Kerstin Maier

Exact and Heuristic Approaches for the Directed Circular Facility Layout Problem

MASTERARBEIT

zur Erlangung des akademischen Grades
Diplom-Ingenieurin

Studium: Masterstudium Technische Mathematik

Alpen-Adria-Universität Klagenfurt
Fakultät für Technische Wissenschaften

Betreuer: MMag. Dipl.Ing. DDr. Philipp Hungerländer Bakk.techn.
Institut für Mathematik, Universität Klagenfurt

Mai 2018

Eidesstattliche Erklärung

Mir ist bekannt, dass die digitale Version der eingereichten wissenschaftlichen Arbeit zur Plagiatskontrolle herangezogen wird.

- die eingereichte wissenschaftliche Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe,
- die während des Arbeitsvorganges von dritter Seite erfahrene Unterstützung, ein- schließlich signifikanter Betreuungshinweise, vollständig offengelegt habe,
- die Inhalte, die ich aus Werken Dritter oder eigenen Werken wortwörtlich oder sinn- gemäß übernommen habe, in geeigneter Form gekennzeichnet und den Ursprung der Information durch möglichst exakte Quellenangaben (z.B. in Fußnoten) ersichtlich gemacht habe,
- die eingereichte wissenschaftliche Arbeit bisher weder im Inland noch im Ausland einer Prüfungsbehörde vorgelegt habe und
- bei der Weitergabe jedes Exemplars (z.B. in gebundener, gedruckter oder digitaler Form) der wissenschaftlichen Arbeit sicherstelle, dass diese mit der eingereichten digitalen Version übereinstimmt.

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

Kerstin Maier e.h. Klagenfurt, 02. Mai 2018

Abstract

In this thesis, we discuss a new facility layout problem, the Directed Circular Facility Layout Problem (DCFLP). The DCFLP seeks for an optimal arrangement of n machines in an unidirectional circular material handling system, such that the total weighted sum of the center-to-center distances between all pairs of machines is minimized. It allows for a wide range of applications and contains several other layout problems that have been discussed extensively in literature as special cases.

We model the DCFLP as a Linear Ordering Problem and solve it using both heuristic and exact approaches. On the one hand, we propose a Tabu Search and a Variable Neighborhood Search heuristic and on the other hand, we apply a Semidefinite and an Integer Linear Programming approach. We set up a comprehensive benchmark library to encourage further research in this field. Finally, we present the results of our approaches in a computational study.

Acknowledgements

First, I would like to express my deepest gratitude to my supervisor DDr. Philipp Hungerländer for his enormous support through the process of researching and writing this thesis.

I also would like to thank my family for their encouragement throughout my study. Without their support this would not have been possible for me.

Finally, I would like to thank my friends for making my years of study an incredible experience.

Contents

Chapter 1

Introduction

In this thesis, we consider a new facility layout problem, namely the Directed Circular Facility Layout Problem (DCFLP). The DCFLP aims to find an optimal layout, such that the total weighted sum of the center-to-center distances between all pairs of machines in an unidirectional circular material handling system is minimized. We show that the DCFLP can be formulated as a Linear Ordering Problem (LOP). Hence, it can be solved efficiently by using exact and heuristic approaches for the LOP. On the one hand, we apply a Semidefinte Programming (SDP) as well as an Integer Linear Programming (ILP) approach and on the other hand, we use a Tabu Search and a Variable Neighborhood Search heuristic, for solving the DCFLP.

An extended abstract covering the key results from this thesis has been published in proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management [44]. Further we will submit an extended version of this abstract to an international journal in the near future. My personal main contributions to these publications were the following: I took part in the development process of the theoretical concepts and the preparation of the material. I implemented both heuristics in Java. For the implementation of the ILP I used Gurobi 7.5.2 as the underlying ILP solver. The mathematical formulation and implementation of the SDP are based on [39, 40]. I ran experiments with the ILP and the heuristics on the proposed benchmark instances and analyzed the produced results.

The thesis is structured as follows. In Chapter 2 we introduce the DCFLP and give an overview of related facility layout problems. In Chapter 3 we present a mathematical formulation for the DCFLP based on binary ordering variables. In Chapter 4 and 5 we briefly recall the best heuristic and exact approaches for the LOP that are additionally easy to implement. In Chapter 6 we summarize the results of our computational study for the DCFLP. Finally, Chapter 7 concludes the thesis and in Chapter 8 we point out possible future research topics.

Chapter 2

Facility Layout Problems

Facility layout problems aim to find the optimal location of machines inside a production plant with respect to a given objective function that considers for example transportation and construction costs or simply pair-wise preferences among machines. Facility layouts are well-known operations research problems and arise in different application areas. Meller and Gau [60] divide facility layout problems into three categories:

- 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 thesis we consider one of the layout problems in the third category, namely the Directed Circular Facility Layout Problem (`DCFLP`) that has recently been suggested by Hungerländer [40]. This category is concerned with problem instances that follow a special structure, such as the arrangement of machines along a production line.

Arranging machines within a Flexible Manufacturing System (FMS) is an essential problem [52], 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 [35]. Machine layouts are in general defined through the required type of material-handling device, such as conveyor systems, Automated Guided Vehicles (AGVs) or robots [66]. The following layout types are the most common ones in practical applications:

- single-row layouts (Figure 2.1),

- double-row and multi-row layouts (Figure 2.2) and

- circular layouts (Figure 2.3).

Figure 2.1: In a.) material is carried between the machines with the help of an AGV which moves in both directions in a straight line. A material-handling industrial robot transports material between the machines in b.).

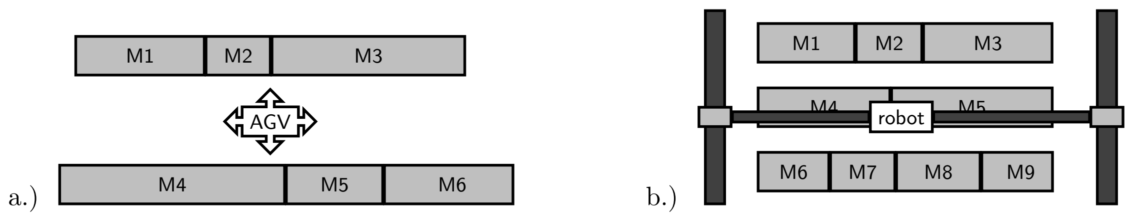


Figure 2.2: In a.) again an AGV carries parts between the machines. In b.) a gantry robot is used in case the space is limited.

Our considered layout problem, the `DCFLP`, aims to find an optimal arrangement of machines in 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. The material handling system moves the material unidirectionally around the circuit taking into account the specified sequence of its process plan. It can be assumed that each machine picks up and processes the material from the material handling system [56]. Circular material handling systems are preferred in practice, because of relative low initial investment costs, high material handling flexibility, space-saving design and the ability of being easily adapted to future introduction of new parts and process changes [2, 50].

