

# Exact and Heuristic Approaches for a New Circular Layout Problem\*

Philipp Hungerländer<sup>†</sup> Kerstin Maier<sup>‡</sup> Veronika Pachatz<sup>§</sup> Christian Truden<sup>¶</sup>

April 4, 2020

## Abstract

We discuss a new facility layout problem, the so-called Directed Circular Facility Layout Problem (DCFLP). The DCFLP aims to find an optimal arrangement of machines on a circular material handling system such that the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction is minimized. Several real-world applications, like for example the optimal arrangement of a set of cutting tools on a tool turret, can be modeled as a DCFLP. Further, the DCFLP generalizes a couple of layout problems that are well-discussed in literature. We show that the DCFLP can be modeled as a Linear Ordering Problem (LOP). Hence, it can be solved efficiently by using exact and heuristic approaches for the LOP. First, we apply a Semidefinite Programming as well as an Integer Linear Programming approach. Moreover, we use a Tabu Search and a Variable Neighborhood Search heuristic, for solving the DCFLP. Finally, we compare the practical performance of our approaches in a computational study.

**Keywords:** Facility Planning and Design Circular Layout Linear Ordering Problem Exact Approaches Heuristics

## 1 Introduction

Facility layout problems (FLPs) aim to find the optimal location of machines inside a production plant with respect to a given objective function that considers for example material-handling, transportation or construction costs, or simply pair-wise preferences among machines. FLPs are well-discussed operations research problems and occur in a great variety of practical applications. Besides the prime example of arranging machines in Flexible Manufacturing Systems (FMSs) (see Subsection 1.1 for more details), it is considered for the alignment of departments within office buildings, rooms in hospitals, etc., see [48]. Meller and Gau [62] divide FLPs into three categories:

---

\*This paper is based on two extended abstracts published in conference proceedings of the IFAC Conference on Manufacturing Modelling, Management, and Control [39] and the IEEE International Conference on Industrial Engineering and Engineering Management [43], respectively.

<sup>†</sup>Laboratory for Information & Decision Systems, MIT, USA, philipp.hungerlaender@aau.at

<sup>‡</sup>MANSIO Karl Popper Kolleg, Alpen-Adria-Universität Klagenfurt, Austria, kerstin.maier@aau.at

<sup>§</sup>Hex GmbH, Austria, veronika.pachatz@hex-solutions.com

<sup>¶</sup>Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Austria, christian.truden@aau.at

- The first category handles different versions of the basic layout problem that asks for an optimal arrangement of a given number of machines within a facility such that the total expected cost of flows inside the facility is minimized. This includes the well-known Quadratic Assignment Problem (QAP) where all machine sizes are equal.
- The second category deals with extensions of unequal-areas layouts that consider several real-world issues, such as time-dependency or uncertain conditions and take into account two or more objectives simultaneously.
- In this paper, we consider one of the layout problems in the third category, namely the Directed Circular Facility Layout Problem (DCFLP). This category is concerned with problem instances that follow a special structure, such as the arrangement of machines along a production line.

## 1.1 Flexible Manufacturing Systems

FMSs are automated production systems, typically consisting of material handling devices under computer control, robots, and Computer Numerically Controlled (CNC) machines that are designed to process a collection of parts. Arranging machines within FMSs is an essential problem [56], as the layout of the machines has enormous impact on material handling costs and time, on throughput, and on overall productivity of the whole system. Poor layouts negatively influence the flexibilities of FMS [34]. Machine layouts are in general defined through the required type of material-handling device, such as conveyor systems, Automated Guided Vehicles (AGVs), or robots [67]. The following layout types are the most common ones in practical applications:

- Single-Row Layouts (see Fig. 1),
- Double-Row and Multi-Row Layouts (see Fig. 2) and
- Circular Layouts (see Fig. 3).

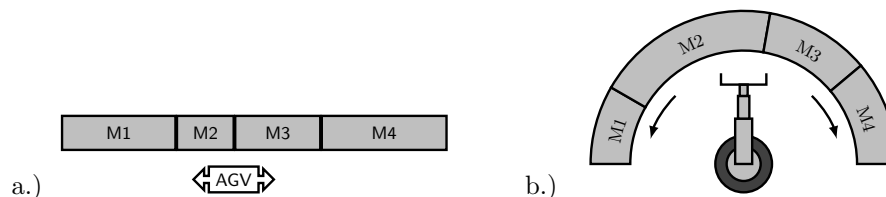


Figure 1: In a.) an AGV transports parts between the machines moving in both directions in a straight line. In b.) a material-handling industrial robot carries parts between the machines.

## 1.2 Directed Circular Facility Layout Problem

Our proposed layout problem, the DCFLP, aims to find an optimal arrangement of machines on a circular material handling system such that the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction is minimized. It is assumed that the

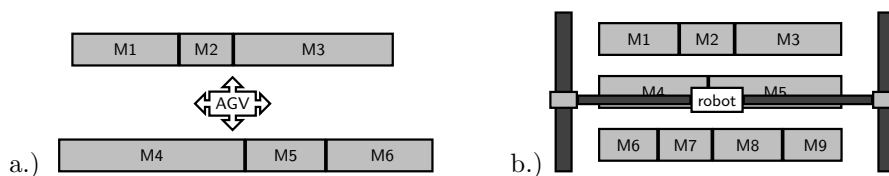


Figure 2: In a.) again an AGV and in b.) a gantry robot is used to move parts between the machines.

material handling system transports the parts unidirectionally around the circuit following the predefined sequence in its process plan. Further, we assume that each machine is capable of picking up and processing the parts from the material handling system [60] and that all parts enter and leave the material handling system through the Loading and Unloading Station (LUS).

Circular material handling systems are in general preferred in practice, due to relative low initial investment costs, space-saving design, high material handling flexibility and the ability of being easily adapted to future introduction of new parts and process changes [1, 50].

A DCFLP instance 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  $\pi$  of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines measured in clockwise direction is minimized, i.e.,

$$\min_{\pi \in \Pi_n} \sum_{i, j \in [n], i \neq j} f_{ij} z_{ij}^{\pi}, \quad (1)$$

where  $\Pi_n$  is the set of all feasible layouts of the machines  $[n] := \{1, 2, \dots, n\}$  and  $z_{ij}^{\pi}$  gives the distance between the centroids of machines  $i$  and  $j$  in the circular layout  $\pi$  (in clockwise direction).

Considering CNC machines, Kiran and Karabati [46] describe the problem of choosing the assignment of a set of cutting tools on a tool turret, where each tool must be located in one of the tool holders. When the CNC machine requires a different tool, the turret rotates unidirectional in order to bring the required cutter into the work envelope of the tool changer. Hence, the problem of choosing the optimal arrangement of the cutters on the tool turret such that the turret travel times are minimized can be modeled as a DCFLP. Thus, this application forms a well-known example for using the DCFLP.

Clearly, the DCFLP is not universally applicable to all facility layout problems. Using machines that strongly differ in length, one can easily encounter cases in which it is impossible to form a closed circle or the machines must be placed at odd angles that can not be realized by the selected material handling system. If this is the case, it is advisable to utilize a different layout that better fits the circumstances, e.g., single-row or multi-row layout.

### 1.3 Modeling the DCFLP as Linear Ordering Problem

In this work, we show that the DCFLP can be modeled as a Linear Ordering Problem (LOP). Considering the matrix version of the LOP we can define it as follows. Given an  $n \times n$  matrix  $W = (w_{ij})$

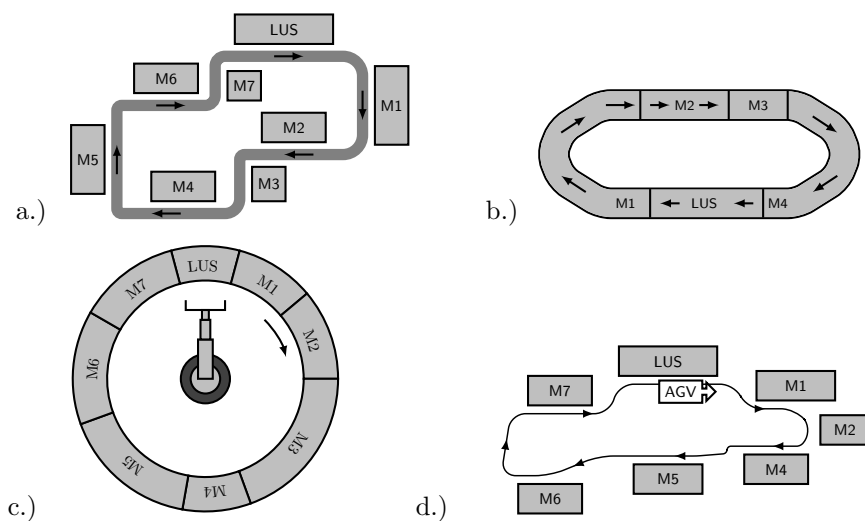


Figure 3: In a.) and b.) a conveyor system moves in a closed-loop rail in one direction transporting parts among the machines. In c.) a material-handling industrial robot rotates unidirectionally and in d.) single loop AGVs transport parts between the machines.

of integers, find a simultaneous permutation  $\pi$  of columns and rows of  $W$  such that

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

is maximized. Moreover,  $w_{ij}$  can be interpreted as weights of a complete directed graph  $G$  with vertex set  $V = [n]$ . A tournament consists of a subset of arcs of  $G$ , which contains for every pair of nodes  $i$  and  $j$  exactly one of the two arcs  $(i, j)$  and  $(j, i)$ . 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 a well-known problem for which a great variety of high-quality exact and heuristic methods exist, see [61]. Hence, the DCFLP can be solved very efficiently with the help of these approaches. In this work, we apply two exact approaches, a Semidefinite Program (SDP) and an Integer Linear Program (ILP), as well as two fast heuristics, namely a Tabu Search (TS) and a Variable Neighborhood Search (VNS), for tackling the DCFLP.

We structure the remainder of this paper as follows. In Section 2, we give an overview on previous and related work. In Section 3, we formally describe the DCFLP and show that it can be modeled as an LOP. In Sections 4 and 5, we give a brief overview of best available exact and heuristic approaches for the LOP which are also easy to implement. In Section 6, the results of our computational experiments are summarized. Finally, in Section 7, we conclude the paper and point out several future research directions.

