Semidefinite Relaxations of Ordering Problems

Alpen-Adria Universität Klagenfurt, Institut für Mathematik, 9020 Klagenfurt, Austria
philipp.hungerlaender@uni-klu.ac.at, franz.rendl@uni-klu.ac.at

January 19, 2011

Abstract

Ordering problems assign weights to each ordering and ask to find an ordering of maximum weight. We consider problems where the cost function is either linear or quadratic. In the first case, there is a given profit if the element u is before v in the ordering. In the second case, the profit depends on whether u is before v and r is before s.

The linear ordering problem is well studied, with exact solution methods based on polyhedral relaxations. The quadratic ordering problem does not seem to have attracted similar attention. We present a systematic investigation of semidefinite optimization based relaxations for the quadratic ordering problem, extending and improving existing approaches. We show the efficiency of our relaxations by providing computational experience on a variety of problem classes.

1 Introduction

Ordering problems associate to each ordering (or permutation) of the set $N := \{1, \dots, n\}$ of n objects a profit and the goal is to find an ordering of maximum profit. In the simplest case of the *linear ordering problem (LOP)*, this profit is determined by those pairs $(u, v) \in N \times N$, where u comes before v in the ordering. Thus LOP can be defined as follows. Given an $n \times n$ matrix $D = (d_{ij})$ of integers, find a simultaneous permutation ϕ of the rows and columns of D such that

$$\sum_{i<j} d_{\phi(i),\phi(j)}$$

is maximized. Equivalently, we can interpret d_{ij} as weights of a complete directed graph G with vertex set N. Then LOP consists of finding a complete acyclic subgraph of G of maximum total edge weight.

$$z_* := \max \left\{ \sum_{i<j} d_{\phi(i),\phi(j)} : \quad \phi \in \Pi \right\} \tag{1}$$

The permutation ϕ gives the ordering of the vertices N of G and the cost function consists of the sum of all edge weights d_{uv} where u comes before v in this ordering. The set of permutations is denoted by Π.

LOP is well known to be NP-complete [16] and arises in a large number of applications in such diverse fields as economy, sociology, graph drawing [29], archaeology, scheduling [7] and also assessment of corruption perception [1]. Two well known examples of LOP are the determination of ancestry relationships [17] and the triangulation of input-output matrices of an economy [11, 33], where the optimal ordering gives some information about the stability of the economic system.

The simplest formulation of LOP uses linear programming in binary variables and will be briefly recalled in section 2. It is the basis for exact methods which use the continuous linear relaxation in an enumerative scheme. The resulting bounds are not always of sufficient quality, making this approach impractical for larger instances. In section 3 we consider semidefinite relaxations and investigate in some detail how the linear description of the problem can be 'lifted' into the semidefinite model to yield tight approximations. Even though some basic semidefinite models have been proposed in the literature, see for instance [5,8,38], we will provide in section 3 a systematic investigation on various ways to derive constraints for the semidefinite model. The semidefinite relaxations are the natural setting for the *quadratic ordering problem (QOP)*, where the profit depends on whether u comes before v and r is before s. In section 4 we describe some nontrivial applications of QOP, notably the 'multilevel crossing minimization problem' and the 'betweenness problem'.

The last part of the paper contains computational results on a variety of different problem types. We use a complete description of the linear ordering polytope in small dimensions to show that the semidefinite relaxation $(\mathtt{SDP}_4)$ below is exact for any instance of size $n \le 6$ and identifies all but one class of facets of the polytope for $n = 7$. This provides an indication of the potential strength of this approach also for larger instances. We also compare the new model with linear-programming based, and much cheaper bounds on LOP instances from the LOLIB library.

Finally, we provide some improved bounding results for linear arrangement problems and report new optimal solutions for bipartite crossing minimization and single-row layout problems.

2 Linear Ordering as a Linear Program in 0-1 variables

The linear ordering problem has a natural formulation as a linear program in 0-1 variables. An instance of the problem is defined by the $n \times n$ matrix $D = (d_{ij})$. We introduce binary variables x_{ij} with $x_{ij} = 1$ if i comes before j and $x_{ij} = 0$ otherwise. Then it is not hard to show that the following constraints describe linear orderings of the set N:

$$x_{ij} + x_{ji} = 1, \ \forall\, i \neq j, \tag{2}$$

$$x_{ij} + x_{jk} + x_{ki} \in \{1, 2\}, \ \forall\, i, j, k, \tag{3}$$

$$x_{ij} \in \{0, 1\}, \ \forall\, i \neq j. \tag{4}$$

The first condition models the fact that either i is before j or j is before i. The second condition rules out the existence of directed 3-cycles and is sufficient to insure that there is no directed cycle. Hence the feasible solutions of these constraints describe complete acyclic digraphs. Maximizing $\sum_{i \neq j} d_{ij} x_{ij}$ over the constraints (2)–(4) therefore solves LOP. The equations (2) are used to eliminate x_{ji} for $j > i$. This leads to the following formulation of LOP as linear program in binary variables, see [19],

$$z_* = \max \left\{ \sum_{i<j} (d_{ij} - d_{ji}) x_{ij} + d_{ji} : \ \ x_{ij} \in \{0, 1\}, \ \ 0 \le x_{ij} + x_{jk} - x_{ik} \le 1, \ \forall\, i < j < k \right\}.$$

The linear relaxation is obtained by leaving out the integrality conditions on the variables. This results in a linear program with $\binom{n}{2}$ variables and $2\binom{n}{3}$ three-cycle inequalities. It poses a serious challenge to standard LP solvers, once $n \approx 200$. State-of-the-art exact algorithms can solve large instances from specific instance classes with up to 150 nodes, while they fail on other much smaller instances with only 50 nodes. For a detailed overview over benchmark instances for LOP solved and not yet solved see [36, Appendix]. The computation time of exact algorithms increases rapidly with problem size. Currently available exact algorithms include a Branch-and-Bound algorithm that uses a linear programming based lower bound by Kaas [31], a Branch-and-Cut algorithm proposed by Grötschel, Jünger and Reinelt [19] and a combined interior-point cutting-plane algorithm by Mitchell and Borchers [37] who explore polyhedral relaxations of the problem and also provide computational results using Branch-and-Cut.

There exist many heuristics and methaheuristics for LOP and some of them are quite good in finding the optimal solution for large instances in reasonable time. For a recent survey and comparison see [36]. Of course these heuristics do not provide an optimality certificate of the solutions found.

