Angewandte Mathematik und Informatik
Universität zu Köln

Report No. 97.299

# A Polyhedral Approach to the Multi-Layer Crossing Minimization Problem

by

Michael Jünger, Eva K. Lee,
Petra Mutzel, Thomas Odenthal

1997

To appear in: G. Di Battista (ed.), Graph Drawing '97 (Proc.), Rome,
Lecture Notes in Computer Science, Springer-Verlag

Institut für Informatik
Universität zu Köln
Pohligstr. 1
50969 Köln

**Keywords:** Crossing number, linear ordering problem, branch-and-cut

# A Polyhedral Approach to the Multi-Layer Crossing Minimization Problem

# (Extended Abstract)

Michael Jünger[1]*, Eva K. Lee[2]**, Petra Mutzel[3]*, and Thomas Odenthal[4]

[1] Institut für Informatik, Universität zu Köln, mjuenger@informatik.uni-koeln.de
[2] Ind. & Sys. Eng., Georgia Institute of Technology, evakylee@isye.gatech.edu
[3] Max-Planck-Institut für Informatik, Saarbrücken, mutzel@mpi-sb.mpg.de
[4] Ind. Eng. & Op. Res., Columbia University, odenthal@ieor.columbia.edu

**Abstract.** We study the multi-layer crossing minimization problem from a polyhedral point of view. After the introduction of an integer programming formulation of the multi-layer crossing minimization problem, we examine the 2-layer case and derive several classes of facets of the associated polytope. Preliminary computational results for 2- and 3-layer instances indicate, that the usage of the corresponding facet-defining inequalities in a branch-and-cut approach may only lead to a practically useful algorithm, if deeper polyhedral studies are conducted.

## 1 Introduction

The *crossing number* of a graph $G$ is the minimum number of edge-crossings needed in any embedding of $G$ in the plane. The *Crossing Number* (CN) problem, that is, to determine the crossing number of a given graph $G$, is well-known to be $\mathcal{NP}$-hard (Garey and Johnson, 1983). Thus, there is little hope to find a polynomial-time algorithm for (CN) on general graphs. Moreover, formulae for the crossing number are only known for restricted types of graphs. We refer the reader to Richter and Thomassen (1994) and Shahrokhi et al. (1995) for surveys.

A (proper) *k-layered* graph is a graph whose node-set is partitioned into $k$ layers with the property that edges join only nodes in consecutive layers. Such graphs arise for example in workflow diagrams and are usually drawn such that all nodes in the same layer are placed on a horizontal line.

A natural combination of the two concepts crossing number and layered graphs is the *multi-layer crossing minimization problem*, which is to minimize the number of edge-crossings of a multi-layered graph. Harary (1969), Harary and Schwenk (1972), Watkins (1970) and Warfield (1977) give first structural results.

* Partially supported by DFG-Grant Ju204/7-1, Forschungsschwerpunkt "Effiziente Algorithmen für diskrete Probleme und ihre Anwendungen" and by ESPRIT LTR Project No. 20244 – ALCOM-IT.

** Supported in part by NSF grants CCR-9501584 and DMS-9527124.

From an algorithmic point of view, there have been extensive studies on crossing minimization for 2-layered graphs. In particular, during the 80's a lot of effort was focused on the design of heuristics for drawing such graphs with few crossings. Among others, we mention the barycenter heuristic (Sugiyama et al., 1981), the median heuristic (Eades and Wormald, 1994) and the greedy heuristics (Eades and Kelly, 1986). As reported in Jünger and Mutzel (1996), the barycenter heuristic yields the best results in terms of number of crossings and solution time.

Other approaches involve formulating the problems as integer programs. In the case of the crossing number problem, the geometric relationships between nodes and edges in the graph are difficult to be represented by an integer program. However, if the graph under consideration is "well-structured", this approach may be applicable and polyhedral theory and branch-and-cut algorithms may be able to provide an exact solution. This approach was first employed in Jünger and Mutzel (1996) for the one-sided *two-layer crossing minimization* problem. This problem is to find the minimum number of edge-crossings in a 2-layered graph, when one node-permutation of the two layers is fixed. Jünger and Mutzel (1996) showed that this problem can be transformed into a linear ordering problem which can then be solved by a branch-and-cut algorithm.

In this paper, we first address the (two-sided) two-layer crossing minimization problem. We first formulate this problem as an integer program, in which the optimal solution corresponds to a crossing-minimal representation of the graph. Next, we extend the integer programming formulation to a more general case – the proper *multi-layer crossing minimization* problem – which is to determine the minimum number of crossings in a (proper) multi-layered graph.