## 2 Related Work

In this section, we describe previous work on the DCFLP and give an overview of related FLPs as well as special cases of the DCFLP. For an overview of various formulations of the FLP, solution approaches, and available software packages we refer to the excellent review papers by Kusiak and Heragu [53], Meller and Gau [62], Singh and Sharma [71]. Moreover, Kulturel-Konak [52] discusses recent literature concerning the FLP under uncertainty.

### 2.1 Previous Work on the DCFLP

The DCFLP is introduced in [39], where an SDP approach is proposed for solving it. In [43], it is first stated that the DCFLP can be modeled as an LOP. Further, an ILP formulation as well as a TS heuristic are suggested for solving the DCFLP. Note that both papers have been published in conference proceedings and hence, the level of detail of the problem description as well as of the solution approaches is limited. Moreover, the computational experiments are reduced to a minimum due to space restrictions.

In this work, we repeat the key findings of our previous published conference proceedings. In addition, we provide a comprehensive literature review, discuss our previously considered solution approaches in more detail, fine-tune the TS, and additionally implement a VNS for solving the DCFLP. Furthermore, a large benchmark library is created, which allows for a detailed performance evaluation of our exact and heuristic approaches. Hence, we now provide an extensive computational study, which is a substantial extension compared to already published material.

To the best of our knowledge, there are currently no other publications focusing on the DCFLP in the literature, except our conference papers, [39, 43].

### 2.2 Single-Row Facility Layout Problem

The well-known Single-Row Facility Layout Problem, see [48] for review, is closely related to the DCFLP. It arises as the problem of ordering machines on a production line where the material flow is handled by an AGV traveling in both directions on a straight-line path [37]. An SRFLP instance 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 formulated as

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

where  $\Pi_n$  is the set of permutations of the machines  $[n]$  and  $\zeta_{ij}^\pi$  is the center-to-center distance between machines  $i$  and  $j$  with respect to a particular permutation  $\pi \in \Pi_n$ .

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. Currently, it is even considered as the FLP with the easiest structure and most straight-forward formulations. The global optimization approaches for the SRFLP are based on relaxations of ILP and SDP formulations. The strongest ones are an LP-based cutting plane algorithm using betweenness variables [5] and an SDP approach using products of ordering variables [42]. In this paper, we suggest a formulation for the DCFLP that is significantly easier (linear terms instead of linear-quadratic terms in ordering variables) than all available formulations for the SRFLP.

Let us further clarify the connections and differences of the SRFLP and the DCFLP 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  $c_{12} = c_{14} = c_{34} = 1$ ,  $c_{13} = c_{24} = 2$  for the SRFLP and pairwise flows  $f_{12} = f_{14} = f_{43} = 1$ ,  $f_{13} = f_{42} = 2$  for the DCFLP. Figure 4 illustrates the optimal layouts and the according costs for both problems.

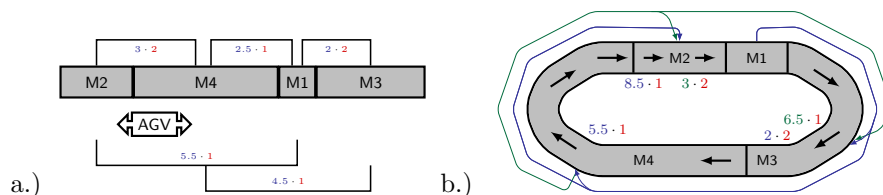


Figure 4: We are given the following data:  $\ell_1 = 1$ ,  $\ell_2 = 2$ ,  $\ell_3 = 3$ ,  $\ell_4 = 4$ ,  $c_{12} = c_{14} = c_{34} = 1$ ,  $c_{13} = c_{24} = 2$ ,  $f_{12} = f_{14} = f_{43} = 1$ ,  $f_{13} = f_{42} = 2$ . In a.) we illustrate the optimal layout for the SRFLP with corresponding costs of  $3 \cdot 2 + 2.5 \cdot 1 + 2 \cdot 2 + 5.5 \cdot 1 + 4.5 \cdot 1 = 22.5$ . In b.) we display the optimal layout for the DCFLP with corresponding costs of  $2 \cdot 2 + 3 \cdot 2 + 5.5 \cdot 1 + 8.5 \cdot 1 + 6.5 \cdot 1 = 30.5$ .

### 2.3 Special Cases of the DCFLP

Additional to its practical relevance for the design of FMSs [56], the DCFLP is a very interesting problem from an academic point of view as it generalizes several other layout problems that are well-studied in literature. The DCFLP is a generalization of the Directed Circular Arrangement Problem (DCAP) that allows the machines to have arbitrary lengths instead of the same lengths. The DCAP was first considered by Liberatore [57] who showed that the problem is NP-hard (hence, also the DCFLP is NP-hard). We refer to Liberatore [57], Bar-Noy et al. [12] and Naor and Schwartz [66] for a great variety of applications in the field of ring networks and server design that can be modeled as DCAP.

Further, the DCFLP is connected to the NP-hard [51] Unidirectional Cyclic Layout Problem (UCFLP) [2]. The UCFLP also considers a circular material handling system with directed flow and the aim is to find an assignment of  $n$  machines to  $n$  predetermined candidate locations such that the total handling costs are minimized. There are two well-known special cases of the UCFLP and hence, also for the DCFLP:

1. In the Balanced Unidirectional Cyclic Facility Layout Problem (BUCFLP) the material flow is conserved at each machine, i.e., total inflow is equal to total outflow at each machine.
2. The Equidistant Unidirectional Cyclic Facility Layout Problem (EUCFLP) considers machine locations which are equally distanced to each other around the unidirectional circular material handling system.

Bozer and Rim [14] have shown that BUCFLP and EUCFLP are equivalent.

While the UCFLP considers distances of the locations, the DCFLP deals with machine lengths. Therefore, the DCFLP can be seen as an adaption of the SRFLP to circular layouts. Considering machine lengths instead of the location distances (i.e., location lengths) is clearly preferable in

many practical applications where the lengths of the machines are the relevant input parameters. Furthermore, it is very hard to solve the UCFLP, because it is a special QAP and QAPs are known to be particularly difficult to solve [58]. Hence, optimizing circular layouts was so far considered to be clearly harder to solve than row layouts. In this work, we show that this is not true, if we determine circular layouts with the help of the DCFLP (formulated as an LOP) instead of the UCFLP (formulated as a QAP).

## 2.4 Further Circular Layout Problems

Another related problem is the (Unidirectional) Loop Design Layout Problem (LDLP), that was first defined by Afentakis [1], which is concerned with minimizing the number of parts traversing a defined point on a conveyor belt. It differs from the DCFLP such that each machine is assigned to one of the predetermined sites along the loop. Moreover, gaps between the machines are allowed as they are connected by a conveyor belt or an AGV and the actual distances between the machines are not considered. Kiran and Karabati [46] derive necessary optimality conditions for the LDLP and propose a Branch-and-Bound algorithm as well as an approximate solution method. Further, they identify the special case where all parts come from and go to the LUS after each operation as being solvable in  $\mathcal{O}(n^2 \log n)$  time. Further characterizations of optimal layouts for the LDLP are given in [55].

Focusing on the balanced case, i.e., the total inflow to each station equals the total outflow from that station, Kiran et al. [47] show that the problem can be reduced to finding a sequence of stations that is independent of the actual locations of the machines.

Although in general, unidirectional systems are preferred due to their operational simplicity, also bidirectional systems have been discussed in the literature. The Bidirectional Circular Layout Problem [15] describes the problem where facilities are arranged around a simple closed loop and flows between facilities occur in clockwise or counterclockwise direction (whichever is shorter) on the loop.

## 2.5 Contribution

The main contributions of this work can be summarized as follows:

1. We show that the DCFLP can be modeled as an LOP.
2. Building on this model we validate that both exact and heuristic methods for the DCFLP are less complex in ordering variables and hence, easier and more efficient to implement than the best current approaches for the SRFLP and the UCFLP.
3. We create an extensive benchmark library for future research in this field and give strong lower and upper bounds for all instances considered.
4. Our computational study provides a detailed performance analysis of our exact and heuristic approaches.

## 3 Mathematical Formulation

For modeling the DCFLP as an LOP we introduce  $\mathcal{O}(n^2)$  binary ordering variables

$$x_{ij} \in \{0, 1\}, i, j \in [n], i < j, \tag{2}$$

with the following interpretations:

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

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

$$0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1, \quad i, j, k \in [n], \quad i < j < k. \quad (3)$$

It is well-known that the 3-cycle inequalities together with integrality conditions on the variables suffice to describe feasible orderings, see e.g., [73].

Next, we introduce the distance variables  $d_{ij}$ ,  $i, j \in [n]$ ,  $i < j$ , which give the distance between machines  $i$  and  $j$ , if all machines are arranged on a straight line. Hence, we can compute  $d_{ij}$  as the difference of the sums of the lengths of the machines arranged before machine  $i$  and machine  $j$ , respectively:

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

Note that  $d_{ij}$  is negative if machine  $i$  is arranged before machine  $j$  and positive, otherwise. Therefore,  $|d_{ij}|$ ,  $i, j \in [n]$ ,  $i < j$ , yields the correct distance between machines  $i$  and  $j$  on a straight line. The  $d_{ij}$  are linear expressions in the ordering variables  $x_{ij}$ . To destroy symmetry and reduce the dimensions of the problem, we fix machine 1 to be first in the ordering. Hence, we set  $x_{1j} = 1$ ,  $j \in [n]$ ,  $j \neq 1$ . Clearly, the solution of the DCFLP is independent of the selection of the first (fixed) machine as the circular arrangement is invariant under rotation. This results (after some additional simplifications) in the following adaption of the distance variables

$$d_{1j} = \frac{\ell_1 - \ell_j}{2} - \sum_{\substack{k \in [n] \\ 1 < k < j}} \ell_k x_{kj} - \sum_{\substack{k \in [n] \\ k > j}} \ell_k (1 - x_{jk}), \quad j \in [n], \quad j \neq 1,$$

$$d_{ij} = \left( \frac{\ell_i}{2} + \sum_{\substack{k \in [n] \\ 1 < k < i}} \ell_k x_{ki} + \sum_{\substack{k \in [n] \\ k > i}} \ell_k (1 - x_{ik}) \right) - \left( \frac{\ell_j}{2} + \sum_{\substack{k \in [n] \\ 1 < k < j}} \ell_k x_{kj} + \sum_{\substack{k \in [n] \\ k > j}} \ell_k (1 - x_{jk}) \right), \quad i, j \in [n], \quad 1 < i < j. \quad (4)$$

Next, we determine 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

$$\begin{aligned} z_{1j} &= -d_{1j}, & z_{j1} &= L + d_{1j}, & j \in [n], j \neq 1, \\ z_{ij} &= -d_{ij} + (1 - x_{ij})L, & z_{ji} &= d_{ij} + x_{ij}L, & i, j \in [n], 1 < i < j, \end{aligned} \quad (5)$$

where  $L = \sum_{k \in [n]} \ell_k$  denotes the sum of the lengths of all machines. Now we can rewrite the objective function (1) with the help of (5) as a linear function in  $dim := \binom{n-1}{2}$  ordering variables:

$$\min_{x \in \{0,1\}^{dim}} f(x) \quad (6)$$

where

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

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 1.** *The LOP defined by (6) subject to (2), (3) and (4) is equivalent to the DCFLP.*

*Proof.* The inequalities (3) together with the integrality conditions on  $x$  suffice to induce a feasible layout on the circle. The equations (4) connect the ordering with the distance variables and finally, the definition of the objective function ensures that the distances between machines are computed correctly and weighted with the appropriate flows.

