n] \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \qquad & i < j < k \in [n],\ l < m \in [n] \end{aligned} \tag{12}$$

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

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

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

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies (12) } \}, \tag{13}$$

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

$$\min \{\ \langle C, Y \rangle + K :\ Y \text{ satisfies (8)},\ Z \in (\mathcal{M} \cap \mathcal{LS}),\ \text{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \tag{$\texttt{SDP}_{\texttt{standard}}$}$$

To make $(\texttt{SDP}_{\texttt{standard}})$ computationally tractable Hungerländer and Rendl [36] suggest to deal with the triangle inequalities (10) and LS-cuts (12) through Lagrangian duality (for details see Subsection 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 [15, 17, 35]. For more details on global optimization approaches for the (SRFLP) we refer to the survey article by Anjos and Liers [10].

3 Formulations and Relaxations for the (kPROP)

Recently Amaral [7] extended his own approach for the (SRFLP) based on distance variables [2] 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 [7]), 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 [4] 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).

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

[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 [8] and the generalization of the approach to the (kPROP).

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 (14) and (15):

$$z_{ij}^y = \frac{1}{2}(\ell_i + \ell_j) + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ki}y_{kj}}{2} + \sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k \frac{1+y_{ik}y_{kj}}{2} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ik}y_{jk}}{2},\ r(i)=r(j), \tag{14a}$$

$$z_{ij}^y = y_{ij}\left(d_j - d_i\right), \qquad r(i)\neq r(j), \qquad d_i = \frac{\ell_i}{2} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k \frac{1+y_{ki}}{2} + \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k \frac{1-y_{ik}}{2}. \tag{14b}$$

To ensure non-negative distances for all feasible layouts we have to introduce the following, additional constraints:

$$z_{ij}^y \geq 0, \qquad i,j\in[n],\ i<j,\ r(i)\neq r(j). \tag{15}$$

In summary we are able to rewrite (2) with the help of ordering variables.

Theorem 1. *Minimizing* $\sum_{i,j\in[n],\ i<j} c_{ij} z_{ij}^y$ *over* $y\in\{-1,1\}^{\binom{n}{2}}$, (8), (14) *and* (15) *solves the* (kPROP).

Proof. The equations (8) model transitivity for $y\in\{-1,1\}^{\binom{n}{2}}$ [15] and hence suffice together with the integrality conditions on y and (15) to induce all feasible layouts. Thus by definition of the distances z_{ij}^y in (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:

$$\min\left\{\langle C_d, Z\rangle\ :\ y\in\{-1,1\}^{\binom{n}{2}},\ y \text{ satisfies (8) and (15)}\right\}, \tag{kPROP}$$

where the cost matrix C_d is deduced by equating the coefficients of the following equation

$$\begin{aligned} 2\langle C_d, Z\rangle \overset{!}{=} &\sum_{\substack{i,j\in[n],\ i<j\\ r(i)=r(j)}} c_{ij}\left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik}y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki}y_{kj}\right) \\ &+ \sum_{\substack{i<j\in[n],\\ r(i)\neq r(j)}} c_{ij}y_{ij}\left(L_{r(i)} - L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik}\right. \\ &\left. - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk}\right) + \sum_{h\in\mathcal{R}}\left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij}\right)\left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i\right)\right], \end{aligned}$$

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 2. *The problem*

$$\min\ \left\{\ \langle C_d, Z\rangle : Z \textit{ satisfies } (8) \textit{ and } (15)\ ,\ Z\in\mathcal{E},\ y\in\{-1,1\}^{\binom{n}{2}}\ \right\}$$

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 1, integrality on Y together with (8) and (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 4.