A branch-and-bound approach (Valls et al., 1996a, Jünger and Mutzel, 1996) and a tabu search (Valls et al., 1996b) were employed to determine the minimum crossing numbers in 2-layered graphs. In this paper, we will focus on polyhedral combinatorics to take advantage of the inherent structural properties of the problem. In Section 3, we define the polytope associated with the 2-layer crossing number problem as $\mathcal{P}_{\mathcal{CROSS}}\ (G)$. Several classes of facet-defining inequalities for $\mathcal{P}_{\mathcal{CROSS}}\ (G)$ will be reported. Most of the inequalities are derived from a combinatorial characterization of 2-layer planar graphs, i.e., graphs that can be drawn on two layers without edge-crossings. In Section 4, separation procedures for each class of facet-defining inequalities are described. Based on these facets, we implemented a branch-and-cut algorithm and compare its performance with a pure branch-and-bound algorithm.

## 2 The Integer Programming Formulations

Let $G = (V_1, V_2, E)$ be a bipartite graph with $|V_1| = n_1$ and $|V_2| = n_2$. The *Two-Layer Crossing Minimization* (TLCM) problem consists of determining the minimum number of crossings among the two layers of $G$ such that nodes can be permuted in both layers and edges are drawn as straight lines. Any solution is uniquely determined by the permutations $\pi_1$ and $\pi_2$ of $V_1$ and $V_2$. We define

$x_{ij} = 1$ if $\pi_1(i) < \pi_1(j)$, 0 otherwise; and $y_{ij} = 1$ if $\pi_2(i) < \pi_2(j)$, 0 otherwise. For a given $\pi_1$ and $\pi_2$, the number of crossings can be expressed as

$$C(\pi_1, \pi_2) = \sum_{i=1}^{n_1-1} \sum_{k=i+1}^{n_1} \sum_{j \in N(i)} \sum_{l \in N(k)} x_{ik} \cdot y_{lj} + x_{ki} \cdot y_{jl}, \tag{1}$$

where $N(v) = \{w \in V \mid e = (v, w) \in E\}$ denotes the set of neighbors of $v \in V = V_1 \cup V_2$ in $G$.

For a fixed permutation $\pi_1$ of $V_1$, this problem can be transformed to a linear ordering problem and solved with a branch-and-cut algorithm in short computation time (Jünger and Mutzel, 1996). The case of two freely permutable layers was handled by a branch-and-bound algorithm in which trivial lower bounds were employed for partial permutations of the smaller layer and the branch-and-cut algorithm for one-sided crossing minimization was applied for complete permutations of the smaller layer. In this paper, we describe an approach for solving (TLCM) directly.

We first present a nonlinear integer program for finding the minimum number of crossings for a 2-layered graph.

$$\begin{array}{llll}
\min & C(\pi_1, \pi_2) & & \\
\text{s.t.} & x_{ij} + x_{jk} + x_{ki} \leq 2 & 1 \leq i < j < k \leq n_1 & (2) \\
& y_{ij} + y_{jk} + y_{ki} \leq 2 & 1 \leq i < j < k \leq n_2 & (3) \\
& x_{ij} + x_{ji} = 1 & 1 \leq i < j \leq n_1 & (4) \\
& y_{ij} + y_{ji} = 1 & 1 \leq i < j \leq n_2 & (5) \\
& x_{ij}, y_{ij} \in \{0, 1\}. & &
\end{array} \qquad \text{(NTLCM)}$$

The 3-cycle-constraints, (2) and (3), constraints (4) and (5) and the binary restriction on the variables are needed to ensure that the resulting vectors $x$ and $y$ are incidence vectors of feasible linear orderings of the nodes on both layers (Grötschel et al., 1985).

To formulate the problem as an integer linear program, we introduce the *crossing variables*, $c_{ijkl}$, to denote if edges $(i, j)$ and $(k, l)$ cross (where $i < j$, $k < l$, $i < k$, and $j \neq l$). This leads to the following integer linear programming formulation for (TLCM).

For notational convenience, we assume without loss of generality that nodes in the first layer have smaller indices than nodes in the second layer and that $i, k \in V_1$ and $j, l \in V_2$. Moreover, since two edges incident at a node cannot result in a crossing, we will employ the notation "for all $(i, j), (k, l) \in E$" to describe the situation in which $i < j$, $k < l$, $i < k$, and $j \neq l$.

$$\begin{array}{llll}
\min & \sum_{(i,j),(k,l) \in E} c_{ijkl} & & \\
\text{s.t.} & x_{ik} + y_{lj} - c_{ijkl} \leq 1 & (i, j), (k, l) \in E & (6) \\
& x_{ki} + y_{jl} - c_{ijkl} \leq 1 & (i, j), (k, l) \in E & (7) \\
& (2),\ (3),\ (4) \text{ and } (5) & & \\
& x_{ij}, y_{ij}, c_{ijkl} \in \{0, 1\}. & &
\end{array} \qquad \text{(ITLCMa)}$$