A `DCFLP`-instance is made up of n one-dimensional machines, located next to each other on a circle, with given positive lengths $\ell_1, \dots, \ell_n$ and pairwise flows $f_{ij},\ i,j \in [n],\ i \neq j$. The aim is to find a permutation π of the machines such that the total weighted sum of the center-to-center distances between all pairs of machines, measured in clockwise direction, is minimized, i.e.,

$$\min_{\pi \in \Pi_n} \sum_{i,j \in [n],\ i \neq j} f_{ij} z_{ij}^{\pi}, \tag{2.1}$$

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

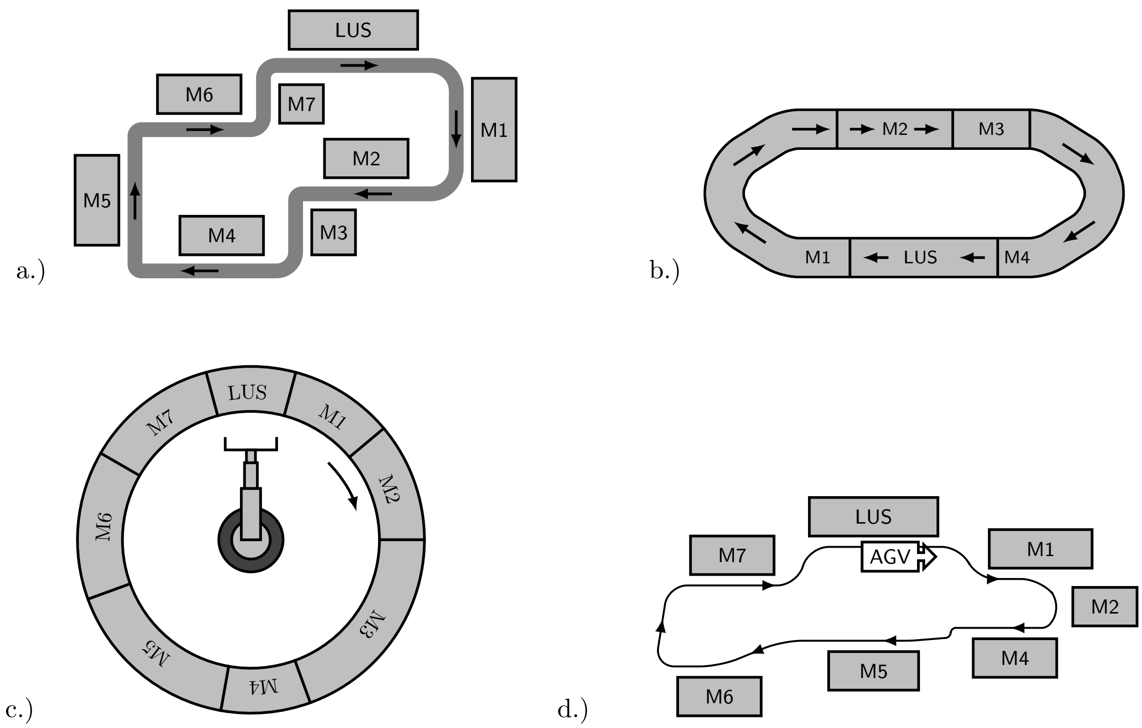


Figure 2.3: In a.) and b.) a conveyor system moves in a closed-loop rail in one direction transporting material among the machines. In c.) a material-handling industrial robot rotates in clockwise direction and single loop AGVs transport material between the machines in d.).

Kiran and Karabati [48] 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 `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.

In this thesis, we show that the `DCFLP` can in fact be formulated as a Linear Ordering Problem (`LOP`), which can be defined as follows. Given a $n \times n$ matrix $W = (w_{ij})$ of integers, find a simultaneous permutation π of the rows and columns of W such that

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

is maximized. Equivalently, w_{ij} can be interpreted as weights of a complete directed graph G with vertex set $V = [n]$. A tournament is made up of a subset of arcs of G containing for each pair of nodes i and j exactly one of the arcs (i, j) or (j, i). Then the LOP seeks to find an acyclic tournament in G, i.e., a tournament without directed cycles, of maximum total edge weight. There exist several high-quality exact methods and heuristics for the LOP. Hence, the DCFLP can be solved very efficiently by applying these approaches. In particular, we apply two exact approaches, a Semidefinite Program and an Integer Linear Program, as well as two fast heuristics, namely a Tabu Search and a Variable Neighborhood Search, for tackling the DCFLP.

Next let us consider the well-known Single-Row Facility Layout Problem (SRFLP) that is closely related to the DCFLP. It arises in the context of ordering machines on a production line where the material flow is handled by an AGV travelling in both directions on a straight-line path [38]. An SRFLP-instance consists of n one-dimensional machines with positive lengths $\ell_1, \dots, \ell_n$, and pairwise connectivities c_{ij}. The optimization problem can be written down as

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

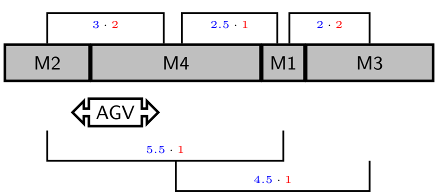
where Π_n is the set of permutations of the machines $[n]$ and ζ^{π}_{ij} is the center-to-center distance between machines i and j with respect to a particular permutation $\pi \in \Pi_n$. It is possible to obtain strong lower bounds and even optimal solutions for reasonable sized SRFLP-instances. The strongest available exact approaches are a LP-based cutting plane algorithm that uses betweenness variables [6] and a SDP approach modelled with products of ordering variables [43]. In summary we suggest a model for the DCFLP that is significantly easier (linear terms instead of linear-quadratic terms in ordering variables) than all available formulations for the SRFLP that to date is generally considered as the facility layout problem with the easiest structure.

Despite its practical relevance for the design of flexible manufacturing systems [52], the DCFLP is a very interesting problem from an academical point of view as it generalizes several well-discussed layout problems from literature. The DCFLP is a generalization of the Directed Circular Arrangement Problem (DCAP), which allows only machines with the same lengths. The DCAP was first considered by Liberatore [53] who showed that the problem is NP-hard (hence also the DCFLP is NP-hard). We refer to Bar-Noy et al. [12], Liberatore [53] and Naor and Schwartz [65] for several interesting applications of the DCAP in the areas of server design and ring networks. The DCAP is related to the Linear Arrangement problem (LA), which is a specialization of the SRFLP that considers machines with uniform lengths. The NP-hard [28] LA problem was originally proposed by Harper [33, 34] to develop error-correcting codes with minimal average absolute errors. Since then it was applied to VLSI design [70], single machine job scheduling [1, 67] and computational biology [46, 63].

