Exact Approaches to Multi-level Vertical Orderings

Markus Chimani[1]* and Philipp Hungerländer[2]

[1] Computer Science, FSU Jena, Germany, markus.chimani@uni-jena.de
[2] Mathematics, Uni Klagenfurt, Austria, philipp.hungerlaender@aau.at

Technical Report TR-FSUJ-CS-AE-11-02, v.2
Algorithm Engineering Group, Computer Science, FSU Jena
December 2011

Abstract. 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 is a quadratic ordering problem and conceptually related to the well-studied problem of Multi-Level Crossing Minimization (MLCM). In contrast to the latter, it can be formulated such that it does not merely consist of multiple sequentially linked bilevel quadratic ordering problems, but as a genuine multi-level problem with dense cost matrix. This allows us to describe the graphs' structures more compactly and therefore obtain solutions for graphs too large for MLCM in practice.
In this paper we give a motivation and mathematical models for MLVO. We formulate linear and quadratic programs, including some strengthening constraint classes, and an SDP relaxation for MLVO. We compare all these approaches both theoretically and experimentally and show that MLVO's properties render linear and quadratic programming approaches inapplicable, even for small sparse graphs, while the SDP works surprisingly well in practice. This is in stark contrast to other ordering problems like MLCM, where such graphs are typically solved more efficiently with integer linear programs. Finally, we also compare our approach to related MLCM approaches.

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 on 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 probably the 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 comment on additional applications in the conclusions of this paper.

In order to motivate MLVO and explain its connection to MLCM we have to consider Sugiyama's framework [33], one of the most common graph drawing paradigms. It is based on the following idea: The aim is to draw a given directed (acyclic) graph "nicely" such that all edges are directed upward. First, we place the nodes of the graph on different *levels* (effectively fixing their vertical positions, i.e., y-coordinates). Edges spanning multiple levels are subdivided into chains of edges such that each edge only spans one level. The second step is to fix orderings of the nodes on their levels such that a certain criterion is optimized. Usually, this means minimizing the number of crossings, which is the aforementioned MLCM problem. As a third step, the nodes are assigned x-coordinates (horizontal positions), consistent with the orderings, such that the number of bends is minimized or the edges' *verticality* is maximized. Edges are thereby always drawn

* Markus Chimani was funded by a Carl-Zeiss-Foundation juniorprofessorship.

as straight lines. An inherent drawback of this scheme is the strict separation of these three stages; an unfortunate optimum solution to step 2 may rule out (visually) good solutions for step 3.

Hence, in this paper we discuss a somehow inverse approach. We want to find a solution for the framework's second step by focusing on the third step's optimization goal: The *Multi-Level Vertical Ordering (MLVO)* problem is to find orderings of the nodes on their levels such that the edges are drawn *as vertical as possible* (see a precise definition below). 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.

MLVO is of course closely related to MLCM, which has received a lot of attention not only within the graph drawing community, but in combinatorial optimization in general. As such, it is worth reviewing the main algorithmic results for the latter. Apart from many heuristic approaches, multiple exact algorithms have been proposed: Jünger and Mutzel [23] presented an integer linear program (ILP) for the two-level (bilevel) 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 [16, 18], and favorably compared to an alternative exact approach based on solving SAT instances by Gange et al. [13]. Buchheim et al. [6] showed that an approach based on semidefinite programs (SDPs) is beneficial over the ILP formulation for (dense) bilevel 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 are quite diverse ILP approaches for the different ordering problems, there exists a universal SDP approach designed to deal with general QOPs [21].

Organization. In the following, we will clarify our main optimization goal (Section 2) and describe a basic mathematical model with linear constraints and a quadratic objective function (Section 3). As our experiments reveal[1], such a model, even after adding some new provably strengthening inequalities, is not directly applicable even to small scale instances. We therefore show how to instead develop a related semidefinite program in Section 4. While the former 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 practically very tight bounds. We demonstrate these effects on well-known graph drawing instances in Section 5.

In this paper we concentrate on exact approaches to MLVO, their polyhedral properties, and their general applicability. The graph drawing aspect constitutes the most direct application field and simplifies speaking about the, in general wider, MLVO problem class. 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 and Variants of MLVO

In the remainder of this paper, we will always consider the following input: Let $G = (V, E)$ with $V = \biguplus_{i=1}^{p} V_i$ (where $\biguplus$ denotes a union of disjunctive sets) 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 (and hence the node's vertical coordinate). Furthermore, let $G' = (V', E')$ with $V' = \biguplus_{i=1}^{p} V'_i$, $E' = \dot{\biguplus}_{i=1}^{p-1} E'_i$, 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.

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

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 positions (i.e., x-coordinates) of its end nodes. Then, $\mathfrak{d}(E) := \sum_{e \in E} \mathfrak{d}(e)$ denotes the overall non-verticality of a solution with edges E. 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 integer coordinates, thereby relating verticality to the drawing's width. Clearly, we only consider adjacent integers 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 such 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.