**Theorem 1.** *(NTLCM) and (ITLCMa) are equivalent formulations for (TLCM).*

*Proof.* Let $(x, y, c)$ be a feasible solution to (ITLCMa). Clearly, $(x, y)$ is also feasible for (NTLCM). From (6) and (7), if $x_{ik} = y_{lj} = 1$ or $x_{ki} = y_{jl} = 1$, then $c_{ijkl} = 1$. On the other hand, if either $x_{ik} = 0$ or $y_{lj} = 0$, then $c_{ijkl}$ can be either 0 or 1. However, for the sake of minimization, we must have $c_{ijkl} = 1$ if and only if $x_{ik} = y_{lj} = 1$ or $x_{ki} = y_{jl} = 1$. In other words, $c_{ijkl} = 1$ if and only if $x_{ik} \cdot y_{lj} = 1$ or $x_{ki} \cdot y_{jl} = 1$.

To prove the other direction, let $(x, y)$ be a feasible solution to (NTLCM). Define $c_{ijkl} = x_{ik} \cdot y_{lj} + x_{ki} \cdot y_{jl}$. Clearly, $(x, y, c)$ is feasible for (ITLCMa). □

Formulation (ITLCMa) can be reduced via variable substitutions from equations (4) & (5) to the following formulation (ITLCMb).

$$
\begin{array}{llll}
\min & \displaystyle\sum_{(i,j),(k,l)\in E} c_{ijkl} & & \\
\text{s.t.} & & & \\
& -c_{ijkl} \le y_{jl} - x_{ik} \le c_{ijkl} & (i,j),(k,l) \in E, j < l & (8) \\
& 1 - c_{ijkl} \le y_{lj} + x_{ik} \le 1 + c_{ijkl} & (i,j),(k,l) \in E, j > l & (9) \\
& 0 \le x_{ij} + x_{jk} - x_{ik} \le 1 & 1 \le i < j < k \le n_1 & (10) \\
& 0 \le y_{ij} + y_{jk} - y_{ik} \le 1 & 1 \le i < j < k \le n_2 & (11) \\
& x_{ij}, y_{ij}, c_{ijkl} \in \{0, 1\}. & &
\end{array}
$$

A generalization of this approach to the (proper) *Multi-Layer Crossing Minimization* (MLCM) problem is straightforward. A (proper) *multi-layered* graph, $G = (V, E)$, is a graph in which the node-set $V$ is partitioned into disjoints sets: $V = V_1 \cup V_2 \cup \ldots \cup V_p$ with $|V_i| = n_i$ such that for all edges $(u, v) \in E$, we must have $u \in V_i$, $v \in V_{i+1}$, for some $i = 1, \ldots, p-1$. In other words, edges are only allowed between successive layers. Indeed, we can write $E = E_1 \cup E_2 \cup \ldots \cup E_{p-1}$ with $E_i$ the edge-set between nodes from layer $i$ and $i+1, i = 1, \ldots, p-1$. Using this property, we only need to couple successive layers with crossing variables. The corresponding integer program for (MLCM) can be written as:

$$
\begin{array}{llll}
\min & \displaystyle\sum_{r=1}^{p-1} \sum_{(i,j),(k,l)\in E_r} c^r_{ijkl} & & \\
\text{s.t.} & & & \\
& -c^r_{ijkl} \le x^{r+1}_{jl} - x^r_{ik} \le c^r_{ijkl} & (i,j),(k,l) \in E_r, j < l & (12) \\
& 1 - c^r_{ijkl} \le x^{r+1}_{lj} + x^r_{ik} \le 1 + c^r_{ijkl} & (i,j),(k,l) \in E_r, j > l & (13) \\
& 0 \le x^r_{ij} + x^r_{jk} - x^r_{ik} \le 1 & 1 \le i < j < k \le n_r & (14) \\
& x^r_{ij}, y^r_{ij}, c^r_{ijkl} \in \{0, 1\}. & &
\end{array}
$$

where $r = 1, \ldots, p-1$.

In a general multi-layered graph the restriction on the edges is relaxed in such a way that edges are allowed to span more than two layers. A solution for (MLCM) can be adapted to a general multi-layered graph by introducing a dummy-node on a layer whenever an edge crosses this layer (see Eades and Wormald (1994) for the transformation). Applying this technique, our integer programming formulation can be employed within popular layout-algorithms (Sugiyama et al., 1981).

