An SDP Approach to Multi-level Crossing Minimization

P. Hungerländer
joint work with
M. Chimani, M. Jünger, P. Mutzel

University of Klagenfurt - Department of Mathematics

15th Combinatorial Optimization Workshop, Aussois,
January 2011
2nd Alpen-Adria Workshop on Optimization

- Will take place in Klagenfurt from 12th to 14th of May 2011
- Is planned as a satellite event to the SIAM conference on Optimization in Darmstadt from 16th to 19th of May, 2011
- There will be a conference dinner and a hiking excursion :)
- In case you plan to attend, please contact us till 28th of February
- Further information is available under http://www.wo2011.uni-klu.ac.at/
Contents

1 Introduction
2 The Standard LP-Approach
3 An SDP-Approach
4 Theoretical Comparison
5 Computational Results
6 Conclusion
Contents

1 Introduction

2 The Standard LP-Approach

3 An SDP-Approach

4 Theoretical Comparison

5 Computational Results

6 Conclusion
Reference

The presentation is based on the paper:
Multi-Level Crossing Minimization (MLCM)

- We have given a proper level graph
  - $p$ horizontal levels/layers with vertex sets $V_r$ for each level $1 \leq r \leq p$
  - edge sets $E_r \subseteq V_r \times V_{r+1}$ for adjacent levels
Multi-Level Crossing Minimization (MLCM)

- We have given a proper level graph
  - $p$ horizontal levels/layers with vertex sets $V_r$ for each level $1 \leq r \leq p$
  - edge sets $E_r \subseteq V_r \times V_{r+1}$ for adjacent levels

- Aim: Reorder the vertices within the levels such that the number of crossings is minimized when the edges are drawn as straight lines
Example Drawing
Literature Review

- Jünger and Mutzel (1997) first solved 2-level MLCM instances
Literature Review

- Jünger and Mutzel (1997) first solved 2-level MLCM instances

- Jünger et al. (1997) suggested the first ILP formulation for general MLCM
Literature Review

- Jünger et al. (1997) suggested the first ILP formulation for general MLCM.
- Healy and Kuusik (1999) extended the ILP formulation using the so-called vertex-exchange graph.
Literature Review

- Jünger and Mutzel (1997) first solved 2-level MLCM instances

- Jünger et al. (1997) suggested the first ILP formulation for general MLCM

- Healy and Kuusik (1999) extended the ILP formulation using the so-called vertex-exchange graph

- Buchheim et al. (2009) suggested the first SDP-based approach for 2-level MLCM using a part the SDP relaxation that we are going to propose
Motivation

- Solving MLCM is the crucial step in the probably most widely used graph drawing scheme, the so-called Sugiyama framework.
Motivation

- Solving MLCM is the crucial step in the probably most widely used graph drawing scheme, the so-called Sugiyama framework

- There are many similarities to other (quadratic) ordering problems
MLCM as a Binary Quadratic Program

We use binary ordering variables $x_{ij}^r$ with

$$x_{ij}^r = \begin{cases} 
1 & \text{if vertex } i \text{ comes before vertex } j \text{ on level } r, \\
0 & \text{otherwise.} 
\end{cases}$$
MLCM as a Binary Quadratic Program

So we ask for

\[ x^r_{ij} \in \{0, 1\}, \quad 1 \leq r \leq p, \quad i < j. \]
MLCM as a Binary Quadratic Program

So we ask for

\[ x_{ij}^r \in \{0, 1\}, \quad 1 \leq r \leq p, \quad i < j. \]  

(1)

Then the following 3-cycle constraints ensure linear orderings on the layers of a given proper level graph:

\[ 0 \leq x_{ij}^r + x_{jk}^r - x_{ik}^r \leq 1, \quad 1 \leq r \leq p, \quad i < j < k \in V_r. \]  

(2)
MLCM as a Binary Quadratic Program

So we ask for

\[ x_{ij}^r \in \{0, 1\}, \quad 1 \leq r \leq p, \quad i < j. \] \hspace{1cm} (1)

