Les Cahiers du GERAD ISSN: 0711–2440

A Semidefinite Optimization Approach to Space-Free Multi-Row Facility Layout

P. Hungerländer
M.F. Anjos

G–2012–03

January 2012

A Semidefinite Optimization Approach to Space-Free Multi-Row Facility Layout

Philipp Hungerländer

Institute of Mathematics
Alpen-Adria Universität of Klagenfurt
A-9020 Klagenfurt, Austria
philipp.hungerlaender@uni-klu.ac.at

Miguel F. Anjos

GERAD & Canada Research Chair in Discrete Nonlinear Optimization in Engineering
École Polytechnique de Montréal
Montréal (Québec) Canada, H3C 3A7
anjos@stanfordalumni.org

January 2012

Les Cahiers du GERAD

G–2012–03

Abstract

Facility layout is a well-known operations research problem that arises in numerous areas of applications. The multi-row facility layout problem is concerned with placing departments along one or several rows so as to optimize objectives such as material handling and space usage. The particular cases of the single-row and double-row facility layout problems are of particular interest in the manufacturing context where materials flow between stations located along linear corridors. Significant progress has been made in recent years on solving single-row problems to global optimality using semidefinite optimization models. The contribution of this paper is the extension of the semidefinite programming approach to the special case of multi-row layout in which all the rows have a common left origin and no empty space is allowed between departments. We call this special case the space-free multi-row facility layout problem. Although this problem may seem overly restrictive, it is a relevant problem in several contexts such as in spine layout design. Our computational results show that for space-free double-row instances the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for instances with up to 15 departments. If the assignment of departments to rows is fixed, then bounds can be computed for instances with up to 70 departments.

Key Words: Facilities planning and design, Flexible manufacturing systems, Cell layout, Semidefinite Programming, Global Optimization.

Résumé

Le layout est un problème bien connu en recherche opérationnelle qui se pose dans de nombreux domaines d'applications. Le problème de layout en lignes doit placer des départements en une ou plusieurs rangées afin d'optimiser un objectif donné. Les cas particuliers de ligne simple et double sont d'un grand intérêt dans le contexte de la fabrication où des flux de matériaux ont lieu en couloirs. Des progrès significatifs ont été réalisés ces dernières années sur l'optimisation globale de layouts en une rangée en utilisant des modèles d'optimisation semi-définie. La contribution de cet article est l'extension de l'approche par programmation semi-définie au cas particulier de layout multi-lignes dans lequel toutes les lignes ont une origine commune et aucun espace vide n'est permis entre les départements. Nous appelons ce cas particulier le problème de layout multi-lignes sans espacement. Nos résultats montrent que l'approche proposée par optimisation semi-définie offre des bornes globales de haute qualité dans des délais raisonnables pour des instances avec un maximum de 15 départements. Si l'attribution des départements aux lignes est fixé d'avance, alors les bornes peuvent être calculées pour les instances avec un maximum de 70 départements.

Acknowledgments: Research of the second author was partially supported by the Natural Sciences and Engineering Research Council of Canada.

1 Introduction

Facility layout is concerned with the optimal location of departments inside a plant according to a given objective function. This is a well-known operations research problem that arises in different areas of applications. For example, in manufacturing systems, the placement of machines that form a production line inside a plant is a layout problem in which one wishes to minimize the total material flow cost. Another example arises in the design of Very Large Scale Integration (VLSI) circuits in electrical engineering. The objective of VLSI floorplanning is to arrange a set of rectangular modules on a rectangular chip area so that performance is optimized; this is a particular version of facility layout. In general, the objective function may reflect transportation costs, the construction cost of a material-handling system, the costs of laying communication wiring, or simply adjacency preferences among departments. Some facility layout problems are dynamic, meaning that the layout may have to change over time (e.g. due to expected changes in the production process).

The variety of applications means that facility layout encompasses a broad class of optimization problems. The survey paper [30] divides facility layout research into three broad categories. The first is concerned with models and algorithms for tackling different versions of the basic layout problem that asks for the optimal arrangement of a given number of departments within a facility so as to minimize the total expected cost of flows inside the facility. This includes the well-known special case of the quadratic assignment problem in which all the departments sizes are equal. The second category is concerned with extensions of unequal-areas layout that take into account additional issues that arise in real-world applications, such as designing dynamic layouts by taking time-dependency issues into account, designing layouts under uncertainty conditions, and computing layouts that optimize two or more objectives simultaneously. The third category is concerned with specially structured instances of the problem, such as the layout of machines along a production line. This paper is concerned with finding global upper and lower bounds for one such type of structured instances, namely the multi-row facility layout problem (MRFLP) in which the departments are to be placed so as to form one or more parallel rows.

Most facility layout problems have a strong combinatorial nature and turn out to be NP-hard. As such, numerous heuristic and metaheuristic approaches have been proposed for the various categories of problems, see e.g. [20]. However, few methods exist that provide global optimal solutions, or at least a measure of nearness to global optimality, for large instances of layout problems. One exception is the case of the single-row facility layout problem (SRFLP). This problem, sometimes called the one-dimensional space allocation problem [31], consists of finding the optimal location of rectangular departments next to each other along one row so as to minimize the total weighted sum of the center-to-center distances between all pairs of departments. It arises for example as the problem of ordering stations on a production line where the material flow is handled by an automated guided vehicle (AGV) travelling in both directions on a straight-line path. The SRFLP has interesting connections to other combinatorial optimization problems such as the maximum-cut problem, the quadratic linear ordering problem, and the linear arrangement problem. We refer the reader to [6] for more details.

Global optimization approaches for the SRFLP are based on relaxations of integer linear programming (ILP) or semidefinite programming (SDP) formulations. Semidefinite programming is the extension of linear programming (LP) from the cone of non-negative real vectors to the cone of symmetric positive semidefinite matrices. SDP includes LP as a special case, namely when all the matrices involved are diagonal. Several excellent solvers for SDP are now available. We refer the reader to the handbooks [37, 5] for a thorough coverage of the theory, algorithms and software in this area, as well as a discussion of many application areas where SDP has had a major impact.

The SRFLP is a special case of the more general MRFLP. Another particular case of the MRFLP is the double-row facility layout problem (DRFLP). The DRFLP is a natural extension of the SRFLP in the manufacturing context when one considers that an AGV can support stations located on both sides of its linear path of travel. This is a common approach in practice for improved material handling and space usage. Furthermore, since real factory layouts most often reduce to double-row problems or a combination of single-row and double-row problems, the DRFLP is especially relevant for real-world applications. A specific example of the application of the DRFLP is in spine layout design. Spine layouts, introduced by Tompkins [36], require departments to be located along both sides of specified corridors along which all the traffic between departments takes place. Although in general some spacing is allowed, layouts with no spacing are much preferable since spacing often translates into higher construction costs for the facility.

Somewhat surprisingly, the MRFLP and DRFLP have received only limited attention in the literature. In the 1980s Heragu and Kusiak [21, 22] proposed a non-linear programming model and obtained locally optimal solutions to SRFLPs and DRFLPs. Recently Chung and Tanchoco [14] (see also [38]) focused exclusively on the DRFLP and proposed a mixed-integer LP (MILP) formulation that was tested in conjunction with several heuristics for assigning the departments to the rows. Algorithms for spine layout design have been proposed, see e.g. [26], but to the best of our knowledge, there are no global optimization methods for spine layout.

In this paper, as a first step towards developing an SDP-based global optimization approach to the general MRFLP, we extend the SDP-based methodology for SRFLP originally proposed in [4] to the particular version of the MRFLP in which all the rows have a common left origin and no empty space is allowed between departments. We call it the *space-free multi-row facility layout problem (SF-MRFLP)*. This is an interesting special case, not only because it is relevant for the application of spine layout design, but also because it captures much of the inherent difficulty of layout problems. This difficulty is characterized by the large number of pairwise interactions between departments reflected by a high density of the objective function matrix in the SDP relaxation: generally speaking, more pairwise interactions lead to SDP approaches outperforming (M)ILP. Our computational results show that for space-free double-row instances the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for instances with up to 15 departments. If the assignment of departments to rows is fixed, then bounds can be computed for instances with up to 70 departments.

This paper is structured as follows. In Section 2 we introduce the SRFLP and discuss the issues to address in extending the SDP models from single-row to multi-row problems. In Section 3 we formally state the SF-MRFLP and present new formulations of it. The SDP relaxations are presented in Section 4 and a heuristic to obtain feasible layouts from the solutions of the SDP relaxations is proposed in Section 5. Computational results demonstrating the strength and potential of our SDP framework are presented in Section 6. Finally, conclusions and future research directions are given in Section 7.

2 From Single-Row to Multi-Row Layout

Our starting point are the most successful models for SRFLP. To introduce these, let $\pi = (\pi_1, \ldots, \pi_n)$ denote a permutation of the indices $[n] := \{1, 2, \ldots, n\}$ of the departments, so that the leftmost department is π_1, the department to the right of it is π_2, and so on, with π_n being the last department in the arrangement. Given a permutation π and two distinct departments i and j (and assuming that there is no space between the departments), the center-to-center distance between i and j is $\frac{1}{2}\ell_i + D_\pi(i,j) + \frac{1}{2}\ell_j$, where ℓ_i is the positive length of department i, and $D_\pi(i,j)$ denotes the sum of the lengths of the departments between i and j in the ordering defined by π. Solving the SRFLP consists of finding a permutation of $[n]$ that minimizes the weighted sum of the distances between all pairs of departments. In other words, the problem is:

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

where c_{ij} is the connectivity between departments i and j, and Π_n denotes the set of all permutations of $[n]$. Since the lengths of the departments are constant, it is clear that the crux of the problem is to minimize $\sum_{i<j} c_{ij} D_\pi(i,j)$ over all permutations $\pi \in \Pi_n$.