## 3 Polyhedral Study

Given a bipartite graph $G = (V_1, V_2, E)$ with $|V_1| = n_1$ and $|V_2| = n_2$, we define a *crossing configuration* to be a vector $(x, y, c)$, where $x \in \mathbb{B}^{\binom{n_1}{2}}$ and $y \in \mathbb{B}^{\binom{n_2}{2}}$ are the linear ordering variables for the first and second layer, respectively, and $c \in \mathbb{B}^t$ is the vector of crossing variables, with $t$ the total number of crossing variables for the graph. Let $s := \binom{n_1}{2} + \binom{n_2}{2} + t$. We define the polytope $\mathcal{P}_{\mathcal{CROSS}}(G) \subseteq \mathbb{R}^s$ as

$$\mathcal{P}_{\mathcal{CROSS}}(G) = conv\{(x, y, c) \in \mathbb{B}^s : (x, y, c) \text{ is a crossing configuration}\}.$$

In this section, we first show that $\mathcal{P}_{\mathcal{CROSS}}(G)$ is full dimensional. We then present the various classes of facet-defining inequalities we obtained thus far. We denote a vector of all 1's by the vector $e$ of appropriate dimension.

**Theorem 2.** *$\mathcal{P}_{\mathcal{CROSS}}(G)$ is full dimensional.*

*Sketch of Proof.* It suffices to show that there exist $s + 1$ affinely independent crossing configurations. Let $\mathcal{P}_{\mathcal{LO}1}$ ($\mathcal{P}_{\mathcal{LO}2}$) be the linear ordering polytope associated with the $x$- ($y$-) variables. Since $\mathcal{P}_{\mathcal{LO}1}$ is of dimension $\binom{n_1}{2}$, there exist $\binom{n_1}{2}$ affinely independent vectors, $x^1, \ldots, x^{\binom{n_1}{2}}$. Similarly, there exist $\binom{n_2}{2}$ affinely independent vectors for $\mathcal{P}_{\mathcal{LO}2}$, $y^1, \ldots, y^{\binom{n_2}{2}}$. Thus the following $s + 1$ vectors are in $\mathcal{P}_{\mathcal{CROSS}}(G)$ and are affinely independent:

i. $(x^i, 0, e)$, $i = 1, \ldots, \binom{n_1}{2}$;
ii. $(0, y^i, e)$, $i = 1, \ldots, \binom{n_2}{2}$;
iii. $(\bar{x}^i, \bar{y}^i, e - e_i)$, $i = 1, \ldots, t$; where $\bar{x}^i, \bar{y}^i$ are chosen so that the vectors satisfy all the constraints for (ITLCMb) after setting exactly one crossing variable $c_i$ to 0;
iv. $(e, e, e)$. □

**Theorem 3.** *The inequalities $c_{ijkl} \geq 0$ are redundant, while $c_{ijkl} \leq 1$ are facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$.*

*Proof.* If we add the two parts of inequality (8) together, we clearly get $c_{ijkl} \geq 0$ for all edges $(i, j), (k, l) \in E, j < l$. Similarly, if we add the two parts of inequality (9) together, we get $c_{ijkl} \geq 0$ for all edges $(i, j), (k, l) \in E, l < j$. Hence $c_{ijkl} \geq 0$ is redundant.

Clearly, $c_{ijkl} \leq 1$ is valid for $\mathcal{P}_{\mathcal{CROSS}}(G)$. To show the dimension, note that for each $ijkl$, all but one vector from Theorem 2 satisfy $c_{ijkl} = 1$, proving that it is facet-defining. □

In the next theorem we show that all facet-defining inequalities for $\mathcal{P}_{\mathcal{LO}\,1}$ ($\mathcal{P}_{\mathcal{LO}\,2}$) are facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$.

**Theorem 4.** *Let $a^T x \leq a_0$ be a facet-defining inequality for $\mathcal{P}_{\mathcal{LO}\,1}$ ($\mathcal{P}_{\mathcal{LO}\,2}$). Then $a^T x \leq a_0$ is also facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$.*

*Sketch of Proof.* Validity is trivial. Without loss of generality, we restrict ourselves to facet-defining inequalities corresponding to the $x$-variables.

Since $a^T x \leq a_0$ is a facet-defining inequality for $\mathcal{P}_{\mathcal{LO}\,1}$, there exist $\binom{n_1}{2}$ affinely independent vectors satisfying $a^T x \leq a_0$ with equality, namely, $x^1, \ldots, x^{\binom{n_1}{2}}$. Now, order the $y$ variables $y_{ij}$ such that $i < j$. Let $D_k$ be the set of the first $k$ indices in vector $y$, and $y^{D_k}$ be the corresponding characteristic vector. Then the following $s$ vectors are affinely independent:

i. $(x^i, 0, e),\ i = 1, \ldots, \binom{n_1}{2}$;
ii. $(x^1, y^{D_k}, e),\ k = 1, \ldots, \binom{n_2}{2}$;
iii. $(x^1, \bar{y}^i, e - e_i),\ i = 1, \ldots, t$, where $\bar{y}^i$ is chosen such that all constraints in (ITLCMb) are satisfied. □

Similar proof techniques are employed in all the theorems regarding facet-defining properties: proving validity for $\mathcal{P}_{\mathcal{CROSS}}(G)$, and showing that the maximum number of affinely independent vectors satisfying the inequality at equality is $s$. For space reasons, we omit proofs for the remaining theorems.