Unlike for MLCM, we cannot evaluate the objective function purely by locally considering relative node orderings. We 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 and therefore a denser SDP cost matrix. Generally, the more SDP structure is needed, the better the SDP performs compared to competing ILP approaches (in the context of ordering problems), as we shall see in Section 5.

Proper vs. Non-proper. Multi-level crossing minimization and multi-level planarization [13, 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: Considering a non-proper level graph G, we only ask for an ordering of the original nodes. When drawing G, each edge is routed closely 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.

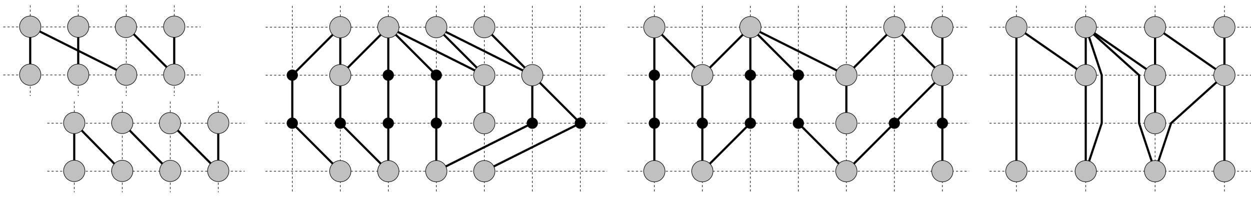
Non-proper MLVO can also be motivated from the graph drawing point of view, as it does away with an intrinsic drawback of considering "properized" graphs instead of the original non-proper ones: LEDs will never be drawn in the resulting drawing and hence it is unreasonable for them to require as much horizontal space as a real node. In fact, to improve overall readability of large graphs, 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) by merging LEDs after Sugiyama's second step (see, e.g., [27]). The non-proper drawing style can hence even be seen as automatic edge bundling.

Non-proper MLVO 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 solutions for graphs that are too large for Proper MLVO or MLCM.

As proper graphs are a special case of non-proper graphs, we will focus on the latter in the following. All tools developed in the following are equivalently suitable for proper graphs and Proper MLVO.

Alignment schemes. 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 scheme, 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 $\{\delta_i, \dots, \delta_i + |V_i| - 1\}$, 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



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

Fig. 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.

addition is the only necessary modification to obtain this alignment scheme[2], we will in the following continue to consider any (non-proper) level graph G, which may or may not be augmented with PDs.

3 Basic Mathematical Models

Consider the level graph G. Similar to the other approaches for multi-level optimization, 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).

$$x_{uv} \in \{0,1\}, \qquad \forall u,v \in V_i,\ 1 \le i \le p,\ u \dot{<} v. \tag{1}$$

Such a variable 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 [34, 35] that feasible orderings can be described via *3-cycle inequalities*

$$0 \le x_{uv} + x_{vw} - x_{uw} \le 1, \qquad \forall u,v,w \in V_i,\ 1 \le i \le p,\ u \dot{<} v \dot{<} w. \tag{2}$$

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

$$d_{(u,v)} \ge X(u) - X(v)\ , \quad d_{(u,v)} \ge X(v) - X(u), \qquad \forall (u,v) \in E. \tag{3}$$

This allows us to give a mathematical model with linear constraints but quadratic objective function to solve the (Non-proper) MLVO problem:

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

Theorem 1 *Every optimal solution to* (DM) *induces an optimal solution to the MLVO problem on (possibly non-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).

Since the objective function is a sum of squares we can write $\mathfrak{d}(E) = \sum_{e \in E} (d_e)^2 = \sum_{(u,v) \in E'} (X(u) - X(v))^2$, without the explicit need for any d-variables. Collecting all variables x_{uv} into a vector x further allows us to observe that there exists a suitably chosen matrix D such that

$$\mathfrak{d}(E) = \begin{pmatrix} 1 \\ x \end{pmatrix}^{\top} D \begin{pmatrix} 1 \\ x \end{pmatrix}. \tag{4}$$

Therefore, in contrast to (DM) which requires distance variables, we obtain a formulation using only ordering variables:

[2] We can trivially force some predefined relative order of all PDs on 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.

