Global Approaches for Facility Layout and VLSI Floorplanning

Miguel F. Anjos[1] and Frauke Liers[2]

[1] Department of Mathematics and Industrial Engineering, École Polytechnique de
   Montréal, Montréal QC, H3C 3A7, Canada `anjos@stanfordalumni.org`
[2] Universität zu Köln, Institut für Informatik, Köln, Germany
   `liers@informatik.uni-koeln.de`

1 Introduction

The facility layout problem consists of partitioning a rectangular facility of
known dimensions into departments with a given (fixed) area so as to min-
imize the total cost associated with the (known or projected) interactions
between these departments. This cost is modeled as the weighted sum of the
center-to-center distances between all pairs of facilities, where the pairwise
weights reflect transportation costs and/or adjacency preferences between de-
partments. If the height and width of the departments vary, then finding their
optimal (rectangular) shapes is also a part of the problem. This is a hard prob-
lem, in particular because any desirable layout must have no overlap among
the areas of the different departments. Versions of the facility layout problem
occur in many environments, such as flexible manufacturing and service center
layout, as well as in other engineering applications, such as the design of Very
Large Scale Integration (VLSI) circuits. Nearly all of the resulting problems
are known to be NP-hard.

Two types of approaches capable of yielding provably optimal solutions
have been proposed in the literature. The first type are graph-theoretic ap-
proaches which assume that the desirability of locating each pair of facilities
adjacent to each other is known. Initially, the area and shape of the depart-
ments are ignored, and each department is simply represented by a node in a
graph. Adjacency relationships between departments can now be represented
by arcs connecting the corresponding nodes in the graph. The objective is then
to construct a graph which maximizes the weight on the adjacencies between
nodes. We refer the reader to Foulds [27] for more details. The second type
of these approaches are mathematical programming formulations with objec-
tive functions based on an appropriately weighted sum of centroid-to-centroid
distances between departments. Exact mixed-integer linear programming for-
mulations were proposed in Montreuil [54], Meller et al. [53], and Sherali
et al. [64]. Non-linear programming formulations include the work of Castillo

et al. [18, 19]. Most recently, Meller et al. [51] solved problems up to 11 departments to global optimality.

Thus, most of the approaches in the literature that tackle realistically sized problems are based on heuristics with no guarantee of optimality. These include genetic algorithms, tabu search, simulated annealing, fuzzy logic, and many others. Mathematical-programming-based heuristics can also be used to provide high-quality solutions for large problems with 20 or more departments [9, 17]. We refer the reader to the extensive bibliographies in the survey papers of Meller and Gau [52], Mavridou and Pardalos [50], and Singh and Sharma [67].

The survey of Meller and Gau [52] divides the research papers on facility layout into three broad areas. The first is concerned with algorithms for tackling the general layout problem as defined above. The second area is concerned with extensions of the problem in order to account for additional issues which arise in applications, such as designing dynamic layouts by taking time dependency issues into account; designing layouts under uncertainty conditions; and achieving layouts which optimize two or more objectives simultaneously. The third area is concerned with specially structured instances of the problem. One such special case that has been extensively studied occurs when all the facilities have equal dimensions and the possible locations for the departments are given a priori; this is the quadratic assignment problem (QAP) formulated by Koopmans and Beckman [42]. Since the possible locations are fixed, the problem reduces to optimize a quadratic objective over all possible assignments of departments to locations. The QAP is NP-hard, and is in general a difficult problem to solve. (Refer to chapter(s) on QAP/Comb Problems.)

In this chapter we will be concerned with two layout problems to which cone optimization approaches have been successfully applied. The first one is a specially structured instance of facility layout, namely the single-row facility layout problem (SRFLP). The SRFLP consists of arranging a given number of rectangular facilities next to each other along a line so as to minimize the total weighted sum of the center-to-center distances between all pairs of facilities. This problem is also known in the literature as the one-dimensional space allocation problem; see, e.g., [58]. The SRFLP also has several interesting connections to other known combinatorial optimization problems such as the maximum-cut problem, the quadratic linear ordering problem, and the linear arrangement problem. We explore some of these connections in Section 2.3. Note that for the SRFLP, numerous approaches based on linear programming (LP) have been proposed in the literature. Using integer LP methods, the most effective such approach for determining exact solutions is a branch-and-cut algorithm based on the recent model by Amaral [5].

The second problem arises from the application of facility layout to fixed-outline floorplanning in VLSI circuit design. Fixed-outline floorplanning consists of arranging a set of rectangular modules on a rectangular chip area so that an appropriate measure of performance is optimized, and is hence a spe-

cialized version of facility layout. The objective is typically to minimize total wire length.

The impact to date of the cone optimization approaches for facility layout problems, excluding its impact for the QAP addressed elsewhere in this book, can be summarized as follows:

- Anjos and Vannelli [10] used a semidefinite programming (SDP) relaxation to provide globally optimal solutions for SRFLP instances in the literature with up to 30 departments that had remained unsolved for nearly 20 years.
- Hungerländer and Rendl [38] show that the SDP relaxation, when augmented with valid inequalities and optimized using a suitable combination of algorithms, can provide global optimal solutions for SRFLP instances with up to 40 departments. Their approach can solve larger instances to optimality than the approach by Amaral [5].
- Anjos and Yen [11] obtained tight global bounds for SRFLP instances with up to 100 facilities using an alternate SDP relaxation. Their approach consistently achieves optimality gaps not greater than 5%.
- Takouda, Anjos and Vannelli [69] provided non-trivial lower bounds for the VLSI macrocell floorplanning problem. Furthermore, their second-order cone programming (SOCP) formulations (described in Section 3.1) for the area and aspect ratio constraints were key to the success of the two-stage convex optimization-based methodology for floorplanning of Luo et al. [49] and of the facility layout model of Jankovits [39], both of which provide computational results outperforming those previously reported in the literature.

This chapter provides an overview of the SOCP/SDP models underpinning this impact, and concludes with a summary of further research directions.

2 Introduction to the Single-Row Facility Layout Problem

An instance of the SRFLP is formally defined by n one-dimensional facilities with given positive lengths $\ell_1, \ldots, \ell_n$, and pairwise non-negative weights c_{ij}. The objective is to arrange the facilities so as to minimize the total weighted sum of the center-to-center distances between all pairs of facilities. If all the facilities have the same length and all the non-zero weights are equal, the SRFLP becomes an instance of the linear arrangement problem, see e.g. [46], which is itself a special case of the QAP; see e.g. [20]. (For a survey on the linear arrangement and other graph layout problems, we refer to [26].) Several practical applications of the SRFLP have been identified in the literature, such as the arrangement of departments on one side of a corridor in supermarkets, hospitals, or offices [65], the assignment of disk cylinders to files [58], the assignment of airplanes to gates in an airport terminal [68], and the arrangement of machines in flexible manufacturing systems, where machines

within manufacturing cells are often placed along a straight path travelled by an automated guided vehicle [36]. We refer the reader to the book of Heragu [34] for more information.

Several heuristic algorithms for the SRFLP have also been proposed. We point out the early work of Hall [30], the application of non-linear optimization methods by Heragu and Kusiak [37], the simulated annealing algorithms proposed independently by Romero and Sánchez-Flores [61] and Heragu and Alfa [35], a greedy heuristic algorithm proposed by Kumar et al. [43], and the use of both simulated annealing and tabu search by de Alvarenga et al. [23]. However, these heuristic algorithms do not provide a guarantee of global optimality, or an estimate of the distance from optimality.

Simmons [65] was the first to state and study the SRFLP, and proposed a branch-and-bound algorithm. His subsequent note [66] mentioned the possibility of extending the dynamic programming algorithm of Karp and Held [40] to the SRFLP, which was done by Picard and Queyranne [58]. Different mixed-integer LP models have been proposed [2, 3, 37, 48], and the first polyhedral study of the so-called distance polytope for SRFLP was carried out by Amaral and Letchford [6]. Most recently, the IP-based cutting-plane algorithm of Amaral [5] can compute optimal solutions for instances with up to 35 facilities within a few hours of computing time. The polyhedral structure of Amaral's model has recently been studied by Sanjeevi and Kianfar [63].

At the time of writing, the state-of-the-art in terms of globally optimal methods are the SDP-based approaches. The first SDP relaxation was proposed in [7] where global bounds for instances with up to 80 facilities were obtained using the spectral bundle solver SB [32, 33]. While this solver is able to handle very large SDP problems, a major drawback is that its convergence slows down significantly after several hours of computation. As a consequence, instances with 80 facilities were the largest for which bounds could be obtained in reasonable time in [7], and some of the optimality gaps were greater than 10%. Anjos and Yen [11] obtained bounds for instances with up to 100 facilities using the solver CSDP [14] and an alternate SDP relaxation. Their approach consistently achieves optimality gaps not greater than 5% for even extremely large instances of SRFLP. Most recently, Hungerländer and Rendl [38] were able to obtain globally optimal solutions for instances with up to 40 facilities.

In the following, we will describe LP- and SDP-based approaches for SR-FLP in more detail. It turns out that the models used are closely related to an important combinatorial optimization problem called the maximum-cut problem in an undirected graph. The latter asks for partitioning the nodes of a graph into two sets such that the sum of the weights of edges with nodes in different partitions is maximum. First, we introduce binary quadratic optimization problems which are well known to be equivalent to the maximum-cut problem. We then show that the convex hull of solutions feasible for SRFLP is a special instance of a quadratic linear ordering problem which is known to induce a face of a cut polytope. This gives some theoretical evidence that

solution approaches for SRFLP based on the maximum-cut problem can lead to effective algorithms. In fact, we describe how the most effective approaches for solving SRFLP instances exploit the underlying cut-polytope structure.

2.1 Binary Quadratic Optimization and the Maximum-Cut Problem

Let us consider some combinatorial optimization problem on a finite set E. We denote the feasible solutions by $\mathcal{I} \subseteq 2^E$. The objective function $c(I) = \sum_{e \in I} c_e$ is linear, where $c_e \in \mathbf{R}$ for all $e \in E$. Without loss of generality, we want to minimize $c(I)$ over all $I \in \mathcal{I}$. Let $P \subseteq \mathbf{R}^E$ denote a polytope with the property $x \in \{0,1\}^E \cap P$ if and only if x is the characteristic vector of a feasible solution.