In this paper we are mostly interested in the lower bound computation by analyzing matrix liftings of the ordering problem. For this purpose it is convenient to reformulate the problem in variables taking the values -1 and 1. The variable transformation

$$y_{ij} = 2x_{ij} - 1 \tag{5}$$

leads to the equivalent problem

$$z_* = \max \left\{ \sum\nolimits_{i<j} (d_{ij} - d_{ji}) \frac{y_{ij}+1}{2} + d_{ji} : \ y_{ij} \in \{-1,1\}, \ |y_{ij} + y_{jk} - y_{ik}| = 1, \ \forall\, i<j<k \right\}. \tag{6}$$

In [22] it is shown that one can easily switch between the $\{0,1\}$ and $\{-1,1\}$ formulations of bivalent problems so that the resulting bounds remain the same and structural properties are preserved. Omitting the integrality condition $y_{ij} \in \{-1,1\}$ gives the linear relaxation

$$z_{LP} := \max \sum_{i<j} (d_{ij} - d_{ji}) \frac{y_{ij}+1}{2} + d_{ji}, \tag{7a}$$

$$\text{subject to} \quad -1 \le y_{ij} + y_{jk} - y_{ik} \le 1, \ \forall\, i<j<k, \tag{7b}$$

$$-1 \le y_{ij} \le 1, \ \forall\, i<j. \tag{7c}$$

The upper bound z_{LP} may lead to gaps between z_* and z_{LP} which are too large for efficient pruning in Branch-and-Bound enumeration. We refer to the column LP-gap in the Tables 3 and 4 below. Thus it would be desirable to have some tighter approximation available. In the next section we take a closer look at relaxations which are based on semidefinite optimization.

3 Semidefinite relaxations

The matrix lifting approach takes a vector y and considers the matrix $Y = yy^T$. We are interested in linear and quadratic orderings and consider the polytope

$$\mathcal{P}_{LQO} := \operatorname{conv} \left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^\top : \ y \in \{-1,1\}, \ y \text{ satisfies } (6) \right\}.$$

The nonconvex equation $Y - yy^T = 0$ is relaxed to the constraint

$$Y - yy^T \succeq 0,$$

which is convex due to the Schur-complement lemma

$$\begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix} \succeq 0 \Leftrightarrow Y - yy^T \succeq 0.$$

Moreover, the main diagonal entries of Y correspond to y_{ij}^2, hence $\operatorname{diag}(Y) = e$, the vector of all ones. We therefore conclude that any $Y \in \mathcal{P}_{LQO}$ satisfies

$$Y - yy^T \succeq 0, \quad \operatorname{diag}(Y) = e. \tag{8}$$

To simplify notation let us introduce

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{9}$$

where $\dim(Z) = \binom{n}{2} + 1 = p$ and $Z = (z_{ij})$. In this case $Y - yy^T \succeq 0 \Leftrightarrow Z \succeq 0$. Hence, the following basic set $\mathcal{B}$ contains $\mathcal{P}_{LQO}$.

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

In order to express constraints on y in terms of Y, they have to be reformulated as quadratic conditions in y. A natural way to do this for $|y_{ij} + y_{jk} - y_{ik}| = 1$ consists in squaring both sides, leading to

$$y_{ij}^2 + y_{jk}^2 + y_{ik}^2 + 2(y_{ij,jk} - y_{ij,ik} - y_{ik,jk}) = 1, \; \forall\, i < j < k. \tag{11}$$

Since $y_{ij}^2 = 1$, this simplifies to

$$y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \; \forall\, i < j < k. \tag{12}$$

In [8] it is shown that these equations formulated in the $\{0, 1\}$ model describe the smallest linear subspace that contains $\mathcal{P}_{LQO}$. We now formulate LOP as a semidefinite optimization problem in bivalent variables.

Proposition 1. *The problem*

$$\max \{\, d^T y : \; Z \text{ satisfies } (9), \;\; Z \in \mathcal{B}, \;\; y_{ij,jk} - y_{ij,ik} - y_{ik,jk} = -1, \; \forall\, i < j < k, \;\; y_{ij} \in \{-1, 1\} \,\}$$

is equivalent to LOP.

Proof. Since $y_{ij}^2 = 1$ we have $\operatorname{diag}(Y - yy^T) = 0$, which together with $Y - yy^T \succeq 0$ shows that in fact $Y = yy^T$. The 3-cycle equations (12) ensure that $|y_{ij} + y_{jk} - y_{ik}| = 1$ holds. The objective value differs from z_* only in an additive constant. □

We are now dropping the integrality condition on y and obtain the following basic semidefinite relaxation of LOP. The objective function assigns currently only costs to y, but not to Y. From now on we do not make any assumptions on the cost function and simply consider a generic cost function $\langle C, Z\rangle$, where C is a symmetric matrix of order p, and Z is given by (9). Thus we get the following basic semidefinite relaxation

$$\max \{\, \langle C, Z\rangle : \; Z \text{ partitioned as in (9) satisfies (12)}, \;\; Z \in \mathcal{B} \,\}. \tag{$\mathtt{SDP}_1$}$$

There are some obvious ways to tighten ($\mathtt{SDP}_1$). First of all we observe that Y, and therefore Z in (9), is actually a matrix with $\{-1, 1\}$ entries in the original LOP formulation. Hence it satisfies the triangle inequalities, defining the metric polytope $\mathcal{M}$

$$\mathcal{M} = \left\{ Z : \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e, \; \forall\, i < j < k \right\}. \tag{13}$$

We note that the metric polytope is defined through $4\binom{p}{3} \approx \frac{1}{12} n^6$ facets. They are used as triangle inequalities of the max-cut polytope in [34, 40, 42]. The basic relaxation ($\mathtt{SDP}_1$) can therefore be improved by asking in addition that $Z \in \mathcal{M}$ yielding ($\mathtt{SDP}_2$).