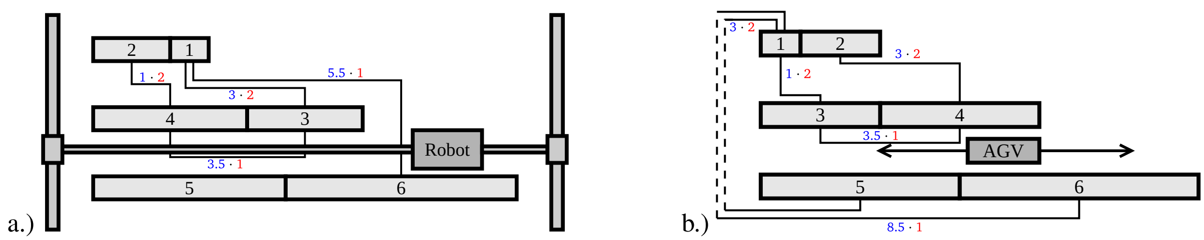

Figure 4: Illustration and comparison of the different distance calculation for departments in non-adjacent rows. We are given the following data: $ell_i = i,\ i \in \{1, \dots, 6\},\ c_{16} = c_{34} = 1,\ c_{13} = c_{15} = c_{24} = 2$. Departments 1 and 2 are assigned to row 1, departments 3 and 4 are 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:

$$
\begin{aligned}
2\langle C_i, Z\rangle \stackrel{!}{=} & \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)}} c_{ij} \left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} \ell_k y_{ik} y_{kj} - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} \ell_k y_{ki} y_{kj} \right) \\
& + \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|=1}} c_{ij} y_{ij} \left(L_{r(i)} - L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} \right. \\
& \left. + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right) + \sum_{\substack{i,j\in[n],\ i<j,\\ |r(i)-r(j)|>1}} c_{ij} \left(L_{r(i)} + L_{r(j)} + \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} \right. \\
& \left. + \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right) + \sum_{h\in\mathcal{R}} \left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij} \right) \left(\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=h}} \ell_i \right) \right].
\end{aligned}
$$

Note that we do not have to concern ourselves with vertical distances because the assignment of the departments to the rows is input data and hence the sum of the vertical 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 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)`.

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:

$$\min \{ \langle C_d, Z\rangle : \ Y \text{ satisfies (8) and (15)},\ Z \in (\mathcal{M} \cap \mathcal{LS}),\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0 \}. \qquad (\mathtt{SDP}_{\mathtt{standard}})$$

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:

$$z^y_{ij} + z^y_{ik} \geq z^y_{jk}, \qquad z^y_{ij} + z^y_{ik} \geq z^y_{jk}, \qquad z^y_{ik} + z^y_{jk} \geq z^y_{ij}, \qquad i,j,k \in [n],\ i<j<k. \qquad (16)$$

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

$$\mathcal{DV} := \{ Z : Z \text{ satisfies (16)} \} \qquad (17)$$

containing the $3\binom{n}{3}$ triangle inequalities relating the distances between 3 or more departments. Adding these constraints to $(\mathtt{SDP}_{\mathtt{standard}})$ yields

$$\min \{ \langle C_d, Z\rangle : \ Y \text{ satisfies (8) and (15)},\ Z \in (\mathcal{M} \cap \mathcal{LS} \cap \mathcal{DV}),\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0 \}. \qquad (\mathtt{SDP}_{\mathtt{full}})$$

It was demonstrated in [35] 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 5 in detail. Instead we will work with the following SDP relaxation

$$\min \{ \langle C_d, Z \rangle : \ Y \text{ satisfies (8) and (15)}, \ Z \in \mathcal{DV}, \ \text{diag}(Z) = e, \ Z \succcurlyeq 0 \}. \qquad (\texttt{SDP}_{\texttt{cheap}})$$

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 [24] 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.

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 [24], which was applied to the max cut problem [46] and several ordering problems [16, 35], and adapt it for the (kPROP). Initially, we only aim at explicitly ensuring the constraints $\text{diag}(Z) = e$ and $Z \succcurlyeq 0$, which can be achieved with standard interior point methods, see, e.g. [30].

All other constraints are handled through Lagrangian duality in the objective function f. Thus the objective function f becomes non-smooth. The bundle method [24] 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 $\text{diag}(Z) = e$ 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 ($\texttt{SDP}_{\texttt{cheap}}$).

4.2 Obtaining Feasible Layouts

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans-Williamson [28] 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 (15) and the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1. \qquad (18)$$

that are well-known [54, 57] 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 [17].

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 (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. [46]).

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 (15) and (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 (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 (SDP$_{\texttt{cheap}}$) with the help of the bundle method and then retry the heuristic (retaining the last vector y').

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 [7] 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 5.1 and then we explain our algorithmic strategies for the (PROP) in detail in Subsection 5.2. Finally we expand our computational results to the case of 3 and more rows in Subsection 5.3.

