An Alternative Method to Crossing Minimization on Hierarchical Graphs*

Petra Mutzel

Max-Planck-Institut für Informatik

Im Stadtwald, D-66123 Saarbrücken, Germany

Abstract. A common method for drawing directed graphs is, as a first step, to partition the vertices into a set of k levels and then, as a second step, to permute the vertices within the levels such that the number of crossings is minimized. We suggest an alternative method for the second step, namely, removing the minimal number of edges such that the resulting graph is k-level planar. For the final diagram the removed edges are reinserted into a k-level planar drawing. Hence, instead of considering the k-level crossing minimization problem, we suggest solving the k-level planarization problem. In this paper we address the case $k = 2$. First, we give a motivation for our approach. Then, we address the problem of extracting a 2-level planar subgraph of maximum weight in a given 2-level graph. This problem is NP-hard. Based on a characterization of 2-level planar graphs, we give an integer linear programming formulation for the 2-level planarization problem. Moreover, we define and investigate the polytope $2\mathcal{LPS}(G)$ associated with the set of all 2-level planar subgraphs of a given 2-level graph G. We will see that this polytope has full dimension and that the inequalities occuring in the integer linear description are facet-defining for $2\mathcal{LPS}(G)$. The inequalities in the integer linear programming formulation can be separated in polynomial time, hence they can be used efficiently in a branch-and-cut method for solving practical instances of the 2-level planarization problem. Furthermore, we derive new inequalities that substantially improve the quality of the obtained solution. We report on extensive computational results.

Key words: 2-Level Graphs, 2-Level Planarization, Integer Linear Programming, Polyhedral Combinatorics, Branch-and-Cut

1. Introduction

Directed graphs are widely used to represent structures in many fields such as economics, social sciences, mathematical and computer science. A good visualization of structural information allows the reader to focus on the information content of the diagram.

A common method for drawing directed graphs has been introduced by Sugiyama et al. [STT81] and Carpano [Car80]. In the first step, the vertices are partitioned into a set of k levels, and in the second step, the vertices within each level are permuted in such a way that the number of crossings is small. We suggest an alternative approach for the second step. From now on let us assume that

* *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*

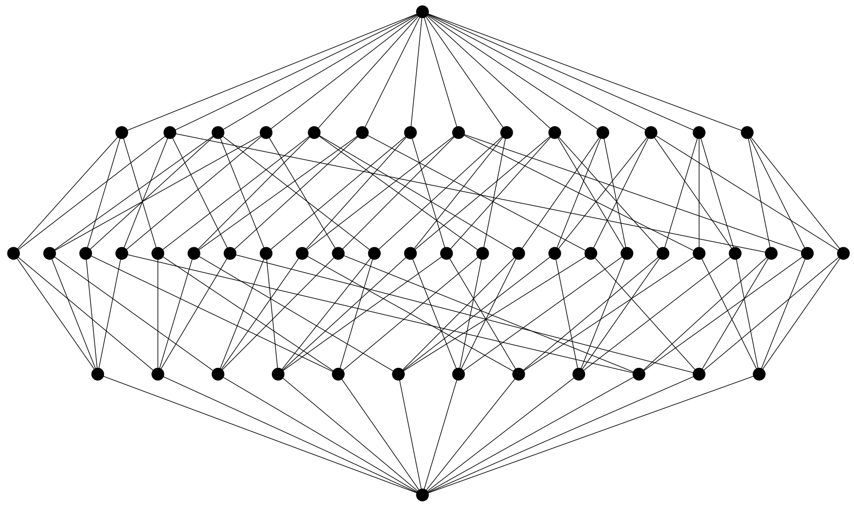


Fig. 1. A real world graph with high crossing number [Fuk96]

we are given a *k-level hierarchy* (*k-level graph*), i.e., a graph $G = (V, E) = (V_1, V_2, \ldots, V_k, E)$ with vertex sets $V_1, \ldots, V_k$, $V = V_1 \cup V_2 \ldots \cup V_k$, $V_i \cap V_j = \emptyset$ for $i \neq j$, and an edge set E connecting vertices in levels V_i and V_j with $i \neq j$ ($1 \leq i, j \leq k$). V_i is called the *i-th level.* A k-level hierarchy is drawn in such a way that the vertices in each level V_i are drawn on a horizontal line L_i with y-coordinate $k - i$, and the edges are drawn as straight lines. In contrary to the definitions of a hierarchy in [STT81,HP96], we do not care about the direction of the edges, since it is irrelevant for the problem considered here. Essentially, a k-level hierarchy is a k-partite graph that is drawn in a special way.

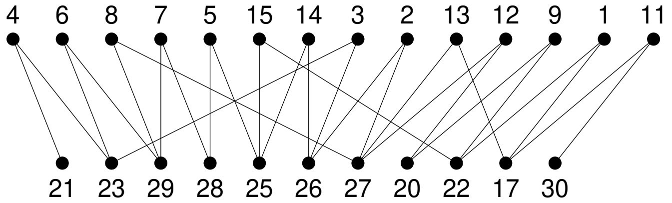
Even for 2-level graphs the straightline crossing minimization problem is NP-hard. Exact algorithms based on branch and bound have been suggested by various authors (see, e.g., [VML96] and [JM96]). For $k \geq 2$, a vast amount of heuristics has been published in the literature (see, e.g., [War77,STT81,EK86,Mäk90,EW94a] and [Dre94]).

Various authors have already asked the following question: Is a hierarchical drawing with the minimal number of crossings always nicer than a drawing that has many more crossings? They ended up with the following answer: "We merely want to draw a reasonably clear picture which has a "relatively small" number of crossings" [Car80].

For graphs that have a relatively small hierarchical crossing number, this statement goes along with our observation. But in some applications, hierarchical graphs arise that have a relatively high hierarchical crossing number, such as the graph shown in Figure 1. For these graphs we have to find a new method that substantially increases the readability of these diagrams.

One approach may be to remove a minimal set of edges such that the remaining k-level graph can be drawn without edge crossings. In the final drawing, the removed edges are reinserted. Since the insertion of each edge may produce many crossings, the final drawing may be far from an edge-crossing minimal drawing.

Figure 2(a) shows a drawing of a graph obtained by 2-level planarization, whereas Figure 2(b) shows the same graph drawn with the minimal number of edge crossings (using the exact algorithm given in [JM96]). Although the drawing in Figure 2(a) has 34 crossings, that is 41% more crossings than the drawing in Figure 2(b) (24 crossings), the reader will not recognize this fact. On the contrary, 90% of the colleagues that we have asked thought that the number of crossings in Figure 2(a) is



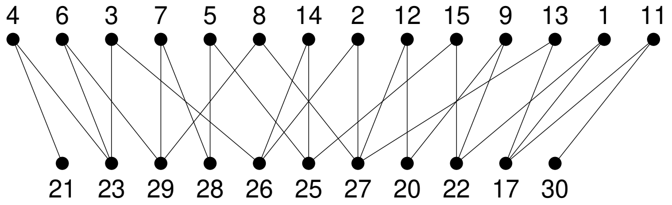


Fig. 2. A graph (a) drawn using k-planarization and (b) drawn with the minimal number of crossings computed by the algorithm in [JM96]

less than in Figure 2(b). This encourages us to study the k-level planarization problem.