The key information to express the quantity $D_\pi(i,j)$ can be encoded using *betweenness variables*. These are $\binom{n}{3}$ binary variables $\zeta_{ijk},\ i,j,k \in [n],\ \ i < j, i \neq k \neq j$ defined by:

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

Not all possible combinations of values of the variables ζ_{ijk} correspond to permutations of $[n]$. Specifically, given three departments i, j, and k, exactly one of them must be located between the other two. This fact is expressed using the following equations:

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

Anjos and Yen [8] show that these equations precisely characterize the combinations of values of the variables ζ_{ijk} that describe permutations of $[n]$.

We collect all the betweenness variables in a vector ζ. Since every permutation $\pi \in \Pi_n$ can be encoded as one such vector ζ, we express the center-to-center distance between departments i and j as

$$D_\pi(i,j) = \sum_{k\in[n],\ i\neq k\neq j} \ell_k \zeta_{ijk}, \quad i,j \in [n],\ i<j,$$

and hence express SRFLP as

$$\sum_{i,j\in[n],\ i<j} \frac{c_{ij}}{2}(\ell_i + \ell_j) + \min_{\zeta\in\mathcal{P}^n_{Btw}} \sum_{i,j\in[n],\ i<j} c_{ij}\left(\sum_{k\in[n],\ i\neq k\neq j} \ell_k \zeta_{ijk}\right),$$

where $\mathcal{P}^n_{Btw}$ is the betweenness polytope:

$$\mathcal{P}^n_{Btw} := \operatorname{conv}\{\zeta \ :\ \zeta \in \{0,1\}^n \text{ and } \zeta_{ijk} + \zeta_{ikj} + \zeta_{jki} = 1, \quad i,j,k \in [n],\ i<j<k\}.$$

Sanjeevi and Kianfar [33] show that the equations (1) describe the smallest linear subspace that contains $\mathcal{P}^n_{Btw}$. Buchheim et al. [9] proved a similar result in the context of quadratic linear ordering problems.

The betweenness polytope is the structure common to most of the recent LP and SDP relaxations for the SRFLP. One visible difference between these two approaches is that SDP approaches define the binary variables in terms of $\{-1,1\}$ instead of $\{0,1\}$. In fact this makes no difference: Helmberg [18] proved that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of binary problems in such a way that the resulting bounds remain the same and structural properties are preserved.

The SDP relaxation proposed in Anjos et al. [4] was used by Anjos and Vannelli [7] to solve SRFLPs with up to 30 departments to global optimality. This was improved on by Amaral [3] who used an LP relaxation of $\mathcal{P}^n_{Btw}$ to solve instances with up to 35 departments. Global lower bounds for very large instances with up to 100 departments were provided by Anjos and Yen [8] using a modified SDP relaxation. More recently, Hungerländer and Rendl [24] provided global optimal solutions for instances with up to 42 departments, and tighter bounds than the Anjos-Yen relaxation for instances with up to 100 departments. The relaxation in [24] achieved the best practical performance to date among all approaches for the SRFLP.

While these approaches work extremely well for the SRFLP, none of them can be applied directly to the MRFLP. This is because there are three modeling issues that arise in the MRFLP but not in the SRFLP:

1. Assigning each department to exactly one row;
2. Expressing the center-to-center distance between departments assigned to different rows;
3. Handling the possibility of empty space between departments.

The fundamental limitation is that the betweenness variables used for the state-of-the-art LP and SDP relaxations are not sufficient to capture these issues.

Models for the DRFLP have been proposed by two groups of authors. Heragu and Kusiak [21] proposed a non-linear programming model that provides locally optimal solutions to SRFLPs and DRFLPs, while Chung and Tanchoco [14] (see also [38]) used a MILP formulation for the DRFLP. The latter approach is only able to provide global optimal solutions for DRFLPs of small sizes. For larger instances, locally optimal solutions were obtained by using the MILP formulation in conjunction with heuristics for assigning the departments to the rows in advance.

This paper proposes an SDP-based model that can provide global upper and lower bounds, and in some cases global optimal solutions, for the SF-MRFLP. The proposed model extends the tight SDP relaxations in [4, 24] and the algorithmic approaches in [24] to the SF-MRFLP. Our SDP relaxations further assume that each department is already assigned to one of the rows but this restriction is overcome by optimizing the relaxations over all row assignments or, for large instances, over a chosen subset of assignments. Our computational results show that our approach obtains high-quality bounds for instances of SF-MRFLP with up to 70 departments in reasonable computational time. The issue of allowing empty space between departments will be addressed in future research.

3 Formulations of the Space-Free Multi-Row Facility Layout Problem

An instance of the SF-MRFLP consists of n one-dimensional departments with given positive lengths $\ell_1, \ldots,$ ℓ_n, pairwise connectivities c_{ij} and a function $r: [n] \to \mathcal{R}$ that assigns each department to one of the m rows $\mathcal{R} := \{1, \ldots, m\}$. The objective is to find permutations $\pi^1 \in \Pi^1, \ldots, \pi^m \in \Pi^m$ of the departments within the rows such that the total weighted sum of the center-to-center distances between all pairs of departments (with a common left origin) is minimized:

$$\min_{\Pi^1,\ldots,\Pi^m} \sum_{i,j\in[n], i\neq j} c_{ij} z_{ij}^{\pi^{r(i)},\pi^{r(j)}}, \qquad \text{(SF-MRFLP)}$$

where $\Pi^1 \times \ldots \times \Pi^m$ denotes the set of all feasible layouts and $z_{ij}^{\pi^{r(i)},\pi^{r(j)}}$ denotes the horizontal distance between the centroids of departments i and j in the layout $\pi^1 \times \ldots \times \pi^m$.

We define the m-row betweenness polytope

$$\mathcal{P}_{Btw}^{n,m} := \text{conv}\,\{\zeta \;:\; \zeta \text{ represents orderings of the } n \text{ departments on the } m \text{ rows}\},$$

and introduce the binary ordering variables $y_{ij},\ i,j \in [n],\ r(i) = r(j),\ i < j$:

$$y_{ij} = \begin{cases} 1, & \text{if department } i \text{ lies before department } j \\ -1, & \text{otherwise.} \end{cases} \tag{2}$$

We can express the betweenness variables ζ as quadratic terms in ordering variables:

$$\zeta_{ijk} = \frac{1 - y_{ik}y_{jk}}{2}, \qquad \zeta_{ikj} = \frac{1 + y_{ij}y_{jk}}{2}, \qquad \zeta_{jki} = \frac{1 - y_{ij}y_{ik}}{2}, \tag{3}$$

for $i,j,k \in [n],\ r(i) = r(j) = r(k),\ i < j < k$, and thus rewrite (1) (generalized for m rows) as

$$y_{ij}y_{jk} - y_{ij}y_{ik} - y_{ik}y_{jk} = -1, \quad i,j,k \in [n],\ r(i) = r(j) = r(k),\ i < j < k. \tag{4}$$

It was shown in [12, 13] that (4) describes the smallest linear subspace that contains the multi-level quadratic ordering polytope

$$\mathcal{P}_{MQO} := \text{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix}\begin{pmatrix} 1 \\ y \end{pmatrix}^{\top} \;:\; y \in \{-1,1\},\ y \text{ satisfies (4)} \right\},$$

where y is a vector collecting the ordering variables. We can also use y to express the center-to-center distances of pairs of departments $i,j \in [n],\ i < j$:

$$z_{ij}^{y} = \begin{cases} D_{ij}, & r(i) = r(j) \\ |d_{ij}|, & r(i) \neq r(j) \end{cases} \tag{5}$$

where

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

and

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

To linearize the absolute value in (5), we introduce binary ordering variables $x_{ij},\ i,j \in [n],\ r(i) \neq r(j), i < j$ for departments in different rows

$$x_{ij} = \begin{cases} 1, & \text{if the center of department } i \text{ lies before the center of department } j \\ -1, & \text{otherwise,} \end{cases}$$

and let x be the vector collecting these linear ordering variables. The following constraints have to hold for x:

$$x_{ij} d_{ij} \leq 0, \quad i,j \in [n],\ r(i) \neq r(j),\ i < j. \tag{6}$$

We can thus rewrite (5) as

$$z_{ij}^{x,y} = \begin{cases} D_{ij}, & r(i) = r(j) \\ -x_{ij} d_{ij}, & r(i) \neq r(j) \end{cases} \tag{7}$$

for all $i,j \in [n],\ i < j$. Now we can rewrite the objective function of SF-MRFLP in terms of x and y:

$$\sum_{i,j\in[n], i<j} c_{ij} z_{ij}^{x,y}. \tag{8}$$

In summary we have deduced a second formulation of SF-MRFLP.