Another generic improvement was suggested by Lovász and Schrijver in [34]. Applied to our problem, this approach suggests to multiply the 3-cycle inequalities (7b)

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

by the nonnegative expressions $(1 - y_{lm})$ and $(1 + y_{lm})$. This results in the following inequalities

$$\begin{aligned} -1 - y_{lm} \le y_{ij} + y_{jk} - y_{ik} + y_{ij,lm} + y_{jk,lm} - y_{ik,lm} \le 1 + y_{lm}, \; \forall\, i < j < k, \; l < m, \\ -1 + y_{lm} \le y_{ij} + y_{jk} - y_{ik} - y_{ij,lm} - y_{jk,lm} + y_{ik,lm} \le 1 - y_{lm}, \; \forall\, i < j < k, \; l < m. \end{aligned} \tag{14}$$

We define the polytope $\mathcal{LS}$

$$\mathcal{LS} := \{\, Z : Z \text{ satisfies (14)} \,\}, \tag{15}$$

consisting of $4\binom{n}{3}\binom{n}{2} \approx \frac{1}{3}n^5$ constraints. The basic relaxation ($\mathtt{SDP}_1$) can therefore also be improved by asking in addition that $Z \in \mathcal{LS}$ yielding ($\mathtt{SDP}_3$). In summary, we get the following tractable relaxation of $\mathcal{P}_{LQO}$, part of which (without the matrix cuts (15)) has been investigated in [8] for bipartite crossing minimization problems and in [5] for single-row layout problems.

$$z_{SDP} := \max \{ \langle C, Z\rangle : \; Z \text{ partitioned as in (9) satisfies (12)}, \; Z \in \mathcal{B}, \; Z \in \mathcal{M}, \; Z \in \mathcal{LS} \} \qquad (\mathtt{SDP}_4)$$

We close this section with a few simple observations. First, it is not hard to verify that any Z feasible for ($\mathtt{SDP}_1$) has in its first column a vector y which satisfies the 3-cycle inequalities (7b). This follows from the semidefiniteness of the following submatrix of Z

$$\begin{pmatrix} 1 & y_{ij} & y_{ik} & y_{jk} \\ y_{ij} & 1 & y_{ij,ik} & y_{ij,jk} \\ y_{ik} & y_{ik,ij} & 1 & y_{ik,jk} \\ y_{jk} & y_{jk,ij} & y_{jk,ik} & 1 \end{pmatrix}.$$

As a consequence, the basic semidefinite relaxation ($\mathtt{SDP}_1$) is at least as strong as the linear relaxation (7).

There are further methods to tighten the relaxation. Instead of the Lovász-Schrijver lifting procedure, we could for instance multiply different pairs of 3-cycle inequalities. For further details on this we refer to the forthcoming dissertation [27].

Remark 2. *The original formulation of the ordering problem was done in dimension $2\binom{n}{2}$, as we introduced variables y_{ij} for $i \neq j$. The equations $y_{ij} + y_{ji} = 0$ were then used to eliminate half the variables, leading to a new model in dimension $\binom{n}{2}$. Would we get a stronger semidefinite relaxation by working with matrices of order $2\binom{n}{2}$ instead of $\binom{n}{2}$? It is not difficult to show that this is not the case.*

Let m linear equality constraints $Ay = c$ be given. If there exists some invertible $m \times m$ matrix B, we can partition the linear system in the following way $Ay = \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = c$ and then solve for v, $v = B^{-1}(c - Cu)$. Therefore $\begin{bmatrix} 1 \\ u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & I \\ B^{-1}c & -B^{-1}C \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix} = D \begin{bmatrix} 1 \\ u \end{bmatrix}$, defining the full column rank matrix D. From this it is clear that $\begin{bmatrix} 1 \\ u \\ v \end{bmatrix} \begin{bmatrix} 1 \\ u \\ v \end{bmatrix}^T = D \begin{bmatrix} 1 \\ u \end{bmatrix} \begin{bmatrix} 1 \\ u \end{bmatrix}^T D^T$ and therefore $\begin{bmatrix} 1 & u^\top \\ u & U \end{bmatrix} \succcurlyeq 0 \Leftrightarrow \begin{bmatrix} 1 & u^\top & v^\top \\ u & U & W^\top \\ v & W & V \end{bmatrix} \succcurlyeq 0$. Thus we do not weaken the relaxation by first moving into the subspace, given by the equations, and then lifting the problem to matrix space.

4 Areas of application

In this section we point out some areas of application of the semidefinite relaxations analyzed above.

We have mentioned already that the cost function of the semidefinite relaxation can model profits that depend on products of variables y_{ij} and therefore on the relative position of two pairs of elements in the ordering.

Going from linear to quadratic objective functions usually makes an optimization problem much harder. For example the binary maximization of a linear function over the hypercube, which is trivial, becomes the *maximum cut (MC)* problem [42] and thus NP-hard. In our case LOP is already NP-hard, nonetheless the practical hardness of the *quadratic ordering problem (QOP)* is significantly higher and classical approaches used for LOP are hopeless for QOP. For the semidefinite approach the linear and quadratic variants of the problem are essentially equally hard to solve. We will demonstrate this for *multi-level crossing minimization (MLCM)* and the *weighted betweenness problem (WB)*, which are two special cases of QOP.

4.1 Multi-level crossing minimization

Let us define a proper level graph [21] as a graph $G(V, E)$, with vertex set $V = V_1 \cup V_2 \cup \ldots \cup V_q,\ V_i \cap V_j = \emptyset,\ i \neq j$, and edge set $E = E_1 \cup E_2 \cup \ldots \cup E_{q-1},\ E_i \subseteq V_i \times V_{i+1}$. Now MLCM consists of drawing a proper level graph such that the number of edge crossings is minimized when the edges are drawn as straight lines connecting the endnodes.

Next we explain how to count the crossings based on quadratic ordering. We need to consider all distinct pairs $s, t \in V_i$ and all distinct pairs $u, v \in V_{i+1}$ for all i.

Let us give an illustrating example by looking at the concrete cost structure of a small bipartite instance with $V_1 = \{1, 2, 3\}$, $V_2 = \{4, 5, 6, 7\}$ and $E_1 = \{(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 7), (3, 4), (3, 6)\}$. The matrix C_1 is indexed rowwise by the pairs $(1, 2), (1, 3), (2, 3)$. The columns are indexed by all pairs of V_2, i.e. $(4, 5), (4, 6), (4, 7), (5, 6), (5, 7), (6, 7)$. Let us set

$$C = \begin{pmatrix} 0 & -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 1 & -1 & 1 & 0 & -1 \end{pmatrix}.$$

The entry $C_{(1,2),(4,5)} = 0$, because independent of the order of $\{1, 2\}$ and $\{4, 5\}$, there will be exactly one crossing in the subgraph induced by these vertices. Thus, the 0 entries in this matrix reflect all subgraphs, where the number of crossings is independent of the ordering of the vertices. Let us now look at $C_{(1,2),(4,6)} = -1$. The order $(1, 2)$ and $(4, 6)$ yields one crossing in this subgraph. It could be avoided if exactly one of the two orderings is reversed, which amounts to asking that $y_{12}y_{46} = -1$. Maximization will try to make the term $y_{12}y_{46}C_{(1,2),(4,6)}$ equal to one, avoiding the crossing. In a similar way we see that the subgraph induced by the vertices $1, 2, 4, 7$ has no crossing precisely if $y_{12}y_{47} = 1$. Thus we set $C_{(1,2),(4,7)} = 1$.