Another motivation for studying k-level planarization arises from the fact that the k-level crossing minimization problem is a very hard problem that cannot be solved exactly or approximately (with some reasonable solution guarantees) in practice. Our experiments in [JM96] showed that for sparse graphs, such as they occur in graph drawing, the heuristic methods used in practice are far from the optimum. We believe that the methods of polyhedral combinatorics that have been successfully applied for the maximum planar subgraph problem [JM93a,JM93b,Mut94], and for the straightline crossing minimization problem on two levels where one level is fixed [JM96], may be helpful for getting some better approximation algorithms in practice. But a lot of effort will be needed to get efficient algorithms that will be able to solve the k-level crossing minimization problem for $k > 2$ and $|V_i| \geq 15$ $(i = 1, \ldots, k)$ to provable optimality.

The k-level planarization problem, however, may be easier to attack. We build our hope on the fact that there is a linear time algorithm for recognizing k-level planar graphs (see [HP96] and [BN88]). Moreover, our computational results on 2-level graphs addressed in this paper support our conjecture.

Besides the application in automatic graph drawing, the 2-level planarization problem comes up in Computational Biology. In DNA mapping, small fragments of DNA have to be ordered according to the given overlap data and some additional information. Waterman and Griggs [WG86] have suggested combining the information derived by a digest mapping experiment with the information on the overlap between the DNA fragments. If the overlap data is correct, the maps can be represented as a 2-level planar graph. But, in practice, the overlap data may contain errors. Hence, Waterman and Griggs suggested solving the 2-level planarization problem (see also [VLM97]). Furthermore, the 2-level planarization problem arises in global routing for row-based VLSI layout (see [Len90,Ull84]).

Section 2 reports on previously known results of the 2-level planarization problem. One of the characterizations of 2-level planar graphs leads directly to an integer linear programming formulation for the 2-level planarization problem. In Section 3 we study the polytope associated with the set of all possible 2-level planar subgraphs of a given 2-level graph. From this we obtain new classes of inequalities that tighten the associated LP-relaxation. In order to get practical use out of these inequalities, we have to solve the "separation problem". This question will be addressed in Section 4, where we also discuss a branch-and-cut algorithm based on those results. First computational results with a branch-and-cut algorithm are presented in Section 5.

2. Characterizing 2-Level Planar Graphs

A *2-level graph* is a graph $G = (L, U, E)$ with vertex sets L and U, called lower and upper level, and an edge set E connecting a vertex in L with a vertex in U. There are no edges between two vertices in the same level. A *2-level planar graph* $G = (L, U, E)$ is a graph that can be drawn in such a way that all the vertices in L appear on a line (the lower line), the vertices in U appear on the upper line, and the edges are drawn as straight lines without crossing each other. The difference between a planar bipartite graph and a 2-level planar graph is obvious. For example, the graph shown in Figure 3 is a planar bipartite graph, but not a 2-level planar graph.

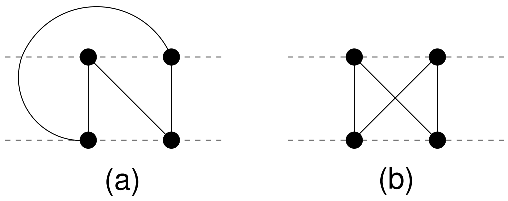


Fig. 3. (a) A planar bipartite graph that is (b) not 2-level planar

Given a 2-level graph $G = (L, U, E)$ with weights $w_e > 0$ on the edges, the *2-level planarization problem* (or *maximum 2-level planar subgraph problem*) is to extract a 2-level planar subgraph $G' = (L, U, F)$, $F \subseteq E$, of maximum weight, i.e., the sum $\sum_{e \in F} w_e$ is maximum.

To our knowledge, only the unweighted ($w_e = 1$ for all $e \in E$) 2-level planarization problem has been considered in the literature so far. It was first mentioned in [TKY77]. The authors introduced the problem in the context of graph drawing. They have given the following nice characterization of 2-level planar graphs based on forbidden subgraphs. The characterization was independently given by [EKW86].

We will call the graph shown in Figure 4(a) a *double claw*. A *caterpillar* is a connected graph $G = (V, E)$ having edges on its backbone $(v_1, v_2, \ldots, v_l)$ and single edges (v_i, w), $w \in V \setminus \{v_1, v_2, \ldots, v_l\}$ (see Figure 4(b)).

Theorem 2.1 [TKY77,EKW86]. *A 2-level graph is 2-level planar if and only if it contains no cycle and no double claw.*

Proof. A graph without any cycles is a set of trees. A tree without any double claws is a set of caterpillars. Caterpillars can be embedded on 2-levels without any crossings (see Figure 4(c)). On the other hand, a 2-level planar graph can contain neither a cycle nor a double claw. $\square$

The following alternative characterization leading to a simple linear time algorithm has been given in [TKY77].

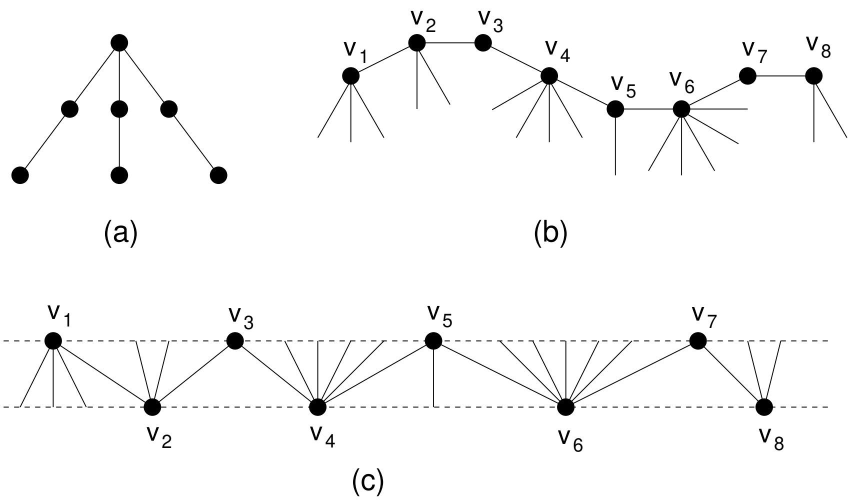


Fig. 4. (a) Double claw. (b) Caterpillar. (c) Caterpillars can be embedded on 2-levels without any crossings.

Theorem 2.2 [TKY77]. *A 2-level graph G is 2-level planar if and only if the graph G^* that is the remainder of G after deleting all vertices of degree one, is acyclic and contains no vertices of degree at least three.*

However, the 2-level planarization problem is NP-hard even for the case when each vertex in U has degree three and each vertex in L has degree two (by reduction to a Hamiltonian path problem) [EW94b]. Therefore, Eades and Whitesides suggested a heuristic based on the search for a longest path which will be used as a "backbone" of the caterpillar to be constructed.

Tomii et al. suggest an algorithm for acyclic 2-level graphs [TKY77]. The algorithm can be seen as an adaptive greedy algorithm. In each step, the edges are labelled according to some rule and the edge with the highest label will be removed. However, this algorithm does not lead to the optimal solution as shown in Figure 5. The algorithm would remove the edge $(0, 14)$ in a first step. The remaining graph still contains two edge-disjoint double-claws that have to be destroyed by removing two more edges, whereas the optimal solution would be to remove the two edges $(0, 11)$ and $(1, 14)$.