The corresponding integer LP problem reads

$$
\text{(P)} \qquad
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & x \in P \\
& x \in \{0,1\}^E.
\end{aligned}
$$

In the following, we focus on objective functions that are quadratic in the variables x, i.e., we consider problems of the form

$$
\text{(QP)} \qquad
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e + \sum_{e,f \in E; e \neq f} c_{ef} x_e x_f \\
\text{s.t.} \quad & x \in P \\
& x \in \{0,1\}^E.
\end{aligned}
$$

For problems defined on a graph $G = (V, E)$ with variables corresponding to edges, and for two edges $e = \{i, j\}$ and $f = \{k, l\}$, we will use the notations c_{ef} and c_{ijkl} interchangeably.

In order to linearize (QP) using the standard linearization, we introduce a binary variable y_{ef} for each pair $\{e, f\}$ with $c_{ef} \neq 0$, modeling $x_e x_f$, along with the constraints $y_{ef} \leq x_e$, $y_{ef} \leq x_f$, and $y_{ef} \geq x_e + x_f - 1$.

The easiest example for (QP) is the quadratic unconstrained binary optimization problem (QUBO), where we optimize in (QP) over the unit hypercube without further constraints. It is well known that QUBO is equivalent to the maximum-cut problem [31, 24]. Given a graph $G = (V, E)$ with edge weights w_e and $W \subseteq V$, the cut $\delta(W)$ is defined as

$$
\delta(W) = \{\{u, v\} \in E \mid u \in W, v \notin W\} \,.
$$

Its weight is $\sum_{e \in \delta(W)} w_e$. The maximum-cut problem asks for a cut of maximum weight and is NP-hard for general graphs. The corresponding cut polytope, i.e., the convex hull of incidence vectors of cuts, is well studied [12, 25], and branch-and-cut implementations that are efficient in practice exist for its solution [45, 60].

In order to establish the equivalence of the maximum-cut problem with QUBO defined on m binary variables, we construct an auxiliary graph

$G_{\text{lin}} = (V_{\text{lin}}, E_{\text{lin}})$ with $m + 1$ nodes [31, 24]. G_{lin} contains a node for each linear variable x_e. For each non-zero quadratic term $x_e x_f$ in the objective, E_{lin} contains an edge between the nodes corresponding to x_e and x_f. Furthermore, an additional root node and edges from this node to all other nodes are introduced. Then there exists a simple linear transformation between the edge variables of G_{lin} in the maximum-cut setting and the linear variables and their products in the unconstrained quadratic optimization setting. Under this transformation, P is isomorphic to the cut polytope of G_{lin}.

If P is the unit hypercube in (QP), solving the resulting QUBO problem thus amounts to determine a maximum cut in G_{lin}, i.e., to optimize over a cut polytope defined in the $|E_{\text{lin}}|$-dimensional space. If P is a strict subset of the unit hypercube, i.e., if additional constraints are present, these constraints can be transformed as well and we derive that P is isomorphic to a cut polytope with further linear constraints. In particular, all inequalities valid for the cut polytope still yield valid inequalities for the linearized binary quadratic problem.

Clearly, intersecting the cut polytope with arbitrary hyperplanes yields in general a non-integer polytope. The structure of the convex hull of integer points in the resulting polytope can be very different from a cut polytope. In this case it is not clear whether the knowledge about the cut polytope can help solving the linearized constrained optimization problem. Some applications exist in which no speed-up is observed when designing a solution algorithm based on the cut polytope if the inequalities cut through the interior of the polytope. However, for the applications studied here and for several further relevant applications, the linear constraints cut out a *face* of the cut polytope. This is an important observation as then the convex hull of the feasible solutions inherits the cut-polytope structure. Therefore, approaches exploiting the cut-polytope structure can lead to effective solution algorithms. This connection has been exploited in the work by Buchheim et al. [15] where several different applications were studied for which the linear constraints always induce faces of some cut polytope. General separation methods for constrained quadratic problems of form (QP) were designed that can complement or replace detailed polyhedral studies of the polytope of the linearized problem and that can be used as a black box. One of these routines consists of exploiting the connection to the maximum-cut problem by separating inequalities known to be valid for the cut polytope. The methods were tested on the quadratic linear ordering problem and the linear arrangement problem, among others. We will introduce these applications in the next section. The main contribution in [15] was to show that these general approaches lead to a dramatic decrease of both the number of nodes in the enumeration tree and the running time when compared to an algorithm that only uses the standard separation routines for the well-studied polytope P from (QP).

In the next section, we introduce the quadratic linear ordering problem which induces a face of a cut polytope. Subsequently, we observe that SRFLP is a special quadratic linear ordering problem.

2.2 The Quadratic Linear Ordering Polytope as a Face of a Cut Polytope

Let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of $[n] := \{1, 2, \ldots, n\}$. The corresponding linear ordering variables $x = (x_{ij}) \in \mathbf{R}^{n(n-1)}$ for $i \neq j$ are defined as

$$x_{ij} = \begin{cases} 1, & \text{if } \pi_i < \pi_j \\ 0, & \text{if } \pi_j < \pi_i. \end{cases} \tag{1}$$

In practice, we can exploit $x_{ij} = 1 - x_{ji}$ and thus eliminate half of the variables and only keep those with $i < j$. Given costs $c_{ij} \in \mathbf{R}$ for each pair of facilities i, j, the linear ordering problem asks for optimizing a linear objective over the convex hull of all valid vectors x. This convex hull is called the linear ordering polytope P_{LO} [29, 59]. This is an NP-hard problem. For a recent survey on the linear ordering problem, we refer to the work by Charon and Hudry [21].

Allowing for products of linear ordering variables in the objective function, the corresponding quadratic linear ordering problem (QLO) in its general form is

$$(QLO) \quad \begin{array}{ll} \min & \sum_{(i,j,k,l) \in I} c_{ijkl} x_{ij} x_{kl} \\ \text{s.t.} & x \in P_{LO} \\ & x_{ij} \in \{0, 1\} \ \forall (i, j) \in J. \end{array}$$

The index set I consists of all quadruples (i, j, k, l) such that $x_{ij} x_{kl}$ occurs as a product in the objective function, while J is the set of all pairs (i, j) for which a linear ordering variable x_{ij} is needed. As $x_{ij}^2 = x_{ij}$ for any binary variable, (QLO) also takes linear terms into account.

In order to linearize the objective function, we introduce a new binary variable y_{ijkl} for each $(i, j, k, l) \in I$, modeling the product $x_{ij} x_{kl}$. (Note that $y_{ijkl} = y_{klij}$.) Applying the standard linearization, the corresponding linearized quadratic linear ordering problem (LQLO) can be written as

$$(LQLO) \quad \begin{array}{ll} \min & \sum_{(i,j,k,l) \in I} c_{ijkl} y_{ijkl} \\ \text{s.t.} & x \in P_{LO} \\ & x_{ij} \in \{0, 1\} & \forall (i, j) \in J \\ & y_{ijkl} \leq x_{ij}, x_{kl} & \forall (i, j, k, l) \in I \\ & y_{ijkl} \geq x_{ij} + x_{kl} - 1 & \forall (i, j, k, l) \in I \\ & y_{ijkl} \in \{0, 1\} & \forall (i, j, k, l) \in I. \end{array}$$

Buchheim et al. [16] introduced the above model for the so-called bipartite crossing minimization problem. Additionally, a quadratic reformulation of the constraints defining P_{LO} was given: it was shown that a 0/1 vector (x, y) satisfying $y_{ijkl} = x_{ij} x_{kl}$ is feasible for (LQLO) if and only if

$$x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} = 0 \ \forall i < j < k \text{ s.t. } (i, j, k, l) \in I \text{ for some } l. \tag{2}$$

Furthermore, the constraints (2) yield a minimum equation system for (LQLO). Equations (2) were used by Lewis et al. [44] in order to derive penalty

functions for the original linear ordering problem. Note that (LQLO) is a quadratic binary optimization problem where the feasible solutions need to satisfy further side constraints, namely those restricting the set of feasible solutions to linear orderings. By proving validity of the corresponding inequality $x_{ik} - y_{ijik} - y_{ikjk} + y_{ijjk} \geq 0$ for each of the equalities appearing in (2), it was shown in [16] that the hyperplanes (2) cut out faces of the cut polytope associated to graph G_{lin}. Exploiting this result, both IP- and SDP-based methods originally designed for maximum-cut problems were used in [16] to solve the quadratic linear ordering problem. It turned out that the SDP-based approach outperformed the IP-based techniques. In the recent work by Hungerländer and Rendl [38], equations (2) were also used for the quadratic linear ordering problem. The SDP-based method from [16] is improved by additionally separating matrixcuts that were introduced by Lovász and Schrijver in [47]. The method is used for different applications of quadratic linear ordering, the SRFLP being one of them. It turns out that the method can effectively solve very large instances to optimality.

2.3 The connection of SRFLP and Related Problems to the Quadratic Linear Ordering Problem

In the following, we present an integer LP formulation of SRFLP. It will turn out that SRFLP is a special instance of a quadratic linear ordering problem.

Let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of the indices $[n]$ of the facilities, so that the leftmost facility is π_1, the facility to the right of it is π_2, and so on, with π_n being the last facility in the arrangement. Given a permutation π and two distinct facilities i and j, the center-to-center distance between i and j with respect to this permutation is $\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j$, where $D_\pi(i,j)$ denotes the sum of the lengths of the facilities between i and j in the ordering defined by π. Solving the SRFLP consists of finding a permutation of the facilities that minimizes the weighted sum of the distances between all pairs of facilities. In other words, the problem is:

$$\min_{\pi \in \Pi_n} \sum_{i<j} c_{ij} \left[\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j \right],$$

where Π_n denotes the set of all permutations of $[n]$. Note that here and in all subsequent formulations of the SRFLP, the assumption that $c_{ij} \geq 0$ ensures that the facilities are placed next to each other, i.e., without holes in the arrangement.