In the general case we will have crossing matrices C_i for each layer $i = 1, \ldots, q - 1$. We can therefore model MLCM as a quadratic ordering problem with cost matrix

$$C = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & C_1 & \ldots & 0 & 0 \\ 0 & C_1^\top & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 0 & C_{q-1} \\ 0 & 0 & 0 & \ldots & C_{q-1}^\top & 0 \end{bmatrix}$$

where the $C_i,\ i \in \{1, \ldots, q - 1\}$ have dimension $\binom{|V_i|}{2} \times \binom{|V_{i+1}|}{2}$ and are determined by the edge set E_i as described above.

4.2 Weighted betweenness problem

An input to WB consists of n objects, a set $\mathcal{B}$ of betweenness conditions and a set $\overline{\mathcal{B}}$ of non-betweenness conditions ($\mathcal{B} \cap \overline{\mathcal{B}} = \emptyset$). The elements of $\mathcal{B}$ and $\overline{\mathcal{B}}$ are triples (i, j, k) with associated costs w_{ijk} for not placing respectively placing object j between objects i and k. Now the task in WB is to find a linear ordering of the objects such that the sum of costs is minimized.

To represent the given cost structure in our model, we use the fact that the statement j is between i and k is equivalent to the statement i is in front of j and j is in front of k or k is in front of j and j is in front of i. In terms of our ordering variables, we have that j is between i and k exactly if $y_{ij}y_{jk} = 1$. Thus the cost matrix C which is indexed by all ordered pairs, has a contribution to $C_{(i,j),(j,k)}$ proportional to w_{ijk}.

The cost matrix C for WB is also quite sparse as for n objects we have $O(n^4)$ matrix entries but only $O(n^3)$ triples.

5 First computational experiments

$(\mathtt{SDP}_4)$ is formulated in the space of symmetric matrices of order $p = \binom{n}{2} + 1$ and has p equality constraints $\mathrm{diag}(Z) = e$ together with $\binom{n}{3}$ 3-cycle equations. Additionally there are $O(n^5)$ inequalities from the Lovasz-Schrijver lifting together with $O(n^6)$ triangle inequalities. A direct solution using standard interior-point based methods is therefore only feasible for very small values of n, such as $n \leq 8$.

We use the MATLAB toolbox SEDUMI for semidefinite optimization to solve this relaxation for small n. As a first experiment we consider the full description of the linear ordering polytope in small dimensions, and try to recover the correct right hand side of the facets for $n \in \{6, 7\}$.

We compare the linear programming (LP) relaxation (7) to the semidefinite relaxations $(\mathtt{SDP}_1)$–$(\mathtt{SDP}_4)$ from Section 3.

In Table 1 we examine all nontrivial facets of the linear ordering polytopes for 6 and 7 nodes. The facets are collected under `http://comopt.ifi.uni-heidelberg.de/software/SMAPO/lop/lop.html`. We also use the same labeling, see column 1. As usual, n denotes the dimension of LOP, and *opt* gives the optimal solution. All relaxations are solved to optimality using the standard settings of SEDUMI [43]. We also include the combinatorial instances Paley 11 and Paley 19, which are notoriously difficult for linear relaxations.

facet	n	opt	(LP)	$(\mathtt{SDP}_1)$	$(\mathtt{SDP}_2)$	$(\mathtt{SDP}_3)$	$(\mathtt{SDP}_4)$
FC3	6	7	7.5	7.35	7	7	7
FC4, 5	6	8	8.5	8.35	8	8	8
FC3	7	7	7.5	7.35	7	7	7
FC4, 20	7	8	8.5	8.35	8	8	8
FC5	7	9	9.5	9.37	9	9.09	9
FC6	7	9	9.5	9.37	9	9	9
FC10, 25	7	9	9.5	9.37	9.06	9	9
FC21	7	9	9.5	9.37	9	9.01	9
FC7, 9, 22, 24	7	10	10.5	10.37	10.11	10	10
FC8, 13, 23	7	10	10.5	10.37	10.19	10	10
FC11	7	10	10.5	10.37	10	10	10
FC12	7	10	10.5	10.37	10	10.03	10
FC14	7	10	10.5	10.35	10.35	10.24	10.22
FC15, 16	7	11	11.5	11.37	11.22	11	11
FC26	7	11	11.5	11.37	11.23	11	11
FC17, 27	7	13	13.5	13.40	13	13	13
FC18	7	14	14.5	14.40	14.17	14.04	14
FC19	7	14	14.5	14.40	14.10	14.01	14
Paley	11	35	36.67	36.03	36.03	35.92	35.92
Paley	19	107	114	110.70	110.70	110.50	110.50

Table 1: Marginal improvement of various semidefinite relaxations as compared to the linear relaxation on facets of the linear ordering polytope for $n = 6$ and $n = 7$

From Table 1 we conclude that the triangle inequalities and the matrix cuts are incomparable, as there are instances where $(\mathtt{SDP}_2)$ is tighter than $(\mathtt{SDP}_3)$ and vice versa. The basic relaxation $(\mathtt{SDP}_1)$ improves upon the pure linear model (LP), but does not give the correct facets for $n = 6$. Adding either (13) or (15) gives the correct facets for $n = 6$. Furthermore the full model $(\mathtt{SDP}_4)$ identifies all except one of the facets correctly for $n = 7$.

As a first conclusion we observe that the semidefinite approach provides a substantial improvement over the polyhedral approach in the approximation of the linear ordering polytope of small dimensions.

It should also be clear that we need to use algorithmic alternatives to solve instances of reasonable size ($n \geq 30$). In the following section we describe a practical implementation to get approximate solutions of $(\texttt{SDP}_4)$ also for larger instances, $n \approx 100$.