Theorem 1 *Minimizing* (8) *over* $x, y \in \{-1, 1\}$, (4) *and* (6) *solves SF-MRFLP.*

Proof. The equations (4) together with the integrality conditions on y suffice to induce all feasible layouts within the rows. The integrality conditions on x together with (6) ensure that we incorporate the distances between departments in different rows with the correct sign in the objective function. □

Next we rewrite (8) in terms of matrices and obtain a matrix-based fomulation:

$$\min \left\{ K + 2c_x^\top x + 2\langle C_V, V\rangle + \langle C_Y, Y\rangle \ :\ x, y \in \{-1, 1\},\ (x,y) \text{ satisfy (4) and (6)} \right\}, \tag{SF-MRFLP}$$

where $K := \frac{1}{2}\sum_{h\in\mathcal{R}} \left[\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} c_{ij}\right)\left(\sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)=h}} \ell_i\right)\right]$, $Y := yy^\top$, $V := xy^\top$. The cost vector c_x and the cost matrices C_Y and C_V are deduced by equating the coefficients of the following equations:

$$2\langle C_Y, Y\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)=r(j)}} c_{ij} \left(\sum_{\substack{k\in[n],\ i<k<j,\\ r(k)=r(i)}} y_{ik}y_{kj}\ell_k - \sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} y_{ki}y_{kj}\ell_k - \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(i)}} y_{ki}y_{kj}\ell_k \right),$$

$$4\langle C_V, V\rangle \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)\neq r(j)}} c_{ij}x_{ij} \left(\sum_{\substack{k\in[n],\ k<i,\\ r(k)=r(i)}} \ell_k y_{ki} - \sum_{\substack{k\in[n],\ k>i,\\ r(k)=r(i)}} \ell_k y_{ik} - \sum_{\substack{k\in[n],\ k<j,\\ r(k)=r(j)}} \ell_k y_{kj} + \sum_{\substack{k\in[n],\ k>j,\\ r(k)=r(j)}} \ell_k y_{jk} \right),$$

$$4c_x^\top x \overset{!}{=} \sum_{\substack{i,j\in[n],\ i<j,\\ r(i)\neq r(j)}} c_{ij}x_{ij}\left(L_{r(i)} - L_{r(j)}\right),$$

where L_i denotes the sum of the length of the departments on row i:

$$L_i = \sum_{k\in[n],\ r(k)=i} \ell_k, \quad i \in \mathcal{R}.$$

In the following section we use matrix-based relaxations to get tight lower bounds for SF-MRFLP.

4 Semidefinite Relaxations

We collect the ordering variables in a vector

$$w := \begin{pmatrix} x \\ y \end{pmatrix},$$

and consider the matrix variable $W = ww^\top$. Our object of interest is the multi-row ordering polytope

$$\mathcal{P}_{MRO} := \operatorname{conv}\left\{ \begin{pmatrix} 1 \\ w \end{pmatrix}\begin{pmatrix} 1 \\ w \end{pmatrix}^\top \ : \ w \in \{-1,1\}, \ w \text{ satisfies (4) and (6)} \right\},$$

We apply standard techniques to construct SDP relaxations. First we relax the nonconvex equation $W - ww^\top = 0$ to the positive semidefinite constraint

$$W - ww^\top \succcurlyeq 0.$$

Moreover, the main diagonal entries of W correspond to squared $\{-1,1\}$ variables, hence $\operatorname{diag}(W) = e$, the vector of all ones. To simplify notation let us introduce

$$Z = Z(w,W) := \begin{pmatrix} 1 & w^\top \\ w & W \end{pmatrix}, \tag{9}$$

where $\dim(Z) = \binom{n}{2} + 1 =: \Delta$. By the Schur complement theorem, $W - ww^\top \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. We therefore conclude that $\mathcal{P}_{MRO}$ is contained in the elliptope

$$\mathcal{E} := \{ \ Z : \ \operatorname{diag}(Z) = e, \ Z \succcurlyeq 0 \ \}, \tag{10}$$

which is studied in detail by Laurent and Poljak [27, 28].

Next we can formulate SF-MRFLP as a semidefinite optimization problem in binary variables.

Theorem 2 *The problem*

$$\min \ \{ \ K + \langle C_Z, Z\rangle : Z \text{ satisfies (4) and (6)}, \ Z \in \mathcal{E}, \ w \in \{-1,1\} \ \}$$

where the cost matrix C_Z is given by

$$C_Z := \begin{pmatrix} 0 & c_x & 0 \\ c_x & 0 & C_V \\ 0 & C_V & C_Y \end{pmatrix},$$

is equivalent to SF-MRFLP.

Proof. Since $w_i^2 = 1, \ i \in \{1,\dots,\Delta-1\}$ we have $\operatorname{diag}(W - ww^\top) = 0$, which together with $W - ww^\top \succcurlyeq 0$ shows that in fact $W = ww^\top$ is integral. By Theorem 1, integrality on W together with (4) and (6) suffice to describe SF-MRFLP. □

Dropping the integrality condition on the first row and column of Z yields the basic semidefinite relaxation of SF-MRFLP:

$$\min \ \{K + \langle C_Z, Z\rangle : Z \text{ satisfies (4) and (6)}, \ Z \in \mathcal{E} \ \}. \tag{SDP$_{\texttt{basic}}$}$$

There are several ways to tighten the above relaxation. This is the topic of the next two subsections.

4.1 Tightening the Semidefinite Relaxation by Exploiting Binarity and Ordering Properites

Every matrix $Z \in \mathcal{E}$ with $\{-1,1\}$ entries also belongs to the metric polytope $\mathcal{M}$:

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \leq e, \;\; 1 \leq i < j < k \leq \Delta \right\}. \tag{11}$$

We note that $\mathcal{M}$ is defined through $4\binom{\Delta}{3} \approx \frac{1}{12}n^6$ facets. They are the triangle inequalities of the max-cut polytope, see e.g. [15].

We can impose a transitivity relation on all ordering variables by imposing the 3-cycle inequalities on w:

$$|w_{ij} + w_{jk} - w_{ik}| = 1, \quad i,j,k \in [n], \; i < j < k.$$

These may rule out some optimal solutions but preserve at least one optimal solution. Squaring them we obtain the 3-cycle equalities

$$w_{ij}w_{jk} - w_{ij}w_{ik} - w_{ik}w_{jk} = -1, \quad i,j,k \in [n], \; i < j < k, \tag{12}$$

which are a strengthening of (4) and together with $Z \succcurlyeq 0$ also ensure the 3-cycle inequalities on w [23, Proposition 4.2].

Another generic improvement was proposed by Lovász and Schrijver [29]. Applied to our problem, their approach suggests to multiply the 3-cycle inequalities

$$1 - w_{ij} - w_{jk} + w_{ik} \geq 0, \qquad 1 + w_{ij} + w_{jk} - w_{ik} \geq 0. \tag{13}$$

by the nonnegative expressions

$$1 - w_{lo} \geq 0, \qquad 1 + w_{lo} \geq 0, \qquad l, o \in [n], \; l < o. \tag{14}$$

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

$$\begin{aligned} -1 - w_{lo} &\leq w_{ij} + w_{jk} - w_{ik} + w_{ij,lo} + w_{jk,lo} - w_{ik,lo} \leq 1 + w_{lo}, \\ -1 + w_{lo} &\leq w_{ij} + w_{jk} - w_{ik} - w_{ij,lo} - w_{jk,lo} + w_{ik,lo} \leq 1 - w_{lo}, \end{aligned} \tag{15}$$

for $i,j,k,l,o \in [n], \; i < j < k, \; l < o$. We define the corresponding polytope $\mathcal{LS}$:

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

We can also deduce lower and upper bounds on the sum of the inter-row ordering variables for each of the $\binom{m}{2}$ pairs of rows:

$$\delta_l^{c_1,c_2} \leq \sum_{\substack{i,j \in [n], i<j, \\ r(i)=c_1 \neq c_2 = r(j)}} x_{ij} \leq \delta_u^{c_1,c_2}, \quad c_1, c_2 \in \mathcal{R}, \; c_1 < c_2, \tag{17}$$

where $\delta_l^{c_1,c_2}$ and $\delta_u^{c_1,c_2}$ are dependent on the given data. We obtain $\delta_l^{c_1,c_2}$ by sorting the departments in row c_1 by decreasing length and the departments in row c_2 by increasing length, then computing the sum of the inter-row ordering variables. Analogously we compute $\delta_u^{c_1,c_2}$ by sorting the departments in row c_1 by increasing length and the departments in row c_2 by decreasing length. If two centers are located exactly below each other we break the symmetry and tighten the bounds by setting the respective variables to +1 in the first case and to -1 in the second case. Thus it can happen that $\delta_l^{c_1,c_2} > \delta_u^{c_1,c_2}$; in this case we set $\delta_l^{c_1,c_2} = \delta_u^{c_1,c_2} := \frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2} + \left(\frac{\delta_l^{c_1,c_2} + \delta_u^{c_1,c_2}}{2} \bmod 2 \right)$ to preserve an optimal solution.

Fact 1 *The constraints* (17) *can rule out some optimal solutions but preserve at least one optimal solution and thus are valid for tightening the semidefinite relaxation.*

We can rewrite (17) as

$$\frac{\delta_l^{c_1,c_2}-\delta_u^{c_1,c_2}}{2} \le \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \frac{\delta_l^{c_1,c_2}+\delta_u^{c_1,c_2}}{2} \le \frac{\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2}}{2}, \quad c_1,c_2\in\mathcal{R},\ c_1<c_2.$$

Squaring yields

$$0 \le \left(\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij}\right)^2 - (\delta_l^{c_1,c_2}+\delta_u^{c_1,c_2}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} + \left(\frac{\delta_l^{c_1,c_2}+\delta_u^{c_1,c_2}}{2}\right)^2 \le \left(\frac{\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2}}{2}\right)^2 \tag{18a}$$

if $(\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2}) \bmod 4 = 0$, and

$$1 \le \left(\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij}\right)^2 - (\delta_l^{c_1,c_2}+\delta_u^{c_1,c_2}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} + \left(\frac{\delta_l^{c_1,c_2}+\delta_u^{c_1,c_2}}{2}\right)^2 \le \left(\frac{\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2}}{2}\right)^2 \tag{18b}$$

if $(\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2}) \bmod 4 = 2$.

To obtain the lower bound we exploit the fact that the inter-row ordering variables are $\{-1,1\}$. Hence if $\delta_u^{c_1,c_2}-\delta_l^{c_1,c_2} \le 2$ then (18) defines equalities on the sum of products of inter-row variables.

