Open access • Journal Article • DOI:10.1007/S10589-012-9505-8

A computational study and survey of methods for the single-row facility layout problem — Source link

Philipp Hungerländer, Franz Rendl

Institutions: Alpen-Adria-Universität Klagenfurt

Published on: 01 May 2013 - Computational Optimization and Applications (Springer US)

Topics: Combinatorial optimization

A Computational Study and Survey of Methods for the Single-Row Facility Layout Problem

Philipp Hungerländer and Franz Rendl

Mathematics, University of Klagenfurt, Austria,
philipp.hungerlaender@uni-klu.ac.at, franz.rendl@uni-klu.ac.at

Technical Report
Alpen-Adria-Universitaet Klagenfurt
April 2012

Abstract. The single-row facility layout problem (SRFLP) is an NP-hard combinatorial optimization problem that is concerned with the arrangement of n departments of given lengths on a line so as to minimize the weighted sum of the distances between department pairs. (SRFLP) is the one-dimensional version of the facility layout problem that seeks to arrange rectangular facilities so as to minimize the overall interaction cost. This paper compares the different modelling approaches for (SRFLP) and applies a recent SDP approach for general quadratic ordering problems from Hungerländer and Rendl to (SRFLP). In particular, we report optimal solutions for several (SRFLP) instances from the literature with up to 42 departments that remained unsolved so far. Secondly we significantly reduce the best known gaps and running times for large instances with up to 100 departments.

1 Introduction

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

$$\min_{\pi \in \Pi} \sum_{i,j \in \mathcal{N}, i<j} c_{ij} z_{ij}^{\pi}, \tag{1}$$

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

Several practical applications of (SRFLP) have been identified in the literature, such as the arrangement of rooms on a corridor in hospitals, supermarkets, or offices [36], the assignment of airplanes to gates in an airport terminal [39], the arrangement of machines in flexible manufacturing systems [23], the arrangement of books on a shelf and the assignment of disk cylinders to files [31].

On the one hand (SRFLP) (also known as one-dimensonal space allocation problem) is a special case of the weighted betweenness problem which is again a special case of the quadratic ordering problem. On the other hand the NP-hard [17] minimum linear arrangement problem is a special case of (SRFLP) where all facilities have the same length and the connectivities are equal to 0 or 1. Hence (SRFLP) is also NP-hard.

Accordingly several heuristic algorithms have been suggested to tackle instances of interesting size of (SRFLP), e.g. [14, 19, 20, 22, 24, 28, 33, 34]. However, these heuristic approaches do not provide any optimality certificate, like an estimate of the distance from optimality, for the solution found.

Several exact approaches to (SRFLP) have also been proposed. Simmons [36] first studied (SRFLP) and suggested a branch-and-bound algorithm. Later on Simmons [37] pointed out the possibility of extending the dynamic programming algorithm of Karp and Held [27] to (SRFLP). This was later on implemented by Picard and Queyranne [31]. A nonlinear model was presented by Heragu and Kusiak [24]. Linear mixed integer programs using distance variables were proposed by Love and Wong [30] and Amaral [1]. Amaral [2] achieved a more efficient linear mixed integer program by linearizing a quadratic model based on ordering variables. However all these models suffer from weak lower bounds and hence have high computation times and memory requirements. But just recently Amaral and Letchford [4] achieved significant progress in that direction through the first polyhedral study of the distance polytope for (SRFLP) and showed that their approach is quite effective for instances with challenging size ($n \geq 30$). Amaral [3] suggested an LP-based cutting plane algorithm using betweenness variables that proved to be highly competitive and solved instances with up to 35 facilities to optimality. Recently Sanjeevi and Kianfar [35] studied the polyhedral structure of Amaral's betweenness model in more detail and identified several classes of facet defining inequalities.

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

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

where the data matrices $A_i, \ i \in \{1, \ldots, m\}$ and C are symmetric. We refer the reader to the handbooks [6, 40] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where semidefinite programming has had a major impact.

Anjos et al. [5] proposed the first SDP relaxation for (SRFLP) yielding bounds for instances with up to 80 facilities. Anjos and Vanelli [8] further tightened the SDP relaxation using triangle inequalities as cutting planes and gave optimal solutions for instances with up to 30 facilities that remained unsolved since 1988. Anjos and Yen [9] suggested an alternative SDP relaxation and achieved optimality gaps no greater than 5 % for large instances with up to 100 facilities. Recently Hungerländer and Rendl [26] proposed a general approach for quadratic ordering problems, where they further improved on the tightness of the above SDP relaxations. They used a suitable combination of optimization methods to deal with the stronger but more expensive relaxations and applied their method among others to some selected medium (SRFLP) instances. Thereby they solved instances with up to 40 facilities to optimality.

The main contributions of this paper are the following: First we describe and compare the most successful modelling approaches to (SRFLP), pointing out their common connections to the maximum cut [10, 21, 38] and the quadratic ordering problem [11, 12]. For further details on this subject see also the recent survey of (SRFLP) by Anjos and Liers [7].