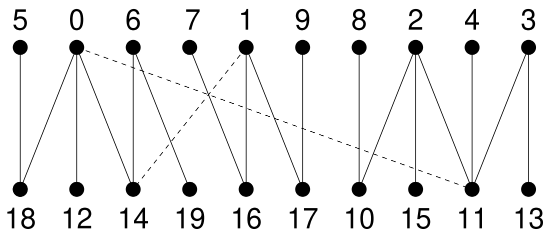


Fig. 5. An acyclic 2-level graph for which the algorithm suggested in [TKY77] leads to a nonoptimal solution

It is an open problem if the 2-level planarization problem can be solved in polynomial time for 2-level acyclic graphs. However, for double claw free graphs, the 2-level planarization problem is equivalent to the maximum forest subgraph problem that can be solved via a simple greedy algorithm.

3. Polyhedral Studies on the 2-Level Planarization Problem

Based on the characterization of 2-level planar graphs in terms of forbidden subgraphs (see Theorem 2.1), it is straightforward to derive an integer linear programming formulation for the maximum 2-level planar subgraph problem. We introduce variables x_e for all edges $e \in E$ of the given 2-level graph $G = (L, U, E)$. We use the following notation: Vectors $\bar{x}$ are column vectors, their transposed vectors $\bar{x}^T$ are row vectors. If $w^T = (w_1, w_2, \ldots, w_m)$ and $\bar{x}^T = (x_1, x_2, \ldots, x_m)$, then $w^T\bar{x} = \sum_{i=1}^{m} w_i x_i$. We use the notation $x(C) = \sum_{e \in C} x_e$ for $C \subseteq E$.

For any set $P \subseteq E$ of edges we define an incidence vector $\chi^P \in R^{|E|}$ with the i-th component $\chi^P(e_i)$ getting value 1 if $e_i \in P$, and 0 otherwise. Any vector $\bar{x}^T = (x_{e_1}, x_{e_2}, \ldots, x_{e_{|E|}})$, that is the incidence vector of a 2-level planar graph satisfies the following inequalities:

$$
\begin{array}{lll}
0 \leq x_e \leq 1, & \text{for all } e \in E, & (1) \\
x(C) \leq |C| - 1, & \text{for all cycles } C \subseteq E & (2) \\
x(T) \leq |T| - 1, & \text{for all double claws } T \subseteq E & (3) \\
x_e \text{ integral}, & \text{for all } e \in E & (4)
\end{array}
$$

and vice versa: any vector $\bar{x}^T = (x_{e_1}, x_{e_2}, \ldots, x_{e_{|E|}})$ satisfying inequalities (1), (2), (3) and (4) corresponds to a 2-level planar subgraph of G. Hence, solving the integer linear system $\{\max\ w^T\bar{x} \mid$ inequalities (1)-(4) hold for $\bar{x}\}$ will give us the solution of the maximum 2-level planar subgraph problem for a given graph $G = (L, U, E)$ with weights w_e on the edges $e \in E$.

Since solving general integer linear programs is NP-hard, we will have to drop the integrality constraints (4), which gives us a relaxation of the original integer linear programming formulation. In polyhedral combinatorics, we try to substitute the missing integrality constraints by additional inequalities.

We define the polytope $2\mathcal{LPS}(G)$ for a given 2-level graph $G = (L, U, E)$ as the convex hull over all incidence vectors of 2-level planar subgraphs of G. The vertices of this polytope correspond exactly to the 2-level planar subgraphs of G and vice versa. If we can describe the polytope $2\mathcal{LPS}(G)$ as the solution set of linear inequalities, we can optimize any given cost function over the set of all 2-level planar subgraphs of G. Of course, because of the NP-hardness of the problem we cannot expect to find such a description, but in practice a partial description may also suffice.

In an irredundant description only facet-defining inequalities are present. An inequality is said to be *facet-defining* for a polytope $\mathcal{P}$ if it is a face of maximal dimension of $\mathcal{P}$. An inequality $c^Tx \leq c_0$ is said to define a *face* of $\mathcal{P}$ if $c^Ty \leq c_0$ for all points $y \in \mathcal{P}$ and if there is at least one point y' in $\mathcal{P}$ with $c^Ty' = c_0$.

So, our task is to find facet-defining inequalities for the polytope $2\mathcal{LPS}(G)$ for a given 2-level graph G. We will first investigate the inequalities given in the integer linear programming formulation. We will see that the linear inequalities (1) and (3) are facet-defining, but only a part of the inequalities (2). But first we will determine the dimension of $2\mathcal{LPS}(G)$.

Let us consider the set $\mathcal{S}$ of all 2-level planar subgraphs of G. The set $\mathcal{S}$ is a *monotone system* (also called *independence system*), since the empty subgraph is 2-level planar and any subgraph of a 2-level planar graph is also 2-level planar. Hence, we easily get the following theorem using the theory for monotone systems.

Theorem 3.1 *Let $G = (L, U, E)$ be a graph on two levels. The dimension of $2\mathcal{LPS}(G)$, the convex hull of incidence vectors of 2-level planar subgraphs of G, is $|E|$. The trivial inequalities $x_e \geq 0$ and $x_e \leq 1$ are facet-defining for $2\mathcal{LPS}(G)$.*

Proof. It is a well known fact, that for a monotone system $(E, \mathcal{S})$ with ground set E the dimension of the associated polyhedron $P_{\mathcal{S}}$ is $|E| - (|E - \bigcup \mathcal{S}|)$ (a proof is contained, e.g., in [GP85]). Moreover, $x_e \geq 0$ defines a facet of $P_{\mathcal{S}}$ iff $e \in \bigcup \mathcal{S}$. Since every single edge is 2-level planar, we have $\bigcup \mathcal{S} = E$. Hence the dimension of the polyhedron $2\mathcal{LPS}(G)$ is $|E|$ and $x_e \geq 0$ is facet-defining for $2\mathcal{LPS}(G)$.

Let P_i be the 2-level planar graphs induced by the edge sets $\{e \cup e_i\}$ for a given edge $e \in E$ and $e_i \in E \setminus \{e\}$ for $i = 1, \ldots, |E| - 1$. The incidence vectors of the graph P induced by the edge e and the graphs P_i for $i = 1, 2, \ldots, |E| - 1$ are linearly independent and they satisfy $x_e = 1$. Hence we have shown that $x_e \leq 1$ is facet-defining for $2\mathcal{LPS}(G)$. □

Next we will see that not all of the inequalities (2) are facet-defining for $2\mathcal{LPS}(G)$.

Theorem 3.2 *Let $G = (L, U, E)$ be a 2-level graph. The cycle inequalities*

$$x(C) \leq |C| - 1$$

where $C \subseteq E$ induces a cycle in G are facet-defining for $2\mathcal{LPS}(G)$ if and only if C induces a cycle without chords in G.