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.*

3.1 Applicability and Linearization

Current mathematical programming software (e.g., CPLEX) can often already deal with models with linear constraints and a quadratic objective function directly. Yet, one naturally may try to linearize the models. In fact, the ILPs for the MLCM problem can be seen as linearized models from the originally quadratic problem, and they are known to outperform SDP approaches for sparse graphs with density $\leq 10\%$ [9]. In this case we observe that most products of two binary variables have coefficients 0 in the objective function and can be omitted; only few products 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 $d_{e,i}$ and requiring

$$d_{e,i} \geq 0, \qquad d_{e,i} \geq d_e - i, \tag{5}$$

for all $1 \leq i < \omega - 1$. Using the smallest feasible numbers for these variables, we have

$$(d_e)^2 = d_e + 2\sum_{1\leq i<\omega-1} d_{e,i}. \tag{6}$$

This can be shown by induction over d_e: The equation holds for $d_e = 0$. When we know the value of some $(d_e)^2$, we can compute the succeeding square $(d_e+1)^2 = (d_e)^2 + 2d_e + 1$ by simply adding $2d_e + 1$ to it. Observe that when d_e increases by one, then so do the values of $d_{e,i}$ for all $1 \leq i \leq d_e$. Hence the right hand side of (6) increases by exactly $1 + 2d_e$, which establishes its correctness.

The objective can hence be written as a linear function and we obtain the ILP

$$\mathfrak{d}^*(E) = \min\left\{\sum_{e\in E}\left(d_e + 2\sum_{1\leq i<\omega-1} d_{e,i}\right), \text{ subject to (1)–(3) and (5)}\right\} \tag{DML}$$

as (DM)'s linearization.

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 already for proper level graphs. 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. For non-proper graphs, the situation turns out to be even worse: the cost matrix D becomes completely dense. The clearly resulting drawback is also supported by the results in [5, Table 2].

3.2 Strengthening Constraints

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 any node $u \in V_i$ and all its adjacent nodes $N \subseteq V_j$ on some level j. Let $\alpha := |N| \geq 2$. We are interested in lower bounds on the non-verticalities of the edges $F := u \times N$. We achieve the smallest non-verticalities over F by placing the nodes N directly next to each other, and centering u below or above them. 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. We can hence introduce the constraints

$$\sum\nolimits_{v\in N} d_{(u,v)} \geq \lfloor \alpha/2 \rfloor \cdot \lceil \alpha/2 \rceil. \tag{7}$$

We can strengthen these linear constraints by allowing quadratic terms: Instead of the horizontal distances, we sum up the arising non-verticalities (i.e., squares of the horizontal distances) as

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

Using $\sum_{i=0}^{n} i^2 = n(n+1)(2n+1)/6$ and a case distinction on whether α is odd or even gives:

$$\sum\nolimits_{v\in N} (d_{(u,v)})^2 \geq \begin{cases} \alpha(\alpha^2-1)/12, & \text{if } \alpha \text{ odd,} \\ \alpha(\alpha^2+2)/12, & \text{if } \alpha \text{ even.} \end{cases} \tag{8}$$

Using the extended d-variables (let $d_{e,0} := d_e$ for notational simplicity), we can linearize (8) as

$$\sum\nolimits_{v\in N} d_{(u,v),i} \geq \lfloor \alpha/2 - i \rfloor \cdot \lceil \alpha/2 - i \rceil, \qquad \forall 0 \leq i < \lfloor \alpha/2 \rfloor. \tag{9}$$

Complete-bipartite constraints. We can generalize the former constraint class by considering complete bipartite subgraphs on two levels. Let $N_i \subseteq V_i$ and $N_j \subseteq V_j$, for some $1 \leq i, j < p$, be two node sets such that $F := N_i \times N_j \subseteq E$. Let $\beta := |N_i|$, $\gamma := |N_j|$ and $\beta \leq \gamma$. We obtain the minimum possible overall horizontal distances for F by tightly packing the node sets on their levels, and centering them over each other.

Placing each node $v \in N_i$ beneath (or above) the center of compactly positioned nodes N_j, 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 position (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 parities of β and γ we obtain

$$\sum\nolimits_{u\in N_i} \sum\nolimits_{v\in N_j} d_{(u,v)} \geq \beta \cdot \lfloor \gamma/2 \rfloor \cdot \lceil \gamma/2 \rceil + \begin{cases} \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil, & \text{if } \gamma \text{ odd,} \\ \lfloor \beta/2 \rfloor \cdot (\lceil \beta/2 \rceil - 1), & \text{if } \gamma \text{ even.} \end{cases} \tag{10}$$

For example, 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 $2 \cdot \frac{1}{2} \cdot \lfloor \beta/2 \rfloor \cdot (\lfloor \beta/2 \rfloor + 1) = \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil$.