Simmons [65] observed that the crux of the problem is to minimize $\sum_{i<j} c_{ij} D_\pi(i,j)$ over all permutations $\pi \in \Pi_n$. It is also clear that $D_\pi(i,j) = D_{\pi'}(i,j)$, where π' denotes the permutation symmetric to π, defined by $\pi'_i = \pi_{n+1-i}, i = 1, \ldots, n$. Hence, it is possible to simplify the problem by considering only the permutations for which, say, facility 1 is on the left half

of the arrangement. This type of symmetry-breaking strategy is important for reducing the computational requirements of most algorithms, including those based on LP or dynamic programming.

The SRFLP can be modeled as the optimization of a linear function over the distance polytope. This polytope is the convex hull of distance vectors $d \in \mathbf{R}_+^{\binom{n}{2}}$ that can be realized by some arrangement of objects. In [6] it is shown that the polytope lies in the cut cone, i.e., the convex cone generated by all incidence vectors of cuts. It was shown that several well-known facet-inducing inequalities for the latter also induce facets for the distance polytope. However, when solving the SRFLP this way, running times can grow quickly.

Alternatively, the sum of the lengths of the facilities between i and j in the ordering defined by π can be expressed in terms of products of linear ordering variables as follows. A facility k lies between i and j if and only if either i is before k and k is before j, or vice versa j is before k and k is before i. Thus, $D(i,j)$ can be written as $D(i,j) = \sum_k \ell_k x_{ik} x_{kj} + \sum_k \ell_k x_{jk} x_{ki}$, where x_{ik} is the usual linear ordering variable modeling whether $\pi_i < \pi_k$ or not. The objective function terms are then summed up appropriately, the resulting cost vector being called c'. Therefore, up to a constant, the SRFLP can be rewritten as a specific quadratic linear ordering problem of the form

$$
\text{(QLO}_2) \qquad
\begin{aligned}
\min \quad & \textstyle\sum_{i \neq j \neq k \neq i} c'_{ij} x_{ik} x_{kj} \\
\text{s.t.} \quad & x \in P_{LO} \\
& x_{ij} \in \{0,1\} \ \forall i,j \in \{1 \dots n\}, i \neq j.
\end{aligned}
$$

If all object lengths and all non-zero weights equal 1, the application (QLO$_2$) is called linear arrangement problem. We noted above that the quadratic linear ordering problem is isomorphic to a face of a cut polytope. As a consequence, the same is true for the SRFLP and the linear arrangement problem.

We note that SRFLP and linear arrangement in the formulation (QLO$_2$) only require products of linear ordering variables of the form $x_{ik} x_{kj}$, whose number is only $O(n^3)$. All other products $x_{ik} x_{lj}$ with pairwise different i,j,k,l have cost coefficients equal to zero.

A more concise model of the SRFLP is achieved when rewriting $D(i,j)$ via betweenness variables. In fact, for the SRFLP it is not necessary to model for each pair of objects their relation in the ordering. It is enough to know for each pair of objects which facilities are *between* them. Betweenness variables ζ_{ijk} for facilities i,j,k are defined as

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

The betweenness polytope is the convex hull of the incidence vectors of feasible solutions, i.e.,

$$
P_{\text{Bet}} = \text{conv}\left\{ \zeta \in \{0,1\}^{n(n-1)(n-2)} : \zeta \text{ arises from a permutation of } 1, \dots, n \right\}.
$$

Thus, by rewriting $D(i,j) = \sum_k \ell_k \zeta_{ijk}$, the SRFLP is the problem of optimizing a linear function over the betweenness polytope. The latter is well understood, see e.g. Christof et al. [22]. A betweenness variable ζ_{ijk} can be written via linear ordering variables as $\zeta_{ijk} = x_{ik}x_{kj} + x_{jk}x_{ki}$. Therefore, the betweenness problem is a quadratic linear ordering problem in which all products $x_{ij}x_{kl}$ with pairwise different i, j, k, l necessarily have zero contribution to the objective function. As the betweenness problem is a specific quadratic linear ordering problem in which only $O(n^3)$ cost coefficients can take nonzero cost value, we immediately obtain that the betweenness polytope induces a face of a cut polytope.

Recently, a different relation of the betweenness polytope to the cut polytope was pointed out by Oswald et al. [57]. To this end, an undirected graph $G_{\mathrm{Bet}} = (V, E)$ is set up that contains a node for each facility. An edge $(i, j) \in E$ between two facilities exists in G_{Bet} if the cost c_{ij} associated with i and j is nonzero. The authors observe that for fixed k, a betweenness incidence vector (ζ_{ijk}) is the incidence vector of a cut in the graph $G_{\mathrm{Bet}} \setminus \{k\}$. In the latter, node k together with all its incident edges is deleted from the graph G_{Bet}. As a consequence, the projection of the betweenness variables ζ_{ijk} for fixed k is isomorphic to the cut polytope associated with graph $G_{\mathrm{Bet}} \setminus \{k\}$. Based on this observation, Oswald et al. [57] suggest an exact solution approach for the linear arrangement problem. One of their separation routines consists of separating odd-cycle inequalities in $G_{\mathrm{Bet}} \setminus \{k\}$, for each k. Odd-cycle inequalities are known to be facet-defining for the cut polytope. A similar exact branch-and-cut approach could be designed for the slightly more general SRFLP.

In summary, there exist different connections of SRFLP to the cut polytope and the cut cone. The model by Letchford et al. [6] relates SRFLP to the cut cone by observing that the latter contains the distance polytope. If the SRFLP is modeled via the quadratic linear ordering problem, a face of a cut polytope is induced. The latter polytope is defined for the graph G_{lin}. Using the betweenness formulation instead, for each object k the projection of the problem is isomorphic to the cut polytope on the graph $G_{\mathrm{Bet}} \setminus \{k\}$.

We note that the model via quadratic linear ordering is more general than the betweenness approach in the sense that every betweenness variable can be expressed via products of linear ordering variables, but not vice versa.

In the next section, we present the currently most effective LP-based approach by Amaral [4] which is based on a betweenness model. In Section 2.5, we obtain matrix-based formulations and SDP relaxations using the quadratic linear ordering model (QLO_2).

2.4 The Betweenness-Based LP Formulation of Amaral for SRFLP

Although not explicitly stated, Amaral's model is implicitly based on the connection to the cut polytope as observed by Oswald et al. [57]. We explain this in more detail in the following. Amaral [4] used the betweenness formulation

based on variables ζ_{ijk} for SRFLP. The objective function of the SRFLP can then be expressed as

$$\sum_{i<j} c_{ij} \left[\frac{1}{2} (\ell_i + \ell_j) + \sum_{k \neq i,j} \ell_k \zeta_{ijk} \right] \tag{3}$$

Let an index of an object be denoted by i, j, k, or d. To obtain an LP relaxation of the SRFLP, Amaral proposes to optimize (3) subject to the following partial description of P_{Bet} in which the objects i, j, k, d are always chosen pairwise different:

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad \text{for all } \{i, j, k\} \subseteq \{1, \ldots, n\}, \tag{4}$$

$$\zeta_{ijd} + \zeta_{jkd} - \zeta_{ikd} \geq 0, \quad \text{for all } \{i, j, k, d\} \subseteq \{1, \ldots, n\}, \tag{5}$$

$$\zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \leq 2, \quad \text{for all } \{i, j, k, d\} \subseteq \{1, \ldots, n\}, \tag{6}$$

$$0 \leq \zeta_{ijk} \leq 1, \text{for all } \{i, j, k\} \subseteq \{1, \ldots, n\}. \tag{7}$$

Equation (4) can easily be understood from the fact that the equalities $\zeta_{ijk} = 1$, $\zeta_{ikj} = 1$, and $\zeta_{jki} = 1$ are mutually exclusive (by definition of ζ_{ijk}). Using the equivalence $\zeta_{ijk} \equiv x_{ik}x_{kj} + x_{jk}x_{ki}$ based on linear ordering variables, it follows that (4) is equivalent to the minimum equation system (2) known for the quadratic linear ordering problem. Without loss of generality, we assume $i < j < k$ and rewrite (4) as

$$\begin{aligned}
\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1 \Leftrightarrow & (x_{ik}x_{kj} + x_{jk}x_{ki}) + (x_{ij}x_{jk} + x_{kj}x_{ji}) \\
& + (x_{ji}x_{ik} + x_{ki}x_{ij}) = 1 \\
\Leftrightarrow & x_{ik}(1 - x_{jk}) + x_{jk}(1 - x_{ik}) + x_{ij}x_{jk} + \\
& (1 - x_{jk})(1 - x_{ij}) + (1 - x_{ij})x_{ik} + (1 - x_{ik})x_{ij} = 1 \\
\Leftrightarrow & 2(x_{ik} - x_{ij}x_{ik} - x_{ik}x_{jk} + x_{ij}x_{jk}) = 0.
\end{aligned}$$

Using the recent observation of Oswald et al. [57] that for a fixed object d, the incidence vectors of betweenness vectors are incidence vectors of cuts in the graph $G_{\text{Bet}} \setminus \{d\}$, it can be checked that (5) and (6) are precisely the triangle inequalities known to be facet-defining for the cut polytope. Indeed, projecting the inequalities accordingly and denoting the variables by ζ^p, (5) and (6) read for pairwise different i, j, k:

$$\zeta_{ij}^p + \zeta_{jk}^p - \zeta_{ik}^p \geq 0, \quad \text{for all } \{i, j, k\} \subseteq \{1, \ldots, n\}, \tag{8}$$

$$\zeta_{ij}^p + \zeta_{jk}^p + \zeta_{ik}^p \leq 2, \quad \text{for all } \{i, j, k\} \subseteq \{1, \ldots, n\}, \tag{9}$$

These are exactly the triangle inequalities stating that a cut either contains zero or two edges from a triangle. They can be generalized to the odd-cycle inequalities that state in algebraic terms that a cycle and a cut can only coincide in an even number of common edges, see e.g. [25]. Alternatively, one can also check that (5) and (6) are the triangle inequalities by using the

equivalence $\zeta_{ijk} \equiv x_{ik}x_{kj} + x_{jk}x_{ki}$ and working through the classical linear transformation between the maximum-cut problem and QUBO.