Using (4) and (6) we can calculate weights  $w_{ij}$ ,  $i, j \in [n]$ ,  $1 < i < j$ , for each pair of machines and rewrite the problem to the following standard LOP formulation:

$$\begin{aligned} \max \quad & \sum_{i,j \in [n], 1 < i < j} w_{ij} x_{ij} \\ \text{subject to} \quad & (2) \text{ and } (3). \end{aligned} \quad (7)$$

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

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

Now the betweenness variables can be further rewritten as linear-quadratic expressions of ordering variables

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

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

## 4 Computing Lower Bounds

Currently available exact algorithms for the LOP are for example a Branch-and-Bound algorithm by Kaas [44], a Branch-and-Cut algorithm developed by Grötschel et al. [33] or a combined interior-point cutting-plane algorithm by Mitchell and Borchers [64]. The working group of Reinelt in Heidelberg proposed the current state-of-the-art Branch-and-Cut algorithm that is based on sophisticated cut generation procedures (see [61] for details). The algorithm is able to solve specific instances with up to 150 objects, but fails on other instances with only 50 objects.

We apply two exact approaches in order to obtain strong lower bounds for the LOP. First, we use a common ILP formulation for solving the LOP, which is stated in (7) in Section 3. Our chosen ILP formulation is easy to implement and thus in particular interesting for practitioners. The minor losses of solution quality, while requiring slightly more run time, compared to the strongest ILP approaches mentioned above are neglectable for our study.

Additionally, we propose an exact algorithm based on SDP that has proven 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 [41]). Further, it is the method of choice for the SRFLP [42] (and other ordering problems [21, 22]). The core of our approach is to solve SDP relaxations of the DCFLP, using the bundle method [26] in combination with interior point methods [36]. The corresponding fractional solutions represent lower bounds for the DCFLP. With the help of a rounding strategy, such fractional solutions can be exploited to obtain upper bounds, i.e., integer feasible solutions that define feasible layouts of the machines on the circle. Thus, in the end we obtain a feasible layout together with a guaranty (achieved by computing the lower bound) of how far this solution is at most away from the true optimum. We refer to Appendix A where we provide further details on how the lower bounds are computed. Further, we describe the applied rounding algorithm for obtaining feasible layouts in Appendix B.

## 5 Computing Feasible Layouts

There exist a great variety of well-performing heuristics for the LOP. The first heuristic for the LOP is a construction heuristic and was introduced by Chenery and Watanabe [20]. Further construction methods are due to Becker and Aujac [11]. 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. However, they are even more important as a powerful concept for the design of metaheuristics. The most important improvement methods for the LOP are local search improvements based on exchanges of objects, Kernighan-Lin improvements [45], and the heuristic of Chanas and Kobylanski [19].

The primary mechanism to move from one solution to another in all relevant heuristics for the LOP and the SRFLP are insert moves. However, the execution costs differ significantly between these two problems. An insertion move for the LOP needs  $O(n)$  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 requires  $O(n^2)$  time because in general all entries of the cost matrix  $C$  have to be queried. Due to this, it can be assumed that heuristics for the DCFLP receive in principle stronger solutions with less computational time than heuristics for the SRFLP. This claim is supported by the comprehensive computational experiments conducted in the literature: While for the LOP high quality solutions for different types of instances with up to 500 objects are produced within seconds

by several heuristics [61] (see also below), the best heuristics for the SRFLP require up to a hundreds of seconds on instances with only 80 objects (on faster computers) to generate solutions of about the same solution quality (see e.g., [23]).

Metaheuristics [31] 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 on an extensive benchmark library with instances consisting of up to 500 objects in [61] and the best ones are the following (in descending order of their rank value obtained by a non-parametric Friedman test):

- Genetic and Memetic Algorithms [27, 63, 69],
- Tabu Search [29, 54],
- Variable Neighborhood Search [28, 65],
- Scatter Search [17, 30] and the
- Greedy Randomized Adaptive Search Procedure [17, 25].

To the best of our knowledge, there are no black-boxes or implementations of metaheuristics for the LOP available online. As the differences with respect to solution quality and runtime among the best methods for the LOP are very small, we decided to implement the two simplest ones that are ranked on second and third position: The Tabu Search (TS) [29, 54] and the Variable Neighborhood Search (VNS) [28, 65] heuristic. For both methods (see also Martí and Reinelt [61]) all required implementation parameters are stated in the papers and the fine-tuning process can be conducted easily.

Glover and Laguna [29] complemented the basic TS procedure with a long-term diversification based on the REVERSE operation proposed by Chanas and Kobylanski [19]. The long-term strategy incorporates frequency information recorded during the application of the short-term phase.

The basic VNS for the LOP introduced by Garca-Gonzlez et al. [28] is based on shaking, improving and updating steps. The method uses up to  $k_{max}$  neighborhoods. A  $k$ -neighborhood of a solution is reached by applying general insertion moves  $k - 1$  times. The shaking step randomly generates a solution within the considered neighborhood. After that a local search method is performed to improve the solution.

Due to clear description of both TS and VNS, we could easily reproduce the computational results presented in the respective papers. This fact is very important for practitioners searching for a profitable layout in their production plant. In summary, we can assume that it is possible to obtain high-quality DCFLP layouts for instances with up to 500 machines with the help of state-of-the-art heuristics for the LOP.

## 6 Computational Experiments

The aims of the following computational study are the following:

1. Building a benchmark library in order to spark further research on the DCFLP.
2. Provide competitive lower and upper bounds for the instances in our benchmark library.
3. Show that existing exact and heuristic approaches can easily be applied for this new way of modeling layouts in FMSs and hence, especially motivate practitioners to consider circular layouts as a valid alternative to row layouts.

We do not aim to develop new exact and heuristic approaches for the LOP as this area is well-studied and hence, strong methods exist. Rather, we want to show the easy applicability of existing methods to the DCFLP and the high quality results obtained by these state-of-the-art approaches.

For generating representative data for our experiments, we take well-known benchmark instances for the SRFLP [3, 4, 5, 7, 9, 10, 42] and the LA [18, 24, 70] and modify them in order to get benchmark instances for the DCFLP and the DCAP, respectively, as follows: We propose two variants for adapting the flows. Either we set  $f_{ij} := c_{ij}$  and  $f_{ji} := 0$  or we decide randomly (with equal probability for both cases) if  $f_{ij} := c_{ij}$ ,  $f_{ji} := 0$  or  $f_{ij} := 0$ ,  $f_{ji} := c_{ij}$ . We denote the first variant as *one-way* and the second one as *random*.

The lengths of the machines are transferred without changes. As explained in Section 3, we fix one machine to be first in the ordering and hence, destroy symmetry and reduce the dimension of the problem. 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 the problem can be remodeled by setting  $\tilde{f}_{ij} := f_{ij} - \min(f_{ij}, f_{ji})$ ,  $\tilde{f}_{ji} := f_{ji} - \min(f_{ij}, f_{ji})$  and adding the constant  $\min(f_{ij}, f_{ji}) \cdot L$  to the objective function. Note that for such problems the additional constant in the objective function just reduces the relative gap and hence, makes it easier to obtain small optimality gaps. Finally, we calculate the corresponding LOP weights as explained in Section 3. Our variants for adapting the flows reflect the following two typical set-ups in FMSs:

- *one-way*: The material flow between the machines follows a natural structure. Hence, there are machines, which should be arranged at the start of a production cycle, e.g., work with raw materials and/or their output is needed by a lot of succeeding machines. Further, there are machines that should be located in the middle as they have balanced input and output flows. Finally, there are machines that receive many work-pieces, which have already been processed by one or more preceding machines, as input and their output is less used by other machines in the production cycle.
- *random*: There is no natural structure of the material flow between the machines.

We report the results for the DCFLP obtained by our exact algorithms based on SDP and ILP, described in Section 4, for obtaining tight lower bounds, and the TS and the VNS heuristic, described in the previous section, for computing strong feasible layouts.

All computations were conducted on a 2.8 GHz Intel Core i7 with 16 GB RAM, running macOS Mojave. The TS and the VNS heuristic were written in Java. We set the TS parameters for the short-term phase as suggested in Martí and Reinelt [61]. After the short-term phase we apply the long-term diversification phase, which is followed by another short-term phase. This procedure is repeated until no further improvement occurs. Our VNS uses up to 15 neighborhoods and stops after 50 consecutive iterations without any further improvement. Preliminary experiments showed that considering more than 15 neighborhoods results in neglectable improvements of the solutions. We apply our heuristics 15 times with different initial solutions for each instance and state the best layout among all solutions. The reported time corresponds to the total solving time for all 15 runs. We use Gurobi 7.5.2, restricted to one thread and a time limit of 24 hours, for solving the ILP. The SDP approach was implemented in MATLAB 7.7. We restrict the bundle method to 500 evaluations for all DCAP instances and DCFLP instances with up to 49 machines and to 250 evaluations for the remaining DCFLP instances with up to 80 machines. All instances and corresponding best-known layouts can be downloaded from <http://tinyurl.com/dcflp-lib>. Note that the LOP is a maximization problem while the DCFLP is a minimization problem. Hence, when using our instances, the objective value of the LOP solution must be multiplied by -1 to obtain the DCFLP objective value.

