Exact Approaches to Multi-level Vertical Orderings

Markus Chimani[1]
Institut of Computer Science, Friedrich-Schiller-University Jena, Germany, markus.chimani@uni-jena.de
Philipp Hungerländer
Institut of Mathematics, Alpen-Adria-University Klagenfurt, Austria, philipp.hungerlaender@uni-klu.ac.at

Technical Report TR-FSUJ-CS-AE-11-02, v.1
Algorithm Engineering Group, Computer Science, FSU Jena
June 2011

We present a semidefinite programming (SDP) approach for the problem of ordering vertices of a layered graph such that the edges of the graph are drawn as vertical as possible. This *multi-level vertical ordering (MLVO)* problem falls into the class of quadratic ordering problems. It is conceptually related to the well-studied problem of multi-level crossing minimization (MLCM), but offers certain interesting novel properties: we not only have to consider the pure relative ordering of the nodes, but their final absolute ranks (i.e., positions) within the ordered levels. Furthermore, MLVO is a natural quadratic problem that does not only consist of multiple sequentially linked bilevel quadratic ordering problems, but is a genuine multi-level quadratic ordering problem. This allows us to describe the graphs' structures more compactly and therefore obtain (near-)optimal, (well-)readable drawings of graphs too large for MLCM.

We show (theoretically and experimentally) that these properties lead to the situation that approaches based on ILPs and QPs are inapplicable, even for small sparse graphs, while the SDP works surprisingly well in practice. This is in stark contrast to other ordering problems as, e.g., MLCM, where such graphs are typically solved more efficiently with ILPs.

In this paper we present a motivation, mathematical models, strengthening constraints for ILPs and QPs, and an SDP relaxation for MLVO. We compare the relevant models from the polyhedral point of view, and conduct a series of experiments (including a comparison to MLCM) to showcase our SDP's applicability. We conclude with sketching further applications in scheduling and ranking problems, where MLVO occurs apart from graph drawing.

Key words: Quadratic ordering problem, ILP and SDP approaches, multi-layer graph drawings, crossing minimization.
History: ...

1. Introduction

In this paper, we study the *Multi-level Vertical Ordering* (MLVO) problem from a polyhedral and experimental point of view. Generally, the problem can be described as a combination of multiple linear ordering problems (each is considered a *level*); instead of (only) having costs between elements within a level, the (main) costs arise from the *positional differences* between elements of distinct levels. For reasons that will become evident below, these latter differences can be considered as *(non-)verticalities*. In the following, we will consider MLVO in a graph drawing setting, where it arises most naturally and allows the probably simplest introduction into the problem class. Concentrating on this specific application furthermore allows comparisons with related research, mainly the well-studied problem of multi-level crossing minimization (MLCM). Yet, note that the MLVO problem by itself is of more general nature; we will showcase some additional applications in Section 6.

One of the most common graph drawing paradigms, known as Sugiyama's framework [35], is based on the following idea: First, we place the nodes of a graph on different *levels* (effectively fixing their y-coordinates). Edges spanning multiple levels are subdivided into a chain of edges such that each edge only spans one level. The second step is to fix an ordering of the nodes on their levels such that a certain optimization goal (usually the number of crossings) is minimized. As a third step, the nodes are assigned x-coordinates, consistent with the ordering, such that the number of bends is minimized or the edges' *verticality* is maximized. Edges are thereby always drawn as straight lines.

In this paper we discuss a somehow inverse approach to the problem of finding a good node order on the levels, focusing on third step's optimization goal. We observe that when thinking about a drawing where the edges are drawn mostly vertical, we will usually also have a low number of crossings. Furthermore, edges tend to cross only on a very "local" scale (i.e., edges will usually not cross over a large horizontal distance), increasing the drawing's readability [28]. Hence, perhaps the combination of maximum verticality and low crossing number leads to (qualitatively) better drawings than the traditional minimum crossing number in conjunction with high verticality.

We call the problem of finding orderings of the nodes on their levels such that the edges are drawn *as vertical as possible* (see a precise definition below) the *Multi-level Vertical Ordering (MLVO)* problem. As we will see, it is a natural quadratic problem. Of course MLVO is NP-hard and closely related to the traditional problem of minimizing the number of crossings in multi-level drawings, which has received a lot of attention not only within the graph drawing community, but in combinatorial optimization in general. Apart from many heuristic approaches, there were also multiple exact algorithms proposed: Jünger and Mutzel [23] presented an integer linear program for the 2-level problem, and the induced special case where the order of one of the two levels is fixed. This approach, based on *linear ordering problems* (LOPs), was generalized to the multi-level case by Jünger et al. [22], later improved by Healy and Kuusik [17, 18], and favorably compared to an alternative exact approach based on solving SAT instances by Gange et al. [14]. Buchheim et al. [6] showed that an approach based on semidefinite programs (SDPs) is beneficial over the ILP formulation for (dense) two-level problems, exploiting the stronger bounds of SDP relaxations for *quadratic ordering problems* (QOPs). Recently, Chimani et al. [9] showed how to further improve on this approach and generalize it to the multi-level case.

MLVO falls into the class of ordering problems. Besides LOP and MLCM, well-studied problems like linear arrangement [7, 30, 31], single row facility layout [2, 4] and weighted betweenness [11] belong to this class. Recently, several ILP and SDP approaches were applied with varying degree of success to deal with different ordering problems [3, 5]. While there exist quite diverse ILP approaches for the different ordering problems, the universal SDP approach is designed to deal with general QOPs [21].

Finally, problems related to MLVO also occur in computational geometry, e.g., when considering "optimal shapes" of towns [12], where n buildings are be placed on a 2-dimensional integer grid, and the aim is to minimize the manhattan distances between any pair of buildings.

In the remainder of this paper, we will always consider the following input: Let $G = (V, E)$ with $V = \dot{\bigcup}_{i=1}^{p} V_i$ be a *level graph*, where we draw the nodes V_i on the i-th level. The function $\ell : V \to \{1, \ldots, p\}$ gives the level on which a node resides. Furthermore, let $G' = (V', E')$ with $V' = \dot{\bigcup}_{i=1}^{p} V'_i$, $E' = \dot{\bigcup}_{i=1}^{p-1} E'_i$, and $E'_i \subseteq V'_i \times V'_{i+1}$ for all $1 \leq i < p$, be the corresponding *proper level graph*. Thereby, the original edges E are subdivided into segments such that each edge in E' connects nodes of adjacent levels. Clearly, we have $V_i \subseteq V'_i$ for all levels i. The additional nodes created by this operation are called *long-edge dummy nodes*, or *LEDs* for short.

Multi-level crossing minimization and multi-level planarization [14, 26] are always applied to *proper* level graphs, as only the introduction of LEDs allows to concisely describe their feasible solutions and objective values. Optimizing these problems means solving $p-1$ dependent, sequentially linked bilevel QOPs (one for each pair of adjacent levels).

We can apply MLVO to proper level graphs (*Proper MLVO*). Furthermore, MLVO seems to be the first optimization problem considered in this realm that can also naturally and reasonably be applied to non-proper level graphs directly (*Non-proper MLVO*), resulting in genuine multi-level QOPs. This is particularly interesting with respect to semidefinite programs: As noted before, SDPs have already shown great potential for MLCM. Yet, when considering the cost matrix, we can observe that it is constructed of (non-zero) sub-matrices along its main diagonal; all other entries of the matrix are 0. Non-proper MLVO seems to be the first multi-level problem using the full SDP structure, resulting in denser cost matrices of smaller dimension. Therefore we can obtain (near-)optimal, (well-)readable drawings of graphs too large for Proper MLVO or MLCM. Furthermore, for both MLVO variants the pure relative node ordering is not sufficient to evaluate the objective function, but we also need to take the resulting absolute position of an element (first, second, etc.) into account. This induces a more complex quadratic cost structure on the involved levels compared to, e.g., crossing minimization, needing more of the implicitly defined SDP variables and therefore leading again to denser SDP cost matrices. But the more SDP variables and therefore SDP structure is needed, the better the SDP performs compared to competing ILP approaches (in the context of ordering problems).

In the following, we will clarify our main optimization goal and describe a basic mathematical model with linear constraints and a quadratic objective function, for Proper and Non-proper MLVO. Yet, as our experiments reveal[2], neither it nor its linearization, even after adding some new provably strengthening inequalities, are directly applicable even to small scale instances. We therefore show how to instead develop a related semidefinite program. While the aforementioned ILP/QP approaches suffer from very weak relaxations and require a very large branch-and-bound tree, our SDP approach does not implement any branching; already the SDP relaxation often suffices to prove optimal solutions or at least comparably tight bounds. We demonstrate these effects on well-known graph drawing instances in Section 5 and then conclude with some further applications and final remarks.

While this paper still focuses on the graph drawing application, as this is the most direct and general application field of MLVO we know of, we herein concentrate on exact appraoches

[2] All instances and results can be found at `http://www.ae.uni-jena.de/Research_Pubs/MLVO.html`

thereto. We refer to the companion paper [8] for a deeper introduction into the practical merits of verticality optimization in the graph drawing setting, as well as for the precise non-proper drawing scheme and algorithm, and for further heuristic approaches to tackle the problem.