Amaral proposes a further class of valid inequalities for P_{Bet} that can be used as cuts to improve the LP relaxation. They are as follows:

Proposition 1 *[4] Let $\beta \leq n$ be a positive even integer and let $S \subseteq \{1, \ldots, n\}$ such that $|S| = \beta$. For each $r \in S$, and for any partition (S_1, S_2) of $S \backslash \{r\}$ such that $|S_1| = \frac{1}{2}\beta$, the inequality*

$$\sum_{t<q,t\in S_1,q\in S_1} \zeta_{tqr} + \sum_{t<q,t\in S_2,q\in S_2} \zeta_{tqr} - \sum_{t\in S_1,q\in S_2} \zeta_{\min\{t,q\},\max\{t,q\},r} \leq 0 \quad (10)$$

is valid for P_{Bet}.

It is easy to check that for $\beta = 4$, (10) is a triangle inequality of the form (5).

Again, the inequalities (10) can be interpreted as inequalities for the cut polytope. For the corresponding graph $G_{\text{Bet}} \backslash \{r\}$, the inequality (10) takes the form

$$\zeta^p(E(S_1)) + \zeta^p(E(S_2)) - \zeta^p(\delta(S_1)) \leq 0 \quad (11)$$

where for a node set U the usual abbreviation $\zeta^p(E(U)) = \sum_{e\in E(U)} \zeta_e^p$ is used. The inequality (11) is a so-called clique inequality [25] that is here switched along the cut $\delta(S_1)$. Clique inequalities are known to be valid and facet-inducing for the cut polytope. Their exact separation is however NP-hard, and so often heuristic separation procedures are used in practice. Amaral does not use a general separation routine for arbitrary values of β in (10) but proposes separating inequalities with $\beta = 6$ by enumeration. The corresponding class of clique inequalities are often called pentagonal inequalities.

Unlike for the LP relaxation, the triangle inequalities are not included explicitly in the SDP relaxations that are presented in Section 2.5. However, they can be added as valid inequalities there, and so can any other valid inequalities for the cut polytope. Furthermore, theoretical results in Section 2.7 below show that when certain triangle inequalities are present in either the LP or the SDP relaxation, a number of the pentagonal and of the so-called hexagonal inequalities automatically hold as well (see Section 2.7 for the definition of these classes of inequalities).

Amaral also shows that the size of the LP relaxation can be reduced by projecting the feasible set into a lower-dimensional space. We refer the reader to [4] for details. Very recently, Sanjeevi and Kianfar [63] have independently shown that several of the valid inequalities from Amaral are indeed facet-defining. The classes of inequalities studied there are well known to be facet-defining for the cut polytope. It is an interesting fact that they remain facet-inducing when considering only the convex hull of the SRFLP incidence vectors which induce a face of this cut polytope.

2.5 Matrix-Based Formulations and SDP Relaxations

In the sections above, we discussed integer LP-based solution approaches where it is convenient to formulate models in binary variables that either take value zero or value one. In this section, we introduce SDP relaxations where the variables are usually assumed to take values in $\{\pm 1\}$. We start by presenting the quadratic formulation of the SRFLP proposed by Anjos et al. [7]. The latter is basically a modeling via the quadratic linear ordering problem given in (QLO_2) but written in ± 1 instead of binary variables. Their model can be described as follows.

For a given permutation π of $[n]$, for each pair of integers ij with $1 \leq i < j \leq n$, define a binary ± 1 variable such that

$$R_{ij} := \left\{ \begin{array}{rl} 1, & \text{if } \pi_j < \pi_i, \\ -1, & \text{if } \pi_i < \pi_j. \end{array} \right.$$

In this definition, the order of the subscripts matters, and $R_{ij} = -R_{ji}$. In fact, R_{ij} determines a linear ordering on the objects.

Given a particular assignment of ± 1 values to the R_{ij} variables, if this assignment represents a permutation of $[n]$, then the condition that

if i is to the right of j and j is to the right of k, then i is to the right of k

must be enforced. Equivalently, if $R_{ij} = R_{jk}$ then $R_{ik} = R_{ij}$. This necessary transitivity condition can be formulated as a set of quadratic constraints:

$$R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \text{ for all triples } 1 \leq i < j < k \leq n. \quad (12)$$

Interestingly, (12) is exactly the set of equations (2), written in ± 1 variables. After (12) was introduced in [7] for SRFLP, it was shown in [16] that the latter is a minimum equation system for the quadratic linear ordering problem (QLO).

The objective function of the SRFLP can be expressed as

$$\sum_{i<j} c_{ij} \left[\frac{1}{2} (\ell_i + \ell_j) + \sum_{k \neq i,j} \ell_k \left(\frac{1 - R_{ki}R_{kj}}{2} \right) \right]$$

upon observing that $R_{ki}R_{kj} = -1$ if and only if facility k is between i and j. The resulting formulation of the SRFLP is:

$$\begin{aligned} \min\ & K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k R_{ki}R_{kj} - \sum_{i<k<j} \ell_k R_{ik}R_{kj} + \sum_{k>j} \ell_k R_{ik}R_{jk} \right] \\ \text{s.t.}\ & \\ & R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \text{ for all triples } i < j < k \\ & R_{ij}^2 = 1 \text{ for all } i < j \end{aligned} \quad (13)$$

where $K := \left(\sum_{i<j} \frac{c_{ij}}{2} \right) \left(\sum_{k=1}^{n} \ell_k \right)$. Note that if every R_{ij} variable is replaced by its negative, then there is no change whatsoever to the formulation. For this reason, the formulation (13) and the subsequent matrix-based formulations and corresponding SDP relaxations implicitly account for the symmetry in the SRFLP.

Let $\rho \in \{\pm 1\}^{\binom{n}{2}}$ denote an assignment of values to the R_{ij} variables. Hence define the set:

$$\mathcal{R}_n := \left\{ \rho \in \{\pm 1\}^{\binom{n}{2}} : R_{ij}R_{jk} - R_{ij}R_{ik} - R_{ik}R_{jk} = -1 \quad \forall 1 \leq i < j < k \leq n \right\}$$

and consider the function $f : \mathcal{R}_n \to \Pi_n$ defined by

$$f(\rho) = (\pi_1, \ldots, \pi_n), \text{ where } \pi_k := \frac{P_k + n + 1}{2}$$

and

$$P_k := \sum_{j \neq k} R_{kj} = \sum_{j<k} -R_{jk} + \sum_{j>k} R_{kj} \qquad \text{for } k = 1, 2, \ldots, n. \qquad (14)$$

We have the following fact:

Theorem 1 *The function $f : \mathcal{R}_n \to \Pi_n$ is a bijection.*

This result follows from Buchheim et al. [16] and the earlier work of Murata et al. [55] who demonstrated that for every layout in two dimensions, there exists a finite representation via so-called sequence-pairs. This technique has been applied with great success in VLSI floorplanning, the original application in [55], and also in two-dimensional facility layout [51, 71]. An independent proof in terms of the variables and structure used here was given in Anjos and Yen [11].

Following [7], we obtain a formulation of the SRFLP in the space of real symmetric matrices by fixing an ordering of all pairs ij such that $i < j$, and defining the vector

$$v := (R_{p_1}, \ldots, R_{p_{\binom{n}{2}}})^T,$$

where p_k denotes the kth pair ij in the ordering. Using v, we construct the rank-one matrix $X := vv^T$ whose rows and columns are indexed by pairs ij. By construction, $X_{p_i, p_j} = R_{p_i} R_{p_j}$ for any two pairs p_i, p_j, and therefore we can formulate the SRFLP as:

$$\min K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k X_{ki,kj} - \sum_{i<k<j} \ell_k X_{ik,kj} + \sum_{k>j} \ell_k X_{ik,jk} \right]$$

$$\text{s.t.}$$

$$X_{ij,jk} - X_{ij,ik} - X_{ik,jk} = -1 \text{ for all triples } i < j < k \qquad (15)$$
$$\text{diag}(X) = e$$
$$\text{rank}(X) = 1$$
$$X \succeq 0$$

where $\mathrm{diag}\,(X)$ represents a vector containing the diagonal elements of X, e denotes the vector of all ones, and $X \succeq 0$ denotes that X is symmetric positive semidefinite. Note that $X \in \mathcal{S}^{\binom{n}{2}}$, the set of symmetric matrices of dimension $\binom{n}{2}$.

Removing the rank constraint from (15) yields an SDP relaxation. Note that in general the SDP problem only provides a lower bound on the optimal value of the SRFLP, and not a feasible solution, unless the optimal matrix X^* happens to have rank equal to one. A standard way to tighten linear or semidefinite relaxations of integer optimization problems is, as mentioned earlier, to add inequalities (such as the triangle inequalities) that are valid for the integer feasible points. The SDP relaxation of (15) with a straightforward scheme to iteratively add violated triangle inequalities was used in [10] to solve SRFLPs with up to 30 facilities to global optimality for the first time.

From a computational perspective, the main limitation of the SDP relaxation of (15) is that it has $O(n^3)$ linear constraints; this limits the size of instances that can be tackled with it. This motivated Anjos and Yen [11] to consider an alternative formulation, obtained by reducing the number of the linear constraints in the following way:

$$\min K - \sum_{i<j} \frac{c_{ij}}{2} \left[\sum_{k<i} \ell_k X_{ki,kj} - \sum_{i<k<j} \ell_k X_{ik,kj} + \sum_{k>j} \ell_k X_{ik,jk} \right]$$

s.t.

$$\sum_{k=1,k\neq i,j}^{n} X_{ij,jk} - \sum_{k=1,k\neq i,j}^{n} X_{ij,ik} - \sum_{k=1,k\neq i,j}^{n} X_{ik,jk} = -(n-2) \text{ for all } i < j$$
$$\mathrm{diag}\,(X) = e$$
$$\mathrm{rank}\,(X) = 1$$
$$X \succeq 0.$$

$$(16)$$

It is straightforward to prove that the two formulations are equivalent (but not their SDP relaxations).