Again, we can strengthen (10) considering squared d-variables and accordingly increased right-hand sides. 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, we obtain the quadratic complete-bipartite constraints:

$$\sum_{u\in N_i, v\in N_j} (d_{(u,v)})^2 \geq \begin{cases} \beta\gamma(\gamma^2 + \beta^2 - 2)/12 & \text{if } \beta \text{ and } \gamma \text{ have the same parity,} \\ \beta\gamma(\gamma^2 + \beta^2 + 1)/12 & \text{otherwise.} \end{cases} \tag{11}$$

Again we can linearize the quadratic constraints by requiring for the extended variables $d_{e,i}$ that their sum is greater than or equal to the sum of distances reduced by i. Simplifying the resulting expression by using (10) gives for all $0 \leq i < \lfloor \gamma/2 \rfloor$:

$$\sum_{u\in N_i, v\in N_j} d_{(u,v),i} \geq \beta \cdot \lfloor \gamma/2 - i \rfloor \cdot \lceil \gamma/2 - i \rceil + \begin{cases} \lfloor \beta/2 \rfloor \cdot \lceil \beta/2 \rceil, & \text{if } \gamma \text{ odd,} \\ \lfloor \beta/2 \rfloor \cdot (\lceil \beta/2 \rceil - 1), & \text{if } \gamma \text{ even.} \end{cases} \tag{12}$$

Lemma 3 *Degree constraints (even in their weaker form* (7)*) strengthen* (cDM)*. Complete-bipartite constraints (even in their weaker form* (10)*) further strengthen the relaxation, even if it already satisfies all (quadratic) degree constraints* (8)*.*

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 constraints force 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 (quadratic) degree constraints. Again all x-variables can be 0.5, despite the fact that the degree constraints force d-variables to be non-zero: the constraints (3) will simply not be tight. 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.

4 Semidefinite Relaxation

In this section we concentrate on bound computations for (Non-proper) MLVO by analyzing matrix liftings of ordering problems. Recall that Proper MLVO can be seen as a special case, and our results are directly applicable to it as well. The linear ordering core of the following SDP formulation has been discussed in the context of MLCM [9]. We recapitulate it here briefly, to be able to discuss MLVO specific considerations.

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

$$-1 \leq y_{uv} + y_{vw} - y_{uw} \leq 1, \qquad \forall u, v, w \in V_i, 1 \leq i \leq p,\ u \dot{<} v \dot{<} w. \tag{13}$$

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 $Y = yy^T$. We are interested in *Multi-Level Quadratic Orderings (MQOs)* and therefore consider the polytope

$$\mathcal{P}_{MQO} := \operatorname{conv}\left\{ \begin{pmatrix} 1 \\ y \end{pmatrix} \begin{pmatrix} 1 \\ y \end{pmatrix}^{\top} \ :\ y \in \{-1, 1\},\ y \text{ satisfies (13)} \right\}.$$

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 $\operatorname{diag}(Y) = \mathfrak{e}$, the vector of all ones. To simplify our notation, we introduce

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

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

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

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 (13) gives $|y_{uv} + y_{vw} - y_{wu}| = 1$. By squaring both sides and using $y_{uv}^2 = 1$ we obtain

$$y_{uv}y_{vw} - y_{uv}y_{uw} - y_{uw}y_{vw} = -1, \qquad \forall u, v, w \in V_i,\ 1 \leq i \leq p,\ u \dot{<} v \dot{<} w. \tag{15}$$

Applying the dimension result for QOP from [6] to MQO, it is easy to deduce that (15) describes the smallest linear subspace containing $\mathcal{P}_{MQO}$.

In remains to discuss the construction of a cost matrix C. First we observe that we can write $\mathfrak{d}(E)$ as a function in y:

$$\begin{aligned}
\mathfrak{d}(E) &= \sum_{(u,v)\in E} (X(u) - X(v))^2 \\
&= \sum_{(u,v)\in E} \left(\delta_{\ell(u)} + \sum_{w\in V_{\ell(u)}\setminus\{u\}} x_{wu} - \delta_{\ell(v)} - \sum_{w\in V_{\ell(v)}\setminus\{v\}} x_{wv} \right)^2 \\
&= \sum_{(u,v)\in E} \left(\delta_{\ell(u)} + \sum_{w\in V_{\ell(u)}\setminus\{u\}} (y_{wu}+1)/2 - \delta_{\ell(v)} - \sum_{w\in V_{\ell(v)}\setminus\{v\}} (y_{wv}+1)/2 \right)^2
\end{aligned} \tag{16}$$

Expanding the result of (16) and using $y_{uv}^2 = 1$ clearly gives a function in y which is a sum of terms; each term is a product of at most two y-variables (and hence an entry in Z) and some coefficient. We can use these coefficients to build a symmetric cost matrix C of order ζ such that $\langle C, Z\rangle = \mathfrak{d}(E)$ for any given feasible ordering y.

Next we show that our basic model $(\mathtt{SDP}_b)$ with integrality constraints on the first row and column of Z is exact.