6 A practical implementation for $(\texttt{SDP}_4)$

Looking at the constraint classes and their sizes in $(\texttt{SDP}_4)$, it should be clear that maintaining explicitely $O(n^3)$ or more constraints is not an attractive option. We therefore take up the approach suggested in [15] and adapt it to our problem. We first note that maintaining the constraints $Z \succeq 0$, $\operatorname{diag}(Z) = e$ explicitely leads to the following running times in seconds on an Intel Xeon 5160 processor with 3 GHz. We use a standard interior-point method to solve this relaxation, see for instance [24], and summarize the results in Table 2.

n	p	time
30	436	3
50	1226	40
70	2416	500
100	4951	3000

Table 2: Average computation times (in seconds) using interior-point methods to optimize over (10), where Z is of order p.

To get approximate solutions to $(\texttt{SDP}_4)$, we only maintain $Z \in \mathcal{B}$ explicitely, and deal with all other constraints through Lagrangian duality. For notational convenience, let us formally denote the 3-cycle equations (12) by

$$e - A(Z) = 0.$$

The remaining $O(n^6)$ inequalities $\mathcal{M} \cap \mathcal{LS}$ are collected in $e - D(Z) \geq 0$. We consider the partial Lagrangian dual defined through the Lagrangian

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

The dual function is thus given by

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

It is not hard to verify that $(\texttt{SDP}_4)$ has strictly feasible points, so strong duality holds, and we get

$$z_{SDP} = \min_{\mu \geq 0, \lambda} f(\lambda, \mu) = f(\lambda^*, \mu^*) \leq f(\lambda, \mu), \ \forall \lambda, \mu \ \geq 0.$$

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 solving a problem over (10), for timings, see Table 2. The primal optimum Z of this semidefinite program also yields an element of the subdifferential of f. Our goal is to use this subgradient information to generate a feasible point (λ, μ) with value close to z_{SDP} with a limited number of function evaluations. To achieve this goal, we use the bundle method, see [26], which is taylored to minimize nonsmooth convex functions. It generates a sequence of iterates (λ_k, μ_k) which converge to an optimal solution. In each iteration, we evaluate f and compute an element of the subdifferential at the iterate. Since this method has a somewhat weak asymptotic convergence behaviour, we limit the number of function evaluations to control the overall computational effort. The computational results in the following sections are obtained with a limited number of such function evaluations, and leave some room for further incremental improvement.

Since inactive inequalities have no influence on f (the corresponding Lagrange multiplier is zero), we concentrate on identifying those inequalities, which are likely to be active at the optimum. Thus the actual set of constraints dualized will change during the iterations of the bundle method. We follow the approach discussed in [15] to add and drop constraints on the fly, and refer the reader interested in further details to this paper.

The high level picture is that we get an acceptable upper bound to z_{SDP} with a few hundred function evaluations of f.

7 Computational results for large-scale applications

In this section we give computational results for large-scale instances of LOP, MLCM and WB. We provide improved upper bounds for some hard LOP instances that substantially reduce the duality gap. Even more promising computational results can be given for MLCM and WB which are active areas of research with many interesting applications. MLCM is applied to the representation of the facets of polytopes [12] and inside the Sugiyama framework [44] that is used for schedules, UML diagrams, and flow charts. WB has applications in computational biology [13] and additionally the *single-row facility layout problem (SRFLP)* and the (weighted) *minimum linear arrangement problem (MinLA)* are special cases. Therefore in Subsections 7.2, 7.3 and 7.4 we just point out some selected improvements achieved and refer to separate papers [12, 28] for an in-depth (polyhedral) analysis of the specific problem structures and extensive computational comparisons with other methods to point out their potential and limitations.

7.1 Large LOP instances

From a purely theoretical point of view, it is clear that $(\mathtt{SDP}_4)$ provides the strongest relaxation. The computational approach described in the previous section typically stops after a preset number of function evaluations and therefore provides only a suboptimal solution. In our preliminary experiments on larger instances, we noticed that it is important to first get 'nearly' feasible with respect to the 3-cycle equations defining $(\mathtt{SDP}_1)$. Once this is achieved we start adding the triangle inequalities and the matrix cuts. Since their number is quite large, we analyzed experimentally their effect and noticed that the inclusion of only the most violated triangle inequalities resulted in the quickest improvement of the bound with only a limited number of function evaluations. We therefore concentrate only on getting good approximations to $(\mathtt{SDP}_2)$.

In Table 3 we summarize upper bounds for some LOP instances for which the optimal solution is not yet known (for details see [36, Tables 10,12,14]). These LOP instances can be downloaded from `http://heur.uv.es/optsicom/LOLIB`. The table identifies the instance by its name and size n. We then provide the best known (integer) solution in the column labeled *bks*. The linear programming bound (7) is given in column z_{LP}, The semidefinite bound $(\mathtt{SDP}_2)$ is (approximately) determined with 250 function evaluations. Finally, we also provide the relative gap between the best known feasible solution and the bounds in the columns *LP-gap* and *SDP-gap*. The gap (in percent) is computed as $gap = 100\frac{bound-bks}{bks}$.

In Table 4 we summarize upper bounds for large-scale instances for which again the optimal solution is not yet known. Here we used $(\mathtt{SDP}_1)$ and allowed 25 function evaluations.

For all instances in Tables 3 and 4 we are able to reduce the gap between lower and upper bound substantially. This reduction of the gaps will also lead to considerably smaller branching trees in a Branch-and-Bound approach. To illustrate this let us mention that applying a Branch-and-Bound algorithm using the lp bounds to the paley graphs 31 and 43 results in bounds still beyond 300 respectively 600 after days of branching.

graph	n	bks	z_{LP}	LP-gap	$(\mathtt{SDP}_2^{250})$	SDP-gap
pal31	31	285	310	8.77	297	4.21
pal43	43	543	602	10.87	569	4.79
pal55	55	1045	1084	3.73	1049	0.38
p50-05	50	42907	44196	3.00	43177	0.63
p50-06	50	42325	43765	3.40	42673	0.82
p50-07	50	42640	43977	3.14	42897	0.60
p50-08	50	42666	44655	4.66	43241	1.35
p50-09	50	43711	45183	3.37	43954	0.56
p50-10	50	43575	45346	4.06	44097	1.20
p50-11	50	43527	45132	3.69	43932	0.93
p50-12	50	42808	44671	4.35	43341	1.25
p50-13	50	43169	44872	3.94	43608	1.02
p50-14	50	44519	46272	3.94	44907	0.87
p50-15	50	44866	46479	3.60	45253	0.86

Table 3: Bounds for medium size LOP instances

graph	n	bks	z_{LP}	LP-gap	$(\mathtt{SDP}_1^{25})$	SDP-gap
N-t1d100.01	100	106852	114468	7.13	110314	3.24
N-t1d100.02	100	105947	114077	7.67	110321	4.13
N-t1d100.03	100	109819	117843	7.31	113926	3.74

Table 4: Bounds for large LOP instances

7.2 Bipartite crossing minimization

Bipartite crossing minimization (BCM) is a special case of MLCM where the number of levels is set to two. An exact algorithm for this problem has been introduced by Jünger and Mutzel [29] which only performs well on small, sparse instances ($n \leq 12$) [8].