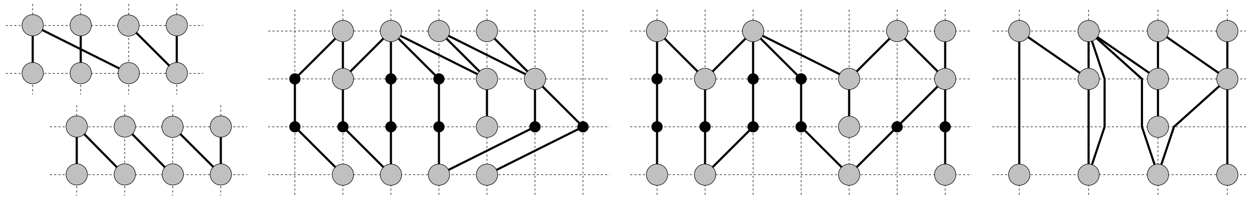
2. Verticality

We define the colloquial term *verticality* via its inverse, *non-verticality*: The non-verticality $\mathfrak{d}(e)$ of a straight-line edge e is the square of the difference in the horizontal coordinates of its end nodes. Then, $\mathfrak{d}(E) := \sum_{e \in E} \mathfrak{d}(e)$ denotes the overall non-verticality of a solution. Using only this notion, we could arbitrarily optimize a drawing by scaling the horizontal coordinates. Hence we consider *grid drawings*, i.e., the nodes' positions are mapped to integral coordinates, thereby relating verticality to the drawing's width. Clearly, we only consider *adjacent* integrals for the y-coordinates. It remains to argue why non-verticality has to be a quadratic term: assume we would only consider a linear function, then even a small example as the one depicted in Fig. 1(a) would result in multiple solutions that are equivalent w.r.t. their objective values, even though the bottom one is clearly preferable from the readability point of view. Intuitively, we prefer multiple slightly non-vertical edges, over few very non-vertical edges. In fact, this argument brings our model in line with the argument of observing crossings only on a local scale.

We can consider two distinct alignment schemes, due to the fact that the node partitions V'_i have different cardinalities. Let $\omega' := \max_{1 \leq i \leq p} |V'_i|$ denote the width of the widest level. In the *narrow* alignment schemes, we require the nodes on the levels to lie on directly adjacent x-coordinates (Fig. 1(b)). Usually, we would like to center the distinct levels w.r.t. each other, i.e, a level i may only use the x-coordinates $\{\delta'_i, \ldots, \delta'_i + |V'_i| - 1\}$, with the level's *width offset* $\delta'_i := \lfloor (\omega' - |V'_i|)/2 \rfloor$.

In the *wide* alignment scheme (Fig. 1(c)), nodes are not restricted to lie on horizontally neighboring grid coordinates. In order to model this in our optimization framework, we expand the graph by adding *positional dummy nodes* (PDs) to each level such that all levels have ω' many nodes. All PDs have degree 0. Since this addition is the only necessary modification to obtain this alignment scheme[3], we will in the following continue to consider

[3]We can trivially force some predefined relative order of all PDs of a common level by fixing the corresponding ordering variables introduced later. This reduces the symmetry of this expansion and is therefore beneficial for branch-and-bound approaches.



(a) Linear Cost (b) Proper, narrow (c) Proper, wide (d) Non-proper, wide

Figure 1: Example drawings regarding verticality maximization: (a) equivalent quality with respect to a linear objective function, (b)–(d) different drawing paradigms, cf. text. Original nodes are drawn as large gray circles, LEDs as black small circles, PDs (on the empty grid points) are omitted for readability.

any (proper) level graph $G^{(\prime)}$, which may or may not be augmented with PDs.

Non-Proper Drawing Scheme. On the one hand, disregarding LEDs in our drawing style and thus considering a smaller graph, potentially improves the running times of our algorithm. On the other hand, this idea also has a foundation in graph drawing applications: As LEDs will never be drawn in the resulting drawing, it is unreasonable for them to require as much horizontal space as a real node. Furthermore, current drawing algorithms even try hard to "bundle" multiple long edges into one dense channel (whose width is constant, disregarding the number of its elements), to improve overall readability of large graphs, see, e.g., [27].

Hence, when considering a non-proper level graph G, we only ask for an ordering of the original nodes (and PDs, if any). Non-verticality is still measured by the horizontal difference of the end nodes. When drawing G, each edge is routed to the left or right of its lower end node, until the level directly below the upper end node. Only there, the edge bends to be drawn as a line with the computed non-verticality, cf. Fig. 1(d); see [8] for a more detailed description of this drawing style and its efficient computation based on a multi-level ordering.

3. Basic Mathematical Models

Consider the proper level graph G'. Similar to the other approaches for multi-level optimizations, we can model the node order by introducing binary variables, assuming some fixed total order $\dot{<}$ of the nodes (e.g., based on their indices).

$$x'_{uv} \in \{0,1\}, \qquad \forall u,v \in V'_i,\ 1 \le i \le p,\ u \dot{<} v. \tag{1}$$

The variables shall be 1 if u is left of v and 0 otherwise. For notational simplicity, we also use the shorthand $x'_{uv} := 1 - x'_{vu}$ for $v \dot{<} u$. It is well-known that feasible orderings can be described via *3-cycle inequalities*

$$0 \leq x'_{uv} + x'_{vw} - x'_{uw} \leq 1, \qquad \forall u, v, w \in V'_i,\ 1 \leq i \leq p,\ u \dot{<} v \dot{<} w. \tag{2}$$

For each edge $(u, v) \in E'$, we introduce a (conceptually integral) variable $d'_{(u,v)}$ measuring the end-nodes' horizontal distance, i.e., $\sqrt{\mathfrak{d}((u,v))}$: Consider the shorthand $X'(u) := \delta'_{\ell(u)} + \sum_{w \in V'_{\ell(u)}} x'_{wu}$ which gives the number of nodes left of u plus the level's width offset, and therefore the x-coordinate of u. Then the horizontal distance between u and v is $|X'(u) - X'(v)|$. In linear terms we can hence require

$$d'_{(u,v)} \geq X'(u) - X'(v)\ , \quad d'_{(u,v)} \geq X'(v) - X'(u), \qquad \forall (u, v) \in E'. \tag{3}$$

This allows us to give a mathematical model with linear constraints but quadratic objective function to solve the Proper MLVO problem:

$$v'_* = \min \left\{ \sum\nolimits_{e \in E'} (d'_e)^2, \text{ subject to } (1) - (3) \right\}. \tag{DM'}$$

Theorem 1. *Every optimal solution to* `(DM')` *induces an optimal solution to the MLVO problem on proper level graphs, and vice versa.*

By replacing the integrality conditions in `(DM')` with 0-1 bounds we obtain a quadratic programming relaxation denoted by `(cDM')`.

Let x' be the vector collecting all variables x'_{uv}. We can write $\mathfrak{d}(E')$ as a linear-quadratic function in x', without the explicit need for any d'-variables:

$$\mathfrak{d}(E') = \sum_{(u,v) \in E'} (X'(u) - X'(v))^2 \stackrel{!}{=} \begin{pmatrix} 1 \\ x' \end{pmatrix}^\top D' \begin{pmatrix} 1 \\ x' \end{pmatrix}, \tag{4}$$

for some suitable matrix D'. Therefore we obtain an equivalent formulation to `(DM')`.

Lemma 2. `(DM')` *and its relaxation* `(cDM')` *give the same values as*

$$\min \left\{ \begin{pmatrix} 1 \\ x' \end{pmatrix}^\top D' \begin{pmatrix} 1 \\ x' \end{pmatrix}, \textit{ subject to } (1) \textit{ and } (2) \right\} \tag{OM'}$$

and its relaxation `(cOM')`, *respectively.*

ILP and Applicability. Modern mathematical programming software can often already deal with models with linear constraints and quadratic objective functions. Yet, one naturally may try to linearize the models. The ILPs for the MLCM problem, e.g., can be seen as linearized models from the originally quadratic problem, and they are known to outperform SDP approaches for sparse graphs [9] with density $\leq 10\%$. Yet, we observe that thereby only few products (especially for sparse graphs) of two binary variables have to be linearized.

In our first model (DM'), we have squares of arbitrary integers, only bounded by $\omega' - 1$. We can linearize any $(d'_e)^2$ by adding variables $d'_{e,i} \geq 0$ and requiring $d'_{e,i} \geq d'_e - i$, for all $1 \leq i < \omega' - 1$. The objective function then becomes $\sum_{e \in E'}(d'_e + 2d'_{e,1} + 2d'_{e,2} + \ldots)$.

In order to obtain an ILP from our second model (OM'), we would have to linearize $\approx \sum_{1 \leq i < p} \binom{|V'_i|}{2}\binom{|V'_{i+1}|}{2}$ products of two binary variables. This number can be compared to MLCM on completely dense graphs, for which, e.g., [9, Table 1] shows that the SDP clearly outperforms the ILP.