In Tables 1 and 2 we summarize the results for up to 36 machines. The SDP and the ILP approach solve all instances considered to optimality, where the ILP approach is on average about two orders of magnitude faster. Additionally, both heuristics find layouts close to optimal for both DCFLP variants within a few seconds. The VNS heuristic is slightly more efficient than the TS heuristic, as the VNS heuristic is faster and produces optimal layouts for all instances except for the *random* variant of *Am35\_03*, while TS is not able to provide optimal layouts for 13 instances considered.

In Tables 3 - 6 we report the computational results for DCFLP instances with 40 to 80 machines. The *one-way* variant can be solved to optimality by SDP and ILP. Additionally, the ILP approach is able to solve all *random* instances with 40 machines to optimality and produces tighter lower bounds than the SDP for *random* instances with 49 machines. For larger instances the SDP approach computes better lower bounds than the ILP. We state the gaps between the lower bounds obtained by SDP or ILP approach and the best layout determined by one of our four approaches for a clear comparison of our two exact approaches with respect to the quality of the lower bound. Note that the SDP rounding algorithm always generates worse feasible layouts compared to ILP, TS, and VNS. Both heuristics produce good results, both with respect to solving time and solution quality, for all instances considered. The VNS heuristic is able to determine the optimal layout for all *one-way* instances. Additionally, VNS is always faster than TS. However, for 14 *random* instances TS is able to find better layouts than VNS.

Tables 7 and 8 show the results obtained for DCAP instances with up to 100 machines. All instances are solved to optimality within 1 hour by our ILP approach. The SDP is slower and not able to close the gap for 3 instances. VNS produces better results than the TS for all instances considered.

In Tables 9 and 10 we state the upper and lower bounds obtained by the initial solutions of our SDP and ILP approach for medium and large sized DCFLP instances with 40 to 80 machines. Further, we report the respective relative gaps between the lower and the upper bounds. The SDP approach starts with gaps between 6% and 62% for the *one-way* instances and gaps between 43% and 85% for the *random* instances. The ILP approach is able to prove optimality of the root node for 34 of 40 *one-way* instances. Further, the ILP obtains initial gaps between 29% and 36% and hence, tighter initial gaps compared to the SDP on all medium and large sized *random* instances considered. Note that for all instances considered, the initial lower bounds produced by the ILP are better than the ones generated by the SDP.

In summary, our computational experiments support the claim that the DCFLP can be solved efficiently with the help of exact and heuristic approaches developed for solving the LOP. Especially, the *one-way* variant proves to be easy to solve for all solution approaches considered.

Instance	$n$	SDP one-way		SDP random		ILP one-way	ILP random
		Optimal layout	Time	Optimal layout	Time	Time	Time
Am15	15	8284.00	2:40	8041.00	3:50	0.01	0.01
Am17	17	12717.00	5:80	13105.00	21:10	0.01	0.02
Am18	18	14450.50	6:80	14726.50	24:10	0.01	0.01
AnVa25_01	25	6221.00	3:56	5921.00	3:54	0.01	2.69
AnVa25_02	25	53933.50	1:00	53162.50	1:36	0.02	0.03
AnVa25_03	25	34784.00	3:16	31698.00	1:23	0.11	0.04
AnVa25_04	25	70468.50	1:19	69385.50	2:54	0.02	0.48
AnVa25_05	25	22256.00	45:50	21540.00	1:05	0.03	0.03
AnVa30_01	30	11067.00	5:00	10216.00	2:55	0.10	0.11
AnVa30_02	30	30890.50	22:52	28194.50	6:16	5.29	0.28
AnVa30_03	30	64275.00	6:21	62721.00	7:23	0.09	0.88
AnVa30_04	30	84051.50	24:04	83023.50	6:51	0.60	1.29
AnVa30_05	30	164981.00	5:15	164662.00	12:18	0.05	0.30
Am33_01	33	84034.50	7:28	79955.50	10:32	0.19	6.72
Am33_02	33	94504.00	2:39	101700.00	12:12	0.14	0.78
Am33_03	33	98414.50	3:08	105357.50	12:58	0.09	1.70
Am35_01	35	95812.50	46:14	90732.50	11:05	1.79	0.53
Am35_02	35	86175.00	15:34	84546.00	20:33	0.31	28.56
Am35_03	35	96865.50	13:14	90215.50	14:49	0.45	4.37
ste36-1	36	13476.00	11:12	19548.00	7:41	0.22	0.30
ste36-2	36	237692.00	33:29	336125.00	28:05	0.25	0.24
ste36-3	36	138237.50	13:38	200207.50	16:01	0.16	0.26
ste36-4	36	131592.50	9:15	184637.50	18:06	0.11	0.25
ste36-5	36	123715.50	7:42	172998.50	11:11	0.12	0.22

Table 1: Results produced by our exact approaches for DCFLP instances with up to 36 machines. The exact approaches solved all instances to optimality within 1 hour. The running times are given in sec and min:sec, respectively.

Instance	$n$	TS one-way			TS random			VNS one-way	VNS random		
		Best layout	Gap (%)	Time	Best layout	Gap (%)	Time	Time	Best layout	Gap (%)	Time
Am15	15	8284.00	0.00	0.10	8041.00	0.00	0.11	0.01	8041.00	0.00	0.02
Am17	17	12717.00	0.00	0.14	13105.00	0.00	0.15	0.02	13105.00	0.00	0.02
Am18	18	14450.50	0.00	0.15	14726.50	0.00	0.20	0.02	14726.50	0.00	0.03
AnVa25_01	25	6227.00	0.10	0.47	5929.00	0.13	0.52	0.10	5921.00	0.00	0.10
AnVa25_02	25	53933.50	0.00	0.49	53162.50	0.00	0.44	0.07	53162.50	0.00	0.10
AnVa25_03	25	34866.00	0.24	0.48	31698.00	0.00	0.52	0.08	31698.00	0.00	0.10
AnVa25_04	25	70468.50	0.00	0.55	69701.50	0.45	0.52	0.08	69385.50	0.00	0.11
AnVa25_05	25	22256.00	0.00	0.50	21540.00	0.00	0.50	0.07	21540.00	0.00	0.12
AnVa30_01	30	11086.00	0.17	0.76	10216.00	0.00	0.94	0.20	10216.00	0.00	0.19
AnVa30_02	30	30982.50	0.30	0.84	28194.50	0.00	1.02	0.16	28194.50	0.00	0.17
AnVa30_03	30	64278.00	0.00	0.95	62766.00	0.07	0.96	0.17	62721.00	0.00	0.24
AnVa30_04	30	84076.50	0.03	0.89	83110.50	0.10	1.02	0.14	83023.50	0.00	0.20
AnVa30_05	30	164981.00	0.00	0.90	164977.00	0.19	1.00	0.16	164662.00	0.00	0.20
Am33_01	33	84073.50	0.05	1.30	79958.50	0.00	1.25	0.19	79955.50	0.00	0.27
Am33_02	33	94504.00	0.00	1.23	101702.00	0.00	1.25	0.17	101700.00	0.00	0.25
Am33_03	33	98414.50	0.00	1.39	105409.50	0.05	1.40	0.19	105357.50	0.00	0.32
Am35_01	35	95995.50	0.19	1.39	90732.50	0.00	1.75	0.33	90732.50	0.00	0.39
Am35_02	35	86353.00	0.21	1.51	84726.00	0.21	1.65	0.26	84546.00	0.00	0.35
Am35_03	35	96917.50	0.05	1.18	90512.50	0.33	1.89	0.32	90512.50	0.33	0.30
ste36-1	36	13558.00	0.60	1.50	19662.00	0.58	2.16	0.32	19548.00	0.00	0.38
ste36-2	36	239235.00	0.64	1.86	338307.00	0.64	1.85	0.41	336125.00	0.00	0.33
ste36-3	36	139001.50	0.55	1.71	202175.50	0.97	2.23	0.31	200207.50	0.00	0.34
ste36-4	36	131678.50	0.07	1.63	186748.50	1.13	1.93	0.24	184637.50	0.00	0.41
ste36-5	36	123849.50	0.11	1.51	173636.50	0.37	1.76	0.30	172998.50	0.00	0.33

Table 2: Results produced by our heuristic approaches for DCFLP instances with up to 36 machines. The running times are given in sec. *Gap* indicates the relative gap between the optimal layout and the upper bound determined by TS and VNS, respectively.