For our experiments, we used the random instances from [8]. These are generated with the Stanford GraphBase generator [32] which is hardware independent. Results are reported for graphs having $n = 14, 16, 18$ vertices on each layer. For each n, we consider graphs with densities $d = 10, 20, \ldots, 90$, (in percent), i.e. with $\left\lfloor \frac{dn^2}{100} \right\rfloor$ edges. For each pair (n, d), we report the average over 10 random instances.

For this type of problem we use the strongest relaxation $(\mathtt{SDP}_4)$ as the number of violated matrix cuts (15) stays manageable for all instances considered. We also refer to Table 10 below.

Once we stop the bound computation, we use the primal solution $\tilde{Z} \in \mathcal{B}$ for hyperplane rounding, as suggested in [18]. Doing this we obtain a $\{-1, 1\}$ vector $\tilde{y}$, which need not be feasible with respect to the cycle equations in (6). An infeasible $\tilde{y}$ can be made feasible in an obvious way by flipping the signs of some of its entries, resulting in a feasible $\tilde{y}$. The most expensive step for this rounding process consists in finding the factorization $\tilde{D}\tilde{D}^T = \tilde{Z}$ of $\tilde{Z}$, to carry out the rounding procedure with $\tilde{D}$. Thus, once $\tilde{D}$ is available, we can easily afford to repeat the rounding procedure, and take the overall best solution, In our case we take the best solution out of a 1000 trials.

It turned out that this heuristic found an optimal solution for all BSM instances under consideration. The computation times were in the order of seconds, hence negligible.

Standard heuristics and also some metaheuristics perform quite poorly for MLCM instances of our size [29], [35], but it would be worth comparing metaheuristics like GRASP and tabu search [35] with our heuristic.

In Table 5 we compare our approach with the two best methods from [8]. These are the CPLEX MIP-

n	d	CPLEX [8]	SDP [8]	($\mathtt{SDP}_4$)
		time	time	time
14	20	40.7	61.9	10.2
14	40	-	97.9	25.3
14	60	-	95.9	20.2
14	80	-	101.9	19.7
16	10	2.1	119.1	2.4
16	20	2701.6	200.9	28.8
16	30	-	432.9	53.5
16	40	-	1432.0	306.3
16	50	-	1181.2	110.2
16	60	-	1186.8	89.0
16	70	-	916.9	79.1
16	80	-	444.9	57.3
16	90	-	224.1	35.6
18	10	5.5	343.2	8.9
18	30	-	1233.7	170.4
18	50	-	-	211.4
18	70	-	2624.98	314.6
18	90	-	601.30	78.6

Table 5: Bipartite crossing minimization: comparison with [8]

solver, applied to the standard linearization of the objective function in combination with the standard integer programming formulation of the linear ordering problem and a Branch-and-Bound approach using SDP that is similar to our approach.

The results show that the SDP approach of [8] allows for substantial improvements, independent of the somewhat slower machine used in [8]. The main reasons for this improvement is on one hand our careful tuning of the bounding routine, and our rounding procedure, which allowed us to prove optimality at the root node, while [8] had to go through a few steps of branching before being able to prove optimality.

For a generalization of the SDP approach to multiple layers, detailed polyhedral studies of the crossing polytope and an extensive computational comparison with a state-of-the-art ILP approach, we refer to the recent paper by Chimani et al. [12].

7.3 Minimum linear arrangement problem

The *minimum linear arrangement problem (MinLA)* is a special case of WB, but can also be defined independently as follows. Given an undirected graph $G(V, E)$ find a permutation $\phi : V \to \{1, \dots, n\}$ minimizing $\sum_{i,j \in E} |\phi(i) - \phi(j)|$.

$$\phi_1(G) := \min_{\phi \in \Pi} \sum_{i,j \in E} |\phi(i) - \phi(j)|.$$

To see that MinLA can be modeled as (quadratic) ordering problem we note that $\phi_1(G)$ can also be expressed as

$$\phi_1(G) = \min \sum_{(i,j) \in E} \sum_k x_{ik} x_{kj},$$

where x_{ij} satisfies (2)–(4). The term $\sum_k x_{ik} x_{kj}$ 'counts', how many nodes k lie between i and j in the ordering ϕ defined by x, as $x_{ik} x_{kj} = 1$ precisely if k lies between i and j.

For MinLA we work again with the strongest relaxation $(\mathtt{SDP}_4)$ as the number of violated matrix cuts (15) stays quite small for all instances considered. Again, we refer to Table 10 below.

To get a first idea of the tightness of the SDP approach we apply it to get bounds for $\phi_1(Q_n)$, where Q_n denotes the n-dimensional Cartesian cube. The value

$$\phi_1(Q_n) = 2^{n-1}(2^n - 1)$$

was determined by Harper [20]. For further comparison we also give the lower combinatorial bounds from [23] and [30] respectively. The results are summarized in Table 6. The relaxation $(\mathtt{SDP}_4)$ correctly identifies $\phi_1(Q_n)$ for all $n \leq 4$.

n	Reference [30]	Reference [23]	$(\mathtt{SDP}_4)$	Optimum $\phi(Q_n)$
2	5	6	6	6
3	21	24	28	28
4	85	99	120	120
5	341	392	493	496
6	1365	1542	2002	2016

Table 6: Bounds for $\phi_1(Q_n)$ on the hypercube Q_n

For MinLA there exists another very recent algorithm of Caprara et al. [9], realized by Schwarz [39], that is preferable to the SDP approach for small graphs and large, sparse graphs.