Furthermore the DCFLP is related to the NP-hard [49] Unidirectional Cyclic Layout Problem (UCFLP) [3]. The UCFLP also considers a circular material handling system with directed flow and the objective is to find an assignment of n machines to n predetermined candidate locations such that the total handling cost is 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 difficult to solve [54]. Therefore optimizing circular layouts was so far considered to be clearly harder than optimizing row layouts. In this thesis we aim to reveal that this is not true, if we determine circular layouts with the help of the DCFLP (formulated as a LOP) instead of the UCFLP (formulated as a QAP).
Finally 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 have given the pairwise connectivities $c_{12} = c_{14} = c_{34} = 1,\ c_{13} = c_{24} = 2$ for the SRFLP and pairwise flows $f_{12} = f_{14} = f_{43} = 1,\ f_{13} = f_{42} = 2$ for the DCFLP. Figure 2.4 depicts the optimal layouts and the corresponding costs for both problems.

a.) 

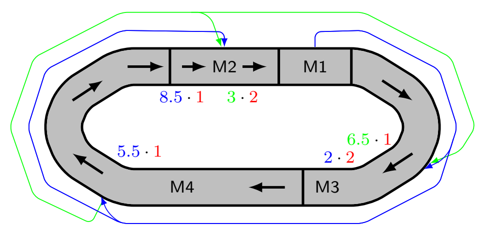
b.) 

Figure 2.4: We have 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 $2 \cdot 2 + 3 \cdot 2 + 5.5 \cdot 1 + 8.5 \cdot 1 + 6.5 \cdot 1 = 30.5$.

In summary the main contributions of this thesis are the following:

1. We prove that the DCFLP can be formulated as a LOP.
2. Building on this formulation we demonstrate that both heuristic and exact methods for the DCFLP are less complex in ordering variables and hence easier 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 provide strong lower and upper bounds for all proposed instances.

Chapter 3

Mathematical Formulation

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

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

A feasible ordering of machines on the circle has to fulfill the well-known 3-cycle inequalities:

$$0 \leq x_{ij} + x_{jk} - x_{ik} \leq 1, \quad i,j,k \in [n],\ i < j < k. \tag{3.1}$$

The 3-cycle inequalities together with integrality conditions on the variables suffice to describe feasible orderings.

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. This arrangement is in fact a single-row layout of the machines that is defined through the ordering variables. Hence we can compute d_{ij} as the difference of the sums of the lengths of the machines arranged left of machine i and machine j respectively:

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

The d_{ij} are linear expressions in the ordering variables x_{ij}. To destroy symmetry and reduce the dimensions of the problem, we fix machine 1 to be first in the ordering. Hence we set $x_{1j} = 1,\ j \in$

$[n],\ j \neq 1$. After some additional simplifications, we can state the distance variables as follows:

$$\begin{aligned} d_{1j} &= \frac{\ell_1 - \ell_j}{2} - \sum_{\substack{k\in[n]\\ 1\le k<j}} \ell_k x_{kj} - \sum_{\substack{k\in[n]\\ k>j}} \ell_k (1-x_{jk}), \quad j\in[n],\ j\neq 1 \\ d_{ij} &= \left(\frac{\ell_i}{2} + \sum_{\substack{k\in[n]\\ 1<k<i}} \ell_k x_{ki} + \sum_{\substack{k\in[n]\\ k>i}} \ell_k (1-x_{ik}) \right) \\ &\quad - \left(\frac{\ell_j}{2} + \sum_{\substack{k\in[n]\\ 1<k<j}} \ell_k x_{kj} + \sum_{\substack{k\in[n]\\ k>j}} \ell_k (1-x_{jk}) \right), \quad i,j\in[n],\ 1<i<j \end{aligned} \tag{3.2}$$

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

$$\begin{aligned} z_{1j} &= -d_{1j}, & z_{j1} &= L + d_{1j}, \quad j\in[n],\ j\neq 1, \\ z_{ij} &= -d_{ij} + (1-x_{ij})L, & z_{ji} &= d_{ij} + x_{ij}L, \quad i,j\in[n],\ 1<i<j, \end{aligned} \tag{3.3}$$

where $L = \sum_{k\in[n]} \ell_k$ denotes the sum of the lengths of all machines. Now we can rewrite the objective function (2.1) with the help of (3.3) as a linear function in $dim := \binom{n-1}{2}$ ordering variables:

$$\min_{x\in\{0,1\}^{dim}} f(x) \tag{3.4}$$

where

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

and x is a vector collecting all the ordering variables.

In summary we obtained the following `DCFLP` formulation, which is based on ordering variables:

Theorem 3.1 *The Linear Ordering Problem* (3.4) *subject to* (3.1) *and* (3.2) *is equivalent to the* `DCFLP`.

Proof. It suffice to induce a feasible layout on the circle with the help of the 3-cycle inequalities (3.1) together with the integrality conditions on x. Equations (3.2) connect the ordering with the distance variables and finally due to the definition of the objective function the distances between the machines are calculated correctly and weighted with the appropriate flows. □

Using (3.4) and (3.2) we can calculate weights $w_{ij}, i,j \in [n],\ 1 < i < j$, for each pair of machines and rewrite the problem as a standard `LOP` formulation:

$$\begin{aligned} \max & \sum_{i,j\in[n],\ 1<i<j} w_{ij} x_{ij} \\ \text{s.t.} & \quad (3.1). \end{aligned} \tag{3.5}$$

We close this chapter by pointing out the connection of the LOP to the SRFLP. The SRFLP is modelled 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

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

In the following chapters we suggest several methods to compute strong feasible layouts as well as lower bounds for the DCFLP.

Chapter 4

Computing Feasible Layouts

There exist many and several well-performing heuristics for the `LOP`. Chenery and Watanabe [19] suggested the first heuristic for the `LOP` based on construction rules in 1958. Further construction heuristics are proposed by Becker and Aujac [11]. Generally, construction heuristics perform bad in practice, hence it is beneficial to look for improvement possibilities after the construction of a feasible ordering. It is possible to compute orderings close to the optimum and even optimal solutions with the help of local improvement methods. However, they are even more important as a powerful concept for the design of metaheuristics. Despite of local search, that uses improvements based on exchanges of objects, the Kernighan-Lin improvements [47] and the heuristic of Chanas and Kobylanski [18] are the most common improvement methods for the `LOP`. In all relevant heuristics for both the `LOP` and the `SRFLP`, insert moves form the basic mechanism to move from one solution to another. However the costs of these insert moves differ significantly on the two considered problems. An insertion for the `LOP` can be done in $O(n)$ time because only the costs in row and column of the moved element have to be considered in the corresponding weight matrix. An insert move for the `SRFLP` needs $O(n^2)$ time because in general it is necessary to view all entries of the cost matrix. Due to that, we assume that heuristics for the `DCFLP` obtain stronger solutions with less computational effort than heuristics for the `SRFLP`. This claim is also supported by the extensive experiments conducted in the literature: While for the `LOP` high quality solutions for different types of instances with up to 500 objects are obtained within seconds by various heuristics [57] (see also below), the strongest heuristics for the `SRFLP` need up to a few hundred seconds on instances with only 80 objects (on faster computers) to obtain solutions of about the same solution quality (see e.g., [22]).