Instance	$n$	SDP one-way		SDP random			ILP one-way	ILP random		
		Optimal layout	Time	Best lower bound	Gap (%)	Time	Time	Best lower bound	Gap (%)	Time
N40_1	40	154285.50	1:43:12	148925.50	1.59*	3:44:15	45.93	151300.50	0.00*	16:36
N40_2	40	135486.00	16:15	144880.00	1.44 <sup>‡</sup>	3:55:13	0.51	146971.00	0.00 <sup>‡</sup>	16:53
N40_3	40	118601.50	1:22:06	120570.50	0.15 <sup>‡</sup>	3:38:52	0.93	120754.50	0.00 <sup>‡</sup>	24.61
N40_4	40	115049.00	5:40:28	118906.00	0.96*	4:07:58	15.21	120042.00	0.00*	2:43
N40_5	40	153266.00	1:11:50	145625.00	0.08*	3:39:24	12.01	145744.00	0.00*	28.78
sko49-1	49	50697.00	35:47	50010.00	2.00*	8:25:36	1.90	50660.00	0.73*	24:00:00
sko49-2	49	573272.00	1:01:02	574750.00	2.35*	8:29:54	1.83	582618.00	1.01*	24:00:00
sko49-3	49	446309.00	50:33	444335.00	3.25*	8:50:43	2.07	449703.00	2.08*	24:00:00
sko49-4	49	330308.50	1:09:11	322226.50	2.09 <sup>‡</sup>	8:30:50	2.54	325978.00	0.95 <sup>‡</sup>	24:00:00
sko49-5	49	896556.00	1:34:58	881914.50	2.54	8:32:15	2.24 <sup>‡</sup>	890138.00	1.63 <sup>‡</sup>	24:00:00
AKV-60-01	60	2042653.00	4:20:38	2479392.00	5.16*	10:22:20	1.59	2373755.00	9.20*	24:00:00
AKV-60-02	60	1224508.00	5:27:32	1345517.00	4.43*	11:07:06	2.50	1296861.00	7.88*	24:00:00
AKV-60-03	60	927361.50	4:20:59	1104305.00	4.10 <sup>‡</sup>	10:34:24	2.01	1084007.50	5.86 <sup>‡</sup>	24:00:00
AKV-60-04	60	610184.00	8:38:01	663133.00	2.61*	9:48:47	6.57	656912.00	3.52*	24:00:00
AKV-60-05	60	451681.00	4:11:27	576075.00	2.83 <sup>‡</sup>	10:15:09	1.88	570072.00	3.84 <sup>‡</sup>	24:00:00
sko64-1	64	122479.00	6:33:07	118187.00	5.01*	16:27:22	11.17	116611.00	6.27*	24:00:00
sko64-2	64	919477.50	10:11:12	913297.00	4.51 <sup>°</sup>	15:31:21	13.48	901642.50	5.73 <sup>°</sup>	24:00:00
sko64-3	64	553264.50	6:50:30	587192.50	4.11 <sup>‡</sup>	15:48:09	7.08	578652.50	5.50 <sup>‡</sup>	24:00:00
sko64-4	64	394514.00	6:06:59	389265.00	4.82*	15:33:45	11.77	383678.00	6.19*	24:00:00
sko64-5	64	679565.50	9:27:46	686896.00	4.47*	15:31:38	9.48	678121.50	5.69*	24:00:00

Table 3: Results produced by our exact methods for DCFLP instances with 40 to 64 machines. The bundle method for the SDP approach is restricted to 500 evaluations for instances with 40 and 49 machines and to 250 evaluations for instances with 60 and 64 machines. We set a time limit of 24 hours for the ILP approach. The running times are given in sec, min:sec, and h:min:sec, respectively. *Gap* indicates the relative gap between the best lower bound determined by SDP or ILP approach and the best layout determined by either TS<sup>°</sup>, VNS\* or ILP<sup>‡</sup>.



Instance	$n$	SDP one-way		SDP random			ILP one-way	ILP random		
		Optimal layout	Time	Best lower bound	Gap (%)	Time	Time	Best lower bound	Gap (%)	Time
AKV-70-01	70	2185508.00	14:38:14	2670853.00	4.52*	23:42:27	6.33	2527338.00	9.65*	24:00:00
AKV-70-02	70	2082572.00	14:00:41	2533911.00	5.60°	30:03:37	3.71	2396326.00	10.72°	24:00:00
AKV-70-03	70	2207952.50	19:30:29	2395187.50	5.79°	36:09:58	5.66	2266829.50	10.84°	24:00:00
AKV-70-04	70	1420836.00	27:38:48	1557019.50	4.95*	33:47:29	9.35	1499856.00	8.44*	24:00:00
AKV-70-05	70	6023331.50	16:49:54	7252641.50	5.86°	31:38:56	2.50	6737883.00	12.54°	24:00:00
sko72-1	72	168547.00	15:00:02	171274.00	5.00°	37:14:22	12.10	167078.00	7.32°	24:00:00
sko72-2	72	918687.00	14:10:09	980256.00	5.65*	30:37:27	12.84	950428.00	8.52*	24:00:00
sko72-3	72	1470058.50	20:12:00	1617738.50	5.32°	37:31:45	12.49	1573984.50	7.88°	24:00:00
sko72-4	72	1255362.50	26:08:17	1368468.50	5.73°	33:42:30	12.65	1331165.50	8.30°	24:00:00
sko72-5	72	542872.50	11:52:34	563290.50	6.04°	31:08:31	14.55	547980.00	8.59°	24:00:00
AKV-75-01	75	3457598.50	17:48:21	4030236.50	4.87°	40:21:01	5.74	3774286.50	10.91°	24:00:00
AKV-75-02	75	6170887.00	19:46:45	7285028.00	5.31*	41:04:15	4.23	6773297.00	11.96*	24:00:00
AKV-75-03	75	1895325.00	48:06:32	2182004.00	6.24°	42:01:47	2:07	2074423.00	10.86°	24:00:00
AKV-75-04	75	5496300.50	19:40:09	6670688.00	4.35*	39:17:19	5.56	6204747.50	11.03*	24:00:00
AKV-75-05	75	2480432.00	16:42:28	2844624.00	5.11*	39:50:32	6.24	2677171.00	10.70*	24:00:00
AKV-80-01	80	2837168.50	26:41:11	3544971.50	5.23°	52:32:09	6.65	3301169.50	11.75°	24:00:00
AKV-80-02	80	2785739.00	25:24:34	3198286.00	5.50°	61:54:05	8.70	2988175.00	11.71°	24:00:00
AKV-80-03	80	5113068.00	31:46:18	5963020.00	4.84°	57:18:22	7.16	5508595.50	12.10°	24:00:00
AKV-80-04	80	5695886.00	37:15:57	6488517.00	5.53*	58:24:15	7.37	5989385.00	12.80*	24:00:00
AKV-80-05	80	2247171.00	32:03:30	2461083.50	6.30°	51:36:19	26.42	2332120.00	11.21°	24:00:00

Table 4: Results produced by our exact methods for DCFLP instances with 70 to 80 machines. The bundle method for the SDP approach is restricted to 250 evaluations. We set a time limit of 24 hours for the ILP approach. The running times are given in h:min:sec. *Gap* indicates the relative gap between the best lower bound determined by SDP or ILP approach and the best layout determined by either TS°, VNS\* or ILP‡.

Instance	$n$	TS one-way			TS random			VNS one-way		VNS random		
		Best layout	Gap (%)	Time	Best layout	Gap (%)	Time	Optimal layout	Time	Best layout	Gap (%)	Time
N40_1	40	154689.50	0.26	2.38	152405.50	0.73	2.73	154285.50	0.65	151300.50	0.00	0.60
N40_2	40	135740.00	0.19	2.56	147578.00	0.41	2.86	135486.00	0.42	147468.00	0.34	0.56
N40_3	40	118764.50	0.14	2.12	120902.50	0.12	2.63	118601.50	0.69	120919.50	0.14	0.64
N40_4	40	115483.00	0.38	2.61	120605.00	0.47	2.44	115049.00	0.42	120042.00	0.00	0.57
N40_5	40	153797.00	0.35	2.73	145868.00	0.09	2.73	153266.00	0.46	145744.00	0.00	0.50
sko49-1	49	50719.00	0.04	5.07	51177.00	1.01	5.47	50697.00	0.89	51031.00	0.73	1.15
sko49-2	49	574617.00	0.23	4.86	593627.00	1.85	4.86	573272.00	0.87	588591.00	1.01	1.09
sko49-3	49	446909.00	0.13	5.17	460409.00	2.33	4.92	446309.00	0.82	459246.00	2.08	1.47
sko49-4	49	331428.50	0.34	5.83	330548.50	1.38	6.06	330308.50	0.93	329527.50	1.08	1.74
sko49-5	49	898870.00	0.26	5.55	906584.00	1.81	5.65	896556.00	0.88	906164.00	1.77	1.45
AKV-60-01	60	2047311.00	0.23	10.88	2621119.00	5.41	10.66	2042653.00	1.14	2614393.00	5.16	2.40
AKV-60-02	60	1226392.00	0.15	12.53	1410749.00	4.62	11.15	1224508.00	1.28	1408717.00	4.49	2.56
AKV-60-03	60	930453.50	0.33	10.35	1154356.50	4.34	11.75	927361.50	1.41	1156496.50	4.51	2.37
AKV-60-04	60	611533.00	0.22	12.46	682692.00	2.86	14.26	610455.00	1.27	680905.00	2.61	2.11
AKV-60-05	60	452592.00	0.20	11.01	595756.00	3.30	9.66	451681.00	2.08	595786.00	3.31	2.92
sko64-1	64	122710.00	0.19	13.28	124462.00	5.04	14.51	122479.00	2.33	124415.00	5.01	3.49
sko64-2	64	920586.50	0.12	13.81	956470.50	4.51	16.26	919477.50	2.74	956895.50	4.56	3.40
sko64-3	64	554397.50	0.20	12.09	613446.50	4.28	17.58	553264.50	1.99	612948.50	4.20	3.21
sko64-4	64	395686.00	0.30	13.74	409759.00	5.00	15.25	394514.00	2.18	408998.00	4.82	3.32
sko64-5	64	680909.50	0.20	14.09	720403.50	4.65	15.82	679565.50	1.88	719042.50	4.47	2.90

Table 5: Results produced by our heuristics for DCFLP instances with 40 to 64 machines. The running times are given in sec. *Gap* indicates the relative gap between the best lower bound computed by either SDP or ILP approach and the upper bound determined by TS and VNS, respectively.