Theorem 2 *[11] The feasible sets of (15) and (16) are equal.*

Removing the rank-one constraint from (16) again yields an SDP relaxation. Although the number of linear constraints in this relaxation is now $O(n^2)$, with a corresponding favourable impact on the computational time and memory requirements of a primal-dual interior-point algorithm, it turns out that the quality of the solution appears to deteriorate only slightly, yielding optimality gaps not greater than 5% for instances with up to $n = 100$. A summary of the results reported in [11] is provided in Table 1.

2.6 SDP-based Heuristic to Obtain a Layout

The SDP relaxations of (15) and (16) are closely related to the basic SDP relaxation for the maximum-cut problem used in the ground-breaking paper

Table 1. Bounds for some extremely large instances

Number of departments	Number of instances	Range of gaps	Average CPU time
56	5	3.49% - 4.47%	3h 03m
64	5	2.80% - 5.15%	9h 00m
72	5	2.89% - 4.03%	23h 20m
81	5	3.44% - 4.97%	2d 0h 24m
100	5	3.18% - 3.81%	9d 17h 0m

of Goemans and Williamson [28]. However, we cannot use their randomized hyperplane rounding procedure because it does not guarantee that we will obtain a valid representation of a permutation. This is because it does not ensure that the equality constraints (12) hold, and hence the result may not represent a valid permutation. One possibility is to use a heuristic to fix the result, as is done in [38]. Alternatively, a different rounding procedure, based on Theorem 1, was proposed in [7] to obtain a permutation from the optimal solution to the SDP relaxation.

The procedure is as follows: If X^* is the optimal solution to the SDP relaxation, then each row of X^* corresponds to a specific pair $i_1 j_1$ of facilities. Therefore, for any row of X^*, if we set $R_{i_1 j_1} = +1$, then we can scan the other entries of the row and assign the value $X_{i_1 j_1, i_2 j_2}$ to the variable $R_{i_2 j_2}$, for every pair $i_2 j_2 \neq i_1 j_1$. (Note that every value assigned to an R_{ij} is in the interval $[-1, 1]$ because every feasible X for the SDP is a correlation matrix.) Using these values, we compute

$$\omega_k = \frac{1}{2} \left(n + 1 + \sum_{j \neq k} R_{kj} \right)$$

for $k = 1, \ldots, n$.

The motivation for the values ω_k comes from the fact that if X^* is rank-one, then $\omega_k = \pi_k, k = 1, \ldots, n$, where the π_k define the bijection in Theorem 1, and hence the ω_k define a permutation of $[n]$. In general, rank $(X^*) > 1$ and thus $\omega_k \in [1, n]$, so the SDP-based heuristic obtains a permutation of $[n]$ by sorting the values ω_k. The sorting can be in either decreasing or increasing order (since the objective value is the same), and since the procedure implicitly sets $R_{i_1 j_1} = +1$, we choose the order that places i_1 to the right of j_1. The output of the heuristic is the best layout found by considering every row in turn.

2.7 On the Facial Structure of the Relaxations

Consider the following set of solutions feasible for both the SDP relaxation of (15) and the LP relaxation (4)-(7) under the aforementioned mapping between

their variables:

$$\mathcal{X}_n := \left\{ X \in \mathcal{S}^{\binom{n}{2}} : -1 \leq X_{ij,kl} \leq 1, \right.$$
$$\left. X_{ij,jk} - X_{ij,ik} - X_{ik,jk} = -1 \quad \forall\, 1 \leq i < j < k \leq n \right\}.$$

It is straightforward to check that a number of triangle inequalities automatically hold for every $X \in \mathcal{X}_n$.

Lemma 1 *If $X \in \mathcal{X}_n$, then for every triple of pairs (i_1, i_2), (i_1, i_3), and (i_2, i_3), where $i_1 < i_2 < i_3$, the entries of X satisfy*

$$X_{i_1 i_2, i_1 i_3} + X_{i_1 i_2, i_2 i_3} + X_{i_1 i_3, i_2 i_3} \geq -1, X_{i_1 i_2, i_1 i_3} - X_{i_1 i_2, i_2 i_3} - X_{i_1 i_3, i_2 i_3} \geq -1,$$
$$-X_{i_1 i_2, i_1 i_3} - X_{i_1 i_2, i_2 i_3} + X_{i_1 i_3, i_2 i_3} \geq -1, -X_{i_1 i_2, i_1 i_3} + X_{i_1 i_2, i_2 i_3} - X_{i_1 i_3, i_2 i_3} \geq -1.$$

Proof: Since $X \in \mathcal{X}_n$, we know $X_{i_1 i_2, i_2 i_3} - X_{i_1 i_2, i_1 i_3} - X_{i_1 i_3, i_2 i_3} = -1$, and therefore the fourth inequality trivially holds. Now,

$$X_{i_1 i_2, i_2 i_3} + X_{i_1 i_2, i_1 i_3} + X_{i_1 i_3, i_2 i_3} = X_{i_1 i_2, i_2 i_3} + X_{i_1 i_2, i_1 i_3} + (1 + X_{i_1 i_2, i_2 i_3} - X_{i_1 i_2, i_1 i_3})$$
$$= 1 + 2 X_{i_1 i_2, i_2 i_3} \geq -1,$$

and hence the first triangle inequality above holds. The other two inequalities follow similarly. ∎

We have thus observed that $4\binom{n}{3}$ triangle inequalities of the form (8-9) automatically hold for all the feasible matrices of $\mathcal{X}_n$. If we consider the addition of the remaining triangle inequalities, we obtain the following (tighter) set of feasible solutions:

$$\mathcal{X}_n^{\triangle} = \mathcal{X}_n \cap \{ X \in \mathcal{S}^{\binom{n}{2}} : X_{p_1,p_2} + X_{p_1,p_3} + X_{p_2,p_3} \geq -1, X_{p_1,p_2} - X_{p_1,p_3} - X_{p_2,p_3} \geq -1,$$
$$-X_{p_1,p_2} - X_{p_1,p_3} + X_{p_2,p_3} \geq -1, -X_{p_1,p_2} + X_{p_1,p_3} - X_{p_2,p_3} \geq -1,$$
$$\forall\, p_1, p_2, p_3 : \{p_1, p_2, p_3\} \neq \{(i_1, i_2), (i_1, i_3), (i_2, i_3)\}$$
$$\text{for any } i_1 < i_2 < i_3 \}.$$

We now show that a number of facet-defining inequalities of the cut polytope are implicitly enforced in $\mathcal{X}_n^{\triangle}$.

First, we consider the pentagonal inequalities. For each subset of pairs $\{p_1, \ldots, p_5\}$ corresponding to rows and columns of X, there are 16 such inequalities and they can be represented as:

$$\sum_{1 \leq i < j \leq 5} \delta_i \delta_j X_{p_i, p_j} \geq -2,$$

where $\delta_k \in \{-1, 1\}, k = 1, 2, 3, 4, 5$. Hence there are $16\binom{\binom{n}{2}}{5}$ valid pentagonal inequalities in total. It is proved in [8] that for $\mathcal{X}_n^{\triangle}$, $90\binom{n}{4}$ of those inequalities automatically hold.

Lemma 2 *[8] Suppose that $X \in \mathcal{X}_n^{\triangle}$, and (from the definition of $\mathcal{X}_n^{\triangle}$) consider any five pairs $p_1, \ldots, p_5$ that satisfy*

$$X_{p_1,p_4} - X_{p_1,p_2} - X_{p_2,p_4} = -1, \tag{17}$$
$$X_{p_1,p_5} - X_{p_1,p_3} - X_{p_3,p_5} = -1. \tag{18}$$

Then for all choices of $\delta_i \in \{-1, 1\}, i = 1, 2, 3, 4, 5$, such that $(\delta_1\delta_2 - 1)(\delta_1\delta_3 - 1)(\delta_1\delta_4 + 1)(\delta_1\delta_5 + 1) = 0$, the pentagonal inequality

$$\sum_{1 \leq i < j \leq 5} \delta_i \delta_j X_{p_i,p_j} \geq -2$$

holds. This gives a total of 15 pentagonal inequalities.

Lemma 2 leads to the following theorem.

Theorem 3 *[8] Suppose that $X \in \mathcal{X}_n^{\triangle}$, and consider any choice $1 \leq i_1 < i_2 < i_3 < i_4 \leq n$. Then for each of the following sets of row indices for X, precisely 15 pentagonal inequalities hold:*

$$\{(i_1, i_2), (i_1, i_3), (i_1, i_4), (i_2, i_3), (i_2, i_4)\},$$
$$\{(i_1, i_2), (i_1, i_3), (i_1, i_4), (i_2, i_3), (i_3, i_4)\},$$
$$\{(i_1, i_2), (i_1, i_3), (i_1, i_4), (i_2, i_4), (i_3, i_4)\},$$
$$\{(i_1, i_2), (i_1, i_3), (i_2, i_3), (i_2, i_4), (i_3, i_4)\},$$
$$\{(i_1, i_2), (i_1, i_4), (i_2, i_3), (i_2, i_4), (i_3, i_4)\}, \quad and$$
$$\{(i_1, i_3), (i_1, i_4), (i_2, i_3), (i_2, i_4), (i_3, i_4)\}.$$

Hence, for $X \in \mathcal{X}_n^{\triangle}$, the $90\binom{n}{4}$ pentagonal inequalities described above automatically hold.

One can also consider the hexagonal inequalities: They are the inequalities obtained from

$$2\sum_{t=2}^{6} \delta_1\delta_t X_{p_1,p_t} + \sum_{2 \leq s < t \leq 6} \delta_s\delta_t X_{p_s,p_t} \geq -4$$

by permutation of the pairs and where $\delta_k \in \{-1, 1\}, k = 1, \ldots, 6$, for every 6-tuple of pairs $p_1, \ldots, p_6$.

Using the same approach as for proving Theorem 3, one can show that a number of hexagonal inequalities are implicitly enforced.