Fact 2 *The smallest subspace containing the multi-row polytope is defined by* (4)*, but we build our semidefinite relaxation on an even smaller subspace that contains at least one optimal solution. This subspace is defined by* (12) *and possibly additional equations from* (17) *and* (18)*.*

We can also multiply

$$\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2} \ge 0, \qquad \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \ge 0,$$

by (13) and (14). This results in the following inequalities

$$\begin{gathered}
\delta_l^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le w_{lo} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - w_{lo}\delta_l^{c_1,c_2} \le \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2},\\
\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_u^{c_1,c_2} \le w_{lo}\delta_u^{c_1,c_2} - w_{lo} \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij},
\end{gathered} \tag{19}$$

for $c_1,c_2\in\mathcal{R},\ c_1<c_2,\ l,o\in[n],\ l<o$ and

$$\begin{gathered}
\delta_l^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le (w_{kl}+w_{lo}-w_{ko}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - (w_{kl}+w_{lo}-w_{ko})\delta_l^{c_1,c_2} \le \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_l^{c_1,c_2},\\
\sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} - \delta_u^{c_1,c_2} \le (w_{kl}+w_{lo}-w_{ko})\delta_u^{c_1,c_2} - (w_{kl}+w_{lo}-w_{ko}) \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij} \le \delta_u^{c_1,c_2} - \sum_{\substack{i,j\in[n],i<j,\\ r(i)=c_1\neq c_2=r(j)}} x_{ij},
\end{gathered} \tag{20}$$

for $c_1,c_2\in\mathcal{R},\ c_1<c_2,\ k,l,o\in[n],\ k<l<o$.

We gather the $O(n^2m^2)$ inequalities based on the bounds on the sum of the inter-row variables and define the polytope $\mathcal{SI}$:

$$\mathcal{SI} := \{\ Z : Z \text{ satisfies } (17) - (20)\ \}. \tag{21}$$

4.2 Tightening the Semidefinite Relaxation through the Distance Variables

We now turn our attention to the distance variables. We can tighten constraints (6) as follows:

$$x_{ij}d_{ij} \le d_{ij}, \quad x_{ij}d_{ij} \le -d_{ij}, \qquad i,j \in [n],\ r(i) \ne r(j),\ i<j, \tag{22}$$

which holds true for $x_{ij} \in \{-1, 1\}$. Similar constraints also hold for the intra-row variables y:

$$y_{ij}d_{ij} \le d_{ij}, \quad y_{ij}d_{ij} \le -d_{ij}, \qquad i,j \in [n],\ r(i) = r(j),\ i<j. \tag{23}$$

We can gather (22) and (23) together into a single expression in terms of w:

$$w_{ij}d_{ij} \le d_{ij}, \quad w_{ij}d_{ij} \le -d_{ij}, \qquad i,j \in [n],\ i<j. \tag{24}$$

Fact 3 *The constraints* (24) *are valid.*

For $i,j \in [n],\ r(i) = r(j),\ i<j$, the expressions $y_{ij}d_{ij}$ and D_{ij} represent different ways to express the distance between two departments within the same row. They are equivalent for integral y but when we look at the semidefinite relaxation, D_{ij} is clearly preferable because it only involves variables that appear in a 3-cycle equality (12) and thus are more tightly constrained in the relaxation. This fact also explains why the various linear and semidefinite relaxations for SRFLP that include these equations on the betweenness variables produce such tight bounds even for very large instances. That is why we expressed the intra-row distances in the objective function (8) via D_{ij}.

Another class of constraints are the triangle inequalities relating the distances between three departments, where we use again D_{ij} to measure the intra-row distances

$$D_{ij} + D_{ik} \ge D_{jk}, \quad D_{ij} + D_{jk} \ge D_{ik}, \quad D_{jk} + D_{ik} \ge D_{ij}, \qquad i<j<k \in [n],\ r(i)=r(j)=r(k), \tag{25a}$$

$$x_{ik}d_{ik} + x_{jk}d_{jk} \ge D_{ij}, \quad x_{ik}d_{ik} + D_{ij} \ge x_{jk}d_{jk}, \quad D_{ij} + x_{jk}d_{jk} \ge x_{ik}d_{ik}, \qquad i<j<k \in [n],\ r(i)=r(j),\ r(i)\ne r(k), \tag{25b}$$

$$x_{ij}d_{ij} + x_{ik}d_{ik} \ge D_{jk}, \quad x_{ij}d_{ij} + D_{jk} \ge x_{ik}d_{ik}, \quad D_{jk} + x_{ik}d_{ik} \ge x_{ij}d_{ij}, \qquad i<j<k \in [n],\ r(i)\ne r(j),\ r(j)=r(k), \tag{25c}$$

$$x_{ij}d_{ij} + x_{ik}d_{ik} \ge x_{jk}d_{jk}, \quad x_{ij}d_{ij} + x_{jk}d_{jk} \ge x_{ik}d_{ik}, \quad x_{jk}d_{jk} + x_{ik}d_{ik} \ge x_{ij}d_{ij}, \qquad i<j<k \in [n],\ r(i)\ne r(j)\ne r(k)\ne r(i). \tag{25d}$$

For the exact SF-MRFLP formulation these constraints implicitly hold.

Fact 4 *The constraints* (25) *are valid.*

Furthermore these constraints imply the distance constraints for more than three departments.

Theorem 3 *The triangle inequalities* (25) *imply all the distance constraints involving more than three departments.*

Proof. Consider a general distance constraint for $\gamma > 3$ departments

$$\sum_{h=1}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}} \ge w_{i_1 i_\gamma} d_{i_1 i_\gamma},$$

where

$$w_{i_h i_{h+1}} d_{i_h i_{h+1}} := \begin{cases} D_{i_h i_{h+1}}, & r(h) = r(h+1), \\ x_{i_h i_{h+1}} d_{i_h i_{h+1}}, & r(h) \ne r(h+1). \end{cases}$$

We show that (25) implies the above inequality. We start out with the left hand side of the inequality and use $w_{i_1 i_2} d_{i_1 i_2} + w_{i_2 i_3} d_{i_2 i_3} \ge w_{i_1 i_3} d_{i_1 i_3}$ to obtain

$$w_{i_1 i_2} d_{i_1 i_2} + w_{i_2 i_3} d_{i_2 i_3} + \sum_{h=3}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}} \ge w_{i_1 i_3} d_{i_1 i_3} + \sum_{h=3}^{\gamma-1} w_{i_h i_{h+1}} d_{i_h i_{h+1}}.$$

Next we use $w_{i_1i_3}d_{i_1i_3} + w_{i_3i_4}d_{i_3i_4} \geq w_{i_1i_4}d_{i_1i_4}$ in the same fashion. The process can be repeated resulting in the chain of inequalities

$$\sum_{h=1}^{\gamma-1} w_{i_hi_{h+1}}d_{i_hi_{h+1}} \geq \ldots \geq w_{i_1,i_{\gamma-1}}d_{i_1,i_{\gamma-1}} + w_{i_{\gamma-1}i_\gamma}d_{i_{\gamma-1}i_\gamma} \geq w_{i_1i_\gamma}d_{i_1i_\gamma}.$$

$\square$

Hence we define the polytope

$$\mathcal{DV} := \{\ Z : Z \text{ satisfies } (25)\ \} \tag{26}$$

using the $3\binom{n}{3}$ triangle inequalities relating the distances between 3 or more departments.

4.3 Tightest Semidefinite Relaxation and Solution Methodology

Gathering all the results in Section 4, we get the following relaxation of $\mathcal{P}_{MRO}$:

$$\min\left\{K + \langle C_Z, Z\rangle : Z \text{ satisfies (12) and (24)},\ Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS} \cap \mathcal{SI} \cap \mathcal{DV})\right\}. \tag{$\texttt{SDP}_{\texttt{full}}$}$$

While theoretically tractable, it is clear that ($\texttt{SDP}_{\texttt{full}}$) has an impractically large number of constraints. Indeed, even including only $O(n^3)$ constraints is not realistic for instances of size $n \geq 20$. For this reason, we adopt an approach originally suggested in [16] and since then applied to the max-cut problem [32] and several ordering problems [25, 12, 10, 11]. Initially, we only explicitly ensure that Z lies in the elliptope $\mathcal{E}$. This can be achieved efficiently with standard interior-point methods, see e.g. [19]. All other constraints are handled through Lagrangian duality.

For notational convenience, let us formally denote the equations in ($\texttt{SDP}_{\texttt{full}}$) by $e - \mathcal{A}(Z) = 0$. Similarly we write the inequalities in ($\texttt{SDP}_{\texttt{full}}$) as $g - \mathcal{D}(Z) \geq 0$. Using the Lagrangian

$$\mathcal{L}(Z, \lambda, \mu) := \langle C, Z\rangle + \lambda^\top(e - \mathcal{A}(Z)) + \mu^\top(g - \mathcal{D}(Z)),$$

we obtain the partial Lagrangian dual

$$f(\lambda, \mu) := \min_{Z\in\mathcal{B}} \mathcal{L}(Z, \lambda, \mu) = e^\top\lambda + g^\top\mu + \min_{Z\in\mathcal{B}}\langle C - \mathcal{A}^\top(\lambda) - \mathcal{D}^\top(\mu), Z\rangle.$$

Since ($\texttt{SDP}_{\texttt{full}}$) has strictly feasible points, strong duality holds and we can solve the relaxation through $\max_{\mu\geq 0,\lambda} f(\lambda, \mu)$.