Proof. Let $C \subseteq E$ be a cycle without chord in G. We will show that there are $|E|$ incidence vectors of 2-level planar subgraphs induced by the edge set F of G that are linearly independent and that satisfy $\chi^F(C) = |C| - 1$. Consider the graphs induced by the edge sets $F_i = C \setminus \{e_i\}$ for $e_i \in C$ for $i = 1, 2, \ldots, |C|$. Moreover, consider the graphs induced by the edge sets $H_j = F_1 \cup f_j$ for $f_j \in E \setminus C$, $j = 1, 2, \ldots, |E| - |C|$. Since the cycle C is chordless, adding any edge $f_j \in E \setminus C$ to F_1 still gives a 2-level planar graph, since neither a cycle nor a double claw destroying 2-level planarity can occur. All the $|E|$ incidence vectors of the 2-level planar graphs induced by F_i for $i = 1, 2, \ldots, |C|$ and H_j for $j = 1, 2, \ldots, |E| - |C|$ are linearly independent and they satisfy inequality (2) with equality. Hence the facet-defining property is shown.

Suppose now, $C = (v_1, v_2, \ldots, v_k, v_1)$ is a cycle with a chord $d = (v_h, v_l) \in E$, $d \notin C$, in G for some $h, l \in \{1, 2, \ldots, k\}$. There exists no 2-level planar graph containing the edge d and $|C| - 1$ edges of C. Hence, there exists no point x in $2\mathcal{LPS}(G)$ with $x_d = 1$, which will prove our claim. □

In the following we will see that all the double claws contained in G are present in an irredundant description of $2\mathcal{LPS}(G)$ by linear inequalities.

Theorem 3.3 *Let $G = (L, U, E)$ be a 2-level graph. The double claw inequalities*

$$x(T) \leq |T| - 1$$

where $T \subseteq E$ induces a double claw in G are facet-defining for $2\mathcal{LPS}(G)$.

Proof. Let $F_i = T \setminus e_i$ for $i = 1, \ldots, 6$. Obviously, the graphs induced by F_i are 2-level planar graphs and satisfy inequality (3) with equality. Moreover, consider the graphs induced by $H_j = T \cup f_j$ for $f_j \in E \setminus T$, $j = 1, 2, \ldots, |E| - |T|$. If H_j contains a cycle C, we can remove any edge in $C \cap T$ in order to get a 2-level planar graph induced by H'_j. In all the other cases there is always an edge we can remove from H_j such that the remaining set H'_j induces a set of caterpillars. Clearly, the incidence vectors of the 2-level planar subgraphs induced by F_i, $i = 1, 2, \ldots, 6$, and H'_j, $j = 1, 2, \ldots, |E| - |T|$ of G are linearly independent and satisfy inequality (3) with equality. □

We can tighten the LP-relaxation of (1)–(3) by introducing new inequalities that are valid and tight in the sense that they are facet-defining for $2\mathcal{LPS}(G)$. First, we generalize the double-claw inequalities to k-double claw inequalities. Considering a double-claw as a claw having three paths of length two, a *generalized k-double claw* is a claw having k paths of length two (see Figure 6(a)).

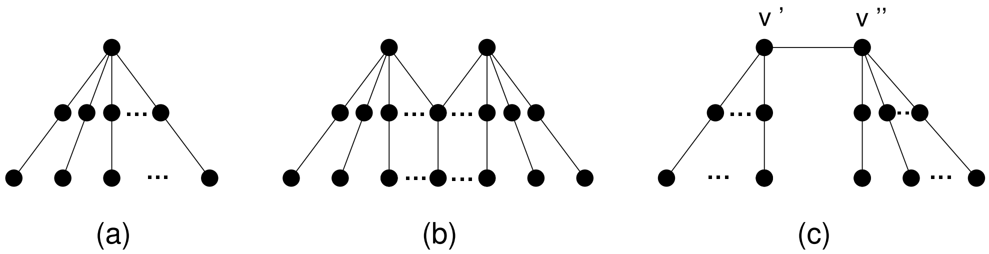


Fig. 6. (a) Generalized k-double claw (b) Combined k-double claw (c) Node-splitted k-double claw

Theorem 3.4 *Let $G = (L, U, E)$ be a 2-level graph. The generalized k-double claw inequalities*

$$x(T) \leq k + 2 \qquad (5)$$

where $T \subseteq E$ induces a k-double claw in G ($k \geq 3$) are facet-defining for $2\mathcal{LPS}(G)$.

Proof. Obviously, the inequality is valid. We denote $x(T) \leq k + 2$ by $c^T x \leq c_0$. Let us assume that there exists an inequality $a^T x \leq a_0$ with $\{x \mid c^T x = c_0\} \subseteq \{x \mid a^T x = a_0\}$. We show that $a_e = \lambda c_e$ and $a_0 = \lambda c_0$ for $\lambda > 0$. Let r be the root of the k-double claw and P denote the subgraph of $G = (V, E)$ induced by the edge set $F := \{(r, w) \mid w \in N(r) \cap V(T)\}$, where $N(r) = \{v \mid (r, v) \in E\}$ is the neighbourhood of r. Adding any two edges $e_1 \neq e_2$ in $T \setminus F$ to P gives a 2-level planar subgraph P' induced by the edge set $F' = \{F \cup e_1 \cup e_2\}$ satisfying $c^T \chi^{F'} = c_0$, hence also $a^T \chi^{F'} = a_0$. Since we can substitute e_1 and e_2 by any of the edges in $T \setminus F'$ we get $a_e = a_f$ for all $e, f \in T \setminus F$. Inserting the edge $e_3 = (w_3, u_3) \in T \setminus F'$ with $w_3 \in N(r) \cap V(T)$ in P' while removing the edge $e'_3 = (r, w_3)$ gives $a_{e_3} = a_{e'_3}$ and finally $a_e = a_f$ for all $e, f \in T$.

For any edge $e \in E \setminus T$ we can find a 2-level planar subgraph induced by the edge set F'' with $e \in F''$ satisfying $c^T \chi^{F''} = c_0$. Hence $a_e = 0$ for all $e \in E \setminus T$. □

We can prove that the *combined k-double claws* give rise to a class of facet-defining inequalities for our polytope. A *combined k-double claw* $T = (T_1, T_2)$ consists of two k_i-double claws T_i, $i = 1, 2$, that share a single edge which has an endnode of degree one (see Figure 6(b)).

Theorem 3.5 *The combined k-double claw inequalities*

$$x(T) \leq k_1 + k_2 + 3 \qquad (6)$$

where $T = (T_1, T_2) \subseteq E$ induces a combined k-double claw in G with parameters $k_1 \geq 3$ and $k_2 \geq 3$ are facet-defining for $2\mathcal{LPS}(G)$ if and only if there exist no edges (r_1, v_2) and (r_2, v_1) in G, where r_i is the root of T_i and $v_i \in T_i$ for $i = 1, 2$.

Proof. Let $e_0 = (l, u)$ denote the edge contained in both k_i-double claws $G_i = (L_i, U_i, T_i)$ and let $T = T_1 \cup \{T_2 \setminus \{e_0\}\}$. We first show validity. Let us assume that there is a 2-level planar subgraph induced by the edge set F violating inequality (6). Let $r_i \in U_i$, and $e_i = (r_i, l)$ for $i = 1, 2$. The set $T_2 \cap F$ cannot contain more than $k_2 + 2$ edges. On the other hand, the set $T_1 \setminus \{e_0, e_1\}$ induces a $(k_1 - 1)$-claw and can contain at most $k_1 - 1 + 2 = k_1 + 1$ edges. Since $T = T_1 \setminus \{e_0, e_1\} \cup T_2 \cup \{e_1\}$, we have $e_1 \in F$, otherwise, (6) is not violated. Symmetrically, we also get $e_2 \in F$. Now, consider the k_2-double claw $T_2' = T_2 \setminus \{e_0\} \cup \{e_1\} \supseteq F$. The set $T_2' \cap F$ cannot contain more than k_2 edges in addition to e_1 and e_2. We get a symmetrical argument for $F \cap T_1'$, where $T_1' = T_1 \setminus \{e_0\} \cup \{e_2\}$. Altogether F cannot contain more than $k_1 + k_2 + 3$ edges, since $F \subseteq T_1' \cup (T_2' \setminus \{e_1, e_2\}) \cup e_0$.