Then the following 3-cycle constraints ensure linear orderings on the layers of a given proper level graph:

\[ 0 \leq x_{ij}^r + x_{jk}^r - x_{ik}^r \leq 1, \quad 1 \leq r \leq p, \quad i < j < k \in V_r. \] \hspace{1cm} (2)

Minimizing \( \sum_{1 \leq r < p} \sum_{1 \leq i < j \leq |V_r|} \sum_{(i,k),(j,l) \in E_r} (x_{ij}^r x_{lk}^{r+1} + x_{ji}^r x_{kl}^{r+1}) \)

over (1) and (2) therefore solves MLCM.
MLCM as a Binary Quadratic Program

So we ask for

\[ x_{ij}^r \in \{0, 1\}, \quad 1 \leq r \leq p, \quad i < j. \]

Then the following 3-cycle constraints ensure linear orderings on the layers of a given proper level graph:

\[ 0 \leq x_{ij}^r + x_{jk}^r - x_{ik}^r \leq 1, \quad 1 \leq r \leq p, \quad i < j < k \in V_r. \]

Minimizing \( \sum_{1 \leq r < p} \sum_{1 \leq i < j \leq |V_r|} \sum_{(i,k),(j,l) \in E_r} (x_{ij}^r x_{lk}^{r+1} + x_{ji}^r x_{kl}^{r+1}) \) over (1) and (2) therefore solves MLCM.

The quadratic relaxation is obtained by relaxing the integrality conditions on the variables.
An QP-Relaxation of MLCM

So we ask for

\[(1) \quad 0 \leq x_{ij}^r \leq 1, \quad 1 \leq r < p, \quad i < j.\]

Then the following 3-cycle constraints ensure linear orderings on the layers of a given proper level graph:

\[(2) \quad 0 \leq x_{ij}^r + x_{jk}^r - x_{ik}^r \leq 1, \quad 1 \leq r \leq p, \quad i < j < k \in V_r.\]

Minimizing \[\sum_{1 \leq r < p} \sum_{1 \leq i < j \leq |V_r|} \sum_{(i,k),(j,l) \in E_r} (x_{ij}^r x_{lk}^{r+1} + x_{ji}^r x_{kl}^{r+1})\]
over (1) and (2) therefore gives a lower bound for MLCM.

The quadratic relaxation is obtained by relaxing the integrality conditions on the variables.
Linearization via Crossing Variables

We can linearize the objective function by introducing binary crossing variables

\[ c_{ij,kl}^r \in \{0, 1\}, \quad 1 \leq r < p, \quad (i, k), (j, l) \in E_r \]

that shall be 1 if the edges \((i, k)\) and \((j, l)\) cross and 0 otherwise.
Linearization via Crossing Variables

We can linearize the objective function by introducing binary crossing variables

\[ c_{ij,kl}^r \in \{0, 1\}, \quad 1 \leq r < p, \quad (i, k), (j, l) \in E_r \]

that shall be 1 if the edges \((i, k)\) and \((j, l)\) cross and 0 otherwise.

To bind the crossing variables with the linear ordering variables, we need the following constraints

\[-c_{ij,kl}^r \leq x_{kl}^{r+1} - x_{ij}^r \leq c_{ij,kl}^r, \quad (i, k), (j, l) \in E_r, \quad k < l,\]

\[1 - c_{ij,kl}^r \leq x_{lk}^{r+1} + x_{ij}^r \leq 1 + c_{ij,kl}^r, \quad (i, k), (j, l) \in E_r, \quad k > l.\]
MLCM as a Binary Linear Program

Let $x$ be the vector collecting the ordering variables $x_{ij}^r$ and $c$ be the vector collecting the crossing variables $c_{ij,kl}^r$. Then we can formulate MLCM as a binary linear program as

$$z^* = \min \left\{ \sum_{1 \leq r < p} \sum_{(i,k),(j,l) \in E_r} c_{ij,kl}^r : (x, c) \text{ integral and feasible} \right\}$$
An LP-Relaxation of MLCM