Instance	$n$	TS one-way			TS random			VNS one-way		VNS random		
		Best layout	Gap (%)	Time	Best layout	Gap (%)	Time	Optimal layout	Time	Best layout	Gap (%)	Time
AKV-60-01	60	2047311.00	0.23	10.88	2621119.00	5.41	10.66	2042653.00	1.14	2614393.00	5.16	2.40
AKV-70-01	70	2190406.00	0.22	21.38	2807170.00	4.86	19.44	2185508.00	2.18	2797368.00	4.52	3.61
AKV-70-02	70	2088367.00	0.28	21.01	2684177.00	5.60	20.32	2082572.00	2.55	2685807.00	5.66	3.80
AKV-70-03	70	2213136.50	0.23	19.89	2542383.50	5.79	22.59	2207952.50	2.11	2557828.50	6.36	4.37
AKV-70-04	70	1423029.00	0.15	21.04	1642995.00	5.23	20.48	1420836.00	2.70	1638157.00	4.95	5.14
AKV-70-05	70	6038563.50	0.25	19.50	7703949.50	5.86	21.69	6023331.50	2.04	7705584.50	5.88	4.88
sko72-1	72	168727.00	0.11	19.81	180281.00	5.00	26.61	168547.00	3.54	180478.00	5.10	5.69
sko72-2	72	919987.00	0.14	23.93	1041345.00	5.87	24.91	918687.00	3.49	1038911.00	5.65	5.54
sko72-3	72	1473829.50	0.26	22.72	1708606.50	5.32	24.10	1470058.50	2.87	1716119.50	5.73	5.94
sko72-4	72	1257451.50	0.17	23.57	1451724.50	5.73	24.64	1255362.50	4.02	1457775.50	6.13	5.73
sko72-5	72	543797.50	0.17	22.70	599502.50	6.04	23.15	542872.50	3.25	601604.50	6.37	6.99
AKV-75-01	75	3462329.50	0.14	28.18	4236641.50	4.87	29.32	3457598.50	3.19	4237311.50	4.89	5.62
AKV-75-02	75	6178481.00	0.12	26.80	7696952.00	5.35	28.73	6170887.00	2.79	7693556.00	5.31	5.67
AKV-75-03	75	1902889.00	0.40	26.60	2327133.00	6.24	27.44	1895444.00	4.02	2343121.00	6.88	5.25
AKV-75-04	75	5512212.50	0.29	28.14	6976729.50	4.39	28.59	5496300.50	2.64	6973983.50	4.35	5.42
AKV-75-05	75	2486228.00	0.23	28.56	3006123.00	5.37	29.21	2480432.00	2.76	2997851.00	5.11	4.38
AKV-80-01	80	2844109.50	0.24	36.33	3740713.50	5.23	34.19	2837168.50	3.07	3746916.50	5.39	6.99
AKV-80-02	80	2789081.00	0.12	32.57	3384420.00	5.50	38.80	2785739.00	4.01	3391813.00	5.71	5.61
AKV-80-03	80	5125726.00	0.25	36.99	6266607.00	4.84	34.91	5113068.00	3.34	6270503.00	4.90	6.60
AKV-80-04	80	5707624.00	0.21	32.62	6881683.00	5.71	39.11	5695886.00	4.40	6868670.00	5.53	6.48
AKV-80-05	80	2254412.00	0.32	33.62	2626486.00	6.30	32.85	2247171.00	4.14	2627385.00	6.33	8.47

Table 6: Results produced by our heuristics for DCFLP instances with 70 to 80 machines. The running times are given in sec. *Gap* indicates the relative gap between the best lower bound computed by either SDP or ILP approach and the upper bound determined by TS and VNS, respectively.

Instance	$n$	Density	SDP one-way		SDP random				ILP one-way	ILP random	
			Optimal layout	Time	Best lower bound	Best layout	Gap (%)	Time	Time	Optimal layout	Time
can_24	24	0.246	311.00	26:50	379.00		0.00	43.2	0.02	379.00	0.03
fidap005	27	0.358	702.00	48:10	981.00		0.00	1:10	0.06	981.00	0.06
pores_1	30	0.236	383.00	1:34	679.00		0.00	2:15	0.05	679.00	0.12
ibm32	32	0.181	638.00	24:34	648.00		0.00	10:54	0.35	648.00	1.59
bcpwr01	39	0.062	131.00	16:07	129.00		0.00	10:35	0.19	129.00	0.21
fidapm05	42	0.277	1989.00	14:31	2745.00		0.00	22:07	0.54	2745.00	1.20
bcpwr02	49	0.050	218.00	58:23	190.00		0.00	46:13	1.44	190.00	0.98
will57	57	0.079	571.00	3:08:23	1120.00		0.00	2:42:52	8.27	1120.00	3.51
dwt_59	59	0.060	327.00	3:01:50	811.00		0.00	3:34:27	2.37	811.00	8.77
impcol_b	59	0.164	2659.00	3:50:44	4193.00	4346.00	3.52	21:00:32	1.99	4202.00	16:34
can_62	62	0.041	271.00	5:55:29	310.00		0.00	4:59:59	3.89	310.00	13.20
gd95c	62	0.076	796.00	2:58:02	1370.00		0.00	6:49:52	2.23	1370.00	17.43
dwt_66	66	0.059	346.00	9:24:23	1025.00		0.00	6:41:40	5.52	1025.00	8.86
dwt_72	72	0.029	175.00	20:50:14	234.00	236.00	0.85	59:05:46	1.41	236.00	39:57
can_73	73	0.057	1304.00	41:55:45	1572.00	1579.00	0.44	57:04:44	67.13	1574.00	6:11
tub100	100	0.029	246.00	222:20:27	934.00		0.00	226:34:24	40.34	934.00	3:25

Table 7: Results produced by our exact methods for DCAP instances with up to 100 machines. The bundle method is restricted to 500 function evaluations. The running times are given in sec, min:sec, and in h:min:sec, respectively. *Gap* indicates the relative gap between the best lower and upper bound determined by our SDP or ILP approach respectively.

Instance	$n$	Density	TS one-way			TS random			VNS one-way			VNS random		
			Best layout	Gap (%)	Time	Best layout	Gap (%)	Time	Best layout	Gap (%)	Time	Best layout	Gap (%)	Time
can_24	24	0.246	311.00	0.00	0.40	379.00	0.00	0.40	311.00	0.00	0.08	379.00	0.00	0.07
fidap005	27	0.358	702.00	0.00	0.68	986.00	0.51	0.64	702.00	0.00	0.10	981.00	0.00	0.12
pores_1	30	0.236	383.00	0.00	0.94	679.00	0.00	0.85	383.00	0.00	0.12	679.00	0.00	0.14
ibm32	32	0.181	638.00	0.00	1.23	656.00	1.22	1.10	638.00	0.00	0.28	648.00	0.00	0.27
bcpwr01	39	0.062	147.00	10.88	2.12	148.00	12.84	2.21	131.00	0.00	0.51	131.00	1.53	0.57
fidapm05	42	0.277	1997.00	0.40	2.62	2757.00	0.44	2.98	1989.00	0.00	0.44	2745.00	0.00	0.61
bcpwr02	49	0.050	224.00	2.68	5.33	211.00	9.95	5.59	218.00	0.00	1.26	190.00	0.00	1.44
will57	57	0.079	644.00	11.34	9.06	1137.00	1.50	8.40	587.00	2.73	1.77	1122.00	0.18	2.20
dwt_59	59	0.060	361.00	9.42	9.25	878.00	7.63	10.14	329.00	0.61	2.04	820.00	1.10	2.83
impcol_b	59	0.164	2661.00	0.08	9.15	4278.00	1.77	11.717	2659.00	0.00	1.22	4209.00	0.17	2.55
can_62	62	0.041	396.00	31.57	12.19	370.00	16.22	11.59	273.00	0.73	2.67	313.00	0.96	2.67
gd95c	62	0.076	943.00	15.59	10.56	1406.00	33.57	12.91	796.00	0.00	2.34	1384.00	32.51	3.20
dwt_66	66	0.059	374.00	7.49	14.91	1120.00	8.48	14.39	346.00	0.00	3.32	1025.00	0.00	3.03
dwt_72	72	0.029	180.00	2.78	22.16	340.00	30.59	21.55	175.00	0.00	2.94	256.00	7.81	6.05
can_73	73	0.057	1361.00	4.19	17.37	1771.00	11.12	25.99	1330.00	1.95	3.46	1664.00	5.41	5.29
tub100	100	0.029	246.00	0.00	102.08	1236.00	24.43	76.91	246.00	0.00	20.37	1077.00	13.28	17.73

Table 8: Results produced by our heuristics for DCAP instances with up to 100 machines. The running times are given in sec. *Gap* indicates the relative gap between the best lower bound computed by either SDP or ILP approach and the upper bound determined by TS and VNS, respectively.

Instance	$n$	SDP one-way			SDP random			ILP one-way			ILP random		
		Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)
N40_1	40	124841.33	171598.50	37.45	115586.33	203501.50	76.06	152179.98	275564.50	44.80	146506.86	225375.50	35.00
N40_2	40	106810.90	148888.00	39.39	114503.64	200557.00	75.15	135486.00	135486.00	0.00	142981.05	221995.00	35.60
N40_3	40	93579.08	133105.50	42.24	92177.54	160704.50	74.34	118394.50	232532.50	49.10	119202.40	182151.50	34.60
N40_4	40	85671.36	127171.00	48.41	90332.09	166419.00	84.23	114731.68	216814.00	47.10	117433.41	177295.00	33.80
N40_5	40	121385.19	168735.00	39.00	107865.16	199141.00	84.62	152544.12	268652.00	43.20	144008.42	224105.00	35.70
sko49-1	49	48188.48	51105.00	6.05	39095.68	66094.00	69.06	50697.00	50697.00	0.00	49281.33	69611.00	29.20
sko49-2	49	468139.27	601357.00	28.46	446121.82	772868.00	73.24	573272.00	573272.00	0.00	566757.19	842539.00	32.70
sko49-3	49	363478.68	363478.68	28.12	350338.21	604761.00	72.62	446309.00	446309.00	0.00	437621.32	629696.00	30.50
sko49-4	49	273686.43	347544.50	26.99	254729.88	412177.50	61.18	330308.50	330308.50	0.00	316969.99	470481.50	32.60
sko49-5	49	737958.58	930723.50	26.12	693466.88	1176368.50	69.63	896556.00	896556.00	0.00	867812.80	1243768.00	30.20
AKV-60-01	60	1655960.40	2303065.00	39.07	2035340.86	3238668.00	59.12	2042653.00	2042653.00	0.00	2339584.00	3347614.00	30.10
AKV-60-02	60	1011356.57	1308449.00	29.37	1062901.43	1805700.00	69.88	1224508.00	1224508.00	0.00	1272856.70	1780298.00	28.50
AKV-60-03	60	723106.23	1040526.00	43.89	853594.63	1440166.00	68.72	927361.50	927361.50	0.00	1059528.50	1520819.50	30.30
AKV-60-04	60	485268.54	724668.00	49.33	488976.72	878481.00	79.66	610107.00	1117746.00	45.40	640853.00	985447.00	35.00
AKV-60-05	60	341069.67	550085.00	61.28	431600.10	755956.00	75.15	451681.00	451681.00	0.00	555289.67	847571.00	34.50
sko64-1	64	101196.56	123415.00	21.96	91995.82	168605.00	83.28	122479.00	122479.00	0.00	115405.00	165965.00	30.50
sko64-2	64	745751.32	962723.00	29.09	710222.64	1181876.00	64.41	919477.50	919477.50	0.00	892083.83	1308356.50	31.80
sko64-3	64	449030.35	577931.50	28.71	522509.74	799669.50	53.04	553264.50	553264.50	0.00	572289.17	881522.50	35.10
sko64-4	64	326201.39	404330.00	23.95	298589.87	520203.00	74.22	394514.00	394514.00	0.00	379914.67	559778.00	32.10
sko64-5	64	560748.87	698666.00	24.59	606293.38	945269.00	55.91	679565.50	679565.50	0.00	671100.83	1012268.50	33.70

Table 9: Upper and lower initial bounds produced by our exact methods for DCFLP instances with 40 to 64 machines. *Gap* indicates the relative gap between the lower and the upper bound at the root node.