Theorem 4 *[8] Suppose that $X \in \mathcal{X}_n^{\triangle}$, and that for the set of pairs $\{p_1, \ldots, p_6\}$ corresponding to rows and columns of X, the following hold:*

$$X_{p_1,p_4} - X_{p_1,p_2} - X_{p_2,p_4} = -1, \quad X_{p_1,p_5} - X_{p_1,p_3} - X_{p_3,p_5} = -1,$$

$$X_{p_2,p_6} - X_{p_2,p_3} - X_{p_3,p_6} = -1, \quad and \ X_{p_4,p_6} - X_{p_4,p_5} - X_{p_5,p_6} = -1.$$

Then for all choices of $\delta_k \in \{-1, 1\}, k = 1, 2, 3, 4, 5, 6$, the hexagonal inequalities

$$2 \sum_{j \neq i, j=1}^{6} \delta_i \delta_j X_{p_i, p_j} + \sum_{j \neq i, k \neq i, 1 \leq j < k \leq 6} \delta_j \delta_k X_{p_j, p_k} \geq -4, \quad i = 1, \ldots, 6,$$

hold.

Hence, $192 \binom{n}{4}$ hexagonal inequalities automatically hold for $X \in \mathcal{X}_n^{\triangle}$.

3 VLSI Floorplanning

In this section, we introduce the application of SDP to the VLSI macrocell floorplanning problem. The general VLSI floorplanning problem consists of arranging a set of rectangular modules on a rectangular chip area so that an appropriate measure of performance is optimized. The resulting layout is called a floorplan. In the macrocell context, the modules are soft, meaning that the area of each rectangular module is assumed to be fixed while its height and width are allowed to vary subject to given aspect ratio constraints [62, 70]. The floorplanning of soft modules is an important problem because it takes advantage of the fact that at this stage of the physical design process, the rectangular modules have not themselves been laid out in detail yet, and so the floorplanner can do a better job if it is allowed to change the dimensions of the modules in a controlled manner.

Specifically, the VLSI macrocell floorplanning problem consists of partitioning a given rectangular chip into N modules with fixed areas so as to minimize the total cost associated with interactions between these modules and with N_p fixed I/O pads located all around the chip. For this problem, Takouda et al. [69] presented the first formulation using a mixed-integer cone optimization model motivated by the success of the SDP approach to SRFLP.

3.1 A Mixed-Integer Cone Optimization Formulation for Floorplanning

The VLSI macrocell surface is modeled as a $w_F \times h_F$ rectangle with given area a_F. We set the origin of our system of coordinates at the center of the chip. A module i is a $w_i \times h_i$ rectangle with given area a_i and centroid (x_i, y_i). We denote by $w_i^{\max}$ (and respectively $w_i^{\min}$, $h_i^{\max}$, $h_i^{\min}$) the given maximum length (resp. minimum length, maximum width, minimum width) of each module. The pads are determined by their coordinates, and are denoted using the letters p, q, r while the modules are denoted by i, j, k. We finally assume that the cost of the connection per unit of distance between modules i, j is denoted by c_{ij} and the cost per unit of distance between a module i and a pad p is γ_{ip}. The variables are (x_i, y_i), h_i and w_i for each module i, and the objective is to minimize the total weighted connection cost.

Area constraints

We relax the area constraint $w_i h_i = a_i$ for each module as

$$w_i h_i \geq a_i. \tag{19}$$

This convex relaxation can be expressed as a semidefinite constraint:

$$\begin{pmatrix} w_i & \sqrt{a_i} \\ \sqrt{a_i} & h_i \end{pmatrix} \succeq 0, \tag{20}$$

which is well defined since $a_i > 0$. Convexifying the area constraints is a reasonable approach for floorplanning in view of the following lemma, and the fact that the condition $\sum_i a_i = a_F$ holds for nearly all macrocell floorplanning problems.

Lemma 3 *[69] If* (20) *holds and* $a_F = \sum_i a_i$, *then the area constraints* $w_i h_i = a_i$ *are satisfied for every* i.

From a practical perspective, we point out that each 2×2 semidefinite constraint (20) can be expressed as a second-order cone constraint of size 3 [41]. Using second-order cone constraints normally leads to lower running times in practice.

Fit-in-the-chip constraints

For module i to lie inside the chip, the following inequalities have to be satisfied:

$$-\tfrac{1}{2}(w_F - w_i) \leq x_i \leq \tfrac{1}{2}(w_F - w_i), \tag{21}$$
$$-\tfrac{1}{2}(h_F - h_i) \leq y_i \leq \tfrac{1}{2}(h_F - h_i). \tag{22}$$

If desired, they can also be expressed as semidefinite constraints since (21) follows from:

$$x_i^2 \leq \left[\frac{1}{2}(w_F - w_i)\right]^2 \Leftrightarrow \begin{pmatrix} \frac{1}{2}(w_F - w_i) & x_i \\ x_i & \frac{1}{2}(w_F - w_i) \end{pmatrix} \succeq 0, \tag{23}$$

provided $0 \leq w_i \leq w_F$. Similarly (22) follows from:

$$\begin{pmatrix} \frac{1}{2}(h_F - h_i) & y_i \\ y_i & \frac{1}{2}(h_F - h_i) \end{pmatrix} \succeq 0, \tag{24}$$

provided $0 \leq h_i \leq h_F$. Computationally, however, the linear constraints (21-22) lead to lower running times.

Aspect ratio constraints

The aspect ratio of department i is defined as $\frac{\max\{h_i, w_i\}}{\min\{h_i, w_i\}}$. In facility layout problems, it is often desirable to set bounds on the aspect ratios of the departments to ensure that no department is excessively narrow (in either direction) in the computed layout. However, as the bounds on the aspect ratios become smaller, the layout problem becomes more constrained and the total cost of the optimal solution increases. VLSI floorplanning benchmarks typically include aspect ratio restrictions on the modules.

Given aspect ratio bounds $\beta_i \geq 1$ for each module i, the aspect ratio requirements are $\frac{w_i}{h_i} \leq \beta_i$ and $\frac{h_i}{w_i} \leq \beta_i$ and can be enforced using linear constraints $w_i \leq \beta_i h_i$ and $h_i \leq \beta_i w_i$ for fixed β_i. Alternatively, since $w_i > 0, h_i > 0, a_i > 0$, and $w_i h_i = a_i$, these constraints are equivalent to $w_i^2 \leq \beta_i a_i$ and $h_i^2 \leq \beta_i a_i$, and therefore to

$$\begin{pmatrix} \beta_i & w_i \\ w_i & a_i \end{pmatrix} \succeq 0, \quad \begin{pmatrix} \beta_i & h_i \\ h_i & a_i \end{pmatrix} \succeq 0. \tag{25}$$

These constraints can also be expressed as second-order cone constraints with small support. The formulation (25) has the advantage that it allows the possibility of letting β_i be a decision variable, which can be useful in practice.

Non-overlap constraints

Finally, we consider the non-overlap constraints. These disjunctive constraints require the modules to be separated in either the x or the y direction. They can be expressed as follows:

$$\underbrace{d_{ij}^x}_{=|x_i-x_j|} \geq \frac{1}{2}(w_i + w_j) \quad \text{or} \quad \underbrace{d_{ij}^y}_{=|y_i-y_j|} \geq \frac{1}{2}(h_i + h_j) \quad \forall i < j.$$

Since only one of the two inequalities must be satisfied for each pair i, j of modules, these constraints are commonly modeled using binary variables [64] or complementarity constraints [9]. The formulation in Takouda et al. [69] uses only two binary variables per pair of modules. The first variable, σ_{ij}, is used to decide the direction in which the non-overlap is enforced:

$$\sigma_{ij} = \begin{cases} +1 & \text{if } i \text{ and } j \text{ are separated along } x, \\ -1 & \text{if } i \text{ and } j \text{ are separated along } y, \end{cases} \tag{26}$$

Once this direction is selected, the variable α_{ij} determines the relative position of modules i and j in that direction:

$$\alpha_{ij} = \begin{cases} +1 & \text{if } i \text{ precedes } j \text{ in the selected direction,} \\ -1 & \text{if } j \text{ precedes } i \text{ in the selected direction.} \end{cases} \tag{27}$$

The following inequalities enforce the separation of modules i and j in the x direction:

$$d_{ij}^x \geq \tfrac{1}{2}(w_i + w_j) - \tfrac{1}{2}(1 - \sigma_{ij})Q_{ij}^x \tag{28}$$

$$d_{ij}^x - 2S_{ij}^x = x_j - x_i \tag{29}$$

$$0 \leq S_{ij}^x \tag{30}$$

$$S_{ij}^x \leq \tfrac{1}{2}\left[(1 - \sigma_{ij}) + \tfrac{1}{2}(1 + \sigma_{ij})(1 - \alpha_{ij})\right] U_{ij}^x \tag{31}$$

$$0 \leq S_{ij}^x + (x_j - x_i) \tag{32}$$

$$S_{ij}^x + (x_j - x_i) \leq \tfrac{1}{2}\left[(1 - \sigma_{ij}) + \tfrac{1}{2}(1 + \sigma_{ij})(1 + \alpha_{ij})\right] U_{ij}^x \tag{33}$$

where $Q_{ij}^x = \min\{w_F, w_i^{\max} + w_j^{\max}\}$ and $U_{ij}^x = w_F - \tfrac{1}{2}w_i^{\min} - \tfrac{1}{2}w_j^{\min}$. The constraints also compute the x component d_{ij}^x of the pairwise distance d_{ij}. A similar set of constraints can be used for the y direction of separation.

Complete formulation

To complete the formulation, we model the rectilinear distances between a module i and an I/O pad p:

$$d_{ip}^r = \underbrace{|x_i - x_p|}_{d_{ip}^x} + \underbrace{|y_i - y_p|}_{d_{ip}^y},$$

where x_p and y_p are given. Since either $|x_p| \geq w_F/2$ or $|y_p| \geq h_F/2$, we have two cases. If $|x_p| \geq w_F/2$:

$$d_{ip}^x = x_p - x_i \text{ or } d_{ip}^x = x_i - x_p \quad \text{(whichever is positive)} \tag{34}$$

$$d_{ip}^y \geq y_p - y_i \tag{35}$$

$$d_{ip}^y \geq y_i - y_p. \tag{36}$$

The case $|y_p| \geq h_F/2$ is handled similarly.