Let $x$ be the vector collecting the ordering variables $x^r_{ij}$ and $c$ be the vector collecting the crossing variables $c^r_{ij,kl}$. Then we can formulate MLCM as a linear program as

$$z_{lp} = \min \left\{ \sum_{1 \leq r < p} \sum_{(i,k),(j,l) \in E_r} c^r_{ij,kl} : (x, c) \text{ feasible} \right\}$$

Replacing the integrality conditions with 0-1 bounds gives the linear relaxation.
Contents

1. Introduction

2. The Standard LP-Approach

3. An SDP-Approach

4. Theoretical Comparison

5. Computational Results

6. Conclusion
Motivation

- The LP-Relaxation is often too weak for efficient pruning in Branch & Bound enumeration
Motivation

- The LP-Relaxation is often too weak for efficient pruning in Branch & Bound enumeration
- Thus it would be desirable to have some tighter approximation available
Motivation

- The LP-Relaxation is often too weak for efficient pruning in Branch & Bound enumeration.
- Thus it would be desirable to have some tighter approximation available.
- For example, relaxations based on semidefinite optimization.
Variable Transformation

For the SDP formulation it is convenient to transform the linear ordering variables $x_{ij}^r$ into variables taking the values $-1$ and $1$:

$$y_{ij}^r = 2x_{ij}^r - 1, \quad 1 \leq r \leq p, \ i, j \in V_r, \ i < j$$
Variable Transformation

For the SDP formulation it is convenient to transform the linear ordering variables $x_{ij}^r$ into variables taking the values $-1$ and $1$:

$$y_{ij}^r = 2x_{ij}^r - 1, \quad 1 \leq r \leq p, \ i, j \in V_r, \ i < j$$

This leads to inequalities equivalent to the 3-cycle constraints

$$-1 \leq y_{ij}^r + y_{jk}^r - y_{ik}^r \leq 1, \quad 1 \leq r \leq p, \ i, j, k \in V_r, \ i < j < k.$$
Matrix lifting

The matrix lifting approach takes a vector $y$ collecting the ordering variables $y_{ij}^r$ and considers the matrix $Y = yy^T$. 
Matrix lifting

The matrix lifting approach takes a vector $y$ collecting the ordering variables $y_{ij}^r$ and considers the matrix $Y = yy^T$.

Our object of interest now is

$$\mathcal{P}_{QC} := \text{conv} \{yy^T : y \in \{-1, 1\}, y \text{ satisfies the 3-cycle constraints}\}.$$
Matrix lifting

The matrix lifting approach takes a vector $y$ collecting the ordering variables $y_{ij}^r$ and considers the matrix $Y = yy^T$.

Our object of interest now is

$$\mathcal{P}_{QC} := \text{conv} \{yy^T : y \in \{-1, 1\}, y \text{ satisfies the 3-cycle constraints}\}.$$  

We relax the nonconvex equation $Y - yy^T = 0$ to the constraint $Y - yy^T \succeq 0$, which is convex due to the Schur-complement lemma.
Matrix lifting

The matrix lifting approach takes a vector \( y \) collecting the ordering variables \( y_{ij} \), and considers the matrix \( Y = yy^T \).

Our object of interest now is

\[
P_{QC} := \text{conv} \{ yy^T : y \in \{-1,1\}, y \text{ satisfies the 3-cycle constraints} \}.\]

We relax the nonconvex equation \( Y - yy^T = 0 \) to the constraint \( Y - yy^T \succeq 0 \), which is convex due to the Schur-complement lemma.

Moreover, the main diagonal entries of \( Y \) correspond to \( y_{ij} y_{ij}^f \), and hence \( \text{diag}(Y) = e \), the vector of all ones.
Further basic notation

To simplify our notation, we introduce

\[ Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \]

where \( \zeta := \dim(Z) = 1 + \sum_{i=1}^{p} (|V_i|) \) and \( Z = (z_{ij}) \).
Further basic notation

To simplify our notation, we introduce

\[ Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \]

where \( \zeta := \dim(Z) = 1 + \sum_{i=1}^{p} (|V_i|) \) and \( Z = (z_{ij}) \).