Secondly we apply the approach from [26] for the first time to a broad selection of small, medium and large instances and compare it computationally to the leading algorithms for the different instance sizes. Thereby we demonstrate that this approach clearly dominates all other methods, permitting significant progress for medium as well as large instances. We can give optimal solutions for several medium instances from the literature with up to 42 facilities that remained unsolved so far and reduce all the best known gaps for large scale instances by a factor varying from 2 to 100.

Finally we relate the two SDP heuristics from [5] and [26] concerning their computational costs and practical performance.

The paper is structured as follows. In Section 2, we put the most competitive algorithms for (SRFLP) into perspective and compare them from a theoretical point of view. In Section 3, we conduct an extensive computational study for the SDP approach of Hungerländer and Rendl [26], achieving significant progress for medium and large instances. Finally some conclusions and current research are summarized in Section 4.

2 The Most Successful Modelling Approaches to (SRFLP)

The most intuitive modelling approach to (SRFLP) using $\binom{n}{2}$ distance variables $z_{ij}^{\pi}, i, j \in \mathcal{N}$ suffers from weak lower bounds of the corresponding LP relaxation and thus large branch-and-bound trees, high computation times and memory requirements. Recently Amaral and Letchford [4] achieved significant progress in that direction by identifying several classes of valid inequalities and using them as cutting planes. Amaral [3] improved the LP relaxation by modelling (SRFLP) via $\binom{n}{3}$ binary betweenness variables. Anjos et. al [5] proposed to model (SRFLP) as a binary quadratic program using $\binom{n}{2}$ ordering variables. They deduced a semidefinite relaxation yielding tighter bounds but being more expensive to compute than the relaxation of Amaral [3]. Later on further SDP approaches have been suggested to improve on the relaxation strength and/or reduce the computational effort involved [8, 9, 26]. In the following subsections we recall the approaches mentioned above and highlight their relations.

2.1 Distance-Based LP Formulation of Amaral and Letchford [4]

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

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

Amaral and Letchford [4] show that the equation

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

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

2.2 Betweenness-Based LP Formulation of Amaral [3]

Amaral [3] introduced binary variables $\zeta_{ijk}(i, j, k \in \mathcal{N}, i < j, i \neq k \neq j)$

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

Amaral [3] collected these betweenness variables in a vector ζ and defined the betweenness polytope

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

In order to formulate `(SRFLP)` via ζ an appropriate objective function is needed. For that purpose Amaral [3] used the relation

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

to rewrite (1) in terms of ζ (for details see [3, Proposition 1 and 2])

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

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

$$\zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i, j, k \in \mathcal{N}, i < j < k. \tag{3}$$

In [35] it is shown that these equations describe the smallest linear subspace that contains $\mathcal{P}^n_{Btw}$. To obtain an LP relaxation of `(SRFLP)`, the integrality conditions on ζ are replaced with 0-1 bounds:

$$0 \leq \zeta_{ijk} \leq 1, \quad i, j, k \in \mathcal{N}, i < j. \tag{4}$$

To further strengthen the relaxation, Amaral [3] came up with additional valid inequalities. Let a subset $\{i, j, k, d\} \subset \mathcal{N}$ be given. On the one hand department d can not be located between the departments i and j, i and k and j and k at the same time. On the other hand if department d is between departments i and k then it also lies between departments i and j or j and k. Thus the inequalities

$$\zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \leq 2, \quad i, j, k, d \in \mathcal{N}, i < j < k \tag{5}$$

and

$$-\zeta_{ijd} + \zeta_{jkd} + \zeta_{ikd} \geq 0, \quad \zeta_{ijd} - \zeta_{jkd} + \zeta_{ikd} \geq 0, \quad \zeta_{ijd} + \zeta_{jkd} - \zeta_{ikd} \geq 0, \quad i, j, k, d \in \mathcal{N}, i < j < k, \tag{6}$$

are valid for $\mathcal{P}^n_{Btw}$. Sanjeevi and Kianfar [35] showed that (6) unlike (5) are facet defining for $\mathcal{P}^n_{Btw}$.

Amaral [3] further generalizes (6) to a more complicated set of inequalities: Let $\beta \leq n$ be an even integer and let $S \subseteq \mathcal{N}$. For each $d \in S$, and for any partition (S_1, S_2) of $S \setminus \{d\}$ such that $|S_1| = \frac{1}{2}\beta$, the inequality

$$\sum_{p,q \in S_1, p<q} \zeta_{pqd} + \sum_{p,q \in S_2, p<q} \zeta_{pqd} \leq \sum_{p \in S_1, q \in S_2, p<q} \zeta_{pqd} \tag{7}$$

is valid [3] and also facet-defining [35] for $\mathcal{P}^n_{Btw}$. Note that (6) is a special case of (7) with $\beta = 4$.