Metaheuristics can be described as 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 [30]. Martí and Reinelt [57] tested all known metaheuristics for the `LOP` on a large variety of benchmark instances with up to 500 objects and the best ones are (in descending order of their rank value obtained by a non-parametric Friedman test) the following:

1. Genetic and Memetic Algorithms [26, 61, 68],
2. Tabu Search [31, 51],
3. Variable Neighborhood Search [27, 64],

4. Scatter Search [16, 29] and the
5. Greedy Randomized Adaptive Search Procedure [16, 24].

All of the above methods produce similar results with respect to solution quality. Due to the results in [59], we can assume that the implementation details, especially the incremental computation of the move value, are more important then the choice of a specific method. This supports our reasoning that heuristics for the LOP compute stronger feasible layouts than heuristics for the SRFLP.

There are no black-boxes or codes of metaheuristics for the LOP (and the SRFLP) 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 two of the best and simplest ones: The Tabu Search (TS) [31, 51] and the Variable Neighborhood Search (VNS) [27, 64] heuristic. For both methods (see also Martí and Reinelt [58]) all relevant implementation parameters are given in the papers and the fine-tuning process can be conducted easily.

Glover and Laguna [31] complemented the basic TS procedure with a long-term diversification based on the REVERSE operation proposed by Chanas and Kobylanski [18]. The long-term strategy incorporates frequency information recorded during the application of the short-term phase.

The basic (VNS) for the LOP introduced by García-González et al. [27] 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. Afterwards a local search method is applied to improve the solution.

Due to clear descriptions of both TS and VNS, we could facilely reproduce the results of the computational experiments presented in the respective papers. Easy reproducibility of state-of-the-art approaches is very important for practitioners looking for efficient layouts in their production plants. We can conclude from the extensive computational studies in the literature that high-quality DCFLP layouts for instances with up to 500 machines can be obtained by state-of-the-art heuristics for the LOP. In our computational study we additionally want to compare the best heuristics for the LOP and the SRFLP on existing facility layout instances. In the next chapter we propose two exact algorithms, one based on Semidefinite Programming and an Integer Linear Programming approach, for obtaining tight lower bounds to the DCFLP.

Chapter 5

Computing Lower Bounds

There exist several well-performing exact algorithms for the LOP, like a Branch-and-Bound algorithm proposed by Kaas [45], a Branch-and-Cut algorithm suggested by Grötschel et al. [32] or a combined interior-point cutting-plane algorithm proposed by Mitchell and Borchers [62]. The working group of Reinelt in Heidelberg developed the current state-of-the-art Branch-and-Cut algorithm for the LOP [59]. The algorithm is able to solve specific instances with up to 150 objects, but fails on other instances with only 50 objects. In this thesis we use a Semidefinite Programming (SDP) approach [39, 40] that has proved to be competitive with the Branch-and-Cut algorithm for the LOP by the working group of Reinelt (see [42] for a detailed analysis of the strengths and weaknesses of both approaches) and is the current state-of-the-art approach for the SRFLP [43] (and other ordering problems [20, 21]). Additionally, we use an Integer Linear Programming (ILP) formulation for solving the standard LOP formulation, see (3.5) in Chapter 4, in order to obtain strong lower bounds for the LOP.

Let us give a description of our SDP approach [39, 40]. SDP extends Linear Programming (LP) to linear optimization over the cone of symmetric positive semidefinite matrices. LP problems are a special case of SDP problems, which consist only of diagonal matrices. A general (primal) SDP can be formulated as the following optimization problem:

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

where the data matrices $A_i, \ i \in \{1, \ldots, m\}$ and C are symmetric. The handbooks [7, 71] give an comprehensive overview of theory, algorithms, software and practical applications of SDP.

In our approach we solve an SDP relaxation of the DCFLP, which uses a bundle method [25] in combination with interior point methods [37]. The corresponding fractional solutions represent lower bounds for the DCFLP.

The fractional solutions can be exploited to provide upper bounds, i.e., feasible layouts of the machines on the circle, with the help of a rounding strategy (see [43] for more details). This results in obtaining feasible solutions together with a certificate on how close this solutions could be to the optimum.

We need the following binary variables $y_{ij},\ i,j \in [n],\ i<j$, for modelling the DCFLP as a SDP:

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

For these variables we can rewrite Theorem 3.1 as follows:

Corollary 5.1 *The problem*

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

subject to:

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

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

and the 3-cycle inequalities

$$-1 \le y_{ij} + y_{jk} - y_{ik} \le 1, \quad i<j<k \in [n], \tag{5.1}$$

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 Chapter 3 and ensure that the distances between machines are computed correctly. It is well-known that the 3-cycle inequalities (5.1) 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

$$\mathcal{P}_{LQO} := \operatorname{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top : y \in \{-1,1\}, y \text{ satisfies } (5.1) \right\}.$$

In the following, standard techniques are used to provide SDP relaxations. First, the nonconvex equation $Y - yy^\top = 0$ is relaxed to the positive semidefinite constraint, i.e.,

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

Furthermore, the main diagonal entries of Y represent squared $\{-1,1\}$ variables, hence $\mathrm{diag}(Y) = e$, i.e., the all-ones vector. For simplifying notation we introduce

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

where $\dim(Z) = \binom{n}{2} + 1 =: \Delta$. $Y - yy^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$ is implied by the Schur complement [13, Appendix A.5.5]. We therefore conclude that $\mathcal{P}_{LQO}$ is contained in the elliptope

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

Next, the constraints on y have to be reformulated as quadratic conditions in order to express them in terms of 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

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \quad i,j,k \in [n],\ i < j < k. \tag{5.2}$$

In [15] it is shown that these 3-cycle equations formulated in the $\{0,1\}$ model[1] describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. The 3-cycle inequalities are implicitly ensured by the 3-cycle equations together with $Z \succcurlyeq 0$ [39, Proposition 4.2].

In Theorem 5.1 we formulate the `DCFLP` as a `SDP` in binary variables.

Theorem 5.1 *The* `DCFLP` *is equivalent to the following formulation:*

$$\min\ \{\ K + \langle C_Z, Z\rangle : Z \text{ satisfies } (5.2),\ Z \in \mathcal{E},\ y \in \{-1,1\}\ \}$$

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

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

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

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

Proof. Since $y_i^2 = 1,\ i \in \{1,\dots,\Delta-1\}$ we have $\mathrm{diag}(Y - yy^\top) = 0$, which together with $Y - yy^\top \succcurlyeq 0$ shows that in fact $Y = yy^\top$ is integral. The 3-cycle equations (5.2) 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. □

The following basic semidefinite relaxation of the `DCFLP` is obtained when removing the integrality condition on the first row and column of Z:

$$\min\ \{K + \langle C_Z, Z\rangle : Z \text{ satisfies } (5.2),\ Z \in \mathcal{E}\ \}. \tag{SDP$_{\texttt{basic}}$}$$

[1] Helmberg [36] showed that bounds and structural properties are preserved when switching between $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems.