The result is a mixed-integer SOCP-SDP formulation for the VLSI macrocell floorplanning problem:

$$\min \sum_{i<j}^N c_{ij}(d_{ij}^x + d_{ij}^y) + \sum_{i=1}^N \sum_{p=1}^{N_p} \gamma_{ip}(d_{ip}^x + d_{ip}^y)$$

s.t. Areas: (20) $\forall i$
   Fit-in-the-chip: (21), (22) $\forall i$
   Aspect ratios: (25) $\forall i$
   Non-overlap & distances (module to module): (28) − (33)
       and similar constraints for the y direction $\forall i < j$
   Distances (module to pad): (34) − (36) $\forall i, p$
   $0 \leq w_i \leq w_F$ and $0 \leq h_i \leq h_F$ $\forall i$
   $d_{ij}^x, d_{ij}^y, d_{ip}^x, d_{ip}^y, S_{ij}^x, S_{ij}^y, x_i, y_i \in \mathbb{R}$ $\forall i < j$
   $\sigma_{ij}, \alpha_{ij} \in \{-1, 1\}$ $\forall i < j$.

It is of course straightforward to alternatively measure the distances using the Euclidean distance by using a second-order cone constraint. We focus on the rectilinear distance because it is the measure of interest in the context of floorplanning.

Additional inequalities can be used to slightly improve this formulation. We refer the reader to [69] for details.

3.2 Deriving an SDP Relaxation

First we expand the non-overlap constraints:

$$S_{ij}^x \leq \tfrac{1}{4}\left(3 - \sigma_{ij} - \alpha_{ij} - \sigma_{ij}\alpha_{ij}\right)U_{ij}^x, \tag{37}$$

$$S_{ij}^x + (x_j - x_i) \leq \tfrac{1}{4}\left(3 - \sigma_{ij} + \alpha_{ij} + \sigma_{ij}\alpha_{ij}\right)U_{ij}^x, \tag{38}$$

$$S_{ij}^y \leq \tfrac{1}{4}\left(3 + \sigma_{ij} - \alpha_{ij} + \sigma_{ij}\alpha_{ij}\right)U_{ij}^y, \tag{39}$$

$$S_{ij}^y + (y_j - y_i) \leq \tfrac{1}{4}\left(3 + \sigma_{ij} + \alpha_{ij} - \sigma_{ij}\alpha_{ij}\right)U_{ij}^y. \tag{40}$$

Thus we need an SDP relaxation that allows us to linearize the products of sigmas and alphas in these expressions.

We also ensure that our formulation satisfies the two-dimensional version of the transitivity property (12) for the SRFLP. We state it as:

> If modules i, j, k are separated in the same direction,
> and if in that direction i precedes j and j precedes k,
> then i precedes k in that direction.

Since

- the modules i, j, k are separated in the same direction if and only if $\sigma_{ij} = \sigma_{jk} = \sigma_{ik}$, and
- the transitivity property in the selected direction is then : $\alpha_{ij} = \alpha_{jk} \Rightarrow \alpha_{ij} = \alpha_{ik}$,

the two-dimensional transitivity property is stated as

$$\begin{pmatrix} \sigma_{ij} = \sigma_{jk} = \sigma_{ik} \\ \text{and} \\ \alpha_{ij} = \alpha_{jk} \end{pmatrix} \Rightarrow \alpha_{ij} = \alpha_{ik}. \tag{41}$$

Proposition 2 *[69] A sufficient condition to enforce* (41) *is*

$$(\sigma_{ij} + \sigma_{jk})(\sigma_{ij} + \sigma_{ik})(\alpha_{ij} + \alpha_{jk})(\alpha_{ij} - \alpha_{ik}) = 0 \tag{42}$$

for all $i < j < k$.

Expanding (42), we obtain:

$$
\begin{aligned}
1 = &\ \alpha_{ij}\alpha_{ik} - \alpha_{ij}\alpha_{jk} + \alpha_{ik}\alpha_{jk} \\
&- \sigma_{ij}\sigma_{ik} - \sigma_{ij}\sigma_{jk} - \sigma_{ik}\sigma_{jk} \\
&+ \sigma_{ij}\sigma_{ik}\alpha_{ij}\alpha_{ik} - \sigma_{ij}\sigma_{ik}\alpha_{ij}\alpha_{jk} + \sigma_{ij}\sigma_{ik}\alpha_{ik}\alpha_{jk} \\
&+ \sigma_{ij}\sigma_{jk}\alpha_{ij}\alpha_{ik} - \sigma_{ij}\sigma_{jk}\alpha_{ij}\alpha_{jk} + \sigma_{ij}\sigma_{jk}\alpha_{ik}\alpha_{jk} \\
&+ \sigma_{ik}\sigma_{jk}\alpha_{ij}\alpha_{ik} - \sigma_{ik}\sigma_{jk}\alpha_{ij}\alpha_{jk} + \sigma_{ik}\sigma_{jk}\alpha_{ik}\alpha_{jk},
\end{aligned}
\tag{43}
$$

for all $i < j < k$. The SDP relaxation must allow us to linearize the products in these expressions as well.

To define the SDP matrix variable, we introduce the vector ξ of $1 + 2\binom{N}{2} + 2\left(\binom{\binom{N}{2}}{2}\right)$ binary variables:

$$
\begin{aligned}
\xi = (1, \sigma_{12}, \ldots, \sigma_{N-1,N}, \alpha_{12}, \ldots, \alpha_{N-1,N}, \sigma_{12}\sigma_{13}, \\
\ldots, \sigma_{N-2,N}\sigma_{N-1,N}, \alpha_{12}\alpha_{13}, \ldots, \alpha_{N-2,N}\alpha_{N-1,N})^T
\end{aligned}
$$

containing all the binary variables and their required monomials. The SDP rank-one formulation and subsequent relaxation are constructed as in Section 2.5 with the vector ξ replacing the vector v.

3.3 Computational Performance of the SDP Relaxation

The MCNC benchmarks are a well-known collection of benchmark problems used by VLSI researchers. Takouda et al. [69] applied the SDP relaxation to three problems from the MCNC benchmark: apte, xerox and hp. Although these are the smallest instances in this benchmark, and this benchmark is now several years old, these problems are quite large from the facility layout perspective, and hence already challenging.

The bounds for varying aspect ratios are reported in Table 2. To give a sense of the computational effort required to solve the relaxations, we report in Table 2 the CPU time taken using CSDP [14] on a 2.0 GHz Dual Opteron with 16Gb of RAM. Finally, to convey a sense of the quality of the bounds, we compare them in Table 3 to the best reported solutions in the literature, namely those of Murata and Kuh [56]. We point out that the xerox instance is particularly challenging. This is true not only for the SDP relaxation but also for macrocell placement algorithms, and is due to the very small number of pads.

Overall, although this is the first time that such lower bounds were computed, we see that the relative gaps to the best known solutions are still large. Takouda et al. improved the bounds slightly by computing full levels of a branch-and-bound tree, but such an approach is of limited use because of the resulting high computational times. It is thus necessary to improve the strength of the SDP relaxations, and to be able to solve them for larger sizes. Current research is making good progress in these directions [1].

Table 2. Results on MCNC benchmark problems

Circuit	Number of modules	Number of pads	Aspect ratio	Lower bound	CPU time (sec)
apte	9	73	2	3135.9	891
			3	3021.6	789
			5	2918.5	848
			8	2848.8	815
			10	2847.7	793
xerox	10	2	2	2434.0	1665
			3	2051.0	2290
			5	1539.8	2930
			8	1217.1	2835
			10	1153.1	2721
hp	11	45	2	976.98	8156
			3	893.25	8230
			5	822.11	8294
			8	783.80	7840
			10	773.23	7855

Table 3. Global bounds versus Murata-Kuh solutions ($\beta_i = 10$)

Circuit	Bound	Solution	Gap (%)
apte	2847.7	4353.5	34.6
xerox	1153.1	4976.5	76.8
hp	773.2	1779.8	56.6

4 Ongoing and Future Research

There is still much potential for the SDP-based approach in the areas of facility layout and floorplanning. First, the recent results of Hungerländer and Rendl [38] suggest that the potential of the SDP approach for the SRFLP is not yet fully developed. While it is generally accepted that memory requirements for SDP are higher than for LP, and that LP can exploit the sparsity in the SRFLP objective better than SDP, the SDP bounds stronger than the LP bounds. It would therefore be interesting to conduct a careful study of the computational limitations and future potential of LP and SDP approaches. Such a study should include an assessment of how the SDP rounding heuristic in Section 2.6 compares to other heuristics applied to the SRFLP in the literature.

Second, extensions of the cone programming approach to other layout problems are still in their infancy. For two-dimensional layout, Adams [1] proposed a novel SDP relaxation that replaces the variables σ_{ij} and α_{ij} described above with quaternary variables. Preliminary results indicate that this approach outperforms the model of Takouda et al. described above. Other im-

portant research directions that remain unexplored include the extension of the SDP models for SRFLP to multi-row problems with applications in the service industry and VLSI design. A second one is the application of SDP to three-dimensional layout problems for which current global optimization methods are of limited use, see e.g. [13] and the references therein. Thus, layout problems remain a very promising opportunity for the application of cone programming techniques to obtain global optimal solutions.