In this case \( Y - yy^T \succeq 0 \iff Z \succeq 0. \)
Further basic notation

To simplify our notation, we introduce

\[ Z = Z(y, Y) := \begin{pmatrix} 1 & y^T \\ y & Y \end{pmatrix}, \]

where \( \zeta := \dim(Z) = 1 + \sum_{i=1}^{p} (|V_i|^2) \) and \( Z = (z_{ij}) \).

In this case \( Y - yy^T \succeq 0 \iff Z \succeq 0 \).

Hence, the ellipotope \( \mathcal{E} \) contains \( \mathcal{P}_{QC} \)

\[ \mathcal{E} := \{ Z : \text{diag}(Z) = e, Z \succeq 0 \}. \]
The Minimum Linear Subspace

In order to express constraints on $y$ in terms of $Y$, where we denote the product $y_{ij} y_{kl}$ by $y_{ij,kl}$. 

A natural way to do this for the 3-cycle constraints $|y_{ij} + y_{jk} - y_{ik}| = 1$ consists in squaring both sides, leading to $y_{ij,kl} - y_{ij,ik} - y_{ik,kl} = -1$, $1 \leq r \leq p$, $i, j, k \in V_r$, $i < j < k$. 

These equations describe the smallest linear subspace that contains $P_{QC}$. 

The Minimum Linear Subspace

In order to express constraints on \( y \) in terms of \( Y \), where we denote the product \( y_{ij}^r y_{kl}^s \) by \( y_{ij,kl}^{r,s} \).

A natural way to do this for the 3-cycle constraints

\[ |y_{ij}^r + y_{jk}^r - y_{ik}^r| = 1 \]

consists in squaring both sides, leading to

\[ y_{ij,jk}^{r,r} - y_{ij,ik}^{r,r} - y_{ik,jk}^{r,r} = -1, \quad 1 \leq r \leq p, \quad i,j,k \in V_r, \quad i < j < k. \]
The Minimum Linear Subspace

In order to express constraints on $y$ in terms of $Y$, where we denote the product $y_{ij}^r y_{kl}^s$ by $y_{ij,kl}^{r,s}$.

A natural way to do this for the 3-cycle constraints $|y_{ij}^r + y_{jk}^r - y_{ik}^r| = 1$ consists in squaring both sides, leading to

$$y_{ij,jk}^r - y_{ij,ik}^r - y_{ik,jk}^r = -1, \quad 1 \leq r \leq p, \ i,j,k \in V_r, \ i < j < k.$$

These equations describe the smallest linear subspace that contains $\mathcal{P}_{QC}$.
MLCM as a Semidefinite Program

Now we can formulate MLCM as a semidefinite optimization problem in bivalent variables, where $C$ is a symmetric matrix of order $\zeta$ assigned to count the number of crossings

$$z^* = \min \{ \langle C, Z \rangle : Z \text{ satisfies the 3-cycle euqations, } Z \in \mathcal{E}, y \in \{-1, 1\} \}.$$
An SDP-Relaxation of MLCM

Now we can formulate an SDP relaxation for MLCM, where $C$ is a symmetric matrix of order $\zeta$ assigned to count the number of crossings

$$\min \{ \langle C, Z \rangle : Z \text{satisfies the 3-cycle equations, } Z \in \mathcal{E} \}. \quad (\text{SDP})_b$$

Dropping the integrality condition on $y$ gives the basic semidefinite relaxation $(\text{SDP})_b$ of MLCM.
Tightening the SDP-Relaxation - Metric Polytope

$Z$ is actually a matrix with $\{-1, 1\}$ entries in the original MLCM formulation.
Tightening the SDP-Relaxation - Metric Polytope

$Z$ is actually a matrix with $\{-1, 1\}$ entries in the original MLCM formulation.