Minimizing (2) over (3)–(6) gives the basic linear relaxation `(LP)`. To construct stronger relaxations from `(LP)` Amaral [3] proposes to use the inequalities $(7)_{\beta=6}$ as cutting planes (for details see Subsection 3.1 below).

2.3 Matrix-Based Relaxations of Anjos et al. [5,8,9]

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

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise,} \end{cases} \tag{8}$$

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

$$\zeta_{ijk} = \frac{1 + y_{ik} y_{kj}}{2}, \; i < k < j, \qquad \zeta_{ijk} = \frac{1 - y_{ki} y_{kj}}{2}, \; k < i < j, \qquad \zeta_{ijk} = \frac{1 - y_{ik} y_{jk}}{2}, \; i < j < k, \tag{9}$$

for $i, j, k \in \mathcal{N}$. Using (9) we can easily rewrite the objective function (2) and equalities (3) in terms of ordering variables

$$K - \sum_{\substack{i,j \in \mathcal{N} \\ i<j}} \frac{c_{ij}}{2} \left(\sum_{\substack{k \in \mathcal{N} \\ k<i}} l_k y_{ki} y_{kj} - \sum_{\substack{k \in \mathcal{N} \\ i<k<j}} l_k y_{ik} y_{kj} + \sum_{\substack{k \in \mathcal{N} \\ k>j}} l_k y_{ik} y_{jk} \right), \tag{10}$$

$$y_{ij} y_{jk} - y_{ij} y_{ik} - y_{ik} y_{jk} = -1, \quad i, j, k \in \mathcal{N}, i < j < k, \tag{11}$$

where $K := \left(\sum_{\substack{i,j \in \mathcal{N} \\ i<j}} \frac{c_{ij}}{2} \right) \left(\sum_{k \in \mathcal{N}} l_k \right)$. In [12] it is shown that the equations (11) formulated in a $\{0, 1\}$ model describe the smallest linear subspace that contains the quadratic ordering polytope

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

To obtain matrix-based relaxations we collect the ordering variables in a vector y and consider the matrix $Y = yy^\top$. The main diagonal entries of Y correspond to y^2_{ij} and hence $\text{diag}(Y) = e$, the vector of all ones. Now we can formulate `(SRFLP)` as the following optimization problem, first proposed in [5]

$$\min \; \{ \; \langle C, Y \rangle + K : \; Y \text{ satisfies (11)}, \;\; \text{diag}(Y) = e, \; \text{rank}(Y) = 1, \; Y \succcurlyeq 0 \; \}, \tag{SRFLP}$$

where the cost matrix C is deduced from (10). Dropping the rank constraint yields the basic semidefinite relaxation of `(SRFLP)`

$$\min \; \{ \; \langle C, Y \rangle + K : \; Y \text{ satisfies (11)}, \;\; \text{diag}(Y) = e, \; Y \succcurlyeq 0 \; \}, \tag{SDP$_1$}$$

providing a lower bound on the optimal value of (SRFLP). To be able to tackle larger instances Anjos and Yen [9] proposed to sum up the $O(n^3)$ constraints (11) over k yielding the $O(n^2)$ constraints

$$\sum_{\substack{k \in \mathcal{N} \\ i \neq k \neq j}} y_{ij} y_{jk} - \sum_{\substack{k \in \mathcal{N} \\ i \neq k \neq j}} y_{ij} y_{ik} - \sum_{\substack{k \in \mathcal{N} \\ i \neq k \neq j}} y_{ik} y_{jk} = -(n-2), \quad i, j \in \mathcal{N}, i < j. \tag{12}$$

They showed that the following optimization problem using (12) instead of (11)

$$\min \{ \langle C, Y \rangle + K : \ Y \text{ satisfies (12)}, \ \operatorname{diag}(Y) = e, \ \operatorname{rank}(Y) = 1, \ Y \succcurlyeq 0 \},$$

is again an exact formulation of (SRFLP). Dropping the rank-one constraint yields a weaker but also cheaper semidefinite relaxation than (SDP_1)

$$\min \{ \langle C, Y \rangle + K : \ Y \text{ satisfies (12)}, \ \operatorname{diag}(Y) = e, \ Y \succcurlyeq 0 \}. \tag{SDP_0}$$

As Y is actually a matrix with $\{-1, 1\}$ entries in the original (SRFLP) formulation, Anjos and Vanelli [8] proposed to further tighten (SDP_1) by adding the triangle inequalities, defining the metric polytope $\mathcal{M}$ and known to be facet-defining for the cut polytope, see e.g. [15]

$$\mathcal{M} = \left\{ Y : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} Y_{i,j} \\ Y_{j,k} \\ Y_{i,k} \end{pmatrix} \leq e, \ 1 \leq i < j < k \leq \binom{n}{2} \right\}. \tag{13}$$