In Table 7 we give some instances from the Boeing Sparse Matrix Collection [14], where we could prove optimality of the upper bounds UB for the first time together with [9] and also improve the best known lower bound for two instances. As we get our lower bounds from the root node relaxation, we are very optimistic to solve these instances when using our bounds in a Branch-and-Bound approach. Table 7 starts with the instance name, the number of nodes n, the density d of the instance and the upper bound (UB) obtained by a multi-start local search routine. All cited algorithms were run on a 2× Xeon CPU with 2.5 GHz. Empty entries indicate that the instance was not tackled by the specific algorithm.

Instance				Reference [39]		Reference [10]		$(\mathtt{SDP}_4)$	
name	n	d	UB	LB	Time	LB	Time	LB	Time
can_24	24	24.6	210	210	4.7	203	2.8	210	66.9
fidap005	27	35.8	414	414	4.1	412	4.2	414	124.4
pores_1	30	23.6	383	383	29.9			383	286.5
ibm32	32	18.1	485	485	1241.3			485	306.1
fidapm05	42	27.7	1003	1003	1516.9	998	805.2	1003	6200.1
bcsstk01	48	15.6	1132	1132	40852.8	972	3848.1	1130	10744.5
impcol_b	59	16.4	2076	2000	limit			2074	51082.6
dwt_59	59	6.0	289	289	39.4	258	55.4	289	37925.3
gd95c	62	7.6	506	506	109.7	443	68.3	506	36647.7
can_73	73	5.7	1100	962	limit	971	2016.8	1088	limit

Table 7: Comparison of several exact approaches to linear arrangement (time limit is 24h)

7.4 The single-row facility layout problem

An instance of the *single-row facility layout problem (SRFLP)* consists of n one-dimensional facilities, with given positive lengths $l_1, \ldots, l_n$, and pairwise weights c_{ij}. Now the task in SRFLP is to find a linear ordering of the departments such that the total weighted sum of the center-to-center distances between all pairs of facilities is minimized. SRFLP is again a special case of WB, where the weights of triples (i, j, k) are set to $w_{ijk} := c_{ik}l_j$ and $\sum_{i<j} \frac{c_{ij}(l_i+l_j)}{2}$ is added as a constant to the objective function.

In Table 8 we compare the strongest relaxation $(\texttt{SDP}_4)$ with another SDP approach from [5]. In [5] they use $(\texttt{SDP}_1)$ as basic relaxation and then add the 300 to 400 most-violated triangle inequalities (13) to the model in every iteration and re-optimize until no more triangle inequalities are violated. The computations in [5] were carried out on a 2.0GHz Dual Opteron.

Using the bundle method instead of cutting planes and additionally adding the matrix cuts (15) yields extremely improved computation times especially for large instances. The problems are collected from different sources and include well-known benchmark instances [25, 41], instances with clearance requirement [25] and random-generated instances [5].[1] For the upper bound computation we used the heuristic described in Subsection 7.2 where again the computation times were in the order of seconds for all instances.

Instance	n	CPU time (h:min:sec) [5]	CPU time $(\texttt{SDP}_4)$ (min:sec)
Lit-3	11	0:00:33	0:02
Lit-4	20	0:26:54	0:54
Lit-5	30	15:50:57	9:07
Lit-Cl-5	12	0:00:33	0:08
Lit-Cl-6	15	0:05:53	0:20
Lit-Cl-7	20	0:41:32	1:16
Lit-Cl-8	30	51:06:53	14:17
Nugent25-01	25	3:44:38	2:48
Nugent25-02	25	4:50:27	5:46
Nugent25-03	25	5:48:21	4:11
Nugent25-04	25	4:04:51	5:33
Nugent25-05	25	8:22:22	3:31
Nugent30-01	30	7:41:06	4:42
Nugent30-02	30	10:41:53	6:08
Nugent30-03	30	19:32:01	10:12
Nugent30-04	30	31:03:11	11:44
Nugent30-05	30	19:54:07	18:30

Table 8: Comparison of two SDP approaches using well-known SRFLP instances

Recently Amaral and Letchford [2,3] proposed a cutting plane algorithm that improved on the algorithm in [5]. Amaral also introduced new instances with 33 and 35 departments, solved them to optimality and pointed out that he could not solve larger instances with his approach as the involved linear programs became too large and too difficult. We can also solve Amaral's new instances [2] to optimality with our approach and even achieve better computation times. For the detailed polyhedral and computational comparison of the two approaches we refer to [28]. There we also successfully apply $(\texttt{SDP}_4)$ to the even larger SRFLP instances with up to 100 departments from [4,6]. In Table 9 we state the optimal values and solution times for five new instances with 40 departments, a density of 50 % and random weights between 1 and 10.

7.5 Model validation for QOPs

Finally we show that including the matrix cuts (15) in the SDP model yields essential improvements for all types of quadratic ordering instances.

First in Table 10 we compare the number of inequalities of the different constraint types that are considered by the bundle method using the strongest relaxation $(\texttt{SDP}_4)$ at the last function evaluation.

[1] All considered instances can be downloaded from `http://flplib.uwaterloo.ca/`.

Instance	n	Optimal cost	CPU time (SDP$_4$) (h:min:sec)
N40_1	40	107348.5	1:01:36
N40_2	40	97693	0:52:52
N40_3	40	78589.5	1:21:40
N40_4	40	76669	1:15:58
N40_5	40	103009	2:20:09

Table 9: Solution values and times for new large SRFLP instances with 40 departments

We examine two LOP instances from Table 3, a random BCM instance with 16 nodes on each layer and a density of 40 % from Table 5 and a MinLA instance from Table 6. For the LOP instances the number of matrix cuts (15) is very large and therefore constricts the overall performance of the algorithm. In contrast the number of matrix cuts stays quite small for the two quadratic ordering instances.

graph	problem type	p	# (12)	# (15)	# (13)
pal 31	LOP	466	5803	746559	14339
N-p50-05	LOP	1226	4874	399720	6800
bcm.16.40	BCM	241	1262	2524	64903
cube5	MinLA	497	5730	192	36768

Table 10: Number of constraints considered by the bundle method

Despite of their small number the matrix cuts help a lot to tighten the relaxation of quadratic ordering instances as can be seen from Table 11. In the last line of the table we give in brackets the number of function evaluations (*fe*) needed to prove optimality for the BCM instance.