**Theorem 5.** *Inequalities (8) and (9) are facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$.*

Other classes of facet-defining inequalities can be derived from the following characterization of "2-layer planar" graphs (these are graphs that can be drawn on two layers without crossings) using forbidden subgraphs.

**Theorem 6. [Harary and Schwenk, 1972, Tomii et al., 1977, Eades et al., 1986]** *A 2-layer graph is 2-layer planar if and only if it contains no cycle and no 3-claw.*

In other words, whenever there is a cycle or a 3-claw (see Figure 1 below) in the graph, we have at least one crossing. Moreover, the exact 2-layer crossing number for cycle graphs is known.

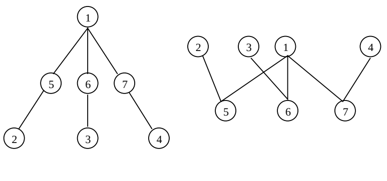


**Fig. 1.** 3-claw graph

**Theorem 7. [Harary and Schwenk, 1972, May and Szkatula, 1988]** *The 2-layer crossing number of a cycle $C$ is $\frac{|C|-2}{2}$.*

In the next theorem, we state the valid inequalities arising from cycles as well as $p$-claws, i.e., claw graphs such as in Figure 1 with $p$ legs.

**Theorem 8.** *The following inequalities are valid for $\mathcal{P}_{\mathcal{CROSS}}(G)$.*

$$\begin{array}{lll} (i.) & \sum_{(i,j),(k,l)\in C} c_{ijkl} \geq \frac{|C|-2}{2} & \textit{for all cycles } C \textit{ in } G. \\ (ii.) & \sum_{(i,j),(k,l)\in W} c_{ijkl} \geq 1 & \textit{for all 3-claws } W \textit{ in } G. \\ (iii.) & \sum_{(i,j),(k,l)\in W_p} c_{ijkl} \geq \begin{cases} \frac{p}{2}(\frac{p}{2}-1) & p \textit{ even} \\ (\frac{p-1}{2})^2 & p \textit{ odd} \end{cases} & \textit{for all } p\textit{-claws } W_p \textit{ in } G. \end{array}$$

Though these inequalities are not facet-defining, we can derive several classes of facets from them, which we will describe next. For notational convenience all integers representing nodes are to be considered modulo $|C|$.

**Theorem 9.** *Let $C = (0, \ldots, i-1, i, i+1, i+2, \ldots, |C-1|, 0)$ be a cycle of length $|C|$ in the graph. For every edge $(i, i+1) \in C$, the following inequality is facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$:*

$$\sum_{(k,l)\in C, k\neq i, l\neq i+1} c_{i,i+1,k,l} + c_{i,i-1,i+1,i+2} \geq 1.$$

Next, we consider any cycle $C = (0, \ldots, |C-1|, 0)$ of length $|C|$ in the graph, where we fix an arbitrary ordering.

**Theorem 10.** *Let $S$ be the set consisting of all pairs of edges $(i,j), (k,l) \in C$ for which edge $(k,l)$ has an odd distance from edge $(i,j)$ in the cycle, i.e., $k = i+2m$ or $i = k+2m$ for $m = 1, 2, \ldots, |C|/2$, respectively. Then the following inequality is facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$:*

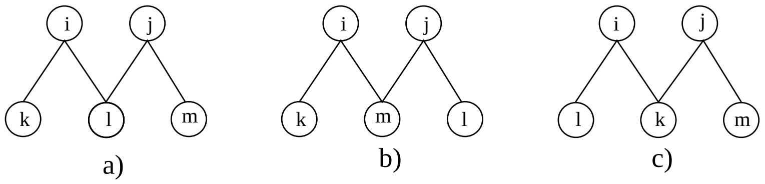
$$\sum_{(i,j),(k,l)\in S} c_{ijkl} \geq \frac{|C|-2}{2}.$$

We now turn to facet-defining inequalities based on 3-claws $W$.