Theorem 4 *(Non-proper) MLVO is equivalent to the problem* $\min\{ \langle C, Z\rangle \ : \ Z \in \mathcal{I}_{MQO} \}$*, where*

$$\mathcal{I}_{MQO} := \{ \ Z \ : \ Z \text{ partitioned as in (14) and satisfies (15)}, \ Z \in \mathcal{E}, \ y \in \{-1, 1\} \ \}.$$

Proof. We summarize the above discussion. Since $y_{uv}^2 = 1$ we have $\mathrm{diag}(Y - yy^T) = 0$, which together with $Y - yy^T \succcurlyeq 0$ shows that in fact $Y = yy^T$. The 3-cycle equations (15) ensure that $|y_{uv} + y_{vw} - y_{uw}| = 1$ holds. Therefore any matrix $Z \in \mathcal{I}_{MQO}$ 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 (Non-proper) MLVO

$$\min \{ \ \langle C, Z\rangle \ : \ Z \text{ partitioned as in (14) and satisfies (15)}, \ Z \in \mathcal{E} \ \}. \tag{$\mathtt{SDP}_b$}$$

Theorem 5 $(\mathtt{SDP}_b)$ *is at least as strong as* $(\mathtt{cDM})$ *together with the quadratic degree constraints* (8) *and the quadratic complete-bipartite constraints* (11).

Proof. First, it is not hard to verify that any Z feasible for $(\mathtt{SDP}_b)$ contains a vector y in its first column that satisfies the 3-cycle inequalities (13) 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. Since the degree constraints are a special case of complete-bipartite constraints, it suffices to consider the latter constraints (11). 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 $N_i \times N_j \subseteq E$, $\beta := |N_i|$, $\gamma := |N_j|$, and $\beta \le \gamma$. In order to prove that the constraints hold, consider the left hand side $\sum_{u\in N_i, v\in N_j} (d_{(u,v)})^2$ of (11).

Assume $\gamma - \beta$ even. Using (16) yields (after some standard algebraic transformations):

$$\begin{aligned}
\sum_{\substack{u\in N_i,\\ v\in N_j}} (d_{(u,v)})^2 = \ & \frac{1}{4} \sum_{u\in N_i, v\in N_j} (\beta + \gamma - 2) \\
& + \frac{1}{2} \sum_{v\in N_j} \left(\sum_{u\dot{<}t\dot{<}w\in N_i} y_{ut}y_{uw} - \sum_{t\dot{<}u\dot{<}w\in N_i} y_{ut}y_{uw} + \sum_{t\dot{<}w\dot{<}u\in N_i} y_{ut}y_{uw} \right) \\
& + \frac{1}{2} \sum_{u\in N_i} \left(\sum_{v\dot{<}t\dot{<}w\in N_j} y_{vt}y_{vw} - \sum_{t\dot{<}v\dot{<}w\in N_j} y_{vt}y_{vw} + \sum_{t\dot{<}w\dot{<}v\in N_j} y_{vt}y_{vw} \right).
\end{aligned}$$

Summing up the equations (15) for all elements in N_i and N_j

$$\begin{aligned}\sum_{u\dot{<}v\dot{<}w\in N_i}(-y_{uv}y_{vw}+y_{uv}y_{uw}+y_{uw}y_{vw}) &= \binom{\beta}{3} = \frac{\beta(\beta-1)(\beta-2)}{6},\\ \sum_{u\dot{<}v\dot{<}w\in N_j}(-y_{uv}y_{vw}+y_{uv}y_{uw}+y_{uw}y_{vw}) &= \binom{\gamma}{3} = \frac{\gamma(\gamma-1)(\gamma-2)}{6},\end{aligned} \tag{17}$$

and applying it to the above gives

$$\begin{aligned}\sum_{\substack{u\in N_i,\\ v\in N_j}}(d_{(u,v)})^2 &= \frac{\beta\gamma(\beta+\gamma-2)}{4}+\frac{\beta\gamma(\beta-1)(\beta-2)}{12}+\frac{\beta\gamma(\gamma-1)(\gamma-2)}{12}\\ &= \frac{\beta\gamma(\gamma^2+\beta^2-2)}{12}.\end{aligned}$$

Now, assume $\gamma-\beta$ odd. Using (16) yields (after some standard algebraic transformations):

$$\begin{aligned}\sum_{u\in N_i,v\in N_j}(d_{(u,v)})^2 = \ & \frac{1}{4}\sum_{u\in N_i,v\in N_j}(\beta+\gamma-1)\\ &+\frac{1}{2}\sum_{v\in N_j}\left(\sum_{u\dot{<}t\dot{<}w\in N_i}y_{ut}y_{uw}-\sum_{t\dot{<}u\dot{<}w\in N_i}y_{ut}y_{uw}+\sum_{t\dot{<}w\dot{<}u\in N_i}y_{ut}y_{uw}\right)\\ &+\frac{1}{2}\sum_{u\in N_i}\left(\sum_{v\dot{<}t\dot{<}w\in N_j}y_{vt}y_{vw}-\sum_{t\dot{<}v\dot{<}w\in N_j}y_{vt}y_{vw}+\sum_{t\dot{<}w\dot{<}v\in N_j}y_{vt}y_{vw}\right).\end{aligned}$$