Although the above models suffice w.r.t. integral solutions, their relaxations can be further strengthened. On the one hand, any polyhedral improvement for the ordering variables directly carries through to MLVO. On the other hand we can add new classes of strengthening inequalities:

Degree Constraints. Consider some node $u \in V'_i$ and all its adjacent nodes N' on some neighboring level j (either above or below). For $\alpha := |N'|$, $\alpha \geq 2$, we can require

$$\sum\nolimits_{v \in N'} d'_{(u,v)} \geq \lfloor \alpha/2 \rfloor \cdot \lceil \alpha/2 \rceil, \tag{5}$$

$$\sum\nolimits_{v \in N'} (d'_{(u,v)})^2 \geq \begin{cases} \alpha(\alpha^2 - 1)/12, & \text{if } \alpha \text{ odd}, \\ \alpha(\alpha^2 + 2)/12, & \text{if } \alpha \text{ even}, \end{cases} \tag{6}$$

based on the fact that the minimum possible overall non-verticality is achieved when tightly packing N' and centering it above/below u. The second (quadratic) constraint thereby is a strengthening of the former (linear) constraint. Using the extended d-variables (let $d'_{\cdot,0} := d'_{\cdot}$ for notational simplicity), we can linearize (6) as

$$\sum\nolimits_{v \in N'} d'_{(u,v),i} \geq \lfloor \alpha/2 - i \rfloor \cdot \lceil \alpha/2 - i \rceil, \qquad \forall 0 \leq i < \lfloor \alpha/2 \rfloor. \tag{7}$$

Complete-Bipartite Constraints. We can generalize the former constraint class by considering complete bipartite subgraphs on consecutive levels. Let $N'_i \subseteq V'_i$ and $N'_{i+1} \subseteq V'_{i+1}$, for some $1 \leq i < p$, be two node sets such that $N'_i \times N'_{i+1} \subseteq E'_i$. Let $\beta := \min\{|N'_i|, |N'_{i+1}|\}$ and $\gamma := \max\{|N'_i|, |N'_{i+1}|\}$. We can require

$$\sum\nolimits_{u \in N'_i} \sum\nolimits_{v \in N'_{i+1}} d'_{(u,v)} \geq \beta \cdot \lfloor \gamma/2 \rfloor \cdot \lceil \gamma/2 \rceil + \begin{cases} \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil, & \text{if } \gamma \text{ odd}, \\ \lfloor \beta/2 \rfloor \cdot (\lceil \beta/2 \rceil - 1), & \text{if } \gamma \text{ even}, \end{cases} \tag{8}$$

where the right hand side gives the minimum possible overall horizontal distances achieved by tightly packing the node sets on their levels, and centering them over each other. Again, we can strengthen (8) considering squared d-variables and the accordingly increased right-hand side instead:

$$\sum_{u \in N_i, v \in N_j} (d'_{(u,v)})^2 \geq \begin{cases} \beta\gamma(\gamma^2 + \beta^2 - 2)/12 & \text{if } \beta \text{ and } \gamma \text{ have the same parity}, \\ \beta\gamma(\gamma^2 + \beta^2 + 1)/12 & \text{otherwise}. \end{cases} \tag{9}$$

Furthermore, we can reuse the correspondingly modified formulas of (8) to again linearize the quadratic constraints:

$$\sum_{u \in N_i, v \in N_j} d'_{(u,v),i} \geq \beta \cdot \lfloor \gamma/2 - i \rfloor \cdot \lceil \gamma/2 - i \rceil + \begin{cases} \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil, & \text{if } \gamma \text{ odd}, \forall 0 \leq i < \lfloor \gamma/2 \rfloor, \\ \lfloor \beta/2 \rfloor \cdot (\lceil \beta/2 \rceil - 1), & \text{if } \gamma \text{ even}, \forall 0 \leq i < \lfloor \gamma/2 \rfloor. \end{cases} \tag{10}$$

See Section 3.1 for further details on both constraint classes and their right hand sides. Next let us point out polyhedral properties of degree and complete-bipartite constraints.

Lemma 3. *Degree constraints (even in their weaker form* (5)*) strengthen* (cDM')*. Complete-bipartite constraints (even in their weaker form* (8)*) further strengthen the relaxation, even if it already satisfies all degree constraints* (6)*.*

Proof. Observe that the ordering constraints allow to set all x'-variables to 0.5. Hence all horizontal node positions are identical, all d'-variables can be 0, and the relaxation has an optimal solution of 0, as well. As long as there is at least one node with two neighbors on the level above (or below), the corresponding degree constraint forces the associated d'-variables to be non-zero and increase the solution value.

Assume we have an LP solution feasible w.r.t. (DM') and all degree constraints; again all x'-variables can be 0.5. Let u_1, u_2 be two nodes on a common level, both adjacent to three nodes v_1, v_2, v_3 on an adjacent level. The degree constraints essentially only require as much horizontal distance (or non-verticality) as achieved by centering both u_1, u_2 at the same center position. Applying the complete-bipartite constraint for this structure hence requires larger values for the corresponding d'-variables, raising the objective value. □

Non-Proper MLVO. We can directly rewrite (DM') to non-proper level graphs, obtaining the formulation (DM): we order all nodes V level-wise via variables x, and measure the non-verticality of the edges E via variables d, using $\omega := \max_{1\leq i\leq p} |V_i|$, $\delta_i := \lfloor(\omega - |V_i|)/2\rfloor$, and $X(u) := \delta_{\ell(u)} + \sum_{v\in V_{\ell(u)}} x_{vu}$:

$$v_* = \min\Big\{ \sum\nolimits_{e\in E} (d_e)^2 : 0 \leq x_{uv} + x_{vw} - x_{uw} \leq 1,\ X(a) - X(b) \leq d_{(a,b)} \geq X(b) - X(a)$$
$$x_{uv} \in \{0,1\}, \quad \forall 1 \leq i \leq p,\ u,v,w \in V_i,\ u\dot{<}v\dot{<}w,\ (a,b) \in E\Big\}. \tag{DM}$$

Analogously to above, replacing the integrality conditions with 0-1 bounds gives (cDM). We can again strengthen (cDM) via corresponding degree and complete-bipartite constraints. Observe that for the former, it suffices that the nodes N lie on *some* common level, not necessarily a neighboring level. Similarly, the node sets for the latter constraints do not have to be on neighboring levels, i.e., we have $N_i \subseteq V_i$, $N_j \subseteq V_j$, for some $1 \leq i < j \leq p$, and $N_i \times N_j \subseteq E$. Note that we can also define (OM) and its relaxation (cOM) analogously to above. But for (OM), the situation turns out to be even worse (compared to the Proper MLVO case) when considering its linearization: the cost matrix D becomes completely dense. The clearly resulting drawback is also supported by the results in [5, Table 2].

3.1 Supplementary: Right Hand Sides of Strengthening Constraints

Right hand side of the degree constraints. In order to obtain the right hand side of the degree constraints, consider all nodes N' being placed directly next to each other, and u centered below or above these nodes. We can sum the arising horizontal distances as

$$\underbrace{0+1+1+2+2+3+\ldots}_{\alpha\text{ many}} = \sum_{i=1}^{\lfloor\alpha/2\rfloor} i + \sum_{i=1}^{\lceil\alpha/2\rceil-1} i = \lfloor\alpha/2\rfloor\lceil\alpha/2\rceil,$$

where the latter function is obtainable via case distinction on whether α is odd or even.

Similarly, we can sum up the arising non-verticalities (i.e., squares of the horizontal distances) as

$$\underbrace{0^2+1^2+1^2+2^2+2^2+3^2+\ldots}_{\alpha\text{ many}} = \sum_{i=1}^{\lfloor\alpha/2\rfloor} i^2 + \sum_{i=1}^{\lceil\alpha/2\rceil-1} i^2,$$

which gives – using $\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$ and a case distinction on whether α is odd or even – the specified right hand side. We can further linearize the quadratic constraint by asking for the extended variables $d_{e,i}$ that their sum is greater or equal the sum of distances reduced by i.

Right hand side of the complete-bipartite constraints. Assume w.l.o.g. that $|N'_i| = \beta$. Placing each node v of N'_i beneath the center of compactly positioned nodes N'_{i+1}, we would attain $\beta \cdot \lfloor \gamma/2 \rfloor \cdot \lceil \gamma/2 \rceil$ overall horizontal distances, according to the degree constraints' right hand sides. Yet, not all nodes N'_i can be placed at the same center positions (or at one of the two center positions for γ even); they have to be tightly grouped around the center. Each shift of a node by one position further away from center position adds 1 to the overall horizontal distances of its incident edges. By case distinction based on the parity of β and γ we obtain the formula. E.g., if both cardinalities are odd, there are exactly two nodes being shifted by i positions, for $1 \leq i \leq \lfloor \beta/2 \rfloor$, resulting in an additional overall sum of horizontal distances of $2 \cdot \frac{1}{2} \cdot \lfloor \beta/2 \rfloor \cdot (\lfloor \beta/2 \rfloor + 1) = \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil$. Analogously, we can sum the minimally occurring non-verticalities as

$$\sum_{i=1}^{\beta} \sum_{j=1}^{\gamma} (\lfloor (\gamma - \beta)/2 \rfloor + i - j)^2.$$

After transforming this, using the above formula for the sum of increasing squares and considering a case distinction on the parities of β and γ, this constitutes the right hand side of the quadratic complete-bipartite constraints.

Again we can further linearize the quadratic constraints by requiring for the extended variables $d_{e,i}$ that their sum is greater or equal the sum of distances reduced by i. Simplifying the resulting expression by using (8) gives the right hand side of the linearized constraints.

4. Semidefinite Relaxation

In this section we concentrate on the bound computation for (Non-)Proper MLVO[4] by analyzing matrix liftings of ordering problems. For this purpose, it is convenient to transform the linear ordering variables x into y-variables taking the values -1 and 1. We use the linear transformation $y_{uv} = 2x_{uv} - 1$ for all $u \dot{<} v \in V$, and rewrite (2) as the equivalent inequalities

$$-1 \leq y_{uv} + y_{vw} - y_{uw} \leq 1, \qquad \forall u, v, w \in V_i, 1 \leq i \leq p,\ u \dot{<} v \dot{<} w. \tag{11}$$

We know from [19] that this switch between $\{0, 1\}$ and $\{-1, 1\}$ formulations of bivalent problems preserves structural properties, in particular, the resulting bounds. We apply a matrix lifting approach to MLVO by taking the vector y and considering the matrix

[4]Both cases are virtually identical for the SDP approach. For notational simplicity, we will use the variable naming scheme of the non-proper setting.