**Theorem 11.** *Let $T$ be the set consisting of all pairs of edges of $W$ except those pairs of edges that are either both within the lower or the upper part of the 3-claw and are not adjacent to each other. Then the following inequality is facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$:*

$$\sum_{(i,j),(k,l)\in T} c_{ijkl} \geq 1.$$

Another class of inequalities can be constructed from the *dome-path*. Each of these structures gives rise to two facet-defining inequalities.



a) b) c)

**Fig. 2.** Dome paths

**Theorem 12.** *For $k < l < m$, the following inequalities are facet-defining for $\mathcal{P}_{\mathcal{CROSS}}(G)$:*

$$\begin{array}{rr} x_{kl} - 2x_{km} + x_{lm} - c_{ikjl} - c_{iljm} \leq 0; & \\ -x_{kl} + 2x_{km} - x_{lm} - c_{ikjl} - c_{iljm} \leq 0; & \textit{Figure (2 a)} \end{array}$$

$$\begin{array}{rr} 2x_{kl} - x_{km} + x_{lm} - c_{ikjm} - c_{imjl} \leq 1; & \\ -2x_{kl} + x_{km} - x_{lm} - c_{ikjm} - c_{imjl} \leq -1; & \textit{Figure (2 b)} \end{array}$$

$$\begin{array}{rr} x_{kl} - x_{km} + 2x_{lm} - c_{iljk} - c_{ikjm} \leq 1; & \\ -x_{kl} + x_{km} - 2x_{lm} - c_{iljk} - c_{ikjm} \leq -1. & \textit{Figure (2 c)} \end{array}$$

Notice that in each of these inequalities, the linear-ordering variable associated with the two endnodes of the dome-path always has coefficient 2.

## 4 Separation Procedures and Computational Experience

In this section, we briefly highlight the separation procedures and their complexity for the classes of facets described in Section 3. We also report our computational experience with a branch-and-cut algorithm based on these facets.

Classes of inequalities from Theorem 9 can be separated in polynomial time. For every fixed edge $(i, j)$, we set up an auxiliary graph $G' = (V', E')$ with $V' = V$ and $E' = E$ and edge-weights $w_{kl} = c_{ijkl}$ for every edge $(k, l)$ not adjacent to $i$ and $j$. Next we solve a series of shortest path problems, searching for the shortest path from node $a$ to node $b$, where $a$ and $b$ are nodes adjacent to $i$ and $j$, respectively. We perform this for all combinations of adjacent edges of node $i$ and $j$. If the length of the obtained path plus the cost of the fixed edge is greater than 1, we have found a violated inequality. If we cannot find such a path, then no inequality of this class is violated. The running time of this separation procedure is $O(|E| \cdot S(|V|, |E|) \cdot 2d)$, where $S(|V|, |E|)$ denotes the time for solving a shortest path problem and $d$ is the maximum node-degree in the graph.

In addition, inequalities derived from dome paths can be checked by enumeration, which can be done in polynomial time. Similarly, $k$-claws can be identified in polynomial time via enumeration, or by computing a matching between the nodes incident to a root-node and their adjacent edges (for a fixed $k$).

We perform a heuristic procedure for separating classes of facets from Theorem 10. We first choose a root node, and apply a breadth-first-search method to identify cycles in the graph. Once a cycle is found, the corresponding inequalities are formed and checked if they are violated by the current fractional solution.

Table 1 shows computational results for a set of 2-layered graphs which have appeared in the literature: [JM96a] and [JM96b] are the graphs of Figures 1 and 4 in Jünger and Mutzel (1996), respectively. [Mut97a] and [Mut97b] are the graphs of Figures 1 and 2 in Mutzel (1997). [KW95a] and [KW95b] are the two bipartite graphs of a 3-partite graph which arises from an illustration of dependencies in constraint negotiation (Kusiak and Wang, 1995). [Him97a] and [Him97b] are the two bipartite graphs of a 3-partite graph visualizing features of a graph-drawing-system. [MS88a] and [MS88b] are again the two bipartite parts of a 3-partite graph arising from computing invariants of matrices (May and Szkatula, 1988). [SM96a] and [SM96b] are two bipartite parts of a multi-layered social networks lattice (Shieh and McCreary, 1996). [Fuk96a] and [Fuk96b] are the two bipartite parts of a 3-partite graph which came up in the study of polyhedral face lattices (Fukuda, 1996). *Name,* $|V_1|, |V_2|, |E|$, *No. Var* denote the name of the problem, the number of nodes in each layer, the number of edges in the graph and the number of variables in the formulation, respectively. *LB* and *Opt. IP* denote the lower bound (the LP-value after adding cuts from Section 3 to the LP-relaxation and no subsequent branching) and the optimal objective value for the integer program, respectively.

For each instance in the table, the original lower bound before addition of cuts is 0. Comparing the values *LB* and *Opt. IP*, we observe that cutting planes helped to close the gap (calculated as (Opt. IP - LB) / Opt. IP) by 50% to 100%. In particular, out of 14 instances, the gap for 6 of them was completely closed after the addition of cuts.

In the rest of the table, computational results for 2-layered graphs for algorithm B-C, a branch-and-cut algorithm incorporating the facets from Section 3, and algorithm B-B, a pure branch-and-bound algorithm, are reported. We used a SUN ULTRA2 (167MHZ) workstation and CPLEX 4.0 as the branch-and-bound solver with strong branching selected as the branching variable selection (We tested various options in CPLEX and strong branching is far superior to other branching rules).