Finally applying (17) to the above gives

$$\begin{aligned}\sum_{u\in N_i,v\in N_j}(d_{(u,v)})^2 &= \frac{\beta\gamma(\beta+\gamma-1)}{4}+\frac{\beta\gamma(\beta-1)(\beta-2)}{12}+\frac{\beta\gamma(\gamma-1)(\gamma-2)}{12}\\ &= \frac{\beta\gamma(\gamma^2+\beta^2+1)}{12}.\end{aligned}$$

In both parity cases of $\gamma-\beta$ we obtain the corresponding right hand side of (11), and hence the theorem holds.

Further tightenings. There are some obvious ways to tighten ($\mathtt{SDP}_b$), analogous to [9]. Since any (integer) feasible matrix $Z \in \mathcal{I}_{MQO}$ is an outer product of a vector with $\{-1,1\}$ entries, it satisfies the triangle inequalities defining the metric polytope

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

The relaxation ($\mathtt{SDP}_b$) 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 (13) (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

$$\mathcal{LS} := \{\ Z : Z \text{ satisfies all LS-cuts}\ \}.^3$$

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

Overall, we get the following tractable relaxation

$$\min \{ \langle C, Z \rangle \ : \ Z \text{ partitioned as in (14) and satisfies (15)}, \ Z \in (\mathcal{E} \cap \mathcal{M} \cap \mathcal{LS}) \}. \qquad (\texttt{SDP}_s)$$

In summary, ($\texttt{SDP}_b$) ensures all constraints from Section 3. We also included $\mathcal{M}$ and $\mathcal{LS}$ to ensure that ($\texttt{SDP}_s$) contains all facets of $\mathcal{P}^7_{LO}$ and $\mathcal{P}^4_{MQO}$ respectively.[4] 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 would be far too expensive.

4.1 Solving ($\texttt{SDP}_s$)

The core of our SDP approach is to solve our SDP relaxation, using the bundle method in conjunction with interior point methods. The resulting fractional solutions constitute lower bounds for the non-relaxed SDP. By the use of a rounding strategy, we can exploit such fractional solutions to obtain upper bounds, i.e., integer feasible solutions that describe solutions to our MLVO problem at hand. Hence, in the end we have some feasible solution, together with a proof how far this solution could possibly be from the true optimum. We will discuss these two steps in more detail in the following.

Lower bound. Looking at the constraint classes and their sizes in the relaxation ($\texttt{SDP}_s$), 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 [12], 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 through Lagrangian duality. Thus the objective function f becomes non-smooth and the evaluation of f for a given feasible point amounts to solving a problem over $\mathcal{E}$. In our experiments, we use a primal-dual interior-point method that also provides a subgradient of f to conduct the function evaluations. Using these ingredients, we get an approximate minimizer of f using subgradient optimization techniques such as the bundle method [12]. 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.

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[5] 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 ($\texttt{SDP}_s$) 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 [14] to D and obtain a $-1/{+1}$ vector $\overline{y}$ (cf. [29]).
3. Compute the induced non-verticality $\mathfrak{d}(\overline{y})$. If $\mathfrak{d}(\overline{y}) \geq \mathfrak{d}(y')$: 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].

---

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

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

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 $f(\lambda, \mu)$: independent of the problem size ζ, the convergence of the bundle method usually slows down already after fewer evaluations; it is not reasonable to spend more computational effort with little to no gain.

We consider input graphs from three different sources, which are often considered in related experimental investigations, e.g., [9, 13, 16, 17, 22]. Table 2 gives the instances' central properties; the graphs are available at `http://www.ae.uni-jena.de/Research/MLVO.html`:

Polytopes. Often, one considers the graphs modeling the incidence relation between faces (corner, edge, 2D-face,...) of a polytope, and hence we are interested in drawing them. 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 [15] is 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 [32] 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.

SDP vs. other models. We start with evaluating the alternatives to the SDP approach. We already argued why the linearization of `(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 `(DML)` (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 notice that this ILP clearly does *not* really solve the MLVO problem as defined previously. We denote this algorithm by *SD*. 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.

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

Table 1. Comparing *DMQ*, *DML*, and *SD* to the SDP approach on selected instances. Recall that *SD* is the simplified ILP which does not solve for ∂ but considers the non-quadratic sum-of-horizontal-distances (shd)—its objective value is hence not directly comparable to ∂. The column *bb* gives the number of branch-and-bound nodes in millions. The 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. ∂ 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.