Using the linear transformations (9) it is straightforward to show the equivalence of a subset of the triangle inequalities with the betweenness constraints (5) and (6) from above. Along the same lines inequalities (7) can be connected to general clique inequalities. Adding the triangle inequalities to (SDP_1), Anjos and Vanelli [8] achieved the following relaxation of (SRFLP)

$$\min \{ \langle C, Y \rangle + K : \ Y \text{ satisfies (11)}, \ Y \in \mathcal{M}, \ \operatorname{diag}(Y) = e, \ Y \succcurlyeq 0 \}. \tag{SDP_2}$$

As solving (SDP_2) directly with an interior-point solver like CSDP gets far too expensive, they suggest to use the $\approx \frac{1}{12} n^6$ triangle inequalities as cutting planes in their algorithmic framework (for details see Subsection 3.1 below). Let us also mention that so far all SDP approaches to (SRFLP) refrained from using other clique inequalities to further tighten the SDP relaxations because of their large number. We will argue in the conclusions that using well-designed subsets of larger clique inequalities, like e.g. pentagonal inequalities, which can be connected to the betweenness constraints $(7)_{\beta=6}$, could be a promising direction to improve current SDP approaches.

2.4 Strengthened Matrix-Based Relaxation

Recently Hungerländer and Rendl [26] suggested a further strengthening of (SDP_2) and an alternative algorithmic approach to solve such large SDP relaxations. To this end we introduce the matrix

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{14}$$

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

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

which is convex due to the Schur-complement lemma. Note that $Z \succcurlyeq 0$ is in general a stronger constraint than $Y \succcurlyeq 0$. Additionally we use an approach suggested by Lovász and Schrijver in [29] to further improve on the relaxation strength. This yields the following inequalities

$$\begin{aligned} -1 - y_{lm} \leq y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \leq 1 + y_{lm}, \ \forall\, i, j, k, l, m \in \mathcal{N}, i < j < k, \ l < m, \\ -1 + y_{lm} \leq y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \leq 1 - y_{lm}, \ \forall\, i, j, k, l, m \in \mathcal{N}, i < j < k, \ l < m, \end{aligned} \tag{15}$$

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

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

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

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies } (15)\ \}, \tag{16}$$

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

$$\min \{\ \langle C, Y\rangle + K :\ Y \text{ satisfies } (11),\ Z \in (\mathcal{M} \cap \mathcal{LS}),\ \operatorname{diag}(Z) = e,\ Z \succcurlyeq 0\ \}. \qquad (\mathtt{SDP}_3)$$

A similar relaxation (without the LS-cuts (15)) was used in [12] for bipartite crossing minimization. In [26] $(\mathtt{SDP}_3)$ is applied to different special cases of the quadratic ordering problem like the linear ordering problem, the linear arrangement problem, multi-level crossing minimization and of course (SRFLP). It is also demonstrated there that adding the LS-cuts to the relaxation pays off in practice.

To make $(\mathtt{SDP}_3)$ computationally tractable Hungerländer and Rendl [26] suggest to deal with the triangle inequalities (13) and LS-cuts (15) through Lagrangian duality (for details see Subsection 3.1 below and [26, Section 6]).

3 Computational Comparison

In this section we give a computational comparison of all state-of-the-art approaches to (SRFLP) on a broad selection of small, medium and large instances from the literature. Using the approach from [26] we solve several instances to optimality for the first time and improve on the gaps of all currently unsolved instances.

3.1 Comparison of Globally Optimal Methods for Small and Medium Instances

In Table 1 we computationally compare the four most competitive approaches to (SRFLP) for small and medium instances. These are the integer linear programming (ILP) approaches of Amaral and Letchford [4] and Amaral [3], the SDP approach of Anjos and Vanelli [8] building on relaxation $(\mathtt{SDP}_2)$ and the SDP approach from [26] building on relaxation $(\mathtt{SDP}_3)$.

Anjos and Vanelli [8] start with the basic relaxation $(\mathtt{SDP}_1)$ and then enhance it with violated triangle inequalities (13) in every iteration (using the interior-point solver CSDP version 5.0) until no more triangle inequalities are violated.

Amaral and Letchford [4] suggest an ILP Branch-and-Cut algorithm based on the distance variables z_{ij}. They use a cheap initial LP relaxation with only $O(n^2)$ non-zero coefficients and apply exact separation routines for triangle and special strengthened pure negative type inequalities and heuristic ones for clique, rounded psd and star inequalities. They suggest a specialised branching rule to avoid the use of additional binary variables and use a primal heuristic based on multi-dimensional scaling to obtain feasible layouts.

Amaral [3] proposes an ILP cutting plane algorithm based on the betweenness variables ζ_{ijk} that improves on the results in [8] and [4]. For computational usage of the betweenness model Amaral [3] suggests to alternate between solving (LP) and strengthening (LP) (by searching for cutting planes $(7)_{\beta=6}$ violated at the optimal solution of the current (LP) and adding them to (LP)). Amaral [3] also introduces new instances with 33 and 35 facilities, solves them to optimality and points out that he cannot solve larger instances with his approach as the involved linear programs become too large and too difficult to solve with the currently available LP solvers.