$Y = yy^T$. We are interested in *multi-level quadratic orderings (MQOs)* and therefore consider the polytope

$$\mathcal{P}_{MQO} := \text{conv}\left\{ \begin{pmatrix}1\\y\end{pmatrix}\begin{pmatrix}1\\y\end{pmatrix}^{\top} : y \in \{-1,1\},\ y \text{ satisfies (11)} \right\}.$$

We relax the non-convex equation $Y - yy^T = 0$ to the constraint $Y - yy^T \succcurlyeq 0$, which is convex due to the Schur-complement lemma. Moreover, the main diagonal entries of Y correspond to y_{uv}^2, and hence $\text{diag}(Y) = e$, the vector of all ones. To simplify our notation, we introduce

$$Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \tag{12}$$

where $\zeta := \dim(Z) = 1 + \sum_{i=1}^{p} \binom{|V_i|}{2}$ and $Z = (z_{ij})$. We have $Y - yy^T \succcurlyeq 0 \Leftrightarrow Z \succcurlyeq 0$. Hence, $\mathcal{P}_{MQO}$ is contained in the elliptope

$$\mathcal{E} := \{\ Z :\ \text{diag}(Z) = e, Z \succcurlyeq 0\ \}. \tag{13}$$

In order to express constraints on y in terms of Y, we reformulate them as quadratic conditions in y. Using $y \in \{-1,1\}$ in (11) gives $|y_{uv} + y_{vw} - y_{wu}| = 1$. By squaring both sides and using $y_{uv}^2 = 1$ we obtain

$$y_{uv}y_{vw} - y_{uv}y_{uw} - y_{uw}y_{vw} = -1, \qquad \forall u, v, w \in V_i,\ 1 \leq i \leq p,\ u \dot{<} v \dot{<} w. \tag{14}$$

Applying the dimension result for QOP from [6] to MQO, it is easy to deduce that (14) describes the smallest linear subspace containing $\mathcal{P}_{MQO}$.

We can hence formulate MLVO as a semidefinite optimization problem in bivalent variables, where Z is given by (12), and C is a symmetric cost matrix of order ζ assigned to give $\mathfrak{d}(E)$ for any given feasible ordering y: We can write $\mathfrak{d}(E)$ as a linear quadratic function in y as

$$\mathfrak{d}(E) = \sum_{(u,v)\in E} (X(u) - X(v))^2 =$$

$$\sum_{(u,v)\in E} \frac{1}{4}\left[\left(\sum_{\substack{t\in V_{\ell(u)}\\ t\neq u}} y_{ut} + g_{\ell(u)}\right) - \left(\sum_{\substack{w\in V_{\ell(v)}\\ w\neq v}} y_{vw} + g_{\ell(v)}\right)\right]^2 \stackrel{!}{=} \langle C, Z\rangle,$$

where $g_{\ell(u)} := (\omega - |V_{\ell(u)}|) \mod 2$. Expanding and using $y_{uv}^2 = 1$ yields

$$\mathfrak{d}(E) = \sum_{(u,v)\in E} \frac{1}{4}\Bigg[(g_{\ell(u)} - g_{\ell(v)})^2 + |V_{\ell(u)}| + |V_{\ell(v)}| - 2 + 2\Bigg((g_{\ell(u)} - g_{\ell(v)}) \sum_{\substack{t\in V_{\ell(u)}\\ t\neq u}} y_{ut} + (g_{\ell(v)} - g_{\ell(u)}) \sum_{\substack{w\in V_{\ell(v)}\\ w\neq v}} y_{vw} + \sum_{\substack{t,w\ \in V_{\ell(u)},\ t\dot{<}w\\ t\neq u,\ w\neq u}} y_{ut}y_{uw} + \sum_{\substack{t,w\ \in V_{\ell(v)},\ t\dot{<}w\\ t\neq v,\ w\neq v}} y_{vt}y_{vw} - \sum_{\substack{t\in V_{\ell(u)},\ t\neq i\\ w\in V_{\ell(v)},\ w\neq v}} y_{ut}y_{vw}\Bigg)\Bigg] \stackrel{!}{=} \langle C, Z\rangle. \tag{15}$$

Now (15) can be directly applied to define the semidefinite cost matrix C for MLVO.

We can show that our basic model ($\texttt{SDP}_1$) with integrality constraints on the first row and column of Z is exact.

Theorem 4. *MLVO is equivalent to the problem* $v_* = \min\{\ \langle C, Z\rangle\ :\ Z \in \mathcal{I}_{MQO}\ \}$, *where*

$$\mathcal{I}_{MQO} := \{\ Z\ :\ Z \text{ partitioned as in } (12) \text{ and satisfies } (14),\ Z \in \mathcal{E},\ y \in \{-1, 1\}\ \}. \tag{16}$$

Proof. Since $y_{uv}^2 = 1$ we have $\operatorname{diag}(Y - yy^T) = 0$, which together with $Y - yy^T \succcurlyeq 0$ shows that in fact $Y = yy^T$. The 3-cycle equations (14) ensure that $|y_{uv} + y_{vw} - y_{uw}| = 1$ holds. Therefore any matrix Z feasible for (16) is bivalent in its entries and represents feasible orderings on all levels. Thus by definition of the cost matrix C, the objective value $\langle C, Z\rangle$ gives $\mathfrak{d}(E)$ for any feasible configuration. □

By dropping the integrality of y, we get the following basic semidefinite relaxation for MLVO

$$\min\{\ \langle C, Z\rangle\ :\ Z \text{ partitioned as in (12) and satisfies (14)},\ Z \in \mathcal{E}\ \}. \tag{$\texttt{SDP}_1$}$$

There are some obvious ways to tighten ($\texttt{SDP}_1$). First, we observe that $Z \in I_{MQO}$ is a matrix with $\{-1, 1\}$ entries. Hence it satisfies the triangle inequalities, defining the metric polytope

$$\mathcal{M} := \left\{ Z\ :\ \begin{pmatrix} -1 & -1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} z_{ij} \\ z_{jk} \\ z_{ik} \end{pmatrix} \le e,\ \forall\ 1 \le i < j < k \le \zeta \right\}. \tag{17}$$

$\mathcal{M}$ is defined through $4\binom{\zeta}{3} = O((\sum_{i=1}^p |V_i|^2)^3)$ facets that are used as triangle inequalities of the max-cut polytope in [24, 32, 34]. ($\texttt{SDP}_1$) can therefore be improved by additionally asking that $Z \in \mathcal{M}$.

Another generic improvement was suggested by Lovász and Schrijver in [24]. Applied to our problem, their approach suggests to multiply the 3-cycle inequalities (11) (on level i, say) by the nonnegative expressions $(1-y_{uv})$, $(1+y_{uv})$, $(1-y_{uv}-y_{vw}+y_{uw})$ and $(1+y_{uv}+y_{vw}-y_{uw})$, respectively, where the nodes $u\dot{<}v\dot{<}w$ are on some (probably different) level j. There are $O((\sum_{i=1}^{p}|V_i|^3)^2)$ such *LS-cuts* and we define

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies all LS-cuts}\ \}.$$ [5]

Overall, we get the following tractable relaxation of $\mathcal{P}_{MQO}$, part of which (without $\mathcal{LS}$) has been investigated in [6] for bipartite crossing minimization, in [4] for single-row layout, and, including part of $\mathcal{LS}$, in [21] for general quadratic linear ordering problems and in [9] for multi-level crossing minimization

$$\min\ \{\ \langle C, Z\rangle\ :\ Z \text{ partitioned as in (12) and satisfies (14)},\ Z \in (\mathcal{E}\cap\mathcal{M}\cap\mathcal{LS})\ \}. \qquad (\mathtt{SDP}_2)$$

We close this section by relating the semidefinite relaxation $(\mathtt{SDP}_1)$ to the quadratic programming relaxation (cDM) incorporating degree and complete-bipartite constraints.

Theorem 5. $(\mathtt{SDP}_1)$ *is at least as strong as* (cDM) *together with the quadratic degree constraints* (6) *and quadratic complete-bipartite constraints.*

Proof. First, it is not hard to verify that any Z feasible for $(\mathtt{SDP}_1)$ contains a vector y in its first column that satisfies the 3-cycle inequalities (11) on the levels. This follows from the semidefiniteness of the following submatrices of Z

$$\begin{pmatrix} 1 & y_{uv} & y_{uw} & y_{vw} \\ y_{uv} & 1 & y_{uv,uw} & y_{uv,vw} \\ y_{uw} & y_{uw,uv} & 1 & y_{uw,vw} \\ y_{vw} & y_{vw,uv} & y_{vw,uw} & 1 \end{pmatrix}, \quad \forall u,v,w \in V,\ u\dot{<}v\dot{<}w.$$

Constraints (3) are implicitly ensured by the definition of the cost matrix C through (15). Next let $N_i \subseteq V_i$ and $N_j \subseteq V_j$, for some $1 \le i < j \le p$, be two node sets such that $N_i \times N_j \subseteq E$. Applying (15) for $\beta := \min\{|N_i|, |N_j|\}$ and $\gamma := \max\{|N_i|, |N_j|\}$ with $\gamma - \beta$

[5] In our computational experiments, we only use a part of $\mathcal{LS}$ that was also considered in previous publications [9, 21].