Acknowledgements

References

1. E.C. Adams. A semidefinite programming model for the facility layout problem. Master's thesis, University of Waterloo, 2010.
2. A.R.S. Amaral. On the exact solution of a facility layout problem. *Eur. J. Oper. Res.*, 173(2):508–518, 2006.
3. A.R.S. Amaral. An exact approach for the one-dimensional facility layout problem. *Oper. Res.*, 56(4):1026–1033, 2008.
4. A.R.S. Amaral. A new lower bound for the single row facility layout problem. *Discrete Appl. Math.*, 157(1):183–190, 2009.
5. A.R.S. Amaral. On the exact solution of a facility layout problem. *Discr. Appl. Math.*, 157(1):183–190, 2009.
6. A.R.S. Amaral and A.N. Letchford. A polyhedral approach to the single-row facility layout problem. Technical report, Department of Management Science, Lancaster University, March 2008.
7. M.F. Anjos, A. Kennings, and A. Vannelli. A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optim.*, 2(2):113–122, 2005.
8. M.F. Anjos and A. Vannelli. Globally optimal solutions for large single-row facility layout problems. Technical report, University of Waterloo, 2006.
9. M.F. Anjos and A. Vannelli. A new mathematical-programming framework for facility-layout design. *INFORMS J. Comp.*, 18(1):111–118, 2006.
10. M.F. Anjos and A. Vannelli. Computing globally optimal solutions for single-row layout problems using semidefinite programming and cutting planes. *INFORMS J. Comp.*, 20(4):611–617, 2008.
11. M.F. Anjos and G. Yen. Provably near-optimal solutions for very large single-row facility layout problems. *Optim. Methods Softw.*, 24(4):805–817, 2009.
12. F. Barahona and A.R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.
13. S. Bernardi. A three-stage mathematical-programming method for the multi-floor facility layout problem. Master's thesis, University of Waterloo, 2010.

14. B. Borchers. CSDP, a C library for semidefinite programming. *Optim. Methods Softw.*, 11/12(1-4):613–623, 1999.
15. C. Buchheim, F. Liers, and M. Oswald. Speeding up IP-based algorithms for constrained quadratic 0-1 optimization. *Mathematical Programming (Series B)*. 124(1-2):513–535, 2010.
16. C. Buchheim, A. Wiegele, and L. Zheng. Exact algorithms for the quadratic linear ordering problem. *INFORMS J. on Computing*, 2(1):168–177, 2010.
17. I. Castillo and T. Sim. A spring-embedding approach for the facility layout problem. *J. Oper. Res. Soc.*, 55:73–81, 2004.
18. I. Castillo, J. Westerlund, S. Emet, and T. Westerlund. Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimixation methods. *Computers and Chemical Engineering*, 30(1):54–69, 2005.
19. I. Castillo and T. Westerlund. An ε-accurate model for optimal unequal-area block layout design. *Comput. Oper. Res.*, 32(3):429–447, 2005.
20. E. Çela. *The Quadratic Assignment Problem*, volume 1 of *Combinatorial Optimization*. Kluwer Academic Publishers, Dordrecht, 1998.
21. I. Charon and O. Hudry. An updated survey on the linear ordering problem for weighted or unweighted tournaments. *Ann. Oper. Res.*, 175:107–158, 2010.
22. T. Christof, M. Oswald, and G. Reinelt. Consecutive ones and a betweenness problem in computational biology. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization, Springer-Verlag Lecture Notes in Computer Science 1412*, 1998.
23. A.G. de Alvarenga, F.J. Negreiros-Gomes, and M. Mestria. Metaheuristic methods for a class of the facility layout problem. *J. Intell. Manuf.*, 11:421–430, 2000.
24. C. De Simone. The cut polytope and the boolean quadric polytope. *Discrete Mathematics*, 79:71–75, 1989.
25. M.M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.
26. J. Díaz, J. Petit, and M. Serna. A survey on graph layout problems. *ACM Computing Surveys*, 34:313–356, 2002.
27. L.R. Foulds. *Graph Theory Applications*. Springer-Verlag, New York, 1991.
28. M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.*, 42(6):1115–1145, 1995.
29. M. Grötschel, M. Jünger, and G. Reinelt. Facets of the linear ordering polytope. *Mathematical Programming*, 33:43–60, 1985.
30. K.M. Hall. An r-dimensional quadratic placement algorithm. *Management Sciences*, 17:219–229, 1970.
31. P. L. Hammer. Some network flow problems solved with pseudo-boolean programming. *Operations Research*, 13:388–399, 1965.
32. C. Helmberg and K.C. Kiwiel. A spectral bundle method with bounds. *Math. Program.*, 93(2, Ser. A):173–194, 2002.
33. C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM J. Optim.*, 10(3):673–696 (electronic), 2000.
34. S. S. Heragu. *Facilities Design*. iUniverse, second edition, 2006.
35. S.S. Heragu and A.S. Alfa. Experimental analysis of simulated annealing based algorithms for the layout problem. *European J. Oper. Res.*, 57(2):190–202, 1992.
36. S.S. Heragu and A. Kusiak. Machine layout problem in flexible manufacturing systems. *Oper. Res.*, 36(2):258–268, 1988.

37. S.S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European J. Oper. Res.*, 53:1–13, 1991.
38. P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. Technical report, Alpen-Adria-Universität Klagenfurt, August 2010.
39. I. Jankovits. An improved convex optimization model for two-dimensional facility layout. Master's thesis, University of Waterloo, 2006.
40. Richard M. Karp and Michael Held. Finite-state processes and dynamic programming. *SIAM J. Appl. Math.*, 15:693–718, 1967.
41. S. Kim, M. Kojima, and M. Yamashita. Second order cone programming relaxation of a positive semidefinite constraint. *Optim. Methods Softw.*, 18(5):535–541, 2003.
42. T.C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25:53–76, 1957.
43. K.R. Kumar, G.C. Hadjinicola, and T. Lin. A heuristic procedure for the single-row facility layout problem. *European J. Oper. Res.*, 87(1):65–73, 1995.
44. M. Lewis, B. Alidaee, F. Glover, and G. Kochenberger. A note on xQx as a modelling and solution framework for the linear ordering problem. *International Journal of Operational Research*, 5(2):152–162, 2009.
45. F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi. *Computing Exact Ground States of Hard Ising Spin Glass Problems by Branch-and-Cut*, pages 47–68. New Optimization Algorithms in Physics. Wiley-VCH, 2004.
46. W. Liu and A. Vannelli. Generating lower bounds for the linear arrangement problem. *Discrete Appl. Math.*, 59(2):137–151, 1995.
47. L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1:166–190, 1991.
48. R.F. Love and J.Y. Wong. On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14(2):139–143, 1976.
49. C. Luo, M.F. Anjos, and A. Vannelli. A nonlinear optimization methodology for VLSI fixed-outline floorplanning. *J. Comb. Optim.*, 16(4):378–401, 2008.
50. T.D. Mavridou and P.M. Pardalos. Simulated annealing and genetic algorithms for the facility layout problem: a survey. *Comput. Optim. Appl.*, 7(1):111–126, 1997.
51. R.D. Meller, W. Chen, and H.D. Sherali. Applying the sequence-pair representation to optimal facility layout designs. *Oper. Res. Lett.*, 35(5):651–659, 2007.
52. R.D. Meller and K.Y. Gau. The facility layout problem: recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 15:351–366, 1996.
53. R.D. Meller, V. Narayanan, and P.H. Vance. Optimal facility layout design. *Oper. Res. Lett.*, 23(3-5):117–127, 1998.
54. B. Montreuil. A modelling framework for integrating layout design and flow network design. In J.A. White and I.W. Pence, editors, *Progress in Material Handling and Logistics*, volume 2, pages 95–116. Springer-Verlag, 1991.
55. H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1518–1524, 1996.
56. H. Murata and E.S. Kuh. Sequence-pair based placement method for hard/soft/pre-placed modules. *Proceedings of the International Symposium on Physical Design*, pages 167–172, 1998.

57. M. Oswald, A. Caprara, E. Traversi, M. Jung, and G. Reinelt. A betweenness approach for solving the linear arrangement problem, 2009. Talk given at 13th Combinatorial Optimization Workshop, Aussois, France.
58. J.-C. Picard and M. Queyranne. On the one-dimensional space allocation problem. *Oper. Res.*, 29(2):371–391, 1981.
59. G. Reinelt. *The Linear Ordering Problem: Algorithms and Applications*. Heldermann Verlag, 1985.
60. F. Rendl, G. Rinaldi, and A. Wiegele. A branch and bound algorithm for Max-Cut based on combining semidefinite and polyhedral relaxations. In *Integer programming and combinatorial optimization*, volume 4513 of *Lecture Notes in Computer Science*, pages 295–309. Springer Verlag, Berlin, 2007.
61. D. Romero and A. Sánchez-Flores. Methods for the one-dimensional space allocation problem. *Comput. Oper. Res.*, 17(5):465–473, 1990.
62. S.M. Sait and H. Youssef. *VLSI physical design automation : theory and practice*. IEEE Press, New York, USA, 1995.
63. S. Sanjeevi and K. Kianfar. A polyhedral study of triplet formulation for single row facility layout problem. *Discrete Appl. Math.*, 158(16):1861–1867, 2010.
64. H.D. Sherali, B.M.P. Fraticelli, and R.D. Meller. Enhanced model formulation for optimal facility layout. *Oper. Res.*, 51(4):629–644, 2003.
65. D.M. Simmons. One-dimensional space allocation: An ordering algorithm. *Oper. Res.*, 17:812–826, 1969.
66. D.M. Simmons. A further note on one-dimensional space allocation. *Oper. Res.*, 19:249, 1971.
67. S.P. Singh and R.R.K. Sharma. A review of different approaches to the facility layout problems. *Intl J. Adv. Manuf. Tech.*, 30(5–6):425–433, 2006.
68. J.K. Suryanarayanan, B.L. Golden, and Q. Wang. A new heuristic for the linear placement problem. *Computers & Operations Research*, 18(3):255–262, 1991.
69. P.L. Takouda, M.F. Anjos, and A. Vannelli. Global lower bounds for the VLSI macrocell floorplanning problem using semidefinite optimization. In *Proceedings of IWSOC 2005*, pages 275–280, 2005.
70. D.F. Wong and C.L. Liu. A new algorithm for floorplan design. In *Proc. of ACM/IEEE Design Automation Conf.*, pages 101–107, 1986.
71. W. Xie and N.V. Sahinidis. A branch-and-bound algorithm for the continuous facility layout problem. *Comput. & Chem. Eng.*, 32:1016–1028, 2008.