Recently Hungerländer and Rendl [26] proposed an algorithm to provide lower bounds to $(\mathtt{SDP}_3)$. Their method is building on subgradient optimization techniques, such as the bundle method [16, 25] and deals with the inequality constraints (13) and (15) through Lagrangian duality. A similar algorithmic approach was successfully applied to the maximum cut problem [32]. In [26] they already demonstrated that their algorithm clearly outperforms the SDP approach suggested in [8] on some selected (SRFLP) instances.

In Table 1 we give a full computational comparison of the four most successful exact approaches to (SRFLP) on all available instances from the literature, including well-known benchmark instances [1–3, 24, 36], instances with clearance requirement [23] and random-generated instances [8]. [1] The table identifies the instance by its name, source and size n and gives the times required by the four approaches to find a layout and prove its optimality.

The computations in [8] were carried out on a 2.0GHz Dual Opteron with 16 GB RAM, Amaral [3] used an Intel Core Duo, 1.73 GHz PC with 1 GB RAM, in [4] a 2.5 GHz Pentium Dual Core PC with 2 GB RAM was employed, whereas for applying the approach from [26] we use an Intel Xeon 5160 processor with 3 GHz and 2 GB RAM.

[1] Most of the instances can be downloaded from http://flplib.uwaterloo.ca/.

For small instances with up to 20 facilities the ILPs are preferable to the SDP approaches whereas the SDP approach from [26] outperforms the other approaches on the larger instances. The difference between the approaches strongly grows with the problem size. Note that we do not take into account the speed of the machines, as it does not differ too much and thus does not affect the conclusions drawn above. Our machine is the quickest and about 2.5 times faster than the one in [3], which is the slowest. [2]

This motivates us to tackle larger instances with the approach from [26]. We summarize the results for the five instances with 40 facilities, a density of 50 % and random lengths and connectivities between 1 and 10 in Table 2.[3]

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

Instance	Source	n	Anjos/Vanelli [8]	Amaral/Letchford [4]	Amaral [3]	Hungerländer/Rendl [26]
S5	[36]	5		0.1	0.1	0.1
S8	[36]	8		0.5	0.1	0.6
S8H	[36]	8	0.2	0.1	0.1	2.3
S9	[36]	9		0.1	0.1	0.7
S9H	[36]	9		2.4	0.1	9.2
S10	[36]	10	3.4	0.4	0.2	0.6
S11	[36]	11	32.6	0.7	0.3	1.3
P15	[1]	15			2.8	19.7
P17	[2]	17			8.4	34.9
P18	[2]	18			13.3	32.5
H_20	[24]	20	26:54	2:22	30.8	54.3
H_30	[24]	30	15:50:57	28:07:49	27:35	9:07
Cl_5	[24]	5	0.1	0.1	0.2	0.1
Cl_6	[24]	6	0.4	0.1	0.1	0.1
Cl_7	[24]	7	1.2	0.3	0.1	0.6
Cl_8	[24]	8	1.8	0.1	0.1	0.4
Cl_12	[24]	12	32.8	4.0	0.6	7.9
Cl_15	[24]	15	5:53	9.6	3.2	19.6
Cl_20	[24]	20	41:32	5:12	40.1	1:16
Cl_30	[24]	30	51:06:53	17:49:43	1:12:19	14:17
N25_01	[8]	25	3:44:38	7:19:44	3:46	2:48
N25_02	[8]	25	4:50:27	38:35	9:59	5:46
N25_03	[8]	25	5:48:21	1:25:41	4:49	4:11
N25_04	[8]	25	4:04:51	39:34	10:19	5:33
N25_05	[8]	25	8:22:22	1:18:10	3:47	3:31
N30_01	[8]	30	7:41:06	34:00:51	25:41	4:42
N30_02	[8]	30	10:41:53	3:56:53	22:43	6:08
N30_03	[8]	30	19:32:01	13:08:12	23:14	10:12
N30_04	[8]	30	31:03:11	58:20	2:19:22	11:44
N30_05	[8]	30	19:54:07	13:03:51	1:05:36	18:30
Am33_01	[3]	33			1:15:57	19:28
Am33_02	[3]	33			2:35:22	48:07
Am33_03	[3]	33			2:22:32	36:33
Am35_01	[3]	35			1:35:04	17:30
Am35_02	[3]	35			5:27:34	41:01
Am35_03	[3]	35			2:17:52	53:14

Table 1. Results for `(SRFLP)` instances with up to 35 facilities. The running times are given in sec, in min:sec or in h:min:sec respectively.

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

[3] These instances and the corresponding optimal orderings are available from `http://flplib.uwaterloo.ca/`.

Instance	n	Optimal cost	Time SDP Hungerländer/Rendl [26]
N40_1	40	107348.5	1:01:36
N40_2	40	97693	52:52
N40_3	40	78589.5	1:21:40
N40_4	40	76669	1:15:58
N40_5	40	103009	2:20:09

Table 2. Results for 5 new (SRFLP) instances with 40 facilities. The running times are given in min:sec or in h:min:sec.