*B-C cuts, B-C time,* and *B-C nodes* denote the total number of cuts added, the total time required (with time limit set to 3600 seconds), and the total number of enumeration nodes needed by algorithm B-C. Finally, *B-B time* and *B-B nodes* denote respectively the time required to solve to optimality and the number of nodes searched by algorithm B-B for solving our formulation of (TLCM) when no cutting planes are included.

The analysis of the computational results has a mixed outcome. For small

**Table 1.** Computational results for 2-layered graphs

| Name | $\|V_1\|$ | $\|V_2\|$ | $\|E\|$ | No. Var. | LB | Opt. IP | B-C | B-C | B-C | B-B | B-B |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | cuts | time | nodes | time | nodes |
| [JM96a] | 8 | 8 | 19 | 193 | 10 | 19 | 45 | 5.18 | 9 | 2.68 | 14 |
| [JM96b] | 10 | 10 | 20 | 249 | 4 | 4 | 20 | 0.58 | 1 | 0.96 | 2 |
| [Mut97a] | 6 | 9 | 14 | 118 | 2 | 2 | 1 | 0.31 | 1 | 0.15 | 1 |
| [Mut97b] | 14 | 11 | 28 | 484 | 14 | 24 | 200 | 246.09 | 173 | 372.11 | 395 |
| [KW95a] | 8 | 7 | 15 | 127 | 9 | 10 | 21 | 0.42 | 3 | 0.41 | 6 |
| [KW95b] | 7 | 3 | 14 | 71 | 15 | 15 | 20 | 0.30 | 1 | 0.07 | 1 |
| [Him97a] | 8 | 16 | 26 | 409 | 10 | 19 | 24 | 8.87 | 7 | 8.17 | 8 |
| [Him97b] | 16 | 3 | 8 | 142 | 0 | 0 | 2 | 0.11 | 1 | 0.007 | 1 |
| [MS88a] | 10 | 15 | 39 | 791 | 35 | 35 | 70 | 10.83 | 3 | 31.77 | 17 |
| [MS88b] | 15 | 12 | 41 | 893 | 52 | 54 | 189 | 15.28 | 3 | 309.97 | 251 |
| [SM96a] | 26 | 18 | 29 | 484 | 0 | 0 | 0 | 0.21 | 1 | 0.012 | 1 |
| [SM96b] | 18 | 17 | 29 | 663 | 8 | 12 | 27 | 56.09 | 23 | 210.33 | 579 |
| [Fuk96a] | 8 | 12 | 24 | 334 | 18 | 35 | 134 | 3600 $^*$ | 2372 | 71.03 | 153 |
| [Fuk96b] | 12 | 6 | 24 | 309 | 23 | 45 | 175 | 3600 $^+$ | 1955 | 597.47 | 1636 |

$^*$Not optimal with lower bound of 31 $^+$Not optimal with lower bound of 39

problems (e.g., [JM96a]) algorithm B-B – although using more enumeration nodes than algorithm B-C – has a faster running time than algorithm B-C. Moreover, we mention that the branch-and-bound algorithm of Jünger and Mutzel (1996) was able to solve instances of (TLCM) to proven optimality when the smaller layer has 15 or fewer nodes in a fast computation time. Since cutting plane procedures can be rather expensive, on smaller problems, it is faster to simply apply a plain branch-and-bound approach.

For problems with a small gap (e.g., [MS88a] and [MS88b]), however, algorithm B-C performs best, both in terms of running time and number of enumeration nodes. On the other hand, we were not able to solve problems [Fuk96a] and [Fuk96b] to optimality with algorithm B-C within the time-limit. For those problems the gap is about 50% and this indicates that a deeper study of the facial structure of $\mathcal{P}_{\mathcal{CROSS}}(G)$ is needed in order to come up with a practical useful algorithm.

However, we caution that the success of a branch-and-cut approach depends on many different factors, to name a few, how often and how many cuts should be generated, how to integrate these cuts effectively within the tree search, and in what order should the cuts be generated. We anticipate further studies to determine the viability of this approach to our application.

We next focus on the (MLCM) problem and report our experience with algorithms B-C and B-B on some 3-partite graphs. These graphs are [KW95], [Him97], [MS88], and [SM96], which are the combined 3-partite graphs of the graphs described above. The contents of Table 2 is similar to that of Table 1.

All cutting planes arising from the two bipartite parts are valid cutting planes for the combined 3-partite graph. The analysis shows that for smaller problems ([KW95] and [Him97]) algorithm B-B has the best performance, whereas for the bigger problems ([MS88] and [SM96]), algorithm B-C is the winner. Both algorithms were not able to solve the combined 3-partite graph of [Fuk96a] and [Fuk96b] to optimality within the time-limit.