even to the left hand side of (9) yields

$$\begin{aligned}\sum_{\substack{u\in N_i,\\ v\in N_j}} d^2_{(u,v)} = \frac{1}{4}\sum_{\substack{u\in N_i,\\ v\in N_j}}\left[\beta+\gamma-2+2\left(\sum_{\substack{t,w\ \in N_i,\ t\lessdot w\\ t\neq u,\ w\neq u}} y_{ut}y_{uw}+\sum_{\substack{t,w\ \in N_j,\ t\lessdot w\\ t\neq v,\ w\neq v}} y_{vt}y_{vw}-\sum_{\substack{t\in N_i,\ t\neq i\\ w\in N_j,\ w\neq v}} y_{ut}y_{vw}\right)\right] = \\ \frac{\beta\gamma(\beta+\gamma-2)}{4}+\frac{1}{2}\sum_{\substack{u\in N_i,\\ v\in N_j}}\left(\sum_{\substack{t,w\ \in N_i,\ t\lessdot w\\ t\neq u,\ w\neq u}} y_{ut}y_{uw}+\sum_{\substack{t,w\ \in N_j,\ t\lessdot w\\ t\neq v,\ w\neq v}} y_{vt}y_{vw}-\sum_{\substack{t\in N_i,\ t\neq i\\ w\in N_j,\ w\neq v}} y_{ut}y_{vw}\right) = \\ \frac{\beta\gamma(\beta+\gamma-2)}{4}+\frac{1}{2}\sum_{v\in N_j}\left(\sum_{\substack{u,t,w\ \in N_i,\\ u\lessdot t\lessdot w}} y_{ut}y_{uw}-\sum_{\substack{u,t,w\ \in N_i,\\ t\lessdot u\lessdot w}} y_{ut}y_{uw})+\sum_{\substack{u,t,w\ \in N_i,\\ t\lessdot w\lessdot u}} y_{ut}y_{uw})\right)+ \\ \frac{1}{2}\sum_{u\in N_i}\left(\sum_{\substack{v,t,w\ \in N_j,\\ v\lessdot t\lessdot w}} y_{vt}y_{vw}-\sum_{\substack{v,t,w\ \in N_j,\\ t\lessdot v\lessdot w}} y_{vt}y_{vw}+\sum_{\substack{v,t,w\ \in N_j,\\ t\lessdot w\lessdot v}} y_{vt}y_{vw}\right)- \\ \frac{1}{2}\left(\sum_{\substack{u\lessdot t\in N_i,\\ v\lessdot w\in N_j}} y_{ut}y_{vw}-\sum_{\substack{t\lessdot u\in N_i,\\ v\lessdot w\in N_j}} y_{tu}y_{vw}-\sum_{\substack{u\lessdot t\in N_i,\\ w\lessdot v\in N_j}} y_{ut}y_{wv}+\sum_{\substack{t\lessdot u\in N_i,\\ w\lessdot v\in N_j}} y_{tu}y_{wv}\right).\end{aligned} \tag{18}$$

The terms in the last line of (18) cancel each other. Summing up $-(14)$ for all elements in N_i and N_j, respectively, and applying it to (18) gives

$$\frac{\beta\gamma(\beta+\gamma-2)}{4}+\frac{\beta\gamma(\beta-1)(\beta-2)}{12}+\frac{\beta\gamma(\gamma-1)(\gamma-2)}{12}=\frac{\beta\gamma(\gamma^2+\beta^2-2)}{12}.$$

Applying (15) for $\gamma-\beta$ odd to the left hand side of (9) yields

$$\begin{aligned}\sum_{u\in N_i, v\in N_j} d^2_{(u,v)} = \frac{1}{4}\sum_{u\in N_i, v\in N_j}(\beta+\gamma-1)+\frac{1}{2}\sum_{u\in N_i, v\in N_j}\left(\sum_{\substack{t,w\ \in N_i,\ t\lessdot w\\ t\neq u,\ w\neq u}} y_{ut}y_{uw}+\sum_{\substack{t,w\ \in N_j,\ t\lessdot w\\ t\neq v,\ w\neq v}} y_{vt}y_{vw}\right)+ \\ \frac{1}{2}\sum_{u\in N_i, v\in N_j}\left(\pm\sum_{\substack{t\in V_{\ell(u)}\\ t\neq u}} y_{ut}\mp\sum_{\substack{w\in V_{\ell(v)}\\ w\neq v}} y_{vw}-\sum_{\substack{t\in V_{\ell(u)},\ t\neq i\\ w\in V_{\ell(v)},\ w\neq v}} y_{ut}y_{vw}\right).\end{aligned} \tag{19}$$

Again the three double sums in the second line of (19) give 0. Summing up $-(14)$ for all elements in N_i and N_j, respectively, and applying it to (19) gives

$$\frac{\beta\gamma(\beta+\gamma-1)}{4}+\frac{\beta\gamma(\beta-1)(\beta-2)}{12}+\frac{\beta\gamma(\gamma-1)(\gamma-2)}{12}=\frac{\beta\gamma(\gamma^2+\beta^2+1)}{12}.$$

As the degree constraints are special complete-bipartite constraints with $\beta=1$, they are also satisfied on any matrix feasible for $(\mathtt{SDP}_1)$. □

In summary, (SDP$_1$) is required to ensure all constraints from Section 3. We also included $\mathcal{M}$ and $\mathcal{LS}$ to ensure that (SDP$_2$) contains all facets of $\mathcal{P}^7_{LO}$ and $\mathcal{P}^4_{MQO}$ respectively.[6] Furthermore, $\mathcal{M}$ is necessary to solve graphs to optimality that only contain the edges required for a degree constraint with $\alpha = 4$ (or, more generally, graphs that only contain the edges required for complete-bipartite constraints with $\gamma - \beta = 3$), where the smaller level is filled up with PDs. For solving analogous graphs with $\gamma - \beta > 3$ odd exactly, we would have to consider additional clique inequalities of size > 3 odd in the relaxation. Yet, separating them is far too expensive, which motivates our model choice.

4.0.1 Solving (SDP$_2$)– Lower Bound.

Looking at the constraint classes and their sizes in the relaxation (SDP$_2$), it is clear that explicitly maintaining $O(\sum_{i=1}^{p} |V_i|^3)$ or more constraints is not an attractive option. We therefore consider an approach originally suggested in [13], which was applied to the max cut problem [29] and several ordering problems [9, 21], and adapt it MLVO. Initially, we only aim at explicitly maintaining that Z lies in the elliptope $\mathcal{E}$, which can be achieved with standard interior point methods, see, e.g., [20].

All other constraints are dealt with through Lagrangian duality. For notational convenience, let us formally denote the 3-cycle equations (14) by $e - \mathcal{A}(Z) = 0$. Similarly we write $\mathcal{M} \cap \mathcal{LS}$ as $e - \mathcal{D}(Z) \geq 0$. Using the Lagrangian $\mathcal{L}(Z, \lambda, \mu) := \langle C, Z \rangle + \lambda^\top (e - \mathcal{A}(Z)) + \mu^\top (e - \mathcal{D}(Z))$, we obtain the partial Lagrangian dual $f(\lambda, \mu) := \min_{Z \in \mathcal{B}} \mathcal{L}(Z, \lambda, \mu) = e^\top \lambda + e^\top \mu + \min_{Z \in \mathcal{B}} \langle C - \mathcal{A}^\top(\lambda) - \mathcal{D}^\top(\mu), Z \rangle$. Since (SDP$_2$) 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 solving a problem over $\mathcal{E}$. In our experiments, we use a primal-dual interior-point method, which also provides a primal optimum $Z_{\lambda,\mu}$ yielding a subgradient of f. Using these ingredients, we get an approximate minimizer of f using subgradient optimization techniques such as the bundle method [13]. Since these methods have a rather weak local convergence behavior, we limit the number of function evaluations to control the overall computational effort. In fact these evaluations constitute the computational bottleneck as they are responsible for more than 95% of the required running time.

[6] We computed the facets of these polytopes with PORTA [10].

4.0.2 Solving $(\texttt{SDP}_2)$– Upper Bound.

Our heuristic exploits information obtained during the bundle method, and follows the general idea sketched in [21]: Initially, we consider a vector y', that encodes a feasible, random ordering on all levels. The algorithm stops after 1000 executions[7] of step 2; y' is then the heuristic solution. If the duality gap is not closed after the heuristic, we continue with further bundle iterations and then retry the heuristic (retaining the last vector y').

1. Let Y'' be the current primal fractional solution of $(\texttt{SDP}_2)$ obtained by the bundle method. Compute the convex combination $R := \lambda(y'y'^\top) + (1-\lambda)Y''$, using some random $\lambda \in [0.3, 0.7]$. Compute the Cholesky decomposition $DD^\top$ of R.
2. Apply Goemans-Williamson hyperplane rounding [15] to D and obtain a $-1/+1$ vector $\overline{y}$ (cf. [29]).
3. Compute the induced crossing number $z(\overline{y})$. If $v(\overline{y}) \geq v(y')$: goto step 2.
4. If $\overline{y}$ satisfies all 3-cycle inequalities: set $y' := \overline{y}$ and goto 2. Else: modify $\overline{y}$ by changing the signs of one of three variables in all violated inequalities and goto step 3.

In practice, it turns out that the repair strategy in the last step is not as critical as one might assume. In fact, we know from MLCM that an analogous heuristic performs astonishingly well in practice, dominating traditional heuristic approaches [9].

5. Experiments

All SDP computations were conducted on an Intel Xeon E5160 (Dual-Core) with 24 GB RAM, running Debian 5.0 in 32bit mode. The algorithm itself runs on top of MatLab 7.7. We restrict the SDP approach to 1500 function evaluations of $f(\lambda, \mu)$, as the convergence process of the bundle method mostly slows down before that point, independent of problem size ζ.

We consider input graphs from three different sources, which are often considered in related experimental investigations, e.g., [9, 14, 17, 18, 22]. Table 1 gives the instances' central properties; the graphs are available at `http://www.ae.uni-jena.de/Research_Pubs/MLVO.html`:

Polytopes. Often, one considers the graphs modeling the incidence relation between faces (corner, edge, 2D-face,...) of an (LP-)polytope, and hence we are interested in drawing them.

[7] Before its 501st execution, we perform step 1 again. As this is quite expensive, we refrain from executing it too often.

We choose from a wide variety starting with a simple 3-dimensional tetrahedron to the face polyhedral body of a soccer ball.