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

In this subsection we compare the most competitive approaches to (SRFLP) for obtaining tight bounds of large instances. These are the algorithms of Anjos and Yen [9] building on relaxations (SDP_0) and (SDP_1) respectively and again the approach of Hungerländer and Rendl [26] building on relaxation (SDP_3). For solving relaxations (SDP_0) and (SDP_1), Anjos and Yen [9] use the interior-point solver CSDP (version 5.0). In Tables 3 and 4 we compare the three SDP approaches on instances with 36 – 100 facilities taken from [5] and [9].[4]

In [26] the constraints $Z \succcurlyeq 0$ and $\mathrm{diag}(Z) = e$ are maintained explicitly. The evaluations of an appropriate function over this set constitute the computational bottleneck and are responsible for more than 99% of the overall running time for large instances. To control the computational effort we restrict the number of function evaluations to 500 for instances with up to 64 departments and to 250 for larger instances. This limitation of the number of function evaluations leaves some room for further incremental improvement.

The SDP relaxations (SDP_0), (SDP_1), (SDP_2) and (SDP_3) are closely related to the standard SDP relaxation for the max-cut problem used in the seminal paper of Goemans and Williamson [18] to obtain high quality feasible solutions providing upper bounds. However the hyperplane rounding idea suggested in [18] cannot be applied directly to (SRFLP) to get a good layout because it yields a $\{-1, 1\}$ vector $\tilde{y}$, which need not be feasible with respect to the three cycle equations (11). That is why Anjos et al. [5] propose a different procedure to obtain a good feasible layout from the optimal solution of the SDP relaxation whereas Hungerländer and Rendl [26] suggest to apply a repair strategy to the infeasible $\tilde{y}$.

Anjos et al. [5] propose to use the entries $y^*_{ij,kl}$ of the optimal matrix Y^* of the SDP relaxation in the following way to obtain a good feasible layout: Fix a row ij and compute the values

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

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

In [26] it is suggested to take the $\{-1, 1\}$ vector $\tilde{y}$ obtained from hyperplane rounding and make it feasible with respect to the 3-cycle inequalities by flipping the signs of some of its entries appropriately. Computational experiments demonstrated that the repair strategy is not as critical as one might assume [13,26]. For example we know from multi-level crossing minimization that the heuristic clearly dominates traditional heuristic approaches.

The heuristic of Anjos et. al [5] is much cheaper than the one in [26] as they have to factorize Y^* to carry out the rounding procedure. Nonetheless the computation times of both heuristics are negligible compared to the computational effort for the lower bound computation. We compared both heuristics concerning the quality of the produced layouts on many test instances and found out that the heuristic from [26] is clearly superior. This is also supported by a comparison of the upper bounds achieved by both approaches in Tables 3 and 4, where the heuristic from [26] improves on the one of Anjos et. al [5] on all instances considered.

[4] Most of the instances can be downloaded from `http://flplib.uwaterloo.ca/`. Our improved gaps and the corresponding orderings are also available there.

When comparing the running times of the three approaches we do not take into account that Anjos and Yen [9] use a machine (2.4GHz Quad Opteron with 16 Gb of RAM) that is more than 1.5 times faster and has 8 times the memory of our machine. [5]

In Table 3 we compare the three approaches for problems with 36 to 56 facilities for which no optimal solution was known before. The table identifies the instance by its name and size n. We then provide the lower bound "lb" and the best layout found "blf" as well as the associated running times for the different approaches. Finally we give the running times that the approach of Hungerländer and Rendl [26] needs to improve on the gaps of the two other approaches "improve gap $(\texttt{SDP}_0)$" and "improve gap $(\texttt{SDP}_1)$".

The results show that the SDP approaches of Anjos and Yen [9] allow for substantial improvement. On the one hand the approach from [26] reduces the difference between best layout and lower bound for all instances by factors that are, except once, > 10 (both lower and upper bounds are improved for all instances). On the other hand it reaches the gaps achieved by the other two approaches considerably faster. Further it is worthwhile to note that all instances with 36 facilities and even one instance with 42 facilities can be solved to optimality for the first time.

In Table 4 we compare the cheaper approach from [9] using relaxation $(\texttt{SDP}_0)$ (the other one gets too expensive for these instances) to the approach from [26] for problems with 60 to 100 facilities.

The results show that the SDP approach of Anjos and Yen [9] again allows for some improvement. On the one hand the SDP approach from [26] reduces the difference between best layout and lower bound for all instances by factors going from clearly above 10 to 2 as the instance sizes grow (again both lower and upper bounds are improved for all instances). On the other hand the gaps achieved by the approach of Anjos and Yen [9] are reached in average in about half the time by the approach from [26].