Instance	$n$	SDP one-way			SDP random			ILP one-way			ILP random		
		Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)	Lower bound	Upper bound	Gap (%)
AKV-70-01	70	1702555.25	2605924.00	53.06	2146066.32	3481809.00	62.24	2185508.00	2185508.00	0.00	2507299.70	3652188.00	31.30
AKV-70-02	70	1681319.03	2324959.00	38.28	2051912.51	3318501.00	61.72	2082572.00	2082572.00	0.00	2371169.30	3437463.00	31.00
AKV-70-03	70	1890057.00	2370558.50	25.42	1915233.87	3100000.50	61.86	2207952.50	2207952.50	0.00	2239093.50	3183726.50	29.70
AKV-70-04	70	1168138.10	1587803.50	35.93	1167886.67	2106376.50	80.36	1420836.00	1420836.00	0.00	1478915.70	2194363.00	32.60
AKV-70-05	70	5030441.57	6766535.50	34.51	6074801.42	8687495.50	43.01	6023331.50	6023331.50	0.00	6686324.80	9548820.50	30.00
sko72-1	72	144900.57	169015.00	16.64	134654.67	231248.00	71.73	168547.00	168547.00	0.00	165931.00	244057.00	32.00
sko72-2	72	760581.23	930384.00	22.33	760377.63	1306061.00	71.76	918687.00	918687.00	0.00	942983.67	1385507.00	31.90
sko72-3	72	1181649.33	1515672.50	28.27	1260577.47	2197924.50	74.36	1470058.50	1470058.50	0.00	1564317.50	2352338.50	33.50
sko72-4	72	1008587.62	1294356.50	28.33	1067132.22	1813884.50	69.97	1255362.50	1255362.50	0.00	1321217.80	1958375.50	32.90
sko72-5	72	461561.27	546371.50	18.37	434795.14	787548.50	81.13	542872.50	542872.50	0.00	544001.83	796912.50	31.70
AKV-75-01	75	2839630.46	3934347.50	38.55	3281612.65	5186132.50	58.04	3457598.50	3457598.50	0.00	3743695.50	5426693.50	31.00
AKV-75-02	75	5224042.91	7169256.00	37.24	6070949.86	9250348.00	52.37	6170887.00	6170887.00	0.00	6715680.70	9886334.00	32.10
AKV-75-03	75	1535659.38	2242736.00	46.04	1687306.11	2909603.00	72.44	1894873.70	3585552.00	47.20	2063152.70	3099122.00	33.40
AKV-75-04	75	4505918.36	6045261.00	34.16	5571484.18	8206555.00	47.29	5496300.50	5496300.50	0.00	6141017.50	9043990.50	32.10
AKV-75-05	75	2135171.35	2760363.00	29.28	2257950.51	3617286.00	60.20	2480432.00	2480432.00	0.00	2653957.70	3953617.00	32.90
AKV-80-01	80	2295246.23	3089308.50	34.60	2938901.10	4707785.50	60.19	2837168.50	2837168.50	0.00	3284713.80	4836117.50	32.10
AKV-80-02	80	2460084.35	3098642.00	25.96	2689335.36	4052757.00	50.70	2785739.00	2785739.00	0.00	2953330.70	4375786.00	32.50
AKV-80-03	80	4412842.32	6058259.00	37.29	5095752.79	7603048.00	49.20	5113068.00	5113068.00	0.00	5451468.00	7771924.00	29.90
AKV-80-04	80	5081748.29	6521919.00	28.34	5541097.73	8245583.00	48.81	5695886.00	5695886.00	0.00	5936876.70	8728520.00	32.00
AKV-80-05	80	1908292.16	2411054.50	26.35	1946108.25	3272756.50	68.17	2247171.00	2247171.00	0.00	2316610.70	3424074.00	32.30

Table 10: Upper and lower initial bounds produced by our exact methods for DCFLP instances with 70 to 80 machines. *Gap* indicates the relative gap between the lower and the upper bound at the root node.

## 7 Conclusion and Outlook

In this work, we analyzed the Directed Circular Facility Layout Problem (DCFLP). The DCFLP represents a new modeling approach for circular layouts and allows for a wide range of applications. We proved that it can be modeled as a special Linear Ordering Problem. Hence, both exact and heuristic methods for the DCFLP are less complex in ordering variables than the best current approaches for the Single Row Facility Layout Problem (SRFLP), which was so far considered as the simplest layout type. This refutes the general assumption that circular layout problems are harder to solve than row layout problems.

We applied two exact approaches, a Semidefinite and an Integer Linear Programming (ILP) approach. Additionally, we solved the problem heuristically using a Tabu Search and a Variable Neighborhood Search (VNS). Finally, we generated an extensive benchmark library for the DCFLP and provided strong lower and upper bounds for all instances considered.

The aim of our work is to propose efficient and easy to implement exact and heuristic approaches. Especially, we motivate practitioners to consider circular facility layouts for their manufacturing systems. However, it could be interesting for future research to implement hybrid approaches, e.g., applying a VNS and exploring the  $k$ -neighborhood with the help of an ILP formulation.

As a worthwhile future research direction we advise to combine and extend the very efficient heuristics for the DCFLP and the SRFLP to the Combined Cell Layout Problem (CCLP). The CCLP is interested in the minimization of the material-handling costs in a cellular manufacturing system with at least two cells in the presence of parts that require processing in more than one cell. The machines of each cell can be arranged differently, e.g., in a row or on a circle. Hence, more complex layout types can be modeled as a CCLP while it still builds on well-discussed and efficiently solvable basic problems. There even exist lower bounds for reasonably sized instances of the CCLP to assess heuristic approaches [8, 40].

As another future research topic we suggest to enhance the heuristics for the LOP (and hence, also for the DCFLP) such that they are able to handle additional practical constraints relevant in the layout context like, e.g., dynamic and stochastic aspects as well as the ability to consider multiple objectives.

## 8 Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

- [1] Panos Afentakis. A loop layout design problem for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 1:175–196, 1989.
- [2] İ. K. Altmel and T. Öncan. Design of unidirectional cyclic layouts. *International Journal of Production Research*, 43(19):3983–4008, 2005.
- [3] André R. S. Amaral. On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508–518, 2006.
- [4] André R. S. Amaral. An exact approach to the one-dimensional facility layout problem. *Operations Research*, 56(4):1026–1033, 2008.



- 
- [5] André R. S. Amaral. A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157(1):183–190, 2009.
- [6] 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.
- [7] 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.
- [8] M. F. Anjos, P. Hungerländer, and K. Maier. An integer linear programming approach for the combined cell layout problem. In *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 705–709, Dec 2018.
- [9] Miguel F. Anjos and Anthony 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.
- [10] Miguel F. Anjos and Ginger Yen. Provably near-optimal solutions for very large single-row facility layout problems. *Optimization Methods and Software*, 24(4):805–817, 2009.
- [11] Henri Aujac. La hiérarchie des industries dans un tableau des échanges interindustriels. *Revue économique*, 11(2):169–238, 1960.
- [12] Amotz Bar-Noy, Joseph (Seffi) Naor, and Baruch Schieber. Pushing dependent data in clients-providers-servers systems. *Wireless Networks*, 9(5):421–430, 2003.
- [13] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [14] Y.A. Bozer and S.C. Rim. Exact solution procedures for the circular layout problem. Technical report, University of Michigan, USA, Report No. 89-33, 1989.
- [15] Yavuz A. Bozer and Rim Suk-Chul. A branch and bound method for solving the bidirectional circular layout problem. *Applied Mathematical Modelling*, 20(5):342–351, 1996.
- [16] Christoph Buchheim, Angelika Wiegele, and Lanbo Zheng. Exact algorithms for the quadratic linear ordering problem. *INFORMS Journal on Computing*, 22(1):168–177, 2010.
- [17] Vicente Campos, Fred Glover, Manuel Laguna, and Rafael Martí. An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, 21(4):397–414, 2001.
- [18] Alberto Caprara and Juan-José Salazar-González. Laying out sparse graphs with provably minimum bandwidth. *INFORMS Journal on Computing*, 17:356–373, 2005.
- [19] Stefan Chanas and Przemyslaw Kobylanski. A new heuristic algorithm solving the linear ordering problem. *Computational Optimization and Applications*, 6(2):191–205, 1996.
- [20] H.B. Chenery and T. Watanabe. International comparisons of the structure of production. *Econometrica*, 26:487–521, 1958.
- [21] 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.
- [22] Markus Chimani and Philipp Hungerländer. Exact approaches to multilevel vertical orderings. *INFORMS Journal on Computing*, 25(4):611–624, 2013.
- [23] Dilip Datta, André R. S. Amaral, and Jos Rui Figueira. Single row facility layout problem using a permutation-based genetic algorithm. *European Journal of Operational Research*, 213(2):388–394, 2011.
- [24] Josep Díaz, Jordi Petit, and Maria Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34:313–356, 2002.