Graphviz gallery. The gallery [16] constitutes a set of various real-world graphs from different applications. We only consider the largest of these graphs, as only they constitute difficult problems for our approach.

Other literature. The two *worldcup* instances, visualizing soccer world cup results of the full history up until the specified year, were proposed in [1, Fig. 12&13]. The graphs MS88 [25] and SM96 [33] are well known instances recurring in multiple publications, and represent certain social networks. Due to the size of the latter, a 3-level subgraph of SM96 is also often considered. As this subgraph includes many originally LEDs even on the lowest and highest level, we cannot reasonably consider this reduced graph w.r.t. Non-proper MLVO.

5.0.3 SDP vs. Other Variants.

We start with evaluating the alternatives to the SDP approach. We already argued why the linearization of `(OM')` and `(OM)` will be unfavorable compared to the SDP. Hence (concentrating on Proper MLVO) it remains to evaluate solving `(DM')` directly (using CPLEX's built in QP solver) and solving its linearization (introducing the variables $d_{e,i}$ described above). We denote these approaches by *DMQ* and *DML*, respectively. Furthermore, we want to investigate the influence of the quadratic objective function w.r.t. to the program's solvability. Therefore we consider the ILP which only minimizes $\sum_{e \in E'} d'_e$, instead of the sum of squares. Yet note that this ILP clearly does *not* really solve the MLVO problem as defined above. We denote this algorithm by *LC*. All these three variants use CPLEX 12.1's branch-and-cut framework and their applicable form of the degree constraints; the 3-cycle inequalities are separated dynamically, implemented in C++. Due to licensing issues, these algorithms are conducted on an Intel Xeon E5520 (Dual-CPU, Quad-Core) with 72 GB RAM, running Debian 6.0 in 32bit mode. Note that this machine/software configuration can safely be considered speed-wise stronger than the one used for the SDP approach.

For both alternative approaches *DMQ* and *DML*, and even for *LC*, we observe running times that are orders of magnitudes larger than the SDP's, cf. Table 2. We observe that this is mainly due to the weak lower bounds and the resulting large number of required branch-and-bound nodes. Recall that the SDP results are always obtained without any branching. In fact, the non-SDP algorithms ran out of memory in all but the smallest instance. At they end, they obtained clearly weaker lower and upper bounds than the SDP approach, although

			Proper								Non-proper							
	Instance	p	$\|V'\|$	$\|E'\|$	ω'	dens.	ζ	d_C	ζ^+	d_C^+	$\|V\|$	$\|E\|$	ω	dens.	ζ	d_C	ζ^+	d_C^+
Polytopes	Tetrahedron	3	14	24	6	0.50	28	0.58	46	0.40	*always proper*							
	Octahedron	3	26	48	12	0.29	110	0.45	199	0.17								
	Cube3	3	26	48	12	0.29	110	0.45	199	0.27								
	Dodecahedron	3	62	120	30	0.13	692	0.24	1306	0.13								
	Icosahedron	3	62	120	30	0.13	692	0.24	1306	0.13								
	Cube4	4	80	208	32	0.14	921	0.25	1985	0.10								
	Soccer ball	3	182	360	90	0.04	6272	0.09	12016	0.05								
Graphviz	switch	6	48	64	8	0.20	169	0.22	169	0.22	*already proper*							
	unix	11	59	66	11	0.19	176	0.16	606	0.04	41	48	7	0.06	77	0.22	232	0.07
	world	9	116	137	20	0.09	815	0.11	1711	0.04	48	69	9	0.07	132	0.27	325	0.10
	profile	9	92	116	28	0.08	846	0.14	3403	0.02	61	85	14	0.06	309	0.21	820	0.05
Other	MS88	3	37	80	15	0.24	217	0.38	316	0.26	*already proper*							
	worldcup86	4	35	55	19	0.19	223	0.30	685	0.08	25	45	11	0.22	92	0.46	221	0.18
	worldcup02	4	50	65	23	0.11	411	0.23	1013	0.07	31	46	14	0.15	149	0.34	365	0.13
	SM96-3L	3	61	58	26	0.07	615	0.16	976	0.11	*not applicable*							
	SM96-full	7	108	179	26	0.10	915	0.13	2276	0.05	63	134	14	0.08	295	0.26	638	0.11

Cube3 and Cube 4 correspond to a 3- and 4-dimensional cube, respectively.

$\omega^{(\prime)}$ denotes the maximum width of any level, *dens.* the graph's density relative to the case of all possible edges.

The columns ζ and d_C (ζ^+ and d_C^+) give the resulting dimension and density of the SDP cost matrix for the narrow (wide) alignment scheme, respectively.

Table 1: Instance properties.

	DMQ			*DML*			*LC*			SDP	
Instance	v_*	bb	time	v_*	bb	time	(v_*)	bb	time	v_*	time
Cube3	262	3.5	1:57:16	262	0.3	0:13:29	(94)	1.4	0:33:59	$261^{+}1$	0:01:34
Icosahedron	115/–	0.5	[174h]	166/3566	0.4	[99h]	(122/510)	0.6	[105h]	$3046^{+}34$	4:51:06
profile	{566/–}	{0.7}	{240h}	583/1565	0.3	[51h]	(177/279)	0.5	[105h]	$1303^{+}9$	7:09:51
SM96-full	{100/–}	{0.4}	{240h}	138/1595	0.3	[42h]	(123/364)	0.4	[57h]	$1212^{+}13$	8:47:37

Table 2: Comparing *DMQ*, *DML*, and *LC* to the SDP approach on selected instances. The column *bb* gives the number of branch-and-bound nodes in millions. Number of hours in square brackets denote when the program runs out of memory (32bit), data in curly brackets denote when the program was terminated after 10 days, as the lower bounds stagnated for over 100h. v^* gives either the optimal solution or the final lower bound and the absolute gap to the upper bound: "$a^{+}b$" means lower bound a, absolute gap b, upper bound $a+b$; when the gaps become large, we write a/c instead, where c is the upper bound.

they required much more CPU time. We conclude that these approaches are no match for the SDP and concentrate on the latter in the following.

5.0.4 SDP Solvability.

We conducted experiments of the SDP approach for Proper and Non-proper MLVO. For both objective functions, we considered the narrow and the wide alignment scheme. Table 3 gives an overview of our results. We observe that the SDP relaxation is tight enough to find and prove optimal solutions, supported by our SDP based upper bound heuristic, for the smaller instances, and gives surprisingly small gaps for the instances of challenging size (on which we concentrate in this abstract).

The approach's running time is mainly dependent on ζ. Hence it is not surprising that the somehow "nicer" wide alignment scheme – requiring a much larger matrix Z – comes at a non-trivial cost w.r.t. the running time. Dropping the LEDs and solving the Non-proper MLVO SDP with a smaller but more tightly packed cost matrix instead, allows us to go well beyond the graph sizes to which the Proper MLVO and MLCM (which cannot be directly applied to a non-proper setting) approaches are restricted.

5.0.5 MLVO vs. MLCM.

As described in the paper's introduction, the standard graph drawing scheme would be to optimize the node orderings w.r.t. the minimum crossing number. As it is not the focus of this optimization oriented abstract, we refrain from an in-depth qualitative comparison between both approaches (yet, at the end of this section we show the potential of MLVO w.r.t. the

		Proper MLVO				Non-proper MLVO			
		narrow		wide		narrow		wide	
	Instance	v_*	time	v_*	time	v_*	time	v_*	time
Polytopes	Tetrahedron	48	2.27	48	2.27	*always proper*			
	Octahedron	261^{+1}	0:02:37	239^{+5}	0:03:28				
	Cube3	261^{+1}	0:01:34	239^{+5}	0:04:11				
	Dodecahedron	3051^{+27}	3:31:58	1815^{+81}	29:55:48				
	Icosahedron	3046^{+34}	4:51:06	1807^{+61}	27:10:23				
	Cube4	6336^{+86}	7:57:46	5279^{+121}	80:49:47				
Graphviz	switch	53^{+1}	0:05:54	53^{+1}	0:05:54	*already proper*			
	unix	111	0:04:27	58^{+5}	1:19:41	49	5.3	30^{+3}	0:10:11
	world	620^{+41}	6:33:10	331^{+95}	54:30:21	129	0:02:06	103^{+7}	0:43:50
	profile	1303^{+9}	7:09:51	876^{+169}	95:45:58	367^{+2}	0:23:56	254^{+5}	3:11:43
Other	MS88	249	0:01:27	155^{+2}	0:52:17	*already proper*			
	Worldcup86	559	0:05:43	349^{+26}	1:44:46	116	19.6	113^{+3}	0:31:30
	Worldcup02	501^{+1}	1:24:56	385^{+15}	7:19:38	167	0:05:26	150^{+1}	0:36:42
	SM96-3L	108^{+4}	2:30:57	43^{+8}	6:52:03	*not applicable*			
	SM96-full	1212^{+13}	8:47:37	658^{+36}	137:21:07	655	0:09:37	408^{+9}	2:16:06

Table 3: SDP approach for different MLVO variants. The time is suitably given either in seconds or as hh:mm:ss. Due to its complexity, we only computed 250 function evaluations of $f(\lambda, \mu)$ for the proper *profile* instance with wide alignment scheme.

drawings' readability on one exemplary instance). Our central interest is to investigate the corresponding optimization strategies. We give a comparison between our MLVO SDP, and the results of the currently strongest SDP and a state-of-the-art ILP approach for MLCM, lifted from [9].[8] Note that for MLCM, the ILP can benefit from the relative sparsity of the linearized variables, and is hence better suited for small and sparse input instances than the SDP approach. We already discussed (and verified experimentally above) that this is not the case for MLVO.