Now, let us assume that there is an inequality $a^T x \leq a_0$ with $\{x \mid c^T x = c_0\} \subseteq \{x \mid a^T x = a_0\}$, where $c^T x \leq c_0$ denotes inequality (6). The set $T_1'' = T_1 \setminus \{e_0, e_1\}$ induces a $(k_1 - 1)$-double claw which is not connected with T_2. Combining any 2-level planar subgraph of size $(k_1 - 1) + 2$ in T_1'' with anyone of size $k_2 + 2$ in T_2 gives a 2-level planar subgraph of size $k_1 + k_2 + 3$ in T. Because of Theorem 3.4, we have that $a_{e''} = a_{f''}$ for all $e'', f'' \in T_1''$, and $a_e = a_f$ for all $e, f \in T_2$. Symmetrical arguments for $T_2'' = T_2 \setminus \{e_0, e_1\}$ and T_1 together with the facet that $e_0 \in \{T_1 \cap T_2\}$ lead to $a_e = a_f$ for all $e, f \in T$.

We have already seen that zero-lifting is possible within the k_i-claws $(i = 1, 2)$. The critical edges to add are those connecting T_1 and T_2. Let $e = (l_2, u_1)$ be such a critical edge with $l_2 \in L_2$, $u_1 \in U_1$. If $u_1 \neq r_1$, we can show that $a_e = 0$. But in the case that $u_1 = r_1$, there exists no 2-level planar subgraph induced by the edge set F with $c^T \chi^F = c_0$ containing the edge e. Hence, in this case, inequality (6) is not facet-defining for $2\mathcal{LPS}(G)$. □

The *node-splitting operation* at vertex v in a graph G substitutes the subgraph induced by the edge set $\{(v, w) \mid w \in N(v)\}$ by a new subgraph induced by $\{(v', w') \mid w' \in W'\} \cup \{(v'', w'') \mid w'' \in W''\} \cup \{(v', v'')\}$, where $N(v)$ is the set of adjacent vertices of v in G, $W', W'' \subseteq N(v)$ with $W' \cup W'' = N(v)$ and $W' \cap W'' = \emptyset$. The vertices v' and v'' are the *duplicates* of v. The resulting graph when splitting the root node of a k-double claw is called *node-split k-double claw* with parameters k_1 and k_2 (see Figure 6(c)). The inequalities derived for those graphs contain a coefficient of two.

Theorem 3.6 *Let $G = (L, U, E)$ be a 2-level graph. The node-split k-double claw inequalities*

$$x(T) + 2x_{(r_1, r_2)} \leq k_1 + k_2 + 4 \qquad (7)$$

where $T \subseteq E$ induces a node-split k-double claw G' in G with parameters $k_1 \geq 2$ and $k_2 \geq 2$ are facet-defining for $2\mathcal{LPS}(G)$.

Proof. Let $e_0 = (r_1, r_2)$ and $T = T_1 \cup T_2 \cup \{e_0\}$, where T_1 and T_2 are the edge sets inducing the two components of $T \setminus \{e_0\}$. We first show validity. Let us assume that there exists a 2-level planar subgraph P induced by the edge set F violating the inequality (7). We know that $T_1 \cap F$ and $T_2 \cap F$ cannot contain more than $k_1 + 2$ and $k_2 + 2$ edges. If $e_0 \notin F$, the inequality cannot be violated by P. But if $e_0 \in F$, either T_1 contains at most k_1 edges, T_2 contains at most k_2 edges, or T_1 and T_2 contain at most $k_1 + 1$ and $k_2 + 1$ edges in order to ensure 2-level planarity of P. Hence, inequality (7) cannot be violated with P and validity is shown.

Now let us assume that there is an inequality $a^T x \leq a_0$ with $\{x \mid c^T x = c_0\} \subseteq \{x \mid a^T x = a_0\}$, where $c^T x \leq c_0$ denotes inequality (7). Let P be the 2-level planar subgraph induced by $k_1 + 2$ edges in T_1 and $k_2 + 2$ edges in T_2 (edge set $F = F_1 \cup F_2$, $F_i \in t_i$ for $i = 1, 2$) not containing e_0.

If $k_i \geq 3$, then any edge in F_i can be substituted by an edge $e_i \in T_i \setminus F_i$ maintaining the 2-level planarity. Hence in this case we have shown that $a_e = a_f$ for all $e, f \in T_i$. Otherwise, let us assume that $k_i = 2$. It is not hard to see that there is a 2-level planar subgraph P' containing e_0 and $k_i + 1$ edges of T_i, $i = 1, 2$ (induced by the edge set F'). Any edge in $F' \cap T_i$ can be substituted by an edge $f \in T_i$, $f \notin F'$ without destroying 2-level planarity. Hence, $a_e = a_f$ for all $e, f \in T_i$. Taking the difference of $a^T\chi^F$ and $a^T\chi^{F'}$ yields $a_{e_0} = a_{e_1} + a_{e_2}$ for $e_i \in T_i$, $i = 1, 2$. Moreover, there is a 2-level planar subgraph induced by e_0, $k_1 + 2$ edges in T_1 and k_2 edges in T_2. Hence, we have shown that $a_{e_1} = a_{e_2}$ for all $e_1 \in T_1$, $e_2 \in T_2$ and $a_{e_0} = 2a_e$ for all $e \in T_1 \cup T_2$ if $k_1, k_2 \geq 2$. Hence, inequality (7) is facet-defining for $2\mathcal{LPS}(G')$.

It remains to show that $a_e = 0$ for all edges $e \in E \setminus T$ if $G' = (V', T)$ with $T \subseteq E$ and $V' \subseteq V$. Since zero-lifting is possible for double claw inequalities, we can restrict our attention to edges $e = (v, w)$ with $v \in G_1$ and $w \in G_2$, where G_1 and G_2 denote the graphs induced by the edge sets T_1 and T_2. For the two possible cases, we can always find a 2-level planar graph containing the edge e_0, an additional edge $e \notin T$, and $k_1 + k_2 + 4$ edges of T in total. □

In the case that the given 2-level graph contains no double claw, the 2-level planarization problem is equivalent to the maximum forest problem. It is well known that this problem can be solved in polynomial time by a simple greedy algorithm. Moreover, the structure of the associated weighted forest polytope has been well studied (see, e.g., [Edm70]). The inequalities of the weighted forest polytope are still valid for our polytope $2\mathcal{LPS}(G)$, even if the graph G contains double claws. And, as we will see in our computational experiments, they are quite useful in practice.

Lemma 3.7 *Let $G = (L, U, E)$ be a 2-level graph. The forest inequalities*

$$x(F) \leq V(F) - 1 \qquad (8)$$

where $F \subseteq E$ and $V(F)$ is the number of vertices contained in the subgraph induced by F are valid for $2\mathcal{LPS}(G)$.