Hence it satisfies the triangle inequalities, defining the metric polytope $\mathcal{M}$.

\[
\mathcal{M} = \{ Z : \begin{pmatrix}
-1 & -1 & -1 \\
-1 & 1 & 1 \\
1 & -1 & 1 \\
1 & 1 & -1
\end{pmatrix} \begin{pmatrix}
Z_{ij} \\
Z_{jk} \\
Z_{ik}
\end{pmatrix} \leq e, \; \forall \; i < j < k \}.
\]
Tightening the SDP-Relaxation - Metric Polytope

$Z$ is actually a matrix with $\{-1, 1\}$ entries in the original MLCM formulation.

Hence it satisfies the triangle inequalities, defining the metric polytope $\mathcal{M}$.

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

The SDP-relaxation can therefore be improved by asking in addition that $Z \in \mathcal{M}$. 
Tightening the SDP-Relaxation - Matrix Cuts

Another generic improvement was suggested by Lovász and Schrijver.
Tightening the SDP-Relaxation - Matrix Cuts

Another generic improvement was suggested by Lovász and Schrijver.

Applied to our problem, their approach suggests to multiply the 3-cycle inequalities

\[ 1 - y_{ij}^r - y_{jk}^r + y_{ik}^r \geq 0, \quad 1 + y_{ij}^r + y_{jk}^r - y_{ik}^r \geq 0, \]

by the nonnegative expressions \( (1 - y_{lm}^s) \) and \( (1 + y_{lm}^s) \).
Tightening the SDP-Relaxation - Matrix Cuts

Another generic improvement was suggested by Lovász and Schrijver.

Applied to our problem, their approach suggests to multiply the 3-cycle inequalities

\[ 1 - y_{ij}^r - y_{jk}^r + y_{ik}^r \geq 0, \quad 1 + y_{ij}^r + y_{jk}^r - y_{ik}^r \geq 0, \]

by the nonnegative expressions \((1 - y_{lm}^s)\) and \((1 + y_{lm}^s)\).

The SDP relaxation can therefore be improved by asking in addition that \(Z \in \mathcal{L}S\).
Strong SDP-Relaxation

In summary, we get the following tractable relaxation of $\mathcal{P}_{QC}$

$$\min \{\langle C, Z \rangle : Z \text{ satisfies the cycle equations, } Z \in \mathcal{E} \cap \mathcal{M} \cap \mathcal{LS} \}. \quad (SDP)$$
Strong SDP-Relaxation

In summary, we get the following tractable relaxation of $\mathcal{P}_{QC}$

$$\min \{ \langle C, Z \rangle : Z \text{ satisfies the cycle equations, } Z \in \mathcal{E} \cap \mathcal{M} \cap \mathcal{LS} \}.$$ (SDP)$_i$

SDP$_i$ is at least as strong as the linear relaxation.
Contents

1. Introduction
2. The Standard LP-Approach
3. An SDP-Approach
4. Theoretical Comparison
5. Computational Results
6. Conclusion
LP vs SDP-Crossing-Polytope

- Several facet classes are known for the LP-Crossing-Polytope (cycles, claws, dome paths)
LP vs SDP-Crossing-Polytope

- Several facet classes are known for the LP-Crossing-Polytope (cycles, claws, dome paths)
- SDP\textsubscript{i} is as least as tight as the linear relaxation with all known facets included except odd claws $\geq 5$
LP vs SDP-Crossing-Polytope

- Several facet classes are known for the LP-Crossing-Polytope (cycles, claws, dome paths).
- SDP$_i$ is at least as tight as the linear relaxation with all known facets included except odd claws ≥ 5.
- All constraints in SDP$_i$ are needed to show this result.
LP vs SDP-Crossing-Polytope

- Several facet classes are known for the LP-Crossing-Polytope (cycles, claws, dome paths).
- SDP\(_i\) is as least as tight as the linear relaxation with all known facets included except odd claws \( \geq 5 \).
- All constraints in SDP\(_i\) are needed to show this result.
- To ensure also odd claws \( \geq 5 \), we had to include clique inequalities of size \( \geq 5 \) which is far too expensive.
LP vs SDP-Crossing-Polytope

- Several facet classes are known for the LP-Crossing-Polytope (cycles, claws, dome paths)

- SDP is as least as tight as the linear relaxation with all known facets included except odd claws \( \geq 5 \)

- All constraints in SDP are needed to show this result

- To ensure also odd claws \( \geq 5 \), we had to include clique inequalities of size \( \geq 5 \) which is far too expensive

- So this supports our model choice
Contents

1 Introduction
2 The Standard LP-Approach
3 An SDP-Approach
4 Theoretical Comparison
5 Computational Results
6 Conclusion
Practical Implementation

- We maintain the elliptope constraint explicitly - this can be achieved by standard interior point methods
Practical Implementation

- We maintain the elliptope constraint explicitly - this can be achieved by standard interior point methods.