Let us finally compare the SDP-based heuristic from [26] with the recent tabu search based heuristic of Samarghandi and Eshghi [34] and the recent permutation-based genetic algorithm of Datta et al. [14] on the 20 "AKV"-instances [5]. On five instances all three heuristics yield the same upper bound, on 5 instances the heuristics from [34] and [14] yield the same best value, on 5 instances the algorithm of Datta et al. [14] generates the best feasible layouts and on 5 instances the approach from [26] produces the best upper bounds. In general the SDP-based heuristic seems to be preferable when $n \leq 70$ and computation time is not a critical factor as its performance depends on the quality of the lower bounds from the SDP relaxation. The "sko"-instances [9] were not considered in [34] and [14], hence for these instances the lower and upper bounds presented in Tables 3 and 4 are the best known ones to date.

4 Conclusions and Current Research

This paper improves on the practical results for $(\texttt{SRFLP})$. The SDP approach of Hungerländer and Rendl [26] provides optimal solutions for several instances with up 42 facilities for the first time. Additionally it significantly reduces the duality gap and running times for large instances with up to 100 facilities. These achievements are the consequence of the interaction of the following three advancements:

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

There are two (combinable) directions to further improve current SDP approaches. On the one hand we could include well-designed subsets of order $\leq O(n^6)$ of larger clique inequalities, like e.g. the $\approx \frac{1}{240}n^{10}$ pentagonal inequalities, in the presented relaxations. On the other hand, we could incorporate the achieved bounds in a branch-and-bound framework.

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

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

Instance	n	SDP Anjos/Yen using (SDP_0) [9]			SDP Anjos/Yen using (SDP_1) [9]			SDP Hungerländer/Rendl [26] - restricted to 500 function evaluations					
		lb	blf	time	lb	blf	time	lb	blf	time	gap in %	improve gap (SDP_0)	improve gap (SDP_1)
ste36-1	36	9851	10328	7:15	10087.5	10301	14:57	10287	10287	14:50	0 %	2:23	2:55
ste36-2	36	170759.5	182649	7:12	175387	181910	14:17	181508	181508	25:25	0 %	2:05	2:39
ste36-3	36	96090	104041.5	7:13	98739	102179.5	13:42	101643.5	101643.5	24:01	0 %	2:35	3:09
ste36-4	36	91103	96854.5	7:16	94650.5	96080.5	14:23	95805.5	95805.5	16:15	0 %	2:01	3:49
ste36-5	36	87688	92563.5	7:19	89533	91893.5	14:25	91651.5	91651.5	17:58	0 %	2:00	3:09
sko42-1	42	24517	25779	20:07	24807	25724	45:21	25521	25525	2:23:09	0.02 %	5:20	7:34
sko42-2	42	207357	218117.5	20:21	210785	217296.5	45:14	216099.5	216120.5	2:43:34	0.01 %	5:57	9:38
sko42-3	42	167783.5	174694.5	20:10	169944.5	173854.5	47:32	173245.5	173267.5	2:47:18	0.01 %	8:01	17:51
sko42-4	42	131536	139630	19:21	133429.5	138829	48:18	137379	137615	2:53:05	0.17 %	4:55	7:24
sko42-5	42	238669.5	250501.5	20:18	242925.5	249327.5	45:41	248238.5	248238.5	1:08:42	0 %	6:37	11:13
sko49-1	49	39333.5	41379	59:55	39794.5	41308	2:48:57	40895	41012	4:36:21	0.29 %	33:33	57:43
sko49-2	49	403024.5	418370	1:03:30	407741.5	418288	2:50:32	416142	416178	8:27:34	0.01 %	19:11	31:43
sko49-3	49	313923.5	326004	1:02:13	317628.0	325747	2:51:45	324464	324512	8:03:03	0.02 %	19:15	26:20
sko49-4	49	229809.5	238380.5	1:05:34	232368	237894.5	2:50:40	236718.5	236755	9:15:14	0.02 %	21:59	32:01
sko49-5	49	645406.5	673303	1:04:13	652638	671508	2:51:45	666130	666143	9:30:22	0.002 %	35:04	35:04
sko56-1	56	61789.5	64454	3:05:19	62496.5	64396	8:40:40	63971	64027	12:36:33	0.09 %	41:55	1:03:12
sko56-2	56	480473.5	499700	3:09:35	486426.5	498836	9:07:10	496482	496561	15:59:27	0.02 %	41:05	1:11:58
sko56-3	56	164609.5	171963	3:08:16	166441.5	171860	8:57:50	169644	171032	16:22:56	0.82 %	1:00:39	1:53:27
sko56-4	56	302572.5	325803	2:55:51	306550.5	315175	9:00:52	312656	313497	15:17:25	0.27 %	52:24	1:26:44
sko56-5	56	575501.5	595593.5	2:56:20	582117.5	594477.5	8:57:53	591915.5	592335.5	17:46:46	0.07 %	1:08:30	1:34:20

Table 3. Results for well-known (SRFLP) instances with 36–56 facilities. n gives the number of facilities, "lb" denotes the lower bound, "blf" gives the objective value of the best layout found and "improve gap (SDP_0)" and "improve gap (SDP_1)" denote the running times that our algorithm based on relaxation (SDP_4) needs to improve on the gaps of the other two approaches. The running times are given in min:sec or in h:min:sec respectively.