The function f is well-known to be convex but non-smooth. For a given feasible point (λ, μ) the evaluation of $f(\lambda, \mu)$ amounts to optimizing over $\mathcal{E}$. We do this using a primal-dual interior-point method which also provides a primal feasible $Z_{\lambda,\mu}$ yielding a subgradient of f. Using these ingredients, we get an approximate minimizer of f using the bundle method [16]. Thanks to the use of the bundle method, we quickly obtain a good initial set of constraints. On the other hand, since the rate of convergence is slow, we limit the number of function evaluations to control the overall computational effort. These evaluations nevertheless constitute the computational bottleneck for larger instances as there they are responsible for more than 95% of the required running time. Detailed computational results are given in Section 6.

We next describe how a feasible layout can be obtained from a solution to any of the SDP relaxations.

5 Obtaining Feasible Layouts

To obtain feasible layouts, we apply the hyperplane rounding algorithm of Goemans-Williamson [17] to the solution of the SDP relaxation. We take the resulting vector $\overline{w}$ and flip the signs of some of its entries to make it feasible with respect to the 3-cycle inequalities

$$|\overline{y}_{ij} + \overline{y}_{jk} - \overline{y}_{ik}| = 1 \tag{27}$$

within the rows and the inequalities for the inter-row variables (6). Computational experiments demonstrated that this repair strategy is not as critical as one might assume. For example, in multi-level crossing minimization this SDP rounding heuristic clearly dominates traditional heuristic approaches [12].

Let us give a more detailed description of the implementation of our heuristic. We consider a vector w' that encodes a feasible layout of the departments in all rows. The algorithm stops after 100 executions of step 2. (Note that before the 51st execution of step 2, we perform step 1 again. As step 1 is quite expensive, we refrain from executing it too often.)

1. Let W'' be the current primal (fractional) solution of $(\mathtt{SDP}_{\mathtt{full}})$ (or some other semidefinite relaxation) obtained by the bundle method or an interior-point solver. Compute the convex combination $R := \lambda(w'w'^\top) + (1-\lambda)W''$ using a randomly generated $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^\top$ of R.
2. Apply Goemans-Williamson hyperplane rounding to D and obtain a $-1/+1$ vector $\overline{w}$ (cf. [32]).
3. Compute the induced objective value $z(\overline{w}) := \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}^\top C_Z \begin{pmatrix} 1 \\ \overline{w} \end{pmatrix}$. If $z(\overline{w}) \geq z(w')$: go to step 2.
4. If $\overline{w}$ satisfies (27) and (6): set $w' := \overline{w}$ and go to 2. Else: modify $\overline{w}$ by first changing the signs of one of three variables in all violated 3-cycle inequalities, afterwards flipping signs of the inter-row ordering variables to satisfy (6) and go to step 3.

The final w' is the heuristic solution. If the duality gap is not closed after the heuristic, we return to the SDP optimization algorithm and then retry the heuristic (retaining the last vector w').

6 Computational Experience

We report the results for different computational experiments with our semidefinite relaxations. All computations were conducted on an Intel Xeon E5160 (Dual-Core) with 24 GB RAM, running Debian 5.0 in 64-bit mode. The algorithm was implemented in Matlab 7.7.

We define DRFLP instances using the data from SRFLP instances in the literature as well as data randomly generated in the same way as in [24], namely with a density of 50% and with lengths and connectivities varying randomly between 1 and 10.

Tables 1 and 9 give the characteristics of the SRFLP instances that we considered. These include well-known benchmark instances from [22, 34, 1, 2, 3], randomly generated instances from [7, 24], and instances with clearance requirements from [21]. All the instances can be downloaded from http://anjos.mgi.polymtl.ca/flplib. We use the latter without taking the clearance requirement into account, hence we could round on 5 as the lengths of the departments are multiples of 10. In general, while for the SRFLP we can round to the nearest integer because 0.5 can only occur in the constant term, for the MRFLP we can round the lower bound only to 0.5 as the inter-row distances do not have to be integer.

For each instance considered, our computational objective is to obtain the best possible solution for a placement of the departments in two rows. We report results only for the two-row case for two reasons: the high computational costs involved in solving the SDP relaxations, and the fact that double-row problems are the most common in practice. Nevertheless we emphasize that our relaxations and methodology are applicable to MRFLP instances with any given number of rows.

6.1 Global Optimization of Small Instances Using $(\mathtt{SDP}_{\mathtt{full}})$

For small DRFLP instances, the relaxation $(\mathtt{SDP}_{\mathtt{full}})$ can be solved for each of the $2^{n-1} - 1$ possible row assignments. From the obtained bounds, we can deduce global upper and lower bounds: these are the minima of all upper and lower bounds respectively. We restricted the running time per instance to 24 hours. The upper bounds were obtained using the SDP rounding heuristic in Section 5. The results are summarized in Table 2. We point out that the lower and upper bound were often obtained from different row assignments.

Table 1: Characteristics of smaller instances with between 5 and 15 departments

Instance	Source	Size (n)	SRFLP Optimal SRFLP solution	Time (sec) [24]
S_8	[34]	8	801	0.6
SH_8	[34]	8	2324.5	2.3
S_9	[34]	9	2469.5	0.7
SH_9	[34]	9	4695.5	9.2
S_10	[34]	10	2781.5	0.6
S_11	[34]	11	6933.5	1.3
H_5	[22]	5	800	0.1
H_6	[22]	6	1480	0.1
H_7	[22]	7	3680	0.6
H_8	[22]	8	4725	0.4
H_12	[22]	12	17945	7.9
H_15	[22]	15	45840	19.6
Rand_5	new	5	147.5	0.1
Rand_6	new	6	420	0.4
Rand_7	new	7	344	0.3
Rand_8	new	8	382	1.3
Rand_9	new	9	1024.5	2.2
Rand_10	new	10	1697	3.1
Rand_11	new	11	1564	2.0
Rand_12	new	12	2088	8.4
Rand_13	new	13	3101.5	7.8
Rand_14	new	14	3653	17.9
Rand_15	new	15	5345.5	19.2
P_15	[1]	15	6305	19.7

Table 2: Computational results for ($\texttt{SDP}_{\texttt{full}}$)

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems			Computational statistics		
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time w/ bundle (sec)	Total time w/o bundle (sec)
H_5	420	450	7.14	7.14	1.54	7	144.2	9.6	8.7
Rand_5	52.5	52.5	0	7.65	1.03	10	141.3	7.9	9.0
H_6	703	720	2.42	10.96	3.30	7	477.4	130.3	384.8
Rand_6	189.5	190.5	0.53	6.18	1.70	7	424.8	94.2	317.6
H_7	1639.5	1700	3.69	14.05	2.69	6	1205.2	6177.5	47619.6
Rand_7	166	166	0	13.21	2.20	16	1016.0	2704.0	19448.3

Looking at the running times and their growth rates, we deduce that this approach is realistic only for instances with fewer than 8 departments within the 24-hour time limit.

The last two columns of Table 2 illustrate the impact of an important computational strategy. We start with the basic relaxation:

$$\min \{K + \langle C_Z, Z\rangle : Z \text{ satisfies } (12) \text{ and } (24),\ Z \in \mathcal{E}\}. \qquad (\texttt{SDP}_{\texttt{basic}})$$

For the results labelled "w/ bundle" we used 10 function evaluations of the bundle method and 3 constraint updates to obtain an initial set of constraints to add to the relaxation ($\texttt{SDP}_{\texttt{basic}}$). We then solved the resulting relaxation using Sedumi [35]; added violated inequality constraints (from all the inequalities in ($\texttt{SDP}_{\texttt{full}}$)); solved again using Sedumi; and repeated this process until no more violations were found. Alternatively one can skip the search for an initial set of inequalities using the bundle method and proceed straight to using Sedumi starting from the relaxation ($\texttt{SDP}_{\texttt{basic}}$). The times for this alternative approach are labelled "w/o bundle". The important observation is that the use of the initial set of inequalities yields a speed-up of one order of magnitude in the running time for the largest instances. The same effect was observed for the linear relaxation of the linear ordering problem for very large instances ($n \geq 150$) [23, Section 10.3].

Table 3: Study of the impact of constraint classes $\mathcal{DV}$ and $\mathcal{SI}$

Instance	$(\texttt{SDP}_{\texttt{basic}})$			$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{DV}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec) (sec)	Gap (%)	Average nbr ineqs added
H_5	5.2	72.75	7.1	4.5	10.43	12.9
Rand_5	4.5	98.11	6.3	4.5	0	26.5
H_6	11.5	171.70	10.6	12.3	8.27	19.5
Rand_6	12.4	126.79	10.0	13.8	1.06	20.9
H_7	32.0	188.14	14.5	31.7	4.36	30.8
Rand_7	30.9	114.19	13.2	31.7	0	31.0
Instance	$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{SI}$			$(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{DV} \cap\ \mathcal{SI}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec) (sec)	Gap (%)	Average nbr ineqs added
H_5	5.4	71.43	25.9	5.3	10.43	30.3
Rand_5	5.0	90.91	23.1	5.1	0	29.3
H_6	13.9	104.26	39.7	15.7	8.27	43.9
Rand_6	15.4	83.17	35.1	14.9	1.06	37.9
H_7	45.4	139.44	79.7	39.4	4.36	62.1
Rand_7	41.8	97.62	49.9	39.0	0	59.8

6.2 Analysis of the Practical Impact of the Various Constraint Classes

Because $(\texttt{SDP}_{\texttt{full}})$ is too expensive to be solved for $n \geq 8$, we examine the efficiency (impact on computation time) and effectiveness (impact on bound quality) of the various constraint classes. The aim is to find a smaller relaxation that contains the most important constraints with respect to bound quality.

Our starting relaxation is again $(\texttt{SDP}_{\texttt{basic}})$. This model reflects the fundamental structure of the original problem in the sense that it would suffice to obtain the optimal solution if we additionally imposed integrality conditions on the ordering variables (see Theorem 1).