The special case when F induces a complete bipartite subgraph of a 2-level graph G leads to forest inequalities that are facet-defining for $2\mathcal{LPS}(G)$. We will call them *crown inequalities.* For $|L'| = |U'| = 2$, the crown inequalities are equivalent to the cycle inequalities for $|C| = 4$. Hence, the crown inequalities are a generalization of this cycle inequality.

Theorem 3.8 *Let $G = (L, U, E)$ be a 2-level graph containing a complete bipartite subgraph $G' = (L', U', E')$, $E' \subseteq E$. The crown inequalities*

$$x(E') \leq |L'| + |U'| - 1 \qquad (9)$$

with $|L'| \geq 2$ and $|U'| \geq 3$ are facet-defining for $2\mathcal{LPS}(G)$.

Proof. The validity follows from Lemma 3.8. Let us assume that there is an inequality $a^T x \leq a_0$ with $\{x \mid c^T x = c_0\} \subseteq \{x \mid a^T x = a_0\}$, where $c^T x \leq c_0$ denotes inequality (9). Let $U' = \{u_1, \ldots, u_{|U'|}\}$, $L' = \{l_1, \ldots, l_{|L'|}\}$, $u \in U'$, and $l \in L'$. The edge set $F = \{(u, l_i) \mid l_i \in L', i = 1, \ldots, |L'|\} \cup \{(l, u_i) \mid u_i \in U' \setminus \{u\}, i = 1, \ldots, |L'|\}$ induces a 2-level planar subgraph satisfying $c^T\chi^F = c_0$, hence also $a^T\chi^F = c_0$. Removing the edge (l, u) from F and adding the edge $\{u_i, l_i\}$, where $u_i \neq u$, $l_i \neq l$, $u_i \in U'$, and $l_i \in L'$, will still leave a 2-level planar graph. Hence, we have $a_{(u,l)} = a_{(u_i,l_i)}$, and since we can choose u and l free among the vertices, we have $a_e = a_f$ for all $e \in E'$. On the other hand, it is always possible to add an extra edge $(l_i, v) \in E \setminus E'$ or $(u_i, v) \in E \setminus E'$ to F without loosing 2-level planarity. Hence, $a_e = 0$ for all edges in $E \setminus E'$, and the theorem is shown. $\square$

In the next section we show how the theoretical results obtained in this section can be used in an algorithm for solving practical instances of the 2-level planarization problem.

4. Separation Problems and a Branch-and-Cut Algorithm

According to results of Grötschel, Lovász, Schrijver [GLS81], Karp and Papadimitriou [KP80], and Padberg and Rao [PR81], we can optimize a linear objective function over a polytope in polynomial time if and only if we can solve the *separation problem* in polynomial time, i.e., given a vector $\bar{x} \in \mathbf{Q}^{|E|}$, decide whether $\bar{x} \in \mathcal{P}$, and, if $\bar{x} \notin \mathcal{P}$, find a vector $d \in \mathbf{Q}^{|E|}$ and a scalar $d_0 \in \mathbf{Q}$ such that the inequality $d^T\bar{x} \leq d_0$ is valid with respect to $\mathcal{P}$ and $d^T\bar{x} > d_0$.

We will see that we can solve the separation problem restricted to the class of inequalities (2) in polynomial time.

Theorem 4.1 *For the cycle inequalities (2) the separation problem can be solved in polynomial time by computing at most $|E|$ shortest path problems.*

Proof. Given a point $\bar{x} \in \mathbf{Q}^{|E|}$, we are searching for a cycle $C \subseteq E$ with $\bar{x}(C) > |C| - 1$. Let us write the inequality in a different way: $|C| - \bar{x}(C) < 1$ which corresponds to $\sum_{e \in C}(1 - x_e) < 1$. For any fixed $e_0 \in E$ we solve a shortest path problem on the graph given by $G - \{e_0\}$ with edge costs $z_e = 1 - x_e$ for $e \in E \setminus \{e_0\}$. Let W be the weight of the shortest path. We only have to test if $W + z_{e_0}$ is less than one. In this case we have found a cycle C violating inequality $\bar{x}(C) > |C| - 1$ of $\bar{x}$. If for no $e_0 \in E$ a violated inequality has been found, we have a proof that all the inequalities of type (2) are satisfied at $\bar{x}$. Hence we have solved the separation problem for (2) in polynomial time. $\square$

The separation problem can also be solved for the double claw inequalities (3) and their generalization to k-double claw inequalities for fixed k.

Theorem 4.2 *The separation problem for the double claw inequalities and the generalized k-double claw inequalities can be solved in polynomial time for fixed k by computing a series of maximum bipartite matching problems on subgraphs of G.*

Proof. Obviously, all k-double claws for fixed k can simply be enumerated in polynomial time. Faster is the following algorithm that is described for the generalized k-double claw inequalities when k is fixed. Given a point $\bar{x} \in \mathbf{Q}^{|E|}$, we are searching for a k-double claw $T \subseteq E$ with $\bar{x}(T) > k + 2$. For any vertex r and any set of k adjacent vertices $w_1, w_2, \ldots, w_k \in N(r)$ let $W := \sum_{i=1}^{k} x_{(r,w_i)}$. We compute a maximum bipartite matching M between the vertex sets $\{w_1, w_2, \ldots, w_k\}$ and $\{N(w_1) \cup N(w_2) \cdots \cup N(w_k)\} \setminus \{r, w_1, w_2, \ldots, w_k\}$. If $W + \sum_{e \in M} x_e \leq k + 2$, then no k-double claw inequality rooted at r with neighbours $w_1, w_2, \ldots, w_k$ is violated. Otherwise, M together with $\{(r, w_i) \mid i = 1, 2, \ldots, k\}$ induces a set T for which the inequality $x(T) \leq k + 2$ is violated (T may be only a part of a k-double claw in case that M contains less than k edges). $\square$

Padberg and Wolsey have already shown that the separation problem for the inequalities occuring in the weighted forest polytope can be solved in polynomial time [PW83].

Theorem 4.3 [PW83]. *The separation problem of the forest inequalities (8) can be solved by computing a minimum cut in a capacitated network G^* constructed from $G = (V, E)$. G^* contains $2(|V| + |E|)$ arcs and $|V| + 2$ vertices.*

We suggest a branch-and-cut algorithm for solving practical instances of the maximum 2-level planar subgraph problem. We will see that our branch-and-cut algorithm is able to find nearly optimal solutions for moderately sized problem instances in reasonable computation time.

We implemented a branch-and-cut algorithm based on the system ABACUS [JRT97,Thi95] using the separation routines mentioned above. In our algorithm we start with the linear system $\{\max w^T \bar{x} \mid x_e \geq 0, x_e \leq 1 \text{ for all } e \in E\}$. Let x^* denote the optimal solution of the LP-system. We solve the separation problem for inequalities (2), (3), (5) and (8) using Theorems 4.1-4.3. We add all the found inequalities to our system and optimize again. The algorithm stops if no violated inequalities of the above mentioned types are found. If x^* is integer, we know that x^* is the incidence vector of a 2-level planar graph. In this case we have found the optimal solution of the 2-level planarization problem. Otherwise, x^* gives us an upper bound to the value of a maximum 2-level planar subgraph of the given instance G. In this case, we branch by setting the value of a fractional variable to zero or one, and try to solve the subproblems like the root node.