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 1. All results obtained are provided by the SDP approach [36], which is the strongest to date, applied to (SDP$_{\texttt{standard}}$) on our machine. Its main features are summarized in Subsection 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.

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 (SDP$_{\texttt{cheap}}$) to obtain an initial set of constraints to add to the relaxation (SDP$_{\texttt{basic}}$). We then solved the resulting relaxation using Sedumi [51]; added all violated inequality constraints from (SDP$_{\texttt{cheap}}$); solved again using Sedumi; and repeated this process until no more violations were found. We also tried to solve (SDP$_{\texttt{cheap}}$) directly but the running times were at least one order of magnitude slower. Additionally we tried to solve (SDP$_{\texttt{full}}$) instead of (SDP$_{\texttt{cheap}}$) 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 (SDP$_{\texttt{full}}$), we have to consider 2745 constraints and the lower bound is 3600.5. For comparison we need only 1 second to compute (SDP$_{\texttt{cheap}}$) as we

Instance	Source	Size (n)	SRFLP			
			Best lower bound	Best layout	Gap (%)	Time (sec) [36]
S11	[49]	11	6933.5		0	1
P15	[2]	15	6305		0	20
P16_a	[7]	16	14829		0	34
P16_b	[7]	16	11878.5		0	34
P17	[3]	17	9254		0	35
P18	[3]	18	10650.5		0	33
H_20	[33]	20	15549		0	54
P20_a	[7]	20	24180.5		0	1:03
P20_b	[7]	20	25218		0	1:07
P21_a	[7]	21	13432.5		0	1:13
P21_b	[7]	21	22964		0	1:21
P22_a	[7]	22	17064		0	1:00
P22_b	[7]	22	29770		0	1:07
P23_a	[7]	23	19589		0	1:14
P23_b	[7]	23	30257.5		0	1:06
N25_05	[11]	25	15623		0	3:31
H_30	[33]	30	44965		0	14:17
N30_05	[11]	30	115268		0	18:30
Am33_03	[4]	33	69942.5		0	36:33
Am35_03	[4]	35	69002.5		0	53:14
ste36.5	[12]	36	91651.5		0	17:58
N40_5	[36]	40	103009		0	2:20:09
sko42-5	[12]	42	248238.5		0	1:08:42
sko49-5	[12]	49	666130	666143	0.002	9:30:22
sko56-5	[12]	56	591915.5	592335.5	0.07	17:46:46
AKV-60-05	[13]	60	318792	318805	0.004	12:39:37
sko64-5	[12]	64	501059.5	502063.5	0.20	13:53:04
AKV-70-05	[13]	70	4213774.5	4218002.5	0.10	28:16:05
sko72-5	[12]	72	426224.5	430288.5	0.95	31:39:43
AKV-75-05	[13]	75	1786154	1791469	0.30	41:10:37
AKV-80-05	[13]	80	1585491	1590847	0.34	58:30:30
sko81-5	[12]	81	1293905	1311166	1.33	58:59:28
sko100-5	[12]	100	1021584.5	1040929.5	1.89	201:29:27

Table 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.

have to consider only 113 constraints and the lower bound obtained is 3563.5. The results obtained are summarized in Table 2, where the optimal solutions are provided by Amaral [7].

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 [7] 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 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

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

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 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. The optimal solutions are provided by Amaral [7].

row. Such balanced row assignments are often of interest in the design of layouts in practice, see e.g. [38]. Our heuristic quickly yields assignments for which the total row lengths are very close; see the second-to-last column of Table 4. We summarize the results averaged over 10 row assignments selected by our heuristic in Table 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 4.2. Additionally we state the average number of inequalities of ($\mathtt{SDP}_{\mathtt{cheap}}$) 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 3: (PROP) results for ($\mathtt{SDP}_{\mathtt{cheap}}$) 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 ($\mathtt{SDP}_{\mathtt{cheap}}$) 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 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 $(\texttt{SDP}_{\texttt{cheap}})$. Also in this case the large number of violated LS-cuts has only little effect on the bound quality but slows down the bundle computation significantly which altogether leads to a computationally better performance of our approach applied to $(\texttt{SDP}_{\texttt{cheap}})$ compared to $(\texttt{SDP}_{\texttt{full}})$. 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 4 below, as well as in Table 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 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 4: (PROP) results for $(\texttt{SDP}_{\texttt{cheap}})$ 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 3 and 4 shows that the lower bounds of the bundle method quickly get close to the exact $(\texttt{SDP}_{\texttt{cheap}})$ bounds even though the number of function evaluations is capped at 125. Furthermore, while the running times in Table 3 grow very quickly with the problem size, the computation times of the bundle method in Table 4 are not so strongly affected by the problem size. Hence this approach yields bounds competitive with the exact optimal value of $(\texttt{SDP}_{\texttt{cheap}})$ at only a fraction of the computational cost. For comparison the (SRFLP) results from Table 1 were obtained by the same algorithmic approach but using a different SDP relaxation ($(\texttt{SDP}_{\texttt{standard}})$ instead of $(\texttt{SDP}_{\texttt{cheap}})$) 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.