First we examine the practical effect of the constraint sets $\mathcal{DV}$ and $\mathcal{SI}$. The computational results are summarized in Table 3. The results support the conclusion that $\mathcal{DV}$ is both effective and efficient. On the other hand, the impact of $\mathcal{SI}$ is much less.

Adding $\mathcal{DV}$ to $(\texttt{SDP}_{\texttt{basic}})$, we obtain a relaxation that is improved but still computationally cheap:

$$\min \{K + \langle C_Z, Z\rangle : Z \text{ satisfies (12) and (24)},\ Z \in (\mathcal{E} \cap \mathcal{DV})\}. \qquad (\texttt{SDP}_{\texttt{cheap}})$$

Next we examine the effects of adding $\mathcal{LS}$ and $\mathcal{M}$ to this new relaxation $(\texttt{SDP}_{\texttt{cheap}})$. The results are summarized in Table 4. We observe that neither $\mathcal{M}$ nor $\mathcal{LS}$ is particularly efficient. We also tested the relaxation $(\texttt{SDP}_{\texttt{basic}}) \cap \mathcal{M} \cap\ \mathcal{LS}$ and found that the overall gaps for these same instances are always over 50%. Furthermore, the running times are much higher than for $(\texttt{SDP}_{\texttt{cheap}})$.

In summary, our computational results in this section strongly suggest that $(\texttt{SDP}_{\texttt{cheap}})$ provides the best tradeoff between computational time and quality of the bounds. Of course, the constraint classes not included in $(\texttt{SDP}_{\texttt{cheap}})$ still help tighten the relaxation but it is more efficient to use them within the improvement strategy proposed in Section 6.5.

6.3 Optimizing Over All Row Assignments Using $(\texttt{SDP}_{\texttt{cheap}})$

We run again the algorithmic approach of Section 6.1 but using the relaxation $(\texttt{SDP}_{\texttt{cheap}})$. The results are reported in Table 5.

Comparing with the performance of $(\texttt{SDP}_{\texttt{full}})$ documented in Table 2, we see that $(\texttt{SDP}_{\texttt{cheap}})$ runs at least one order of magnitude faster for instances of size $n = 6$ and $n = 7$ with only a mild deterioration of the lower bounds, and hence of the gap.

Using $(\texttt{SDP}_{\texttt{cheap}})$, we are able to compute bounds for instances of sizes up to $n = 14$ within the 24-hour time limit. We observe that the quality of the bounds does not deteriorate as the size increases, and that the running time increases by a factor of 3 for each unit increase in n.

Table 4: Study of the impact of constraint classes $\mathcal{M}$ and $\mathcal{LS}$

Instance	($\texttt{SDP}_{\texttt{cheap}}$)			($\texttt{SDP}_{\texttt{cheap}}$) $\cap$ $\mathcal{LS}$		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	4.5	10.43	12.9	6.6	7.27	70.0
Rand_5	4.5	0	26.5	6.6	0	71.2
H_6	12.3	8.27	19.5	35.4	4.65	198.8
Rand_6	13.8	1.06	20.9	29.9	1.06	177.5
H_7	31.7	4.36	30.8	255.1	4.26	415.0
Rand_7	31.7	0	31.0	232.0	0	407.4
Instance	($\texttt{SDP}_{\texttt{cheap}}$) $\cap$ $\mathcal{M}$			($\texttt{SDP}_{\texttt{full}}$)		
	Total time (sec)	Gap (%)	Average nbr ineqs added	Total time (sec)	Gap (%)	Average nbr ineqs added
H_5	8.3	7.27	92.9	9.6	7.14	144.2
Rand_5	7.9	0	98.4	7.9	0	141.3
H_6	70.4	2.42	339.8	130.3	2.42	477.4
Rand_6	58.2	0.53	310.5	94.2	0.53	424.8
H_7	1821.5	3.72	940.1	6177.5	3.69	1205.2
Rand_7	1006.0	0	780.4	2704.0	0	1016.0

Table 5: Computational results for ($\texttt{SDP}_{\texttt{cheap}}$)

	Global bounds (over all row assignments)			Statistics for the $2^{n-1}-1$ subproblems			Computational statistics	
Instance	Lower bound	Upper bound	Gap (%)	Largest gap (%)	Average gap (%)	Nbr times zero-gap	Average nbr active inequalities	Total time w/ bundle (sec)
H_5	407.5	450	10.43	10.43	2.52	5	12.9	4.5
Rand_5	52.5	52.5	0	8.93	1.78	7	15.1	4.5
H_6	665	720	8.27	16.95	5.78	4	19.5	12.3
Rand_6	188.5	190.5	1.06	6.65	2.59	3	20.9	13.8
H_7	1629	1700	4.36	15.17	3.75	4	30.8	31.7
Rand_7	166	166	0	13.82	3.16	11	31.0	31.7
H_8	2351	2385	1.45	21.22	5.80	1	42.9	86.8
S_8	380.5	408	7.23	20.10	5.87	2	44.1	91.6
SH_8	990.5	1135.5	14.64	17.00	10.85	4	56.1	87.4
Rand_8	192	205	6.77	28.10	4.77	6	42.8	82.2
S_9	1163	1181.5	1.59	13.63	3.42	6	64.2	253.4
SH_9	1974.5	2294.5	16.21	18.87	11.31	0	80.6	251.9
Rand_9	447.5	492.5	10.06	19.73	5.61	3	55.8	252.6
S_10	1314	1374.5	4.60	10.77	4.20	7	82.7	713.0
Rand_10	779	838	7.57	15.16	5.68	0	78.7	698.2
S_11	3325.5	3439.5	3.43	14.92	5.16	6	106.8	2127.0
Rand_11	643.5	708	10.02	23.95	5.76	9	103.4	2048.5
H_12	8446.5	8995	6.49	17.31	6.19	0	125.1	6189.5
Rand_12	775.5	799	3.03	17.69	6.21	0	128.8	6389.5
Rand_13	1058	1070	1.13	19.11	5.98	0	159.5	20636.9
Rand_14	1335.5	1393.5	4.34	20.25	6.59	1	172.5	60845.6

6.4 Using Bounds in the Enumeration

It is possible to further reduce the computational effort within the enumeration scheme using previously acquired lower-bound knowledge. This is because the computation of a lower bound can be stopped if its current value is already above the current global lower bound.

The impact of this strategy depends on the order in which we look at the row assignments; those with the weakest lower bounds should be computed first. We propose the following heuristic to obtain a reasonably good ordering:

- Order the row assignments in increasing difference of the sums of the lengths of the departments in each row

- If two or more assignments are tied, further sort them in increasing difference between the sum of connectivities within the rows and the sum of connectivities between the rows. Specifically for the two-row case, we have:

$$\left|\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=1}} c_{ij} - \sum_{\substack{i<j\in[n],\\ r(i)=r(j)=2}} c_{ij}\right| + \left|\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=1}} c_{ij} - \sum_{\substack{i,j\in[n],\\ r(i)=1,r(j)=2}} c_{ij}\right| + \left|\sum_{\substack{i<j\in[n],\\ r(i)=r(j)=2}} c_{ij} - \sum_{\substack{i,j\in[n],\\ r(i)=1,r(j)=2}} c_{ij}\right|.$$

The intuition behind this heuristic is that small differences in both cases are generally good: it is desirable that the sum of lengths of departments in the rows should be equal, and that connectivities should be spread equally.

Table 6 summarizes the results obtained for the enumeration using bound information for instances with $n \leq 15$ departments. Comparing with the performance of using $(\texttt{SDP}_{\texttt{cheap}})$ without bound information documented in Table 5, we see that using the bounds makes a dramatic reduction in the running time without any effect on the quality of the results. As a consequence, we are able to compute bounds for instances with up to $n = 15$ within the 24-hour time limit. Nevertheless, the running time still increases by a factor of 3 for each unit increase in n. For the instances with $n = 15$, the rapid growth of the computational effort required to handle the 3-cycle equations is clear.

We point out that for the results in Table 6, we change our strategy for $n \geq 10$ by doing 20 function evaluations (instead of 10) and 5 constraint updates (instead of 3) in the bundle method. Not only do the Sedumi iterations become more expensive compared to bundle iterations for $n \geq 10$, but also because in these tests we use the bounds for pruning, running the bundle method longer reduces the overall computation time as we often can prune the lower bound computation before switching to Sedumi. As a consequence we do not have to go to Sedumi for many assignments (see the sixth and seventh columns in Table 6).

We checked the quality of the order of the row assignments obtained by our heuristic using the data obtained by the complete enumeration approach above. The results are summarized in the last two columns of Table 6 which give the average position of the best assignment with respect to the lower and the upper bound respectively. The impact of the heuristic is measured by comparing the average percentages we obtain with the expected value of 50% for a random ordering; our smaller percentages show that the heuristic generally has the desired effect. Note that the quality of the heuristic cannot be evaluated for instances with more than 14 departments since the exact lower bounds for all row assignments of these instances could not be computed in the previous subsection.

6.5 A Strategy for Further Improvement of the Bound Quality

The relaxation $(\texttt{SDP}_{\texttt{cheap}})$ is more efficient than $(\texttt{SDP}_{\texttt{full}})$ but is also weaker. We can often improve the quality of the global lower bounds for an instance by finding the row assignment with the weakest lower bound; taking the optimal solution of its $(\texttt{SDP}_{\texttt{cheap}})$ relaxation as reported in Table 5; adding to the relaxation the violated inequalities from those present in $(\texttt{SDP}_{\texttt{full}})$ and resolving with Sedumi until we get the optimal solution of $(\texttt{SDP}_{\texttt{full}})$ for the selected row assignment; update the global lower bound of the instance accordingly. We repeat this process until the weakest lower bound comes from a row assignment for which we have already improved the $(\texttt{SDP}_{\texttt{cheap}})$ relaxation. We also use the current overall lower bound to stop the lower bound computation when it becomes irrelevant.