- Maintaining explicitly $O(\sum_{i=1}^{p} |V_i|^3)$ or more constraints is not an attractive option.
Practical Implementation

- We maintain the ellipotope constraint explicitly - this can be achieved by standard interior point methods.

- Maintaining explicitly $O\left(\sum_{i=1}^{p} |V_i|^3\right)$ or more constraints is not an attractive option.

- Therefore all (in)equality constraints are dealt through Lagrangian duality by using a dynamic bundle method.
Upper Bound Computation

To get feasible, (near)-optimal solutions we first apply the Goemans-Williamson hyperplane rounding to a matrix $\in \mathcal{E}$ obtained by the lower bound computation.
Upper Bound Computation

- To get feasible, (near)-optimal solutions we first apply the Goemans-Williamson hyperplane rounding to a matrix $\in \mathcal{E}$ obtained by the lower bound computation.

- Doing this we obtain a $\{-1, 1\}$ vector that can be made feasible with respect to the 3-cycle equations by flipping the signs of some entries.
Upper Bound Computation

- To get feasible, (near)-optimal solutions we first apply the Goemans-Williamson hyperplane rounding to a matrix $\in \mathcal{E}$ obtained by the lower bound computation.

- Doing this we obtain a $\{-1, 1\}$ vector that can be made feasible with respect to the 3-cycle equations by flipping the signs of some entries.

- In practice the quality of the solutions of this SDP heuristic is very high and the computation times are quite low compared to other (meta)heuristics.
Implementation Details

- We use a newly written ILP implementation with CPLEX 12.1 as a B&C framework - for the SDP we do not use a Branch & Bound.
Implementation Details

- We use a newly written ILP implementation with CPLEX 12.1 as a B&C framework - for the SDP we do not use a Branch & Bound.

- It turned out that separating the 3-cycle constraints on the fly but abstaining from searching further facet defining constraints is most efficient.
Implementation Details

- We use a newly written ILP implementation with CPLEX 12.1 as a B&C framework - for the SDP we do not use a Branch & Bound.
- It turned out that separating the 3-cycle constraints on the fly but abstaining from searching further facet defining constraints is most efficient.
- We could favorable compare our ILP implementation to previous ILP and SAT approaches from the literature.
Implementation Details

- We use a newly written ILP implementation with CPLEX 12.1 as a B&C framework - for the SDP we do not use a Branch & Bound.

- It turned out that separating the 3-cycle constraints on the fly but abstaining from searching further facet defining constraints is most efficient.

- We could favorable compare our ILP implementation to previous ILP and SAT approaches from the literature.

- We could also favorable compare our SDP implementation to a previous SDP approach restricted to bipartite crossing minimization by Buchheim, Wiegele and Zheng (2009).
Benchmark Set

We set up a benchmark library containing all considered instances and optimal solutions under http://www.ae.uni-jena.de/Research_Pubs/MLCM.html
Graphs With Varying Densities - Random Instances