In addition, we try to find good solutions to the problem. After each optimization process, we may get new solutions x^* to the problem, most of which are fractional. Fractional solutions x^* may give us a hint about good solutions to the problem. We try to use this information in our heuristics that we apply in each iteration.

5. Computational Results

For our experiments we used the branch-and-cut algorithm described above. The algorithm stops if either the optimal solution is found or no violated cycle, double claw, generalized double claw or forest inequality can be detected. Moreover, we put a time limit of five minutes (=300 seconds) to our program. In any case, the program gives a 2-level planar subgraph together with an upper bound of the optimal solution.

Table 1. Computational Results for graphs on 20 vertices per level

$\lvert V_i \rvert$	$\lvert E \rvert$	Gar	Time	Cycles	2Claw	kClaw	Forest
20	20	0.00	0.01	0.18	0.52	0.00	0.00
20	25	0.00	0.03	0.81	1.65	0.15	0.00
20	30	0.00	0.17	2.36	4.60	0.87	0.05
20	35	0.00	0.52	5.04	16.08	4.52	0.16
20	40	0.00	5.81	11.39	55.69	22.20	1.09
20	45	0.03	25.56	19.38	116.73	92.80	3.93
20	50	0.67	100.38	32.62	185.72	234.21	9.13
20	55	0.53	81.17	25.14	194.21	297.81	8.11
20	60	0.37	56.04	23.69	167.63	277.66	5.78
20	65	0.32	54.25	23.22	188.07	320.39	5.66
20	70	0.13	25.97	18.47	103.06	159.25	1.79
20	75	0.13	21.69	18.33	76.45	139.69	0.92
20	80	0.03	12.37	17.01	61.09	67.38	0.11
20	85	0.10	20.28	18.16	75.91	111.53	0.34
20	90	0.02	7.47	13.46	29.05	34.66	0.09
20	95	0.00	3.94	11.66	15.81	16.77	0.12
20	100	0.00	4.08	10.83	13.90	15.72	0.02

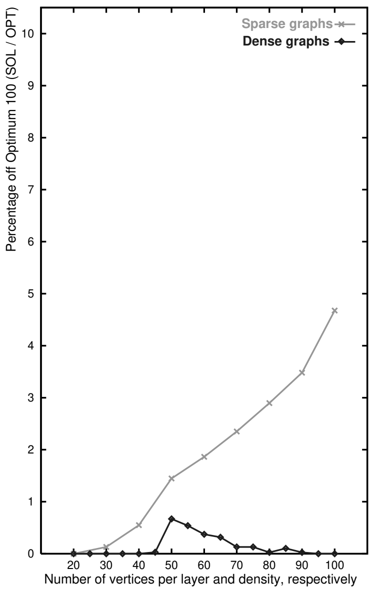


Fig. 7. Results for 100 instances

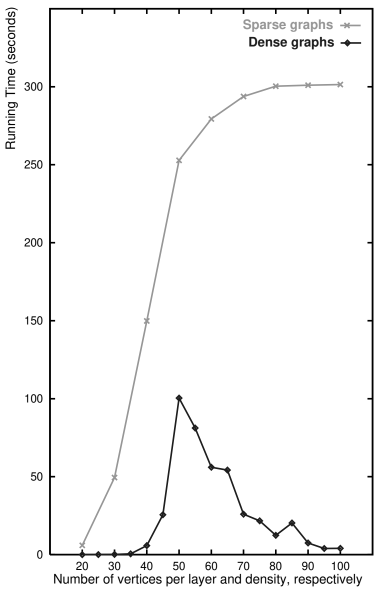


Fig. **8.** Average running times

Table 1 shows computational results for 100 instances of 2-level graphs with 20 vertices at each level with increasing density. The columns show the number of vertices per level, the number of edges, and the average guarantee of the solution value, i.e., if Sol denotes the number of edges remaining in a found 2-level planar subgraph and UpBound denotes the value determined by the linear programming relaxation, then the solution guarantee Gar is $(\frac{\text{UpBound}-\text{Sol}}{\text{UpBound}})\times 100\%$. Column 4 shows the time on a SUN Ultra 2/2x200 in seconds. Columns 5 to 8 show the average number of found violated cycle, double claw, generalized k-double claw and forest inequalities.

The average quality of the solution value is visualized in Figure 7. The results are surprisingly good. On the average, the solution we found is very close (below 0.7% on average) to the optimal one. Figure 8 shows the average running time in seconds.

Furthermore, we ran 100 instances on a series of sparse graphs. The results are promising also for these cases (see Table 2). Our solution is at most 5% away from the optimal solution. Figure 7 and Figure 8 visualize the average guarantee of the solution and the running times also for sparse graphs. Practical instances in graph drawing typically have up to 40 vertices per layer. Hence, regarding the running times, our algorithm is competitive with the classical heuristics used for crossing minimization in graph drawing.

Table 2. Computational Results for sparse graphs

$\lvert V_i\rvert$	$\lvert E\rvert$	Gar	Time	Cycles	2Claw	kClaw	Forest
20	40	0.00	5.97	11.39	55.69	22.20	1.09
30	60	0.13	49.41	15.17	136.79	83.24	2.25
40	80	0.55	149.85	17.46	227.30	137.60	2.99
50	100	1.45	252.81	20.26	309.59	178.53	3.51
60	120	1.86	279.38	22.63	395.82	230.21	1.81
70	140	2.35	293.73	25.65	441.42	222.76	1.64
80	160	2.90	300.37	28.68	534.30	248.17	1.02
90	180	3.48	300.97	31.18	589.81	241.98	0.33
100	200	4.67	300.39	35.26	687.14	237.27	0.31

Consider the graph shown in Figure 2. Our branch-and-cut algorithm solved the 2-level planarization problem for the given instance provably optimal within 0.01 seconds. During the run 5 violated cycle constraints were found, 10 double claw inequalities, 1 generalized k-double claw inequalities and no forest inequality.

Acknowledgements

I started to get interested in the 2-level planarization problem when Peter Eades visited the Max-Planck-Institut für Informatik (Saarbrücken) in June 1995. Thanks to Peter for initiating my interest in this beautiful problem. I am very grateful to René Weiskircher for providing the implementation of the branch-and-cut algorithm and for running all the computational experiments.