There are several ways to tighten $\texttt{SDP}_{\texttt{basic}}$. We consider two ways that have been successfully applied to the SRFLP.

First, we observe that Z is generated as the outer product of the vector $\begin{pmatrix}1 & y\end{pmatrix}$ which holds only $\{-1, 1\}$ entries in the non-relaxed SDP formulation. Therefore, each feasible solution of the DCFLP belongs to the metric polytope $\mathcal{M}$ which can be defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets:

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

Lovász and Schrijver [55] proposed a second class of strengthening constraints. They suggest to multiply the 3-cycle inequalities

$$1 - y_{ij} - y_{jk} + y_{ik} \geq 0, \qquad 1 + y_{ij} + y_{jk} - y_{ik} \geq 0, \tag{5.3}$$

by the nonnegative expressions

$$1 - y_{lo} \geq 0, \qquad 1 + y_{lo} \geq 0, \qquad l, o \in [n],\ l < o. \tag{5.4}$$

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

$$\begin{aligned} -1 - y_{lo} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lo} + y_{jk,lo} - y_{ik,lo} \leq 1 + y_{lo}, \qquad & i, j, k, l, o \in [n], \\ -1 + y_{lo} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lo} - y_{jk,lo} + y_{ik,lo} \leq 1 - y_{lo}, \qquad & i < j < k,\ l < o. \end{aligned} \tag{5.5}$$

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

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (5.5)\ \}. \tag{5.6}$$

In summary, the following semidefinite relaxation of the DCFLP is obtained:

$$\min \left\{ K + \langle C_Z, Z \rangle : Z \text{ satisfies } (5.2), Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \right\}. \tag{$\texttt{SDP}_{\texttt{strong}}$}$$

A lot of the facets which are used for separation in Branch-and-Cut approaches for the LOP are implicitly included in $\texttt{SDP}_{\texttt{strong}}$ [39, Proposition 4.2]. Hence it is not surprising that $\texttt{SDP}_{\texttt{strong}}$ yields essentially stronger bounds, of course at higher expenses, than Linear Programming relaxations in practice.

In the next chapter we state the results of our exact and heuristic approaches for solving the DCFLP.

Chapter 6

Computational Experiments

The aims of the following computational study are the following:

1. Building a benchmark library in order to facilitate further research on the `DCFLP`.
2. Provide competitive lower and upper bounds for the instances in our benchmark library.
3. Show that existing heuristics and exact approaches can easily be applied for this new way of modelling 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 heuristics and exact 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 the generation of reasonable data we use well-known benchmark instances for the `SRFLP` [4, 5, 6, 8, 9, 10, 43] and the `LA` problem [17, 23, 69] and modify them in order to obtain benchmark instances for the `DCFLP` and the `DCAP` respectively as follows: We propose two variants for adapting the flows. In the first variant, which we denote as *one-way*, we just set $f_{ij} := c_{ij}$ and $f_{ji} := 0$ and in *random* variant we decide with equal probability for both cases if $f_{ij} := c_{ij}, f_{ji} := 0$ or $f_{ij} := 0, f_{ji} := c_{ij}$. The lengths of the machines are transferred without changes. To destroy symmetry and reduce the dimensions of the problem, we fix one machine to be first in the ordering. 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 remodelled by setting $f^n_{ij} := f_{ij} - \min(f_{ij}, f_{ji})$, $f^n_{ji} := f_{ji} - \min(f_{ij}, f_{ji})$ and adding the constant $\min(f_{ij}, f_{ji}) \cdot L$ to the objective function. Note that the additional constant in the objective function only reduces the relative gap.

Our variants for adapting the flows reflect the following two practical issues in FMSs:

- *one-way*: The material flow between the machines follow a natural structure. 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 are already processed by one ore 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 using TS and VNS described in Chapter 4 for computing feasible layouts and the exact algorithms based on SDP and ILP described in the previous chapter for obtaining tight lower bounds.

All computations were conducted on an 2.8 GHz Intel Core i7 with 16 GB RAM, running macOS. 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 [58]. The short-term phase is followed by applying the long-term diversification phase 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. 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 evaluation 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*.

In Tables 6.1 and 6.2 we summarize the results for up to 36 machines. The SDP and the ILP approach are able to solve all considered instances with up to 36 machines to optimality. Additionally both heuristics are able to find close to optimal layouts 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 considered instances.

In Tables 6.3, 6.4 and 6.5 we report the computational results for DCFLP instances between 40 and 80 machines. The *one-way* variant can be solved to optimality by SDP, ILP and VNS. 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 the larger instances the SDP approach returns better lower bounds than the ILP. Both heuristics return efficient results, both with respect to solving time and solution quality, for all considered instances. The VNS heuristic is faster than the TS heuristic. However, in some cases TS is able to find better layouts than VNS.

Tables 6.6 and 6.7 show 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 considered instances.

In summary, the computational study supports our assumption that the DCFLP can be solved very efficiently by applying exact and heuristic approaches developed for solving the LOP. Especially, the *one-way* variant proves to be easy to solve for both exact and heuristic approaches.

Instance	n	SDP one-way		SDP random		ILP one-way	ILP random
		Optimal layout	Time	Optimal layout	Time	Time	Time
P15	15	8284.00	2.40	8041.00	3.50	0.01	0.01
P17	17	12717.00	5.80	13105.00	21.10	0.01	0.02
P18	18	14450.50	6.80	14726.50	24.10	0.01	0.01
N25_01	25	6221.00	3:56	5921.00	3:54	0.01	2.69
N25_02	25	53933.50	1:00	53162.50	1:36	0.02	0.03
N25_03	25	34784.00	3:16	31698.00	1:23	0.11	0.04
N25_04	25	70468.50	1:19	69385.50	2:54	0.02	0.48
N25_05	25	22256.00	45.50	21540.00	1:05	0.03	0.03
N30_01	30	11067.00	5:00	10216.00	2:55	0.10	0.11
N30_02	30	30890.50	22:52	28194.50	6:16	5.29	0.28
N30_03	30	64275.00	6:21	62721.00	7:23	0.09	0.88
N30_04	30	84051.50	24:04	83023.50	6:51	0.60	1.29
N30_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 6.1: 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 or 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
P15	15	8284.00	0.00	0.10	8041.00	0.00	0.11	0.01	8041.00	0.00	0.02
P17	17	12717.00	0.00	0.14	13105.00	0.00	0.15	0.02	13105.00	0.00	0.02
P18	18	14450.50	0.00	0.15	14726.50	0.00	0.20	0.02	14726.50	0.00	0.03
N25_01	25	6227.00	0.10	0.47	5929.00	0.13	0.52	0.10	5921.00	0.00	0.10
N25_02	25	53933.50	0.00	0.49	53162.50	0.00	0.44	0.07	53162.50	0.00	0.10
N25_03	25	34866.00	0.24	0.48	31698.00	0.00	0.52	0.08	31698.00	0.00	0.10
N25_04	25	70468.50	0.00	0.55	69701.50	0.45	0.52	0.08	69385.50	0.00	0.11
N25_05	25	22256.00	0.00	0.50	21540.00	0.00	0.50	0.07	21540.00	0.00	0.12
N30_01	30	11086.00	0.17	0.76	10216.00	0.00	0.94	0.20	10216.00	0.00	0.19
N30_02	30	30982.50	0.30	0.84	28194.50	0.00	1.02	0.16	28194.50	0.00	0.17
N30_03	30	64278.00	0.00	0.95	62766.00	0.07	0.96	0.17	62721.00	0.00	0.24
N30_04	30	84076.50	0.03	0.89	83110.50	0.10	1.02	0.14	83023.50	0.00	0.20
N30_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 6.2: 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 or VNS respectively.