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 [7]. We use again the bundle method in conjunction with Sedumi applied to ($\mathrm{SDP}_{\mathrm{cheap}}$) because this approach clearly outperforms the other two variants discussed above (solving ($\mathrm{SDP}_{\mathrm{cheap}}$) directly or using ($\mathrm{SDP}_{\mathrm{full}}$)) 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 3.1, for an illustration see Figure 4. The results obtained are summarized in Table 5 for 3 rows and in Table 6 for 4 and 5 rows. The upper bounds are provided by the heuristic described in Subsection 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 5: (kPROP) results with $k = 3$ for ($\mathrm{SDP}_{\mathrm{cheap}}$) 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 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 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 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 7. The upper bounds are provided by the heuristic described in Subsection 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 $\langle C_d, Z \rangle$.
- 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.

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 [34].

References

[1] H. Ahonen, A. G. de Alvarenga, and A. R. S. Amaral. Simulated annealing and tabu search approaches for the corridor allocation problem. *European Journal of Operational Research*, 232(1):221–233, 2014.

[2] A. R. S. Amaral. On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508–518, 2006.

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 6: (kPROP) results with $k = 4$ and $k = 5$ for ($\mathrm{SDP}_{\mathrm{cheap}}$) 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.

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 difference of row lengths	Average time
P17	3083.5	3417	8.75	19.99	14.24	1.3	30	4491.5	4715	2.62	6.73	4.31	1.3	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	1.7	32
H_20	4950	5718	10.48	18.04	13.74	1.5	55	7150.5	7731	1.77	8.13	4.48	1.5	51
N25_05	4737	5507	11.11	17.69	15.07	0.7	2:24	7282.5	7912	1.08	8.64	4.07	0.7	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	1.5	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	2.2	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	1.5	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	1.0	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	1.5	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	0.7	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	0.8	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	2.4	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	0.7	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	1.1	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	0.7	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	1.4	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 difference of row lengths	Average time
P17	1883	2125	9.61	16.73	13.29	1.9	32	4113	4142	0.09	1.15	0.62	1.9	23
P18	2095	2465.5	8.52	19.45	13.76	1.9	40	4553	4562.5	0.11	1.50	0.64	1.9	29
H_20	3190	3641	12.64	20.45	16.00	1.4	1:01	6648	6668	0.21	1.74	0.80	1.4	45
N25_05	2885	3301	12.19	21.90	17.16	0.9	2:40	6700	6760	0.13	1.02	0.63	0.9	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	1.3	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	2.2	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	1.5	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	1.1	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	1.4	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	0.6	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	1.2	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	2.7	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	1.0	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	1.2	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	0.8	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	1.6	14:56:07

Table 7: (kPROP) results with $k = 3$ and $k = 5$ for (SDP$_{\texttt{cheap}}$) 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.

[3] A. R. S. Amaral. An exact approach to the one-dimensional facility layout problem. *Operations Research*, 56 (4):1026–1033, 2008.

[4] A. R. S. Amaral. A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157(1):183–190, 2009.

[5] A. R. S. Amaral. Optimal solutions for the double row layout problem. *Optimization Letters*, pages 1–7, 2011.

[6] A. R. S. Amaral. The corridor allocation problem. *Computers & Operations Research*, 39(12):3325–3330, 2012.

[7] A. R. S. Amaral. A parallel ordering problem in facilities layout. *Computers & Operations Research*, 40(12): 2930–2939, 2013.