The results we obtained using this improvement approach on instances with $n = 7, 8, 9$ are reported in Table 7. (We omit Rand_7 since $(\texttt{SDP}_{\texttt{cheap}})$ is already optimal for it.) As the direct solution for instances with 8 or more departments is far too expensive, the improvement strategy proves to be a very valuable tool. For instance, for the H_7 instance, we were able to compute the global lower bound from $(\texttt{SDP}_{\texttt{full}})$ in only $31.7 + 80.1 < 112$ seconds instead of the 6177.5 seconds needed in Table 2. Furthermore, for the H_8 instance, we closed the gap and hence proved global optimality. Overall we see that most of the gaps are reduced by between 0.2% and 1.5% with respect to those in Table 6. But there is significant variability: while for the H_8 instance we closed the gap and hence proved global optimality, the lower bound for SH_8 was not

Table 6: Results using $(\texttt{SDP}_{\texttt{cheap}})$ and using bounds for pruning in enumeration

	Global bounds (over all row assignments)			Computational statistics			Validation of the ordering heuristic	
Instance	Lower bound	Upper bound	Gap (%)	Total time w/ bundle (sec)	% of instances stopped early by: bundle only	bundle or Sedumi	Position of best lower bound	Position of best upper bound
H_5	407.5	450	10.43	2.6	66.7	80.0	46.67	26.67
Rand_5	52.5	52.5	0	2.0	86.7	86.7	13.33	13.33
H_6	665	720	8.27	5.6	83.9	87.1	41.94	41.94
Rand_6	188.5	190.5	1.06	6.4	83.9	90.3	12.90	16.13
H_7	1629	1700	4.36	13.5	90.5	95.2	7.94	7.94
Rand_7	166	166	0	14.9	76.0	90.5	22.22	22.22
H_8	2351	2385	1.45	44.2	66.1	96.1	22.05	5.51
S_8	380.5	408	7.23	47.8	60.6	96.1	18.11	21.26
SH_8	990.5	1135.5	14.64	47.9	60.0	99.2	0.79	10.24
Rand_8	192	205	6.77	47.2	49.6	93.7	36.22	36.22
S_9	1163	1181.5	1.59	126.2	60.4	98.8	1.57	1.57
SH_9	1974.5	2294.5	16.21	135.9	49.4	99.6	3.92	3.53
Rand_9	447.5	492.5	10.06	109.1	80.4	98.0	10.20	2.35
S_10	1314	1374.5	4.60	333.2	82.6	98.8	2.35	4.50
Rand_10	779	838	7.57	398.4	71.4	98.8	11.74	4.31
S_11	3325.5	3439.5	3.43	1221.2	54.5	99.4	25.22	12.51
Rand_11	643.5	708	10.02	556.6	96.6	99.2	20.14	20.14
H_12	8446.5	8995	6.49	3245.2	52.2	99.7	24.72	5.18
Rand_12	775.5	799	3.03	1428.3	97.8	99.7	16.27	23.50
Rand_13	1058	1070	1.13	3444.5	98.5	99.8	30.99	30.99
Rand_14	1335.5	1393.5	4.34	9941.4	97.4	99.9	39.58	39.58
H_15	16066	16640	3.57	69181.3	44.2	99.9	-	-
P_15	3046	3195	4.89	69622.8	42.8	99.9	-	-
Rand_15	2461	2643.5	7.42	64097.9	48.9	99.9	-	-

Table 7: Computing the $(\texttt{SDP}_{\texttt{full}})$ bounds starting with the $(\texttt{SDP}_{\texttt{cheap}})$ relaxation

	Improvement statistics		Global bounds (over all row assignments)			Computational statistics	
Instance	Nbr of instances with gap > 0	Nbr of instances improved	Lower bound	Upper bound	Gap (%)	Total time (sec)	Average final nbr of ineq constraints
H_7	3	1	1639.5	1700	3.69	80.1	1320.0
H_8	4	4	2385	2385	0	911.2	2457.5
S_8	26	1	380.5	408	7.23	1.9	336.0
SH_8	58	2	999.0	1135.5	13.66	20.3	1013.5
Rand_8	17	1	193	205	6.22	1713.0	3360.0
S_9	6	2	1168.0	1181.5	1.16	432.9	2888.0
SH_9	131	9	2009.5	2294.5	14.18	8806.9	2599.7
Rand_9	22	1	448.5	492.5	9.81	77.7	1516.0

reduced at all. Similarly the computational times vary significantly even for instances of the same size. For instance, computing the $(\texttt{SDP}_{\texttt{full}})$ bound for Rand_9 in this manner required only 77.7 seconds (less time than it took to compute the $(\texttt{SDP}_{\texttt{cheap}})$ bound for it in Table 6) while the $(\texttt{SDP}_{\texttt{full}})$ bound for SH_9 took over 9000 seconds.

We can control the computational effort involved in improving the lower bounds by considering a relaxation between $(\texttt{SDP}_{\texttt{cheap}})$ and $(\texttt{SDP}_{\texttt{full}})$ in the sense of setting a limit on the total number of inequality constraints that can be present in the relaxation. Motivated by the results from Tables 3 and refreslsmet, we consider the inequalities in the following order: (24), $\mathcal{DV}$, $\mathcal{LS}$, $\mathcal{M}$, $\mathcal{SI}$. We summarize the computational results, where we use at most 2000, 4000 or 6000 inequality constraints in Table 8. Whenever the fourth column of Table 8 has a zero, this means that we effectively solved $(\texttt{SDP}_{\texttt{full}})$. Therefore we do not test those instances for larger values of the maximum number of inequalities.

Comparing the results for the instances with 9 departments in Tables 7 and 8 shows that limiting the number of constraints helps to reduce the computation time considerably without too much impact on the

Table 8: Improved bounds starting from ($\mathtt{SDP_{cheap}}$) and with limits on the number of inequality constraints

Instance	Nbr of instances with gap > 0	Nbr of instances improved	Nbr of instances for which max nbr of ineqs added	Lower bound	Upper bound	Gap (%)	Total time (sec)	Average final nbr of ineq constraints
Maximum of 2000 inequality constraints								
S_9	6	2	1	1168.0	1181.5	1.16	262.7	1150.0
SH_9	131	8	4	2008	2294.5	14.27	822.5	1129.5
S_10	19	1	0	1319.5	1374.5	4.17	115.1	1760.0
Rand_10	84	1	1	788	838	6.35	128.1	2000.0
S_11	62	2	2	3334	3439.5	3.16	560.8	2000.0
Rand_11	5	1	1	649.5	708	9.01	155.0	2000.0
H_12	478	10	3	8482	8995	6.05	710.9	1079.4
Rand_12	3	1	1	777.5	799	2.77	216.3	2000.0
Rand_13	3	1	1	1061	1070	0.85	368.5	2000.0
Rand_14	6	2	1	1349	1393.5	3.30	307.0	1546.5
Maximum of 4000 inequality constraints								
Rand_10	84	2	1	792.5	838	5.74	1779.4	2305.0
S_11	62	2	0	3337.5	3439.5	3.06	9932.8	3761.0
Rand_11	5	1	1	652.5	708	8.51	1366.9	4000.0
H_12	478	10	3	8485	8995	6.01	5086.4	1694.3
Rand_12	3	1	1	783.5	799	1.98	1465.3	4000.0
Rand_13	3	1	1	1062.5	1070	0.71	2054.0	4000.0
Rand_14	6	2	1	1353.5	1393.5	2.96	1880.7	2546.5
Maximum of 6000 inequality constraints								
Rand_10	84	2	1	793	838	5.68	9891.8	3306.0
Rand_11	5	1	1	655.5	708	8.01	4173.0	6000.0
H_12	478	10	3	8485.5	8995	6.00	23844.3	2285.8
Rand_12	3	1	1	785.5	799	1.72	5968.7	6000.0
Rand_13	3	1	1	1063	1070	0.66	6905.5	6000.0
Rand_14	6	2	1	1355.5	1393.5	2.80	5588.9	3537.0

quality of the lower bounds. Changing the limit from 2000 to 4000 and from 4000 to 6000 constraints we observe that while the lower bound improves a little, the computation time grows significantly. When allowing at most 6000 constraints the computation times get already quite large and hence we do not consider adding even more constraints.

Starting the improvement strategy with ($\mathtt{SDP_{basic}}$) is not an attractive option because the bounds are much weaker than the ($\mathtt{SDP_{cheap}}$) bounds, and the number of relevant inequalities in $\mathcal{DV}$ is very small compared to the $\binom{n}{3}$ 3-cycle equalities.

6.6 Medium and Large Instances

As a final test of our SDP relaxations, we consider the DRFLP using the data from selected SRFLP instances with between 17 and 70 departments. For each value of n we chose one instance in the literature. Table 9 lists the characteristics of the instances. All the instances can be downloaded from http://anjos.mgi.polymtl.ca/flplib.

Given the size of these instances, we can only solve the relaxation for a selection of row assignments. We select the row assignments using the following simple heuristic: We first randomly assign 25% of the departments to each of the two rows; then the remaining 50% of the departments are added one at a time by taking the longest remaining department and adding it to the shorter row. This heuristic quickly yields assignments for which the total row lengths are very close; see the second-to-last column of Table 11. Similar row lengths are often of interest in the design of layouts in practice, see e.g. [26].

We summarize in Table 10 the results averaged over 10 row assignments selected by our heuristic. We do this using ($\mathtt{SDP_{cheap}}$) and solve it exactly with Sedumi using the same algorithmic approach as proposed in Section 6.1. We have to call Sedumi 3 times on average to solve ($\mathtt{SDP_{cheap}}$) exactly. It is interesting to note that in all our experiments, the gap changes only marginally after the first call to Sedumi.