- [25] Thomas A. Feo and Mauricio G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.
- [26] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105:451–469, 2006.
- [27] J.F. Frenzel. Genetic algorithms. *Potentials, IEEE*, 12(3):21–24, 1993.
- [28] Carlos Garca-Gonzalez, Dionisio Prez-Brito, Vicente Campos, and Rafael Marti. Variable neighborhood search for the linear ordering problem. *Computers & OR*, 33:3549–3565, 12 2006.
- [29] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.
- [30] Fred Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977.
- [31] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.
- [32] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
- [33] Martin Grötschel, Michael Jünger, and Gerhard Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.
- [34] M. M. D. Hassan. Machine layout problem in modern manufacturing facilities. *International Journal of Production Research*, 32(11):2559–2584, 1994.
- [35] C. Helmberg. Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3):952–969, 2000.
- [36] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.
- [37] Sunderesh S. Heragu and Andrew Kusiak. Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268, 1988.
- [38] P. Hungerländer. *Semidefinite Ordering Problems*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2012.
- [39] 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.
- [40] P. Hungerländer and M. F. Anjos. An exact approach for the combined cell layout problem. In *Operations Research Proceedings 2012*, pages 275–281, 2013.
- [41] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming*, 140(1):77–97, 2013.
- [42] Philipp Hungerländer and Franz Rendl. A computational study and survey of methods for the single-row facility layout problem. *Computational Optimization and Applications*, 55(1):1–20, 2013.
- [43] P. Hungerländer, K. Maier, J. Pcher, and C. Truden. On a New Modelling Approach for Circular Layouts and Its Practical Advantages. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 1581–1585, Dec 2017.
- [44] R. Kaas. A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, 8(4):355–362, 1981.
- [45] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.
- [46] Ali S. Kiran and Selcuk Karabati. Exact and approximate algorithms for the loop layout

- problem. *Production Planning & Control*, 4(3):253–259, January 1993.
- [47] Ali S. Kiran, Ali Tamer Unal, and Selcuk Karabati. A location problem on unicyclic networks: Balanced case. *European Journal of Operational Research*, 62(2):194–202, 1992.
  - [48] Ravi Kothari and Diptesh Ghosh. The single row facility layout problem: state of the art. *OPSEARCH*, 49(4):442–462, December 2012. ISSN 0030-3887, 0975-0320.
  - [49] Ravi Kothari and Diptesh 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.
  - [50] P Kouvelis, W-C Chiang, and AS Kiran. A survey of layout issues in flexible manufacturing systems. *Omega*, 20(3):375–390, 1992.
  - [51] Panagiotis Kouvelis and Michael W. Kim. Unidirectional loop network layout problem in automated manufacturing systems. *Operations Research*, 40:533–550, 1992.
  - [52] Sadan Kulturel-Konak. Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century. *Journal of Intelligent Manufacturing*, 18(2):273–284, April 2007.
  - [53] Andrew Kusiak and Sunderesh S. Heragu. The facility layout problem. *European Journal of Operational Research*, 29(3):229 – 251, 1987.
  - [54] Manuel Laguna, Rafael Martí, and Vicente Campos. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computer & Operations Research*, 26(12):1217–1230, 1999.
  - [55] Shine-Der Lee, Kuo-Hsu Huang, and Chih-Ping Chiang. Configuring layout in unidirectional loop manufacturing systems. *International Journal of Production Research*, 39(6):1183–1201, January 2001.
  - [56] Janny Leung. Polyhedral structure and properties of a model for layout design. *European Journal of Operational Research*, 77(2):195–207, 1994.
  - [57] Vincenzo Liberatore. Circular arrangements. In Peter Widmayer, Stephan Eidenbenz, Francisco Triguero, Rafael Morales, Ricardo Conejo, and Matthew Hennessy, editors, *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 782–783. Springer Berlin Heidelberg, 2002.
  - [58] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.
  - [59] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.
  - [60] B. Malakooti. Unidirectional loop network layout by a LP heuristic and design of telecommunications networks. *Journal of Intelligent Manufacturing*, 15(1):117–125, 2004.
  - [61] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*. Number 175 in Applied Mathematical Sciences. Springer-Verlag Berlin Heidelberg, 2011. ISBN 978-3-642-16728-7.
  - [62] 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.
  - [63] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1994.
  - [64] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In H. Frenk, K. Roos, T. Terlaky, and S. Zhang, editors, *High Performance Optimization*, pages 349–366. Kluwer Academic Publishers, 2000.

- [65] N. Mladenovi and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- [66] Joseph (Seffi) Naor and Roy Schwartz. The directed circular arrangement problem. *ACM Transactions on Algorithms*, 6:47:1–47:22, 2010.
- [67] Andreas C. Nearchou. Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, 101(2):312–328, 2006.
- [68] Hamed Samarghandi and Kourosh Eshghi. An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*, 205(1):98 – 105, 2010.
- [69] Tommaso Schiavinotto and Thomas Stützle. The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 3(4):367–402, 2005.
- [70] H. Seitz. *Contributions to the Minimum Linear Arrangement Problem*. PhD thesis, University of Heidelberg, Germany, 2010.
- [71] S. P. Singh and R. R. K. Sharma. A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology*, 30(5):425–433, September 2006.
- [72] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.
- [73] D. H. Younger. Minimum feedback arc sets for a directed graph. *IEEE Transactions on Circuit Theory*, 10(2):238–245, 1963.

## A Computing Lower Bounds for the Directed Circular Facility Layout Problem with Semidefinite Programming Relaxations

Semidefinite Programming (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 general (primal) SDP model can be expressed as the following optimization problem:

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

where the data matrices  $A_i$ ,  $i \in \{1, \dots, m\}$  and  $C$  are symmetric. We refer the reader to the handbooks by Wolkowicz et al. [72] and Anjos and Lasserre [6] 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.

For applying an SDP approach it is helpful to model the Directed Circular Facility Layout Problem (DCFLP) with the help of 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 1 in Section 3 as follows:

**Corollary 2.** *The problem*

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

subject to:

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

the 3-cycle inequalities

$$-1 \leq y_{ij} + y_{jk} - y_{ik} \leq 1, \quad i < j < k \in [n], \quad (8)$$

and integrality conditions

$$y_{ij} \in \{-1, 1\}, \quad i, j \in [n], \quad i < j$$

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 3 and ensure that the distances between machines are computed correctly. The 3-cycle inequalities (8) together with the integrality conditions on  $y$  suffice to induce a feasible layout on the circle.

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

$$\mathcal{P}_{LQO} := \text{conv} \left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top : y \in \{-1, 1\}, y \text{ satisfies (8)} \right\}.$$

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

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

Moreover, the main diagonal entries of  $Y$  correspond to squared  $\{-1, 1\}$  variables, hence,  $\text{diag}(Y) = e$ , 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 [13, 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 ellipsope

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

In order to express constraints on  $y$  in terms of  $Z$ , 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. Now applying  $y_{ij}^2 = 1$  to the resulting equations gives

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i, j, k \in [n], \quad i < j < k. \quad (9)$$

In [16] it is shown that these 3-cycle equations formulated in the  $\{0, 1\}$  model<sup>1</sup> describe the smallest linear subspace that contains  $\mathcal{P}_{LQO}$ . The 3-cycle inequalities are implicitly ensured by the 3-cycle equations together with  $Z \succcurlyeq 0$  [38, Proposition 4.2].

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

**Theorem 3.** *The following optimization problem is equivalent to the DCFLP:*

$$\min \{ K + \langle C_Z, Z \rangle : Z \text{ satisfies (9)}, Z \in \mathcal{E}, y \in \{-1, 1\} \}$$

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:

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

*Proof.* Since  $y_i^2 = 1$ ,  $i \in \{1, \dots, \Delta - 1\}$  we have  $\text{diag}(Y - yy^\top) = 0$ , which together with  $Y - yy^\top \succcurlyeq 0$  shows that in fact  $Y = yy^\top$  is integral. The 3-cycle equations (9) 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:

$$\min \{ K + \langle C_Z, Z \rangle : Z \text{ satisfies (9)}, Z \in \mathcal{E} \}. \quad (\text{SDP}_{\text{basic}})$$

There are several ways to tighten  $\text{SDP}_{\text{basic}}$ . 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  $(1 \ y)$  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:

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

<sup>1</sup>In [35] 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.

A second class of strengthening constraints for our problem was proposed by [59]. They suggest to multiply the 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \quad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0, \quad (10)$$

by the nonnegative expressions

$$1 - y_{lo} \geq 0, \quad 1 + y_{lo} \geq 0, \quad l, o \in [n], \quad l < o. \quad (11)$$

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

$$\begin{aligned} -1 - y_{lo} &\leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lo} + y_{jk,lo} - y_{ik,lo} \leq 1 + y_{lo}, & i, j, k, l, o \in [n], \\ -1 + y_{lo} &\leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lo} - y_{jk,lo} + y_{ik,lo} \leq 1 - y_{lo}, & i < j < k, \quad l < o. \end{aligned} \quad (12)$$

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

$$\mathcal{LS} := \{ Z : Z \text{ satisfies (12)} \}. \quad (13)$$

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

$$\min \{ K + \langle C_Z, Z \rangle : Z \text{ satisfies (9)}, Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \}. \quad (\text{SDP}_{\text{strong}})$$

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  $\text{SDP}_{\text{strong}}$  [38, Proposition 4.2]. Hence, it is not surprising that  $\text{SDP}_{\text{strong}}$  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.

## B Rounding Heuristic for Computing Upper Bounds for the Directed Circular Facility Layout Problem

In this section we propose an adapted rounding algorithm (see [41], [42]) that is used in our Semidefinite Programming approach (SDP) for obtaining upper bounds for the Directed Circular Facility Layout Problem (DCFLP). The upper bounds constitute feasible layouts of the machines on the circle.

We apply the hyperplane rounding algorithm of [32] to the solution of our SDP relaxation stated in Appendix A. We take the resulting vector  $\bar{y}$  and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities (see (8) in Appendix A). Computational experiments showed that the repair strategy is not as critical as one might assume [21]. For example for the Single-Row Facility Layout Problem the SDP-based rounding heuristic performs comparably to the strongest heuristics [23, 49, 68].

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 machines 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  $\bar{y}$ .
3. Compute the induced objective value  $z(\bar{y})$ . If  $z(\bar{y}) \geq z(y')$ : go to step 2.
4. If  $\bar{y}$  satisfies all 3-cycle inequalities: set  $y' := \bar{y}$  and goto 2. Else: modify  $\bar{y}$  by changing the signs of one of three variables in all violated inequalities and go to step 3.

The final  $y'$  is a feasible solution of the DCFLP. As this algorithm is applied in an (SDP) approach, we check if the duality gap is closed. If it is closed we stop as we found the optimal solution otherwise, we return to the SDP optimization algorithm for computing a new lower bound (see Appendix A) and then retry the heuristic (retaining the last vector  $y'$ ).