[8] A. R. S. Amaral and A. N. Letchford. A polyhedral approach to the single row facility layout problem. *Mathematical Programming*, pages 1–25, 2012.

[9] 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, 2011.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] C. Buchheim, A. Wiegele, and L. Zheng. Exact Algorithms for the Quadratic Linear Ordering Problem. *INFORMS Journal on Computing*, 22(1):168–177, 2010.

[16] M. Chimani and P. Hungerländer. Exact approaches to multilevel vertical orderings. *INFORMS Journal on Computing*, 25(4):611–624, 2013.

[17] 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.

[18] J. Chung and J. Tanchoco. The double row layout problem. *International Journal of Production Research*, 48 (3):709–727, 2010.

[19] 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.

[20] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer Verlag, Berlin, 1997.

[21] J. Dickey and J. Hopkins. Campus building arrangement using topaz. *Transportation Research*, 6(1):59–68, 1972.

[22] A. N. Elshafei. Hospital layout as a quadratic assignment problem. *Operational Research Quarterly*, 28(1): 167–179, 1977.

[23] 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.

[24] 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.

[25] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, 1(3):237 – 267, 1976.

[26] M. Gen, K. Ida, and C. Cheng. Multirow machine layout problem in fuzzy environment using genetic algo-

rithms. *Computers & Industrial Engineering*, 29(1–4):519–523, 1995.
[27] 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.
[28] 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.
[29] M. M. D. Hassan. Machine layout problem in modern manufacturing facilities. *International Journal of Production Research*, 32(11):2559–2584, 1994.
[30] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.
[31] S. S. Heragu and A. Kusiak. The facility layout problem. *European Journal of Operational Research*, 29(3): 229–251, 1987.
[32] S. S. Heragu and A. Kusiak. Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268, 1988.
[33] S. S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1):1–13, 1991.
[34] P. Hungerländer and M. Anjos. A semidefinite optimization approach to space-free multi-row facility layout. Cahier du GERAD G-2012-03, GERAD, Montreal, QC, Canada, 2012.
[35] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming*, 140(1):77–97, 2013.
[36] 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.
[37] 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.
[38] A. Langevin, B. Montreuil, and D. Riopel. Spine layout design. *International Journal of Production Research*, 32(2):429–442, 1994.
[39] G. Laporte and H. Mercure. Balancing hydraulic turbine runners: A quadratic assignment problem. *European Journal of Operational Research*, 35(3):378–381, 1988.
[40] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.
[41] R. F. Love and J. Y. Wong. On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14:139–143, 1967.
[42] 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.
[43] A. C. Nearchou. Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, 101(2):312–328, 2006.
[44] J.-C. Picard and M. Queyranne. On the one-dimensional space allocation problem. *Operations Research*, 29 (2):371–391, 1981.
[45] M. Pollatschek, N. Gershoni, and Y. Radday. Optimization of the typewriter keyboard by computer simulation. *Angewandte Informatik*, 10:438–439, 1976.
[46] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.
[47] 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.
[48] S. Sanjeevi and K. Kianfar. A polyhedral study of triplet formulation for single row facility layout problem. *Discrete Applied Mathematics*, 158:1861–1867, 2010.
[49] D. M. Simmons. One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17: 812–826, 1969.
[50] L. Steinberg. The backboard wiring problem: A placement algorithm. *SIAM Review*, 3(1):37–50, 1961.
[51] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization*

Methods and Software, 11–12:625–653, 1999.

[52] J. Suryanarayanan, B. Golden, and Q. Wang. A new heuristic for the linear placement problem. *Computers & Operations Research*, 18(3):255–262, 1991.

[53] J. A. Tompkins. Modularity and flexibility: dealing with future shock in facilities design. *Industrial Engineering*, pages 78–81, 1980.

[54] A. W. Tucker. On directed graphs and integer programs. Technical report, IBM Mathematical Research Project, 1960.

[55] 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.

[56] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.

[57] D. H. Younger. Minimum feedback arc sets for a directed graph. *IEEE Transactions on Circuit Theory*, 10(2): 238–245, 1963.

[58] Z. Zhang and C. Murray. A corrected formulation for the double row layout problem. *International Journal of Production Research*, 2011.