<table>
<thead>
<tr>
<th>( p )</th>
<th>( n, \zeta )</th>
<th>SDP ( d = 0.1 )</th>
<th>SDP ( d = 0.5 )</th>
<th>SDP ( d = 0.9 )</th>
<th>ILP ( d = 0.1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>20, 381</td>
<td>✓, 45.1</td>
<td>✓, 478.6</td>
<td>✓, 319.6</td>
<td>10, 3.40</td>
</tr>
<tr>
<td></td>
<td>22, 463</td>
<td>10, 117.8</td>
<td>7, 3621.3</td>
<td>10, 923.4</td>
<td>10, 28.15</td>
</tr>
<tr>
<td>11</td>
<td>10, 495</td>
<td>10, 97.1</td>
<td>9, 2539.3</td>
<td>10, 722.4</td>
<td>10, 0.45</td>
</tr>
<tr>
<td></td>
<td>12, 726</td>
<td>10, 615.1</td>
<td>8, 8490.7</td>
<td>10, 2959.9</td>
<td>10, 50.52</td>
</tr>
<tr>
<td>20</td>
<td>8, 561</td>
<td>10, 13.5</td>
<td>10, 1145.9</td>
<td>10, 591.5</td>
<td>10, 0.01</td>
</tr>
<tr>
<td></td>
<td>10, 901</td>
<td>10, 1000.9</td>
<td>4, 11407.5</td>
<td>10, 6976.2</td>
<td>10, 2.46</td>
</tr>
</tbody>
</table>

The ILP could not solve any instance for \( d \geq 0.2 \) for any of the given parameter settings.
Real-World Graphs

- Rome and North Graphs - commonly used benchmark sets in the area of graph drawing
Real-World Graphs

- Rome and North Graphs - commonly used benchmark sets in the area of graph drawing
- All instances are sparse ($d \leq 0.1$)
Real-World Graphs

- Rome and North Graphs - commonly used benchmark sets in the area of graph drawing
- All instances are sparse ($d \leq 0.1$)
- ILP usually faster
Real-World Graphs

- Rome and North Graphs - commonly used benchmark sets in the area of graph drawing
- All instances are sparse ($d \leq 0.1$)
- ILP usually faster
- But SDP is stronger with respect to overall solvability
  - The SDP fails on only 2 graphs out of $\approx 10000$, the ILP fails on 21, including the aforementioned 2
  - On these 2 graphs the SDP bounds are significantly tighter ($853/854$ vs $336/854$)
Contents

1 Introduction
2 The Standard LP-Approach
3 An SDP-Approach
4 Theoretical Comparison
5 Computational Results
6 Conclusion
Conclusion

- For MLCM SDP relaxations provide theoretically and practically essentially tighter bounds than LP relaxations.
Conclusion

- For MLCM SDP relaxations provide theoretically and practically essentially tighter bounds than LP relaxations.
- The higher computation times surely pay off in Branch & Bound enumeration for graphs with density $> 0.1$. 
Conclusion

- For MLCM SDP relaxations provide theoretically and practically essentially tighter bounds than LP relaxations.

- The higher computation times surely pay off in Branch & Bound enumeration for graphs with density $> 0.1$.

- The SDP approach provides essentially better bounds and therefore smaller gaps for hard instances of any density.
Conclusion

- For MLCM SDP relaxations provide theoretically and practically essentially tighter bounds than LP relaxations.
- The higher computation times surely pay off in Branch & Bound enumeration for graphs with density $> 0.1$.
- The SDP approach provides essentially better bounds and therefore smaller gaps for hard instances of any density.
- These results mainly carry over to other quadratic ordering problems like SRFLPs, Linear Arrangement and Betweeness Problems.
Outlook

- There are three (combinable) directions to enhance the presented SDP based relaxations
Outlook

- There are three (combinable) directions to enhance the presented SDP based relaxations
  - Include further constraint classes to tighten the relaxation
Outlook

- There are three (combinable) directions to enhance the presented SDP based relaxations
  - Include further constraint classes to tighten the relaxation
  - Incorporate the SDP based bounds in a Branch & Bound framework
Outlook

- There are three (combinable) directions to enhance the presented SDP based relaxations
  - Include further constraint classes to tighten the relaxation
  - Incorporate the SDP based bounds in a Branch & Bound framework
  - Speed-up the computation over the elliptope using low-rank methods with restart ability
An SDP Approach to Multi-level Crossing Minimization

P. Hungerländer
joint work with
M. Chimani, M. Jünger, P. Mutzel

University of Klagenfurt - Department of Mathematics

15th Combinatorial Optimization Workshop, Aussois, January 2011