Instance	n	SDP one-way		SDP random			
		Optimal layout	Time	Best lower bound	Best layout	Gap (%)	Time
N40_1	40	154285.50	1:43:12	148925.50	151300.50	1.59	3:44:15
N40_2	40	135486.00	16:15	144880.00	146971.00	1.44	3:55:13
N40_3	40	118601.50	1:22:06	120570.50	120754.50	0.15	3:38:52
N40_4	40	115049.00	5:40:28	118906.00	120042.00	0.96	4:07:58
N40_5	40	153266.00	1:11:50	145625.00	145744.00	0.08	3:39:24
sko49-1	49	50697.00	35:47	50010.00	51031.00	2.04	8:25:36
sko49-2	49	573272.00	1:01:02	574750.00	588591.00	2.41	8:29:54
sko49-3	49	446309.00	50:33	444335.00	457742.00	3.02	8:50:43
sko49-4	49	330308.50	1:09:11	322226.50	329119.50	2.14	8:30:50
sko49-5	49	896556.00	1:34:58	881914.50	904923.50	2.61	8:32:15
AKV-60-01	60	2042653.00	4:20:38	2479392.00	2608666.00	5.21	10:22:20
AKV-60-02	60	1224508.00	5:27:32	1345517.00	1406480.00	4.53	11:07:06
AKV-60-03	60	927361.50	4:20:59	1104305.00	1148412.00	3.99	10:34:24
AKV-60-04	60	610184.00	8:38:01	663133.00	680905.00	2.68	9:48:47
AKV-60-05	60	451681.00	4:11:27	576075.00	591985.00	2.76	10:15:09
sko64-1	64	122479.00	6:33:07	118187.00	123706.00	4.67	16:27:22
sko64-2	64	919477.50	10:11:12	913297.00	951454.00	4.18	15:31:21
sko64-3	64	553264.50	6:50:30	587192.50	610927.50	4.04	15:48:09
sko64-4	64	394514.00	6:06:59	389265.00	407336.00	4.64	15:33:45
sko64-5	64	679565.50	9:27:46	686896.00	717466.00	4.45	15:31:38
AKV-70-01	70	2185508.00	14:38:14	2670853.00	2797161.00	4.73	23:42:27
AKV-70-02	70	2082572.00	14:00:41	2533911.00	2674120.00	5.53	30:03:37
AKV-70-03	70	2207952.50	19:30:29	2395187.50	2532646.50	5.74	36:09:58
AKV-70-04	70	1420836.00	27:38:48	1557019.50	1634485.50	4.98	33:47:29
AKV-70-05	70	6023331.50	16:49:54	7252641.50	7667993.50	5.73	31:38:56
sko72-1	72	168547.00	15:00:02	171274.00	179425.00	4.76	37:14:22
sko72-2	72	918687.00	14:10:09	980256.00	1038406.00	5.93	30:37:27
sko72-3	72	1470058.50	20:12:00	1617738.50	1697744.50	4.95	37:31:45
sko72-4	72	1255362.50	26:08:17	1368468.50	1443741.50	5.50	33:42:30
sko72-5	72	542872.50	11:52:34	563290.50	595530.50	5.72	31:08:31
AKV-75-01	75	3457598.50	17:48:21	4030236.50	4216271.50	4.62	40:21:01
AKV-75-02	75	6170887.00	19:46:45	7285028.00	7664408.00	5.21	41:04:15
AKV-75-03	75	1895325.00	48:06:32	2182004.00	2316993.00	6.18	42:01:47
AKV-75-04	75	5496300.50	19:40:09	6670688.00	6946526.00	4.14	39:17:19
AKV-75-05	75	2480432.00	16:42:28	2844624.00	2990000.00	5.11	39:50:32
AKV-80-01	80	2837168.50	26:41:11	3544971.50	3723924.50	5.05	52:32:09
AKV-80-02	80	2785739.00	25:24:34	3198286.00	3370233.00	5.38	61:54:05
AKV-80-03	80	5113068.00	31:46:18	5963020.00	6240366.00	4.65	57:18:22
AKV-80-04	80	5695886.00	37:15:57	6488517.00	6856745.00	5.68	58:24:15
AKV-80-05	80	2247171.00	32:03:30	2461083.50	2610297.50	6.06	51:36:19

Table 6.3: Results produced by our SDP approach for DCFLP instances with between 40 and 80 machines. The bundle method for the SDP approach is restricted to 500 evaluations for the instances with 40 and 49 machines and to 250 evaluations for instances with between 60 and 80 machines. The running times are given in sec, min:sec or h:min:sec respectively. *Gap* indicates the relative gap between the best lower and upper bound determined by our SDP.