For both alternative approaches *DMQ* and *DML*, and even for *SD*, we observe running times that are orders of magnitudes larger than the SDP's, cf. Table 1. 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

	Instance	p	Proper								Non-proper							
			$\|V'\|$	$\|E'\|$	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

Table 2. Instance properties. Cube3 and Cube4 correspond to a 3- and 4-dimensional cube, respectively. ω and ω' denote 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.

always obtained without any branching. In fact, the non-SDP algorithms ran out of memory in all but the smallest instance. Upon termination, they obtained clearly weaker lower and upper bounds than the SDP approach, although 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.

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, supported by our SDP based upper bound heuristic, is tight enough to find and prove optimal solutions for the smaller instances, and gives surprisingly small gaps for the instances of challenging size (on which we concentrate in this paper).

The approach's running time is mainly dependent on ζ. Hence it is not surprising that the visually "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.

		Proper MLVO				Non-proper MLVO			
		narrow		wide		narrow		wide	
	Instance	∂	time	∂	time	∂	time	∂	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.

MLVO vs. MLCM. As described in the 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 paper, 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 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, extracted from [9].[6] 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 optimality

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

	Instance	MLCM					MLVO			MLCVO			
		cr^*	$\mathfrak{d}$	d_C	time	ILP	cr	$\mathfrak{d}^*$	time	b_*	cr	$\mathfrak{d}$	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

Table 4. Comparing MLVO with MLCM, and combining them to obtain MLCVO. The columns cr^*, $\mathfrak{d}^*$, and b^* give the optimal solutions (or final bounds) for MLCM, MLVO, and MLCVO, respectively. The columns cr and $\mathfrak{d}$ 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.

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 $C_{\text{MLCM}}, C_{\text{MLVO}}$ are balanced via coefficients $c_{\text{cr}}, c_{\mathfrak{d}}$, 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 $c_{\text{cr}} = 10$ and $c_{\mathfrak{d}} = 1$ (such that $c_{\text{cr}}/c_{\mathfrak{d}} \approx \sum \mathfrak{d} / \sum cr$, where cr stands for the number of crossings achieved by MLCM). 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. It becomes obvious that considering the wide instead of the narrow alignment scheme is beneficial from the aesthetic point of view. Another line of research could be to only solve the narrow alignment scheme and find a (practically) good strategy to insert PDs as a postprocessing.

6 Conclusion

We presented the, intrinsically 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 take into account the absolute positions 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. By using our non-proper drawing scheme, we can vastly reduce the size of the associated optimization problem and consequently can obtain drawings for graphs much too large for exact Proper MLVO and MLCM approaches.

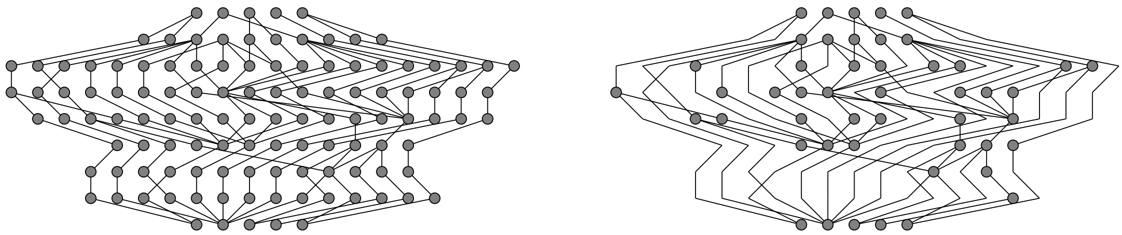
The paper's aim was to discuss optimization strategies and compare them to the better understood MLCM paradigm. We did not aim at a qualitative comparison between the resulting drawings generated by MLVO or MLCM, respectively, as this would be beyond the scope of this paper. Only now, after having promising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

Further applications. 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, e.g., 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.

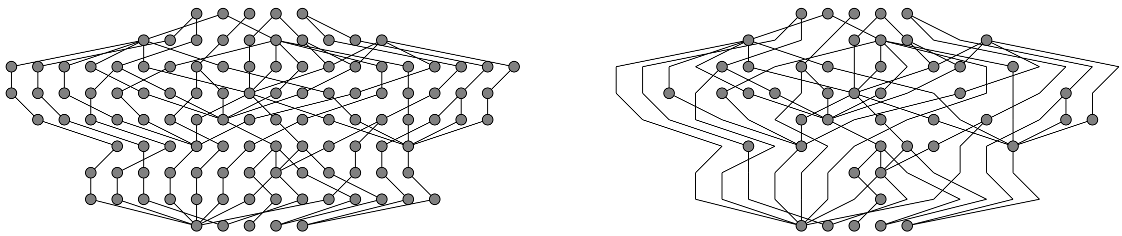
Another application, also giving a (Non-)Proper MLVO instance, can be found in *multiple ranking*, where we have groups of objects, objects have relationships (e.g., similarities) with objects from other groups, and we want to (linearly) rank the objects within their groups such that related objects 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.