graph	# fe	opt	(SDP$_2$)	(SDP$_4$)
cube5	300	496	490.28	491.47
	400		490.91	492.11
	500		491.21	492.32
	600		491.32	492.41
bcm.16.40	300	1397	1395.18	1395.42
	400		1395.62	1395.88
			1396.01 (595)	1396.01 (455)

Table 11: Gain of relaxation tightness through matrix cuts

8 Conclusions

In this paper, we have presented a systematic investigation and comparison of SDP based relaxations to ordering problems. We demonstrated that semidefinite relaxations provide substantially tighter bounds than linear programming relaxations for the linear ordering problem. As semidefinite relaxations are the natural setting for quadratic ordering problems, we also applied them to crossing minimization, single-row layout and linear arrangement problems. For all these different fields of application with their rich, mainly independent bodies of literature, we could produce new superior bounding results compared with the multiple, diverse approaches developed so far. This generality distinguishes the semidefinite approach. We only have to adapt the cost function to compute all kinds of quadratic ordering problems and also combinations of them. Another mentionable feature of our approach is that the computation times mainly depend on the number of nodes in the underlying graphs and are not so much influenced by their density structures, which can be a great advantage compared to the other methods for special types of quadratic ordering problems that all depend on exploiting sparsity.

Therefore it seems to be worthwhile to think about ways to further improve the presented approach. There are two (combinable) directions to enhance the presented SDP based relaxations of ordering problems. On the one hand, we could include further constraint classes to further tighten the relaxation and on the other hand, we could incorporate the SDP based bounds in a Branch-and-Bound framework.

Acknowledgement: We thank Gerhard Reinelt and Marcus Oswald from Universität Heidelberg for many fruitful discussions and practical hints regarding the polyhedral approach to LOP.

References

[1] H. Achatz, P. Kleinschmidt, and J. Lambsdorff. Der corruption perceptions index und das linear ordering problem. *ORNews*, 26:10–12, 2006.

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

[3] 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*, 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 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.

[6] 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.

[7] K. Boenchendorf. *Reihenfolgenprobleme/Mean-flow-time sequencing.* Mathematical Systems in Economics 74. Verlagsgruppe Athenaum, Hain, Scriptor, 1982.

[8] C. Buchheim, A. Wiegele, and L. Zheng. Exact Algorithms for the Quadratic Linear Ordering Problem. *INFORMS Journal On Computing*, pages 168–177, 2009.

[9] A. Caprara, M. Jung, M. Oswald, G. Reinelt, and E. Traversi. A betweenness approach for solving the linear arrangement problem. *In preparation*, 2010.

[10] A. Caprara, A. N. Letchford, and J.-J. Salazar-Gonzales. Decorous lower bounds for minimum linear arrangement. *to appear in: INFORMS Journal on Computing*, 2010.

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

[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. to appear.

[13] T. Christof, M. Oswald, and G. Reinelt. Consecutive ones and a betweenness problem in computational biology. In *Proceedings of the 6th Conference on Integer Programming and Combinatorial Optimization (IPCO 1998)*, Lecture Notes in Computer Science 1412, pages 213–228. Springer, 1998.

[14] I. S. Duff, R. G. Grimes, and J. G. Lewis. Users' guide for the harwell-boeing sparse matrix collection. Technical report, CERFACS, Toulouse, France, 1992.

[15] 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.

[16] 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, NY, USA, 1974. ACM.

[17] F. Glover, T. Klastorin, and D. Klingman. Optimal weighted ancestry relationships. *Management Science*, 20:1190–1193, 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] M. Grötschel, M. Jünger, and G. Reinelt. A Cutting Plane Algorithm for the Linear Ordering Problem. *Operations Reasearch*, 32(6):1195–1220, 1984.

[20] L. H. Harper. Optimal assignments of numbers to vertices. *SIAM Journal on Applied Mathematics*, 12(1):131–135, 1964.

[21] P. Healy and A. Kuusik. The vertex-exchange graph: A new concept for multilevel crossing minimisation. In *Proceedings of the Symposium on Graph Drawing [GD'99]*, pages 205–216. Springer, 1999.

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

[23] C. Helmberg, B. Mohar, S. Poljak, and F. Rendl. A spectral approach to bandwidth and separator problems in graphs. *Linear and Multilinear Algebra*, 39:73–90, 1995.

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

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

[26] J.-B. Hiriart-Urruty and C. Lemarechal. *Convex Analysis and minimization algorithms (vol. 1 and 2).* Springer, 1993.

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

[28] P. Hungerländer and F. Rendl. Improvements for the single-row facility layout problem. *In preparation*, 2011.

[29] M. Jünger and P. Mutzel. 2-layer straightline crossing minimization: performance of exact and heuristic algorithms. *Journal of Graph Algorithms and Applications*, 1:1–25, 1997.

[30] M. Juvan and B. Mohar. Optimal linear labelings and eigenvalues of graphs. *Discrete Applied Mathematics*, 36(2):153–168, 1992.

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

[32] D. E. Knuth. *The Stanford GraphBase: a platform for combinatorial computing.* ACM, New York, NY, USA, 1993.

[33] W. Leontief. Quantitative input-output relations in the economic system of the united states. *The Review of Economics and Statistics*, 18, 1936.

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

[35] R. Martí and M. Laguna. Heuristics and meta-heuristics for 2-layer straight line crossing minimization. *Discrete Applied Mathematics*, 127(3):665–678, 2003.

[36] R. Mart, G. Reinelt, and A. Duarte. A Benchmark Library and a Comparison of Heuristic Methods for the Linear Ordering Problem. *to appear in: Computational Optimization and Applications*, 2010.

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

[38] A. Newman. Cuts and orderings: On semidefinite relaxations for the linear ordering problem. In K. Jansen, S. Khanna, J. Rolim, and D. Ron, editors, *Lecture Notes in Computer Science*, volume 3122, pages 195–206. Springer, 2004.

[39] R. Schwarz. *A branch-and-cut algorithm with betweenness variables for the linear arrangement problems.* Diploma Thesis, Heidelberg, 2010.

[40] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[41] D. M. Simmons. One-Dimensional Space Allocation: An Ordering Algorithm. *Operations Research*, 17(5):812–826, 1969.

[42] C. D. Simone. The cut polytope and the Boolean quadric polytope. *Discrete Mathematics*, 79(1):71–75, 1990.

[43] J. Sturm. *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones (Updated for version 1.05).* Available under http://sedumi.ie.lehigh.edu, 2001.

[44] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981.