We compare the MLCM approaches to their closest relative in the MLVO setting: Proper MLVO with narrow alignment scheme. Table 4 gives an overview. We observe that MLVO is harder than MLCM from the SDP point of view: in general, MLVO requires more computing time and cannot close the optimiality gap as often. Being able to optimize both MLCM and MLVO using an SDP on common variables also gives rise to the idea of combining them using an objective function where the cost matrices $C_{\mathrm{MLCM}}, C_{\mathrm{MLVO}}$ are balanced via coefficients $c_{\mathrm{z}}, c_{\mathrm{v}}$, respectively. We should be careful when choosing these coefficients to still allow some kind of rounding of lower bounds. To demonstrate the applicability of this approach (not arguing over the visual merits of certain blending coefficients), we choose integrals $c_{\mathrm{z}} = 10$

[8]The MLCM SDP (ILP) experiments were run on the same machine/setting as our MLVO SDPs (ILPs, respectively).

and $c_\mathrm{v} = 1$ (such that $c_\mathrm{z}/c_\mathrm{v} \approx \sum v_*/\sum z_*$). We denote this combined problem by *MLCVO*. Table 4 shows that MLCVO is in general harder than MLCM but easier than MLVO. The resulting solutions seem to yield quite convincing compromises between both objectives.

Example Drawings.

Although not the focus of this paper, we showcase the visual results and relative benefits of the various problem solutions in Figures 2 and 3.

6. Applications Beyond Graph Drawing

We want to conclude with noting that MLVO can also be directly applied to other seemingly very different problem classes unrelated to graph drawing: Consider a *scheduling problem* with multiple machines, where each machine has multiple pre-assigned jobs. The jobs are related to each other in such a way that certain jobs should be finished at similar times. Modeling machines as levels, jobs as nodes, time as horizontal coordinates, and job relations as edges, we directly obtain a Non-proper MLVO problem.

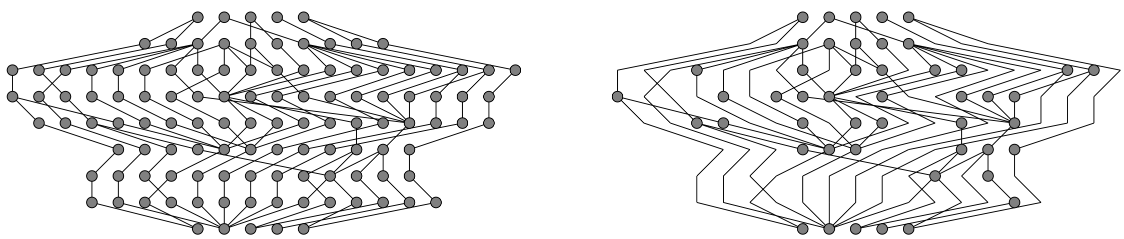
Another application, also giving a (Non-)Proper MLVO instance can be found in *multiple ranking*, where we have groups of objects, objects have relationships (e.g., similarities) with objects from other groups, and we want to (linearly) rank the objects within their groups such that related objects are ranked similarly over all groups. This can be seen as a generalization of maximum weight matchings, where the relative positions of all objects are considered in a quadratic cost setting.

In the following, we will discuss some problem variants less abstractly, with the focus on showcasing the problem's versatility: As a tongue-in-cheek example, consider a restaurant that offers a menu which lists *food categories* (e.g., soup, main dish, side dish, etc.) and allows to choose one or more *kinds* per category (e.g., the main dish may be steak, turkey, or fish). After the guests have ordered, the following problem arises: Although the restaurant has one cook per food category, each cook wants to prepare all items of the same kind (e.g., all ordered steaks), before preparing a different kind (e.g., before preparing fish). Assume that we do not want the guests to wait long between separate courses, and recognize that, e.g., the main dish should always be accompanied with the side dish at the same time. In which order should the cooks prepare their items (kinds, in fact), such that the guests get their menu with all kinds being resonable warm/fresh?

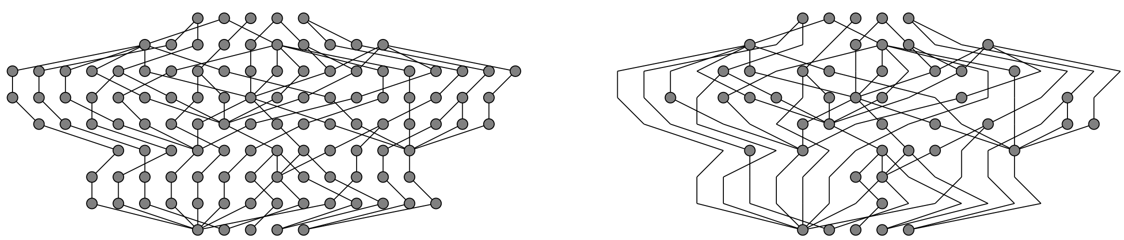
		MLCM					MLVO			MLCVO			
	Instance	z_*	v	d_C	time	ILP	z	v_*	time	b_*	z	v	time
Polytopes	Tetrahedron	22	48	0.245	0.08	0.12	24	48	2.27	268	22	48	2.83
	Octahedron	80	264	0.077	10.66	2.62	81	$261^{+}1$	0:02:37	$1063^{+}1$	80	264	0:04:46
	Cube3	80	264	0.077	10.93	3.14	81	$261^{+}1$	0:01:34	$1063^{+}1$	80	264	0:05:52
	Dodecahedron	$393^{+}1$	3096	0.014	4:40:09	(132/427)	399	$3051^{+}27$	3:31:58	$6972^{+}48$	394	3080	2:49:21
	Icosahedron	$393^{+}1$	3148	0.014	4:37:25	(174/401)	395	$3046^{+}34$	4:51:06	$6968^{+}52$	394	3080	4:18:09
	Cube4	$1192^{+}3$	6594	0.017	7:10:19	(197/1334)	1247	$6336^{+}86$	7:57:46	$18416^{+}128$	1195	6594	9:02:52
	Soccer ball	1627^{+} 726	72648	0.002	91:34:16	(118/2630)	2681	52392^{+} 21759	141:56:52	82898^{+} 15900	2538	73418	123:23:29
Graphviz	switch	20	56	0.024	1.92	0.66	23	$53^{+}1$	0:05:54	1064	20	56	0:01:34
	unix	0	141	0.011	0.25	0.01	7	111	0:04:27	126	0	126	0:03:19
	world	46	847	0.003	1:13:49	16.08	83	$620^{+}41$	6:33:10	$1221^{+}13$	50	734	8:39:12
	profile	37	2767	0.003	0:53:34	2.84	75	$1303^{+}9$	7:09:51	$1835^{+}2$	45	1387	7:19:20
Other	MS88	91	300	0.053	2.79	5.02	109	249	0:01:27	1209	91	299	22.65
	Worldcup86	49	762	0.020	25.3	1.12	72	559	0:05:43	1131	52	611	0:05:39
	Worldcup02	45	790	0.009	0:01:33	6.66	63	$501^{+}1$	1:24:56	1051	51	541	1:39:45
	SM96-3L	13	388	0.004	0:01:26	0.18	16	$108^{+}4$	2:30:57	246	13	116	3:20:54
	SM96-full	162	1491	0.006	0:53:29	3:03:05	222	$1212^{+}13$	8:47:37	3010	163	1380	7:15:34

The columns z_*, v_*, and b_* give the optimal solutions (or final bounds) of MLCM, MLVO, and MLCVO, respectively.
The columns z, v give the crossing number and non-verticality of the found solution.
The column d_C gives the density of MLCM's cost function.
The column "ILP" gives the time required by the ILP if successful, or the final bounds if it ran out of memory after 51, 16, 18, and 52h, respectively.
Due to its complexity, we only computed 50 function evaluations of $f(\lambda, \mu)$ for soccer ball.

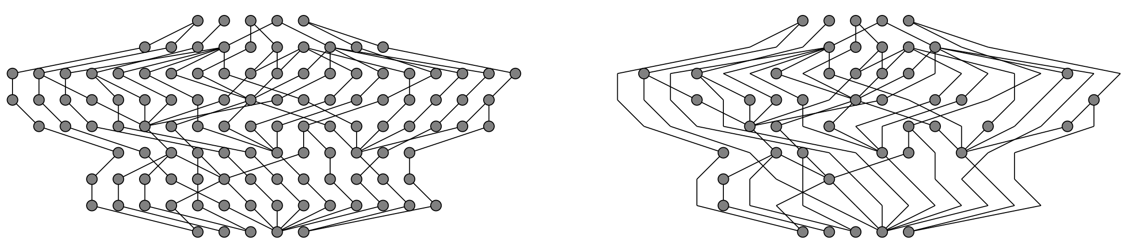
Table 4: Comparing MLVO with MLCM, and combining them to obtain MLCVO.



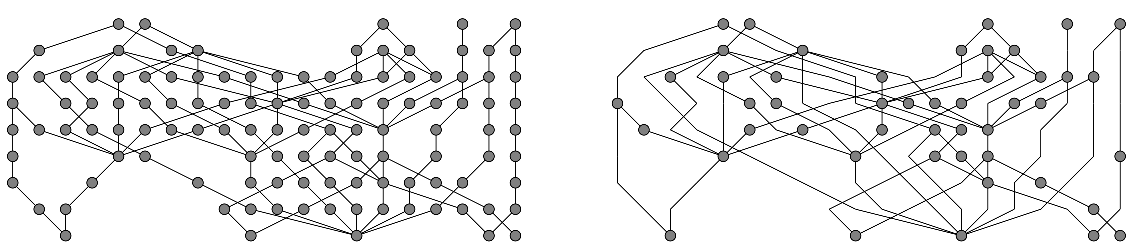
(a) MLCM, with and without explicitly drawn LEDs.