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

Table 4. Results for well-known (SRFLP) instances with 60–100 facilities. n gives the number of facilities, "lb" denotes the lower bound, "blf" gives the objective value of the best layout found and "improve gap (SDP$_0$)" denotes the running times that our algorithm based on relaxation (SDP$_4$) needs to improve on the gaps of the approach by Anjos and Yen. The running times are given in h:min:sec.

References

1. A. R. S. Amaral. On the exact solution of a facility layout problem. *European Journal of Operational Research*, 173(2):508 – 518, 2006.
2. A. R. S. Amaral. An Exact Approach to the One-Dimensional Facility Layout Problem. *Operations Research*, 56(4):1026–1033, 2008.
3. A. R. S. Amaral. A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157(1):183–190, 2009.
4. A. R. S. Amaral and A. N. Letchford. A polyhedral approach to the single row facility layout problem. *In preparation, preprint available from http://www.optimization-online.org/ DB_FILE/2008/03/1931.pdf*, 2011.
5. 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.
6. M. F. Anjos and J. Lasserre, editors. *Handbook of Semidefinite, Conic and Polynomial Optimization*. International Series in Operations Research & Management Science. Springer-Verlag, to appear in 2011.
7. M. F. Anjos and F. Liers. Global approaches for facility layout and vlsi floorplanning. In M. F. Anjos and J. B. Lasserre, editors, *Handbook of Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*. To appear.
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. F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.
11. C. Buchheim, F. Liers, and M. Oswald. Speeding up ip-based algorithms for constrained quadratic 0–1 optimization. *Mathematical Programming*, 124:513–535, 2010.
12. C. Buchheim, A. Wiegele, and L. Zheng. Exact Algorithms for the Quadratic Linear Ordering Problem. *INFORMS Journal on Computing*, pages 168–177, 2009.
13. 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]*, 2011.
14. 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.
15. M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*, volume 15 of *Algorithms and Combinatorics*. Springer Verlag, Berlin, 1997.
16. 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.
17. M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, New York, 1974.
18. M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.
19. A. Gomes de Alvarenga, F. J. Negreiros-Gomes, and M. Mestria. Metaheuristic methods for a class of the facility layout problem. *Journal of Intelligent Manufacturing*, 11:421–430, 2000.
20. K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.
21. P. Hammer. Some network flow problems solved with pseudo-Boolean programming. *Operations Research*, 13:388–399, 1965.
22. S. S. Heragu and A. S. Alfa. Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research*, 57(2):190 – 202, 1992.
23. S. S. Heragu and A. Kusiak. Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research*, 36(2):258–268, 1988.
24. S. S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1):1 – 13, 1991.
25. J.-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and minimization algorithms (vol. 1 and 2)*. Springer, 1993.
26. P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming B*, 2011. accepted, preprint available at http://www.optimization-online.org/DB_HTML/2010/08/2696.html.
27. R. M. Karp and M. Held. Finite-state processes and dynamic programming. *SIAM Journal on Applied Mathematics*, 15(3):693–718, 1967.
28. K. R. Kumar, G. C. Hadjinicola, and T. li Lin. A heuristic procedure for the single-row facility layout problem. *European Journal of Operational Research*, 87(1):65 – 73, 1995.
29. L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.
30. R. F. Love and J. Y. Wong. On solving a one-dimensional space allocation problem with integer programming. *INFOR*, 14:139–143, 1967.

31. J.-C. Picard and M. Queyranne. On the one-dimensional space allocation problem. *Operations Research*, 29(2):371–391, 1981.
32. F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.
33. D. Romero and A. Sánchez-Flores. Methods for the one-dimensional space allocation problem. *Computers & Operations Research*, 17(5):465 – 473, 1990.
34. H. Samarghandi and K. Eshghi. An efficient tabu algorithm for the single row facility layout problem. *European Journal of Operational Research*, 205(1):98 – 105, 2010.
35. S. Sanjeevi and K. Kianfar. Note: A polyhedral study of triplet formulation for single row facility layout problem. *Discrete Appl. Math.*, 158:1861–1867, August 2010.
36. D. M. Simmons. One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17:812–826, 1969.
37. D. M. Simmons. A further note on one-dimensional space allocation. *Operations Research*, 19:249, 1971.
38. C. D. Simone. The cut polytope and the Boolean quadric polytope. *Discrete Mathematics*, 79(1):71–75, 1990.
39. J. Suryanarayanan, B. Golden, and Q. Wang. A new heuristic for the linear placement problem. *Computers & Operations Research*, 18(3):255 – 262, 1991.
40. H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer Academic Publishers, Boston, MA, 2000.