Table 9: Characteristics of larger instances with between 17 and 70 departments

<table>
<tr><th rowspan="2">Instance</th><th rowspan="2">Source</th><th rowspan="2">Size (n)</th><th colspan="3">SRFLP</th></tr>
<tr><th>Best lower bound</th><th>Best layout</th><th>Time (sec) w/ bundle [24]</th></tr>
<tr><td>P17</td><td>[2]</td><td>17</td><td colspan="2">9254</td><td>35</td></tr>
<tr><td>P18</td><td>[2]</td><td>18</td><td colspan="2">10650.5</td><td>33</td></tr>
<tr><td>H_20</td><td>[22]</td><td>20</td><td colspan="2">15549</td><td>54</td></tr>
<tr><td>N25_05</td><td>[7]</td><td>25</td><td colspan="2">15623</td><td>211</td></tr>
<tr><td>H_30</td><td>[22]</td><td>30</td><td colspan="2">44965</td><td>547</td></tr>
<tr><td>N30_05</td><td>[7]</td><td>30</td><td colspan="2">115268</td><td>1110</td></tr>
<tr><td>Am33_03</td><td>[3]</td><td>33</td><td colspan="2">69942.5</td><td>2193</td></tr>
<tr><td>Am35_03</td><td>[3]</td><td>35</td><td colspan="2">69002.5</td><td>3194</td></tr>
<tr><td>ste36.5</td><td>[8]</td><td>36</td><td colspan="2">91651.5</td><td>1078</td></tr>
<tr><td>N40_5</td><td>[24]</td><td>40</td><td colspan="2">103009</td><td>8409</td></tr>
<tr><td>sko42-5</td><td>[8]</td><td>42</td><td colspan="2">248238.5</td><td>4122</td></tr>
<tr><td>sko49-5</td><td>[8]</td><td>49</td><td>666130</td><td>666143</td><td>34222</td></tr>
<tr><td>sko56-5</td><td>[8]</td><td>56</td><td>591915.5</td><td>592335.5</td><td>64006</td></tr>
<tr><td>AKV-60-05</td><td>[4]</td><td>60</td><td>318801</td><td>318805</td><td>99106</td></tr>
<tr><td>sko64-5</td><td>[8]</td><td>64</td><td>501342.5</td><td>502063.5</td><td>119158</td></tr>
<tr><td>AKV-70-05</td><td>[4]</td><td>70</td><td>4213774.5</td><td>4218002.5</td><td>101765</td></tr>
</table>

Table 10: Double-row results over 10 row assignments using $(\mathtt{SDP_{cheap}})$ and Sedumi

Instance	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average nbr of inequalities	Average time (sec)
P17	4501.5	4722	2.68	10.05	5.82	265.7	41.5
P18	5153	5503.5	3.85	11.51	8.36	298.6	67.4
H_20	7520	8046	4.97	10.86	7.70	400.7	242.6
N25_05	7385	7986	5.62	11.56	8.79	659.1	1386.3
H_30	21028	22848	6.64	13.74	9.63	1057.6	7949.8
N30_05	53854	58221	5.89	13.46	9.27	1201.3	9439.3
Am33_03	32847	35904.5	7.59	13.88	9.31	1580.7	21133.3
Am35_03	32142	35273	8.64	12.89	9.74	1666.3	37676.7
ste36.5	44786.5	46794.5	1.36	5.54	3.66	1633.6	45615.5

For even larger instances, say $n \geq 40$, we continue to use $(\mathtt{SDP_{cheap}})$ but apply only the bundle method (without Sedumi) to obtain reasonable bounds. We report results only for instances with up to 70 departments as the experiments quickly become too time consuming. This is evidenced by the growth of the running times in Table 11 below, as well as in Table 9 for solving the simpler SRFLP relaxation. We restrict the bundle method to 125 function evaluations of $f(\lambda, \mu)$. This limitation of the number of function evaluations sacrifices some possible incremental improvement of the bounds.

Table 11 summarizes the results we obtained. Comparing the results in Tables 10 and 11 shows that the lower bounds of the bundle method quickly get close to the exact $(\mathtt{SDP_{cheap}})$ bounds even though the number of function evaluations is capped at 125. Furthermore, while the running times in Table 10 grow very quickly with the problem size, the computation times of the bundle method in Table 11 are not so strongly affected by the problem size. Hence this approach yields bounds competitive with the exact optimal value of $(\mathtt{SDP_{cheap}})$ at only a fraction of the computational cost.

7 Conclusions and Future Research

We proposed a new semidefinite programming approach for the space-free multi-row facility layout problem. This is the special case of multi-row layout in which all the rows have a common left origin and no empty space is allowed between departments. Our computational results show that for space-free double-row instances the proposed semidefinite optimization approach provides high-quality global bounds in reasonable time for instances with up to 15 departments. If the row assignment is fixed, then bounds can be computed for instances with up to 70 departments.

Table 11: Double-row results over 10 row assignments using ($\mathtt{SDP_{cheap}}$) and the bundle method

Instance	Lower bound	Upper bound	Minimum gap (%)	Maximum gap (%)	Average gap (%)	Average difference of row lengths	Average time (sec)
P17	4435	4737	4.68	9.62	7.29	1.8	24.9
P18	5080	5462.5	5.09	14.32	9.63	1.0	32.0
H_20	7402	8149	8.54	12.40	10.03	2.0	48.5
N25_05	7254	7945	6.37	15.33	10.45	0.4	128.7
H_30	20659.5	22801	9.18	18.70	13.34	2.0	313.2
N30_05	52756.5	58425	7.29	13.55	10.45	1.8	310.4
Am33_03	32058	35958.5	10.45	20.41	15.39	1.6	554.3
Am35_03	31521	34794.5	8.77	18.48	14.83	1.2	720.1
ste36.5	41409.5	47259.5	7.14	19.94	12.91	1.0	808.2
N40_5	48212.5	56204	11.13	20.39	16.90	1.0	1524.2
sko42-5	113606	127639.5	11.36	19.43	15.54	1.0	1959.6
sko49-5	291004.5	349137	17.46	23.10	20.20	2.0	4904.0
sko56-5	261686	306133.5	15.91	22.54	19.66	1.0	11849.1
AKV-60-05	145702	171280	17.56	22.42	19.41	1.0	17162.7
sko64-5	219646	261257.5	18.95	24.78	21.56	1.0	22828.3
AKV-70-05	1861211	2196942.5	18.04	21.36	19.62	1.2	45232.4

Because the number of possible assignments of departments to rows grows exponentially, future research will seek better heuristics to quickly find high-quality assignments to which the proposed SDP approach can then be applied. Other issues to address in future research are the incorporation of spacing within the rows in the optimization process, and the use of the SDP approach within a suitable enumeration scheme to globally optimize instances of double-row and multi-row layout.

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] 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.

[5] M. F. Anjos and J. B. Lasserre, editors. *Handbook on Semidefinite, Conic and Polynomial Optimization Theory, Algorithms, Software and Applications*. International Series in Operations Research & Management Science. Springer-Verlag, New York, 2011.

[6] M. F. Anjos and F. Liers. Global approaches for facility layout and VLSI floorplanning. In M. F. Anjos and J. B. Lasserre, editors, *Handbook on Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications*, International Series in Operations Research and Management Science. Springer, New York, 2011.

[7] 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.

[8] 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.

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

[10] M. Chimani and P. Hungerländer. Exact approaches to multi-level vertical orderings. Technical report, Uni Jena, Computer Science, Algorithm Engineering Group, TR-FSUJ-CS-AE-11-02. `www.ae.uni-jena.de/Research_Pubs/MLVO.html`, 2011.

[11] M. Chimani and P. Hungerländer. Multi-level verticality optimization: Concept, strategies, and drawing scheme. Technical report, Uni Jena, Computer Science, Algorithm Engineering Group, TR-FSUJ-CS-AE-11-01. `www.ae.uni-jena.de/Research_Pubs/MLVO.html`, 2011.

[12] 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.

[13] M. Chimani, P. Hungerländer, M. Jünger, and P. Mutzel. An SDP approach to multi-level crossing minimization. Technical report, submitted (extended version of the proceedings paper with the same title), 2011.

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

[15] 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. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

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

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

[20] S. S. Heragu. *Facilities Design*. iUniverse, 2nd edition, 2006.

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

[22] S. S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European Journal of Operational Research*, 53(1):1–13, 1991.

[23] P. Hungerländer. *Semidefinite Approaches to Ordering Problems*. PhD thesis, Alpen-Adria Universität Klagenfurt, 2011.

[24] P. Hungerländer and F. Rendl. A computational study and survey of methods for the single-row facility layout problem. Technical report, submitted, preprint available at http://www.optimization-online.org/DB_HTML/2011/05/3029.html, 2011.

[25] P. Hungerländer and F. Rendl. Semidefinite relaxations of ordering problems. *Mathematical Programming B*, 2011. accepted, preprint available at www.optimization-online.org/DB_HTML/2010/08/2696.html.

[26] A. Langevin, B. Montreuil, and D. Riopel. Spine layout design. *International Journal of Production Research*, 32(2):429–442, 1994.

[27] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224:439–461, 1995.

[28] M. Laurent and S. Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17:530–547, 1996.

[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. Meller and K.-Y. Gau. The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems*, 5(5):351–366, 1996.

[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] S. Sanjeevi and K. Kianfar. A polyhedral study of triplet formulation for single row facility layout problem. *Discrete Applied Mathematics*, 158:1861–1867, 2010.

[34] D. M. Simmons. One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17:812–826, 1969.

[35] J. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11–12:625–653, 1999.

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

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

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