Instance	n	ILP one-way		ILP random			
		Optimal layout	Time	Best lower bound	Best layout	Gap (%)	Time
N40_1	40	154285.50	45.93	151300.50		0.00	16:36
N40_2	40	135486.00	0.51	146971.00		0.00	16:53
N40_3	40	118601.50	0.93	120754.50		0.00	24.61
N40_4	40	115049.00	15.21	120042.00		0.00	2:43
N40_5	40	153266.00	12.01	145744.00		0.00	28.78
sko49-1	49	50697.00	1.90	50660.00	51031.00	0.73	24:00:00
sko49-2	49	573272.00	1.83	582618.00	588591.00	1.01	24:00:00
sko49-3	49	446309.00	2.07	449703.00	459246.00	2.08	24:00:00
sko49-4	49	330308.50	2.54	325978.00	329119.50	0.95	24:00:00
sko49-5	49	896556.00	2.24	890138.00	904923.00	1.63	24:00:00
AKV-60-01	60	2042653.00	1.59	2373755.00	2625986.00	9.61	24:00:00
AKV-60-02	60	1224508.00	2.50	1296861.00	1421067.00	8.74	24:00:00
AKV-60-03	60	927361.50	2.01	1084007.50	1151473.50	5.86	24:00:00
AKV-60-04	60	610184.00	6.57	656912.00	691079.00	4.94	24:00:00
AKV-60-05	60	451681.00	1.88	570072.00	592864.00	3.84	24:00:00
sko64-1	64	122479.00	11.17	11611.00	124693.00	6.48	24:00:00
sko64-2	64	919477.50	13.48	901642.50	957159.50	5.80	24:00:00
sko64-3	64	553264.50	7.08	578652.50	612339.50	5.50	24:00:00
sko64-4	64	394514.00	11.77	383678.00	415466.00	7.65	24:00:00
sko64-5	64	679565.50	9.48	678121.50	721638.50	6.03	24:00:00
AKV-70-01	70	2185508.00	6.33	2527338.00	2835266.00	10.90	24:00:00
AKV-70-02	70	2082572.00	3.71	2396326.00	2777651.00	13.70	24:00:00
AKV-70-03	70	2207952.50	5.66	2266829.50	2561982.50	11.50	24:00:00
AKV-70-04	70	1420836.00	9.35	1499856.00	1662331.00	9.77	24:00:00
AKV-70-05	70	6023331.50	2.50	6737883.00	7796281.50	13.60	24:00:00
sko72-1	72	168547.00	12.10	167078.00	181929.00	8.16	24:00:00
sko72-2	72	918687.00	12.84	950428.00	1078881.00	11.90	24:00:00
sko72-3	72	1470058.50	12.49	1573984.50	1758093.50	10.47	24:00:00
sko72-4	72	1255362.50	12.65	1331165.50	1461299.50	8.91	24:00:00
sko72-5	72	542872.50	14.55	547980.00	612858.50	10.06	24:00:00
AKV-75-01	75	3457598.50	5.74	3774286.50	4304658.50	12.30	24:00:00
AKV-75-02	75	6170887.00	4.23	6773297.00	7930903.00	14.60	24:00:00
AKV-75-03	75	1895325.00	2:07	2074423.00	2388249.00	13.10	24:00:00
AKV-75-04	75	5496300.50	5.56	6204747.50	7213379.50	14.00	24:00:00
AKV-75-05	75	2480432.00	6.24	2677171.00	3074658.00	12.90	24:00:00
AKV-80-01	80	2837168.50	6.65	3301169.50	3896864.50	15.30	24:00:00
AKV-80-02	80	2785739.00	8.70	2988175.00	3494574.00	14.50	24:00:00
AKV-80-03	80	5113068.00	7.16	5508595.50	6540966.00	15.78	24:00:00
AKV-80-04	80	5695886.00	7.37	5989385.00	7175081.00	16.52	24:00:00
AKV-80-05	80	2247171.00	26.42	2332120.00	2749578.00	15.18	24:00:00

Table 6.4: Results produced by our ILP approach for DCFLP instances with between 40 and 80 machines. We set a time limit of 24 hours for the ILP approach. The running times are given in sec, min:sec or h:min:sec respectively. *Gap* indicates the relative gap between the best lower and upper bound determined by our 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
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 6.5: Results produced by our heuristics for DCFLP instances with between 40 and 80 machines. The running times are given in sec. *Gap* indicates the relative gap between the best lower bound of our SDP and ILP approach and the upper bound determined by TS or VNS respectively.

Instance	n	Density	SDP one-way		SDP random			ILP one-way	ILP random	
			Optimal layout	Time	Best lower bound	Best layout	Time	Time	Optimal layout	Time
can_24	24	0.246	311.00	26.50	379.00		43.2	0.02	379.00	0.03
fidap005	27	0.358	702.00	48.10	981.00		1:10	0.06	981.00	0.06
pores_1	30	0.236	383.00	1:34	679.00		2:15	0.05	679.00	0.12
ibm32	32	0.181.00	638.00	24:34	648.00		10:54	0.35	648.00	1.59
bcspwr01	39	0.062	131.00	16:07	129.00		10:35	0.19	129.00	0.21
fidapm05	42	0.277	1989.00	14:31	2745.00		22:07	0.54	2745.00	1.20
bcspwr02	49	0.050	218.00	58:23	190.00		46:13	1.44	190.00	0.98
will57	57	0.079	571.00	3:08:23	1120.00		2:42:52	8.27	1120.00	3.51
dwt_59	59	0.060	327.00	3:01:50	811.00		3:34:27	2.37	811.00	8.77
impcol_b	59	0.164	2659.00	3:50:44	4193.00	4346.00	21:00:32	1.99	4202.00	16:34
can_62	62	0.041	271.00	5.55:29	310.00		4:59:59	3.89	310.00	13.20
gd95c	62	0.076	796.00	2:58:02	1370.00		6:49:52	2.23	934.00	17.43
dwt_66	66	0.059	346.00	9:24:23	1025.00		6:41:40	5.52	1025.00	8.86
dwt_72	72	0.029	175.00	20:50:14	234.00	236.00	59:05:46	1.41	236.00	39:57
can_73	73	0.057	1304.00	41:55:45	1572.00	1579.00	57:04:44	67.13	1574.00	6:11
tub100	100	0.029	246.00	222:20:27	934.00		226:34:24	40.34	934.00	3:25

Table 6.6: Results produced by our exact methods for DCAP instances with up to 100 machines. We restricted the bundle method to 500 function evaluations. The running times are given in sec, min:sec or 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
bcspwr01	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
bcspwr02	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 6.7: 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 of our SDP and ILP approach and the upper bound determined by TS or VNS respectively.

Chapter 7

Conclusion

In this thesis we analyzed a new modelling approach for cyclic layouts, the so-called Directed Circular Facility Layout Problem (DCFLP). The DCFLP allows for a great variety of applications and can be solved very efficiently, as it can be modelled as a special Linear Ordering Problem (LOP).

We applied two heuristic approaches, namely a Tabu Search and a Variable Neighborhood Search and additionally we solved the problem with a Semidefinite and an Integer Linear Programming approach. Finally, we generated an extensive benchmark library for the DCFLP and provided strong lower and upper bounds for the considered instances.

An extended abstract covering the key results from this thesis has been published in proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management [44]. Further we will submit an extended version of this abstract to an international journal in the near future.

Chapter 8

Outlook

As a future research topic we suggest to combine and extend the very efficient heuristics and the exact methods for the Directed Circular Facility Layout Problem and the Single Row Facility Layout Problem in order to solve the Combined Cell Layout Problem (`CCLP`). The `CCLP` tries to minimize the material-handling costs in a cellular manufacturing system with two or more cells where materials occur, which require processing in more than one cell. The alignment of the machines in each cell can follow a linear or a circular structure. Therefore, the `CCLP` allows to model more complex layout types while building on well-studied and efficiently solvable basic problems. Hungerländer and Anjos [41] give lower bounds for reasonably sized instances of the `CCLP` to assess heuristic approaches.

Furthermore, it would be interesting to adapt the heuristics and the exact approaches for the Linear Ordering Problem and therefore for the Directed Circular Facility Layout Problem with dynamic and stochastic aspects as well as the ability to consider multiple objectives.

Bibliography

[1] D. Adolphson. Singlemachine job sequencing with precedence constraints. *SIAM Journal on Computing*, 6:40–54, 1977.