**Table 2.** Computational results for 3-layered graphs

| Name | $\|V_1\|$ | $\|V_2\|$ | $\|V_3\|$ | $\|E_1\|$ | $\|E_2\|$ | No. | LB | Opt. | B-C | B-C | B-C | B-B | B-B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Var. | | IP | cuts | time | nodes | time | nodes |
| [KW95] | 8 | 7 | 3 | 15 | 14 | 198 | 24 | 27 | 153 | 2.57 | 15 | 0.76 | 9 |
| [Him97] | 8 | 16 | 3 | 26 | 8 | 551 | 10 | 21 | 82 | 21.61 | 15 | 8.55 | 7 |
| [MS88] | 10 | 15 | 12 | 39 | 41 | 1684 | 87 | 91 | 469 | 268.54 | 19 | 3553.85 | 908 |
| [SM96] | 26 | 18 | 17 | 29 | 29 | 1347 | 8 | 13 | 35 | 331.63 | 21 | 530.33 | 1579 |

## References

Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Algorithms for drawing graphs: An annotated bibliography. Computational Geometry: Theory and Applications **4** (1994) 235–282.

Eades, P., Kelly, D.: Heuristics for Reducing Crossings in 2-Layered Networks. Ars Combinatoria **21-A** (1986) 89–98.

Eades, P., Wormald, N.C.: Edge Crossings in Drawings of Bipartite Graphs. Algorithmica **11** (1994) 379–403.

Eades, P., McKay, B.D., Wormald, N.C.: On an edge crossing problem. Proc. 9th Australian Computer Science Conference, Australian National University (1986) 327–334.

Fukuda, K.: Face Lattices. Personal Communication (1996).

Garey, M.R., Johnson, D.S.: Crossing Number is NP-Complete. SIAM Journal on Algebraic and Discrete Methods **4** (1983) 312–316.

Grötschel, M., Jünger, M., Reinelt, G.: Facets of the linear ordering polytope. Mathematical Programming **33** (1985) 43–60.

Harary, F.: Determinants, permanents and bipartite graphs. Mathematical Magazine **42** (1969) 146–148.

Harary, F., Schwenk, A.: A new crossing number for bipartite graphs. Utilitas Mathematica **1** (1972) 203–209.

Himsolt, M.: Personal Communication (1997).

Jünger, M., Mutzel, P.: 2-Layer Straightline Crossing Minimization: Performance of Exact and Heuristic Algorithms. Journal of Graph Algorithms and Applications (JGAA), (http://www.cs.brown.edu/publications/jgaa/), No. 1, Vol. 1, (1997) 1–25.

Kusiak, A., Wang, J.: Dependency Analysis in Constraint Negotiation. IEEE Trans. Sys. Man, Cybern. **25** (1995) 1301–1313.

May, M., Szkatula, K.: On the bipartite crossing number. Control and Cybernetics **17** No.1 (1988) 85–97.

Mutzel, P.: An Alternative Method for Crossing Minimization. Lecture Notes in Computer Science LNCS 1190 (1997) 318–333.

Richter, B.R., Thomassen, C.: A survey on crossing numbers. Manuscript, Carleton University and The Technical University of Denmark (1994).

Shahrokhi, F., Szelky, L.A., Vrtô, I.: Crossing Number of Graphs, Lower Bound Techniques and Algorithms: A Survey. Lecture Notes in Computer Science LNCS 894 (1995) 131–142.

Shieh, F., McCreary, C.,: Directed Graphs Drawing by Clan-Based Decomposition. Lecture Notes in Computer Science LNCS 1027 (1996) 472–482.

Sugiyama, K., Tagawa, S., Toda, M.: Methods for Visual Understanding of Hierarchical System Structures. IEEE Trans. Syst. Man, Cybern. SMC-11 (1981) 109–125.

Tomii, N., Kambayashi, Y., Shunzo, Y.: On Planarization Algorithms of 2-Level Graphs. Papers of tech. group on electronic computers, IECEJ, EC77–38 (1977) 1–12.

Valls, V., Marti, R., Lino, P.: A Branch and Bound Algorithm for Minimizing the Number of Crossing Arcs in Bipartite Graphs. Journal of Operational Research **90** (1996a) 303–319.

Valls, V., Marti, R., Lino, P.: A tabu thresholding algorithm for arc crossing minimization in bipartite graphs. Annals of Operations Research **63** (1996b) 223–251.

Warfield, J.N.: Crossing Theory and Hierarchy Mapping. IEEE Trans. Syst. Man, Cybern. SMC-7 (1977) 505–523.

Watkins, M.E.: A special crossing number for bipartite graphs: a research problem. Annals of New York Academy of Sciences **175** (1970) 405–410.

This article was processed using the LaTeX macro package with LLNCS style