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.

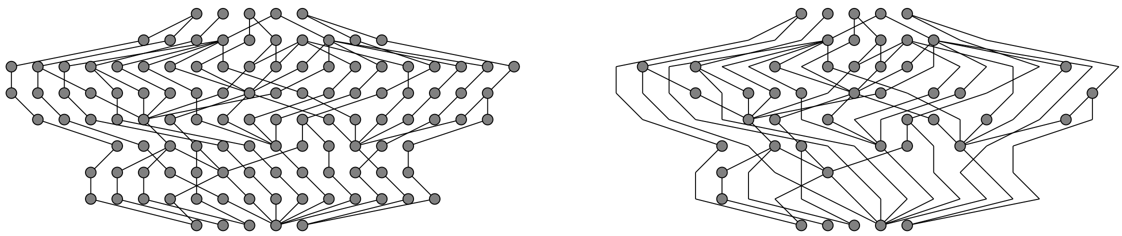
M. Chimani was funded by a Carl-Zeiss-Foundation juniorprofessorship.



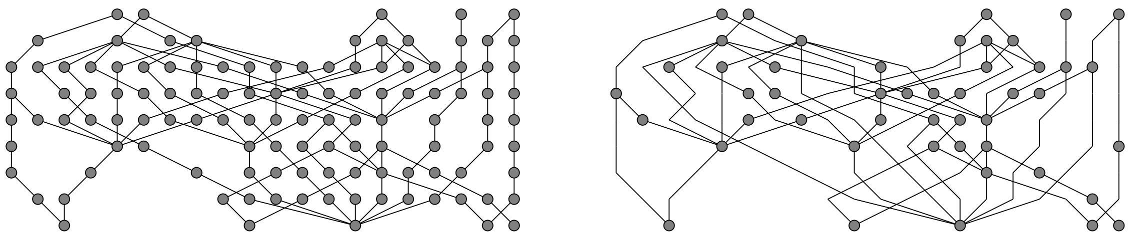
(a) MLCM, with and without explicitly drawn LEDs.



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

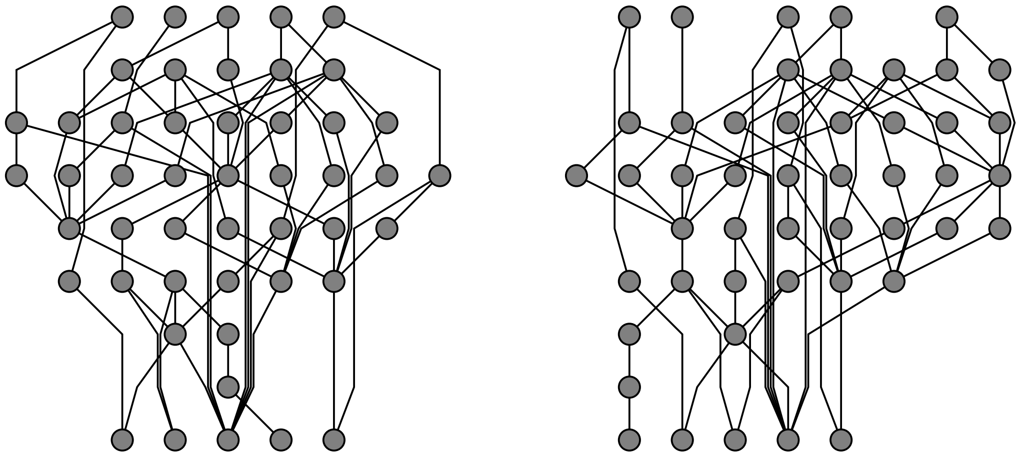


(c) Combined MLCVO, with and without explicitly drawn LEDs.



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

Fig. 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

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

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-Gonzalez. Decorous lower bounds for minimum linear arrangement. *INFORMS Journal on Computing*, 23(1):26–40, 2011.

[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.

[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] 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.

[13] G. Gange, P. Stuckey, and K. Marriott. Optimal k-level planarization and crossing minimization. In U. Brandes and S. Cornelsen, editors, *Graph Drawing*, volume 6502 of *Lecture Notes in Computer Science*, pages 238–249. Springer Berlin / Heidelberg, 2011.

[14] 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.

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

[16] 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.

[17] 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.

[18] P. Healy and A. Kuusik. Algorithms for multi-level graph planarity testing and layout. *Theoretical Computer Science*, 320(2–3):331–344, 2004.

[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, 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] 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.

[33] 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.

[34] A. W. Tucker. On directed graphs and integer programs. Technical report, IBM Mathematical Research Project, 1960.

[35] D. H. Younger. Minimum feedback arc sets for a directed graph. *IEEE Transactions on Circuit Theory*, 10(2):238–245, 1963.