(b) Proper MLVO, narrow alignment scheme, with and without explicitly drawn LEDs.

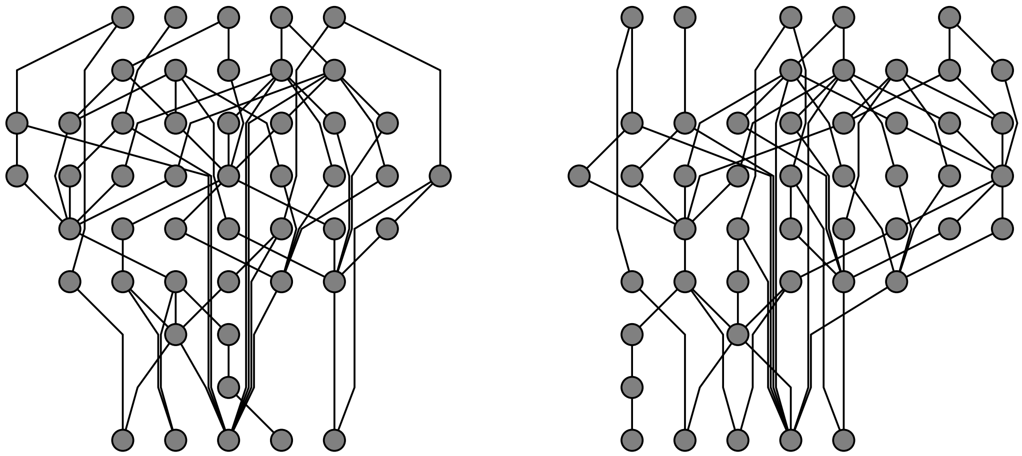


(c) Combined MLCVO, with and without explicitly drawn LEDs.



(d) Proper MLVO, wide alignment scheme, with and without explicitly drawn LEDs.

Figure 2: Example of different (near-)optimal solutions for different problem paradigms on proper level graphs. Instance: *world*.



(a) Non-proper MLVO, narrow alignment scheme

(b) Non-proper MLVO, wide alignment scheme

Figure 3: Example of (near-)optimal solutions for Non-Proper MLVO. Instance: *world*.

This question obviously leads to a weighted (possible Non-proper) MLVO problem (with wide alignement scheme) where the categories are levels, and the kinds are nodes. Note that weights can be directly added to our ILP/SDP approaches. The quadratic cost structure reflects the preference to accept several small delays rather than some big ones. — While this example seems far fetched, or course, it can be seen as a naïve interpretation of the following problem in logistics:

Consider a worker at a storehouse, who has to pack items onto pallets. Each pallet is a separate *purchase* (we omit the term "order" to avoid confusion) of multiple, prespecified items. Within the storehouse, items are categorized by coarse type (e.g., heavy, small, electronics, etc.) and stored at different locations, according to this type. Now, we have a conveyor belt (or forklift) for each such storage location serving the worker items of the corresponding type. Whenever an item arrives at the worker, he packs it onto the corresponding pallet. Our goal is that each purchase is packed within a small timeframe, and hence the worker does not have to deal with many started-but-incomplete purchases/pallets simultaneously. By modeling the items as nodes on levels corresponding to their respective item type, we again obtain an MLVO problem.

Finally we examine team-building, a problem in business studies. Consider a company that wants to build interdisciplinary teams, taking the team members' preferences into account. The different disciplines involved (like cost accounting, financing, or taxation) are the levels, the employees are the nodes, and weighted edges represent the preferences of employees for collaborations. This gives a weighted, Non-proper MLVO problem with wide alignment scheme. The optimal solution of the multiple ranking can guide the chief executives in their final team-building decisions. For example, employees in the center generally have higher esteem and could be chosen as team leaders; employees far away from each other should not be in the same team. The quadratic cost structure reflects the common notion of fairness, i.e., we prefer to violate multiple preferences slightly, than to violate some very strongly.

7. Conclusion

In this paper, we presented the naturally quadratic MLVO problem and showed the effectiveness of an SDP-based approach over ILP and QP techniques. In particular, we also introduced two non-trivial classes of strengthening inequalities for the ILPs and QPs and showed that the SDP satisfies them automatically. Our problem combines multiple QOPs in a tighter manner than the traditional chain of bilevel problems (as is most prominently known from MLCM). Furthermore, it is not sufficient to only consider the relative order of the elements on the levels, but we also have to consider the absolute position of the ordered elements to compare them among multiple levels. This results in much denser cost matrices with different properties than the matrices studied before. Additionally, by using our non-proper drawing scheme, we can vastly reduce the dimension of the constraints matrices and consequently can obtain (near-)optimal, (well-)readable drawings of graphs much too large for exact Proper MLVO and MLCM approaches.

The paper's aim was to discuss optimization strategies and compare them to the better understood MLCM paradigm. We did not aim at a qualitative comparison between the resulting drawings generated by MLVO or MLCM, respectively, as this would be beyond the scope of this paper. Only now, after having promising solving strategies at our disposal, we can start such investigations and discover methods to combine both optimization goals in the best manner.

Finally, we showed that MLVO is not only interesting from the graph drawing perspective, but is also at the core of other, seemingly very different, problems in scheduling, logistics and business studies.

Acknowledgments.

We would like to thank Petra Mutzel and Michael Jünger for introducing us to each other, to graph drawing and crossing numbers, and in particular for useful discussions on the topic of multi-level graph drawing and exact approaches thereto.

References

[1] A. Ahmed, X. Fu, S.-H. Hong, Q. H. Nguyen, and K. Xu. Visual analysis of history of world cup: A dynamic network with dynamic hierarchy and geographic clustering. In *Proceedings of the Visual Information Communications International [VINCI'2009]*, pages 25–39. Springer, 2010.

[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] M. F. Anjos and F. Liers. Global approaches for facility layout and VLSI floorplanning. *Submitted; preprint available at http://www.zaik.uni-koeln.de/ paper/unzip.html?file=zaik2010-604.pdf*, 2010.

[4] 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.

[5] C. Buchheim, F. Liers, and M. Oswald. Speeding up IP-based algorithms for constrained quadratic 01 optimization. *Mathematical Programming*, 124:513–535, 2010.

[6] C. Buchheim, A. Wiegele, and L. Zheng. Exact algorithms for the quadratic linear ordering problem. *INFORMS Journal On Computing*, pages 168–177, 2009.

[7] 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.

[8] 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.

[9] 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.

[10] T. Christof. *Low-dimensional 0/1-polytopes and branch-and-cut in combinatorial optimization.* PhD thesis, Ruprecht-Karls-Universität Heidelberg, Aachen, 1997.

[11] T. Christof, M. Oswald, and G. Reinelt. Consecutive ones and a betweenness problem in computational biology. In *Proceedings of the Conference on Integer Programming and Combinatorial Optimization [IPCO'98]*, volume 1412 of *LNCS*, pages 213–228. Springer, 1998.

[12] E. D. Demaine, S. P. Fekete, G. Rote, N. Schweer, D. Schymura, and M. Zelke. Integer point sets minimizing average pairwise l1 distance: What is the optimal shape of a town? *Computational Geometry*, 44(2):82 – 94, 2011. Special issue of selected papers from the 21st Annual Canadian Conference on Computational Geometry.

[13] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle method for semidefinite cutting plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105:451–469, 2006.

[14] G. Gange, P. J. Stuckey, and K. Marriott. Optimal k-level planarization and crossing minimization. In *Proceedings of the Symposium on Graph Drawing [GD'10]*, LNCS. Springer, 2010. to appear.

[15] 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.

[16] Graphviz gallery, `http://www.graphviz.org/Gallery.php`, November 2010.

[17] 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]*, volume 1731 of *LNCS*, pages 205–216. Springer, 1999.

[18] P. Healy and A. Kuusik. The vertex-exchange graph and its use in multi-level graph layout. Technical Report UL-CSIS-99-1, Department of Computer Science and Information Systems, University of Limerick, Ireland, 1999.

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

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

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

[22] M. Jünger, E. K. Lee, P. Mutzel, and T. Odenthal. A polyhedral approach to the multi-layer crossing minimization problem. In *Proceedings of the Symposium on Graph Drawing [GD'97]*, volume 1353 of *LNCS*, pages 13–24. Springer, 1997.

[23] 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.

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

[25] M. May and K. Szkatula. On the bipartite crossing number. *Control and Cybernetics*, 72:85–97, 1988.

[26] P. Mutzel and R. Weiskircher. Two-layer planarization in graph drawing. In *Proceedings of the International Symposium on Algorithms and Computation [ISAAC'98]*, volume 1533 of *LNCS*, pages 69–78. Springer, 1998.

[27] L. Nachmanson, S. Pupyrev, and M. Kaufmann. Improving layered graph layouts with edge bundling. In *Proceedings of the Symposium on Graph Drawing [GD'2010]*, volume 6502 of *LNCS*. Springer, 2010.

[28] H. Purchase. Which aesthetic has the greatest effect on human understanding? In *Proceedings of the Symposium on Graph Drawing [GD'97]*, volume 1353 of *LNCS*, pages 248–259. Springer, 1997.

[29] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 212:307–335, 2010.

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

[31] H. Seitz. *Contributions to the minimum linear arrangement problem*. PhD thesis, University of Heidelberg, Germany, 2010.

[32] 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.

[33] F. Shieh and C. McCreary. Directed graphs drawing by clan-based decomposition. In *Proceedings of the Symposium on Graph Drawing [GD'95]*, volume 1027 of *LNCS*, pages 472–482. Springer, 1996.

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

[35] 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.