[2] P. Afentakis. A loop layout design problem for flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 1:175–196, 1989.

[3] I. K. Altınel and T. Öncan. Design of unidirectional cyclic layouts. *International Journal of Production Research*, 43(19):3983–4008, 2005.

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

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] H. Aujac. La hiérarchie des industries dans un tableau des échanges interindustriels. *Revue économique*, 11(2):169–238, 1960.

[12] A. Bar-Noy, J. S. Naor, and B. Schieber. Pushing dependent data in clients-providers-servers systems. *Wireless Networks*, 9(5):421–430, 2003.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[14] Y. Bozer and S. Rim. Exact solution procedures for the circular layout problem. Technical report, University of Michigan, USA, Report No. 89-33, 1989.

[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] V. Campos, F. Glover, M. Laguna, and R. Martí. An experimental evaluation of a scatter search for the linear ordering problem. *Journal of Global Optimization*, 21(4):397–414, 2001.

[17] A. Caprara and J.-J. Salazar-González. Laying out sparse graphs with provably minimum bandwidth. *INFORMS Journal on Computing*, 17:356–373, 2005.

[18] S. Chanas and P. Kobylanski. A new heuristic algorithm solving the linear ordering problem. *Computational Optimization and Applications*, 6(2):191–205, 1996.

[19] H. Chenery and T. Watanabe. International comparisons of the structure of production. *Econometrica*, 26:487–521, 1958.

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

[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] 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.

[23] J. Díaz, J. Petit, and M. Serna. A survey of graph layout problems. *ACM Computing Surveys*, 34:313–356, 2002.

[24] T. A. Feo and M. G. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[25] 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.

[26] J. Frenzel. Genetic algorithms. *Potentials, IEEE*, 12(3):21–24, 1993.

[27] C. García-González, D. Pérez-Brito, V. Campos, and R. Marti. Variable neighborhood search for the linear ordering problem. 33:3549–3565, 12 2006.

[28] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.

[29] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977.

[30] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13(5):533–549, 1986.

[31] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, 1997.

[32] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Reasearch*, 32(6):1195–1220, 1984.

[33] L. H. Harper. Optimal assignments of numbers to vertices. *SIAM Journal on Applied Mathematics*, 12:131–135, 1964.

[34] L. H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1:385–393, 1966.

[35] M. M. D. Hassan. Machine layout problem in modern manufacturing facilities. *International Journal of Production Research*, 32(11):2559–2584, 1994.

[36] C. Helmberg. Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3):952–969, 2000.

[37] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM Journal on Optimization*, 6:342–361, 1996.

[38] S. S. Heragu and A. Kusiak. Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268, 1988.

[39] P. Hungerländer. *Semidefinite Approaches to Ordering Problems*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2012.

[40] 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.

[41] P. Hungerländer and M. F. Anjos. An exact approach for the combined cell layout problem. In *Operations Research Proceedings 2012*, pages 275–281, 2013.

[42] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming*, 140(1):77–97, 2013.

[43] 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.

[44] P. Hungerländer, K. Maier, J. Pöcher, 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. doi: 10.1109/IEEM.2017.8290159.

[45] R. Kaas. A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, 8(4):355–362, 1981.

[46] R. M. Karp. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 278–285, New York, USA, 1993. ACM.

[47] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[48] A. S. Kiran and S. Karabati. Exact and approximate algorithms for the loop layout problem. *Production Planning & Control*, 4(3):253–259, Jan. 1993. ISSN 0953-7287, 1366-5871. doi: 10.1080/09537289308919443. URL https://www.tandfonline.com/doi/full/10.1080/09537289308919443.

[49] P. Kouvelis and M. W. Kim. Unidirectional loop network layout problem in automated manufacturing systems. *Operations Research*, 40:533–550, 1992.

[50] P. Kouvelis, W.-C. Chiang, and A. Kiran. A survey of layout issues in flexible manufacturing systems. *Omega*, 20(3):375–390, 1992.

[51] M. Laguna, R. Martí, and V. Campos. Intensification and diversification with elite tabu search solutions for the linear ordering problem. *Computer & Operations Research*, 26(12):1217–1230, 1999.

[52] J. Leung. Polyhedral structure and properties of a model for layout design. *European Journal of Operational Research*, 77(2):195–207, 1994.

[53] V. Liberatore. Circular arrangements. In P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy, editors, *Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 782–783. Springer Berlin / Heidelberg, 2002.

[54] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176 (2):657–690, 2007.

[55] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[56] B. Malakooti. Unidirectional loop network layout by a LP heuristic and design of telecommunications networks. *Journal of Intelligent Manufacturing*, 15(1):117–125, 2004.

[57] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*. Applied Mathematical Sciences. Springer, 2011.

[58] R. Martí and G. Reinelt. *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization*. Applied Mathematical Sciences. Springer Berlin Heidelberg, 2011. ISBN 9783642167294. URL https://books.google.at/books?id=4Lk8pPYYI3cC.

[59] R. Martí, G. Reinelt, and A. Duarte. A benchmark library and a comparison of heuristic methods for the linear ordering problem. *Computational Optimization and Applications*, 51 (3):1297–1317, 2011.

[60] 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.

[61] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, 1994.

[62] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In T. T. H. Frenk, K. Roos and S. Zhang, editors, *High Performance Optimization*, pages 349–366. Kluwer Academic Publishers, 2000.

[63] G. Mitchison and R. Durbin. Optimal numberings of an N N array. *SIAM Journal on Algebraic and Discrete Methods*, 7:571–582, 1986.

[64] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097 – 1100, 1997. ISSN 0305-0548. doi: https://doi.org/10.1016/S0305-0548(97)00031-2. URL http://www.sciencedirect.com/science/article/pii/S0305054897000312.

[65] J. S. Naor and R. Schwartz. The directed circular arrangement problem. *ACM Transactions on Algorithms*, 6:47:1–47:22, 2010.

[66] A. C. Nearchou. Meta-heuristics from nature for the loop layout design problem. *International Journal of Production Economics*, 101(2):312–328, 2006.

[67] R. Ravi, A. Agrawal, and P. Klein. Ordering problems approximated: Single-processor scheduling and interval graphs connection. In J. L. Albert, B. R. Artalejo, and B. Monien, editors, *18th International Colloquium on Automata, Languages and Programming*, volume 150 of *Lecture Notes in Computer Science*, pages 751–762. Springer-Verlag New York, 1991.

[68] T. Schiavinotto and T. Stützle. The linear ordering problem: Instances, search space analysis and algorithms. *Journal of Mathematical Modelling and Algorithms*, 3(4):367–402, 2005.

[69] H. Seitz. *Contributions to the Minimum Linear Arrangement Problem*. PhD thesis, University of Heidelberg, Germany, 2010.

[70] A. Vanelli and G. S. Rowan. An eigenvector based approach for multistack VLSI layout. In *Proceedings of the Midwest Symposium on Circuits and Systems*, volume 29, pages 135–139, 1986.

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