References

[BN88] G. Di Battista and E. Nardelli: Hierarchies and Planarity Theory. *IEEE Transactions on Systems, Man and Cybernetics* 18 (1988) 1035 – 1046

[Car80] Carpano, M.J.: Automatic display of hierarchized graphs for computer aided decision analysis. *IEEE Trans. Syst. Man and Cybern.,* SMC-10, no. 11 (1980) 705 – 715

[Edm70] Edmonds, J.: Submodular functions, matroids and certain polyhedra. *in: Combinatorial Structures and Their Applications,* Gordon and Breach, London (1970) 69 – 87

[Dre94] Dresbach, S.: A New Heuristic Layout Algorithm for DAGs. *in: Derigs, Bachem & Drexl (eds.), Operations Research Proceedings 1994,* Springer Verlag, Berlin (1994) 121 – 126

[EK86] Eades, P., and D. Kelly: Heuristics for Reducing Crossings in 2-Layered Networks. *Ars Combinatoria* 21-A (1986) 89 – 98

[EKW86] Eades, P., B.D. McKay, and N.C. Wormald: On an edge crossing problem. *Proc. 9th Austral. Comp. Sci. Conf.,* Australian National University (1986) 327 – 334

[EW94a] Eades, P., and N.C. Wormald: Edge crossings in Drawings of Bipartite Graphs. *Algorithmica* 10 (1994) 379 – 403

[EW94b] Eades, P. and S. Whitesides: Drawing graphs in two layers. *Theoretical Computer Science* 131 (1994) 361 – 374

[Fuk96] Fukuda, A.: Face Lattices. Personal Communication (1996)

[GLS81] Grötschel, M., L. Lovász, and A. Schrijver: The Ellipsoid Method and its Consequences in Combinatorial Optimization. *Combinatorica* 1 (1981) 169 – 197

[GP85] Grötschel, M. and M.W. Padberg: Polyhedral theory. *In E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan, and D.B. Shmoys (eds.), The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization,* Wiley-Interscience (1985)

[HP96] Heath, L.S. and S.V. Pemmaraju: Recognizing Leveled-Planar Dags in Linear Time. *Lect. Notes Comp. Sci. 1027, F. Brandenburg (ed.), Proc. Graph Drawing '95,* (1996) 300 – 311

[JM93a] Jünger, M. und P. Mutzel: Solving the maximum planar subgraph problem by branch and cut. *In: L.A. Wolsey and G. Rinaldi (eds.),* Proceedings of the 3rd IPCO Conference, Erice (1993) 479 – 492

[JM93b] Jünger, M. und P. Mutzel: Maximum planar subgraphs and nice embeddings: Practical layout tools. *Algorithmica* 16, No. 1, Special Issue on Graph Drawing, G. Di Battista and R. Tamassia (eds.), (1996) 33 – 59

[JM96] Jünger, M. und P. Mutzel: Exact and Heuristic Algorithms for 2-Layer Straightline Crossing Minimization. *Lecture Notes in Comp. Sci. 1027, in: F. Brandenburg (ed.), Proc. on Graph Drawing '95,* Passau (1996) 337 – 348

[JRT97] Jünger, M., G. Reinelt, and S. Thienel: ABACUS - A Branch And CUt System. *Technical Report, Institut für Informatik, Universität zu Köln, in preparation,* (1997)

[KP80] Karp, R.M. and C.H. Papadimitriou: On Linear Characterizations of Combinatorial Optimization Problems. *Proc. 21st Ann. Symp. Found. of Comp. Sci. IEEE* (1980) 1 – 9

[Len90] Lengauer, T.: Combinatorial algorithms for integrated circuit layout. John Wiley & Sons, Chichester, UK (1990)

[Mäk90] Mäkinen, E.: Experiments on Drawing 2-Level Hierarchical Graphs. *Intern. J. Computer Math.* 37 (1990) 129 – 135

[Mut94] Mutzel, P.: The Maximum Planar Subgraph Problem. Dissertation, Universität zu Köln (1994)

[PR81] Padberg, M.W. and M.R. Rao: The Russian Method for Linear Inequalities III: Bounded Integer Programming. GBA Working Paper 81 – 39, New York University (1981)

[PW83] Padberg, M.W. and L.A. Wolsey: Trees and Cuts. *Annals of Discrete Mathematics* 17 (1983) 511 – 517

[STT81] Sugiyama, K., S. Tagawa, and M. Toda: Methods for Visual Understanding of Hierarchical System Structures. *IEEE Trans. Syst. Man, Cybern.,* SMC-11 (1981) 109 – 125

[Thi95] Thienel, S: ABACUS - A Branch And CUt System. Dissertation, Univ. zu Köln, 1995

[TKY77] Tomii, N., Y. Kambayashi, and Y. Shuzo: On Planarization Algorithms of 2-Level Graphs. Papers of tech. group on electronic comp., IECEJ, EC77-38 (1977) 1 – 12

[Ull84] Ullman, J.D.: Computational Aspects of VLSI. Comp. Sci. Press, Rockville, MD (1984)

[VLM97] Vingron, M., H.-P. Lenhof, and P. Mutzel: Computational Molecular Biology. *In: Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, S. Martello (eds.),* Chapter 23, to appear (1997)

[VML96] Valls, V., R. Marti, and P. Lino: A Branch and Bound Algorithm for Minimizing the Number of Crossing Arcs in Bipartite Graphs. *J. of Operational Res.* 90 (1996) 303 – 319

[War77] Warfield, J.N.: Crossing Theory and Hierarchy Mapping. *IEEE Trans. Syst. Man, Cybern.,* SMC-7 (1977) 505 – 523

[WG86] Waterman, M.S. and J. R. Griggs: Interval graphs and maps of DNA. *Bull. Math. Biology* 48, no. 2 (1986) 189 – 195

I N F O R M A T I K

Below you find a list of the most recent technical reports of the research group *Efficient Algorithms* at the Max-Planck-Institut für Informatik. They are available by anonymous ftp from our ftp server `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Birgit Hofmann
Im Stadtwald
D-66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-97-1-008	P. Mutzel	An Alternative Method to Crossing Minimization on Hierarchical Graphs
MPI-I-97-1-007	F.J. Brandenburg, M. Jünger, P. Mutzel	Algorithmen zum automatischen Zeichnen von Graphen
MPI-I-97-1-006	W.H. Cunningham, Y. Wang	Restricted 2-Factor Polytopes
MPI-I-97-1-005	Y. Wang	Bicriteria Job Sequencing with Release Dates
MPI-I-97-1-004	H. Lenhof	New Contact Measures for the Protein Docking Problem
MPI-I-97-1-003	B. Jung, H. Lenhof, P. Müller, C. Rüb	Parallel Algorithms for MD-Simulations of Synthetic Polymers
MPI-I-97-1-002	N. Garg	Approximating Sparsest Cuts
MPI-I-97-1-001	J.F. Sibeyn, M. Kaufmann	BSP-Like External-Memory Computation
MPI-I-96-1-033	M. Seel	A Runtime Test of Integer Arithmetic and Linear Algebra in LEDA
MPI-I-96-1-032	U. Finkler, K. Mehlhorn	Runtime Prediction of Real Programs on Real Machines
MPI-I-96-1-031	L. Gasieniec, J. Jansson, A. Lingas, A. Östlin	On the complexity of computing evolutionary trees
MPI-I-96-1-030	L. Gasieniec, P. Indyk, P. Krysta	External Inverse Pattern Matching
MPI-I-96-1-029	P.G. Bradford, K. Reinert	Lower Bounds for Row Minima Searching
MPI-I-96-1-028	K. Reinert, H. Lenhof, P. Mutzel, K. Mehlhorn, J.D. Kececioglou	A Branch-and-Cut Algorithm for Multiple Sequence Alignment
MPI-I-96-1-027	T. Christof, M. Jünger, J.D. Kececioglou, P. Mutzel, G. Reinelt	A branch-and-cut approach to physical mapping with end-probes
MPI-I-96-1-026	S. Albers, J. Westbrook	A Survey of Self-Organizing Data Structures
MPI-I-96-1-025	M. Jünger, P. Mutzel	2-Layer straigthline crossing minimization: Performance of Exact and Heuristic Algorithms
MPI-I-96-1-024	K. Diks, T. Hagerup	More geneal parallel tree contraction: Register allocation and broadcasting in a tree