Solving the Maximum Weight Planar Subgraph Problem by Branch and Cut

by

Michael Jünger and Petra Mutzel

Institut für Informatik, Universität zu Köln
Pohligstraße 1, 5000 Köln 51, Germany
mjuenger@informatik.uni-koeln.de
mutzel@informatik.uni-koeln.de

In this paper we investigate the problem of identifying a planar subgraph of maximum weight of a given edge weighted graph. In the theoretical part of the paper, the polytope of all planar subgraphs of a graph G is defined and studied. All subgraphs of a graph G, which are subdivisions of K_5 or $K_{3,3}$, turn out to define facets of this polytope.

We also present computational experience with a branch and cut algorithm for the above problem. Our approach is based on an algorithm which searches for forbidden substructures in a graph that contains a subdivision of K_5 or $K_{3,3}$. These structures give us inequalities which are used as cutting planes.

Key words: Maximum planar subgraph, planar subgraph polytope, facets, branch and cut

1. Introduction

A graph $G = (V, E)$ is said to be planar, if it can be drawn on the plane such that no two edges intersect geometrically except at a vertex at which they are both incident. According to Kuratowski's Theorem, planar graphs are exactly the graphs that contain no subdivisions of K_5 or $K_{3,3}$. Given a nonplanar weighted graph with edge weights w_e for $e \in E$ we want delete a set of edges F to obtain a planar subgraph $G' = (V, E \setminus F)$ such that the sum of all edge weights $\sum_{e \in E \setminus F} c_e$ of G' is maximum. In the unweighted case, where $c_e = 1$ for all edges $e \in E$, the problem consists of finding the minimum number of edges whose deletion from a nonplanar graph gives a planar subgraph.

In either case the problem is NP-hard [GJ79]. The problem can be solved in polynomial time if G is already planar, since planarity testing can be done in linear time [HT74]. If $G = K_n$, the complete graph on n nodes, or $G = K_{m,n}$, the complete bipartite graph on $n + m$ nodes, it is easy to construct a solution which contains $3n - 6$, resp. $2n - 4$ edges, and so the unweighted problem is solved in linear time.

A related problem to the unweighted maximum planar subgraph problem is the maximal planar subgraph problem. It consists of finding a planar subgraph $G' = (V', E')$ such that for all edges $e \in E \setminus E'$ the addition of e to G' destroys the planarity of G'. Recently Cai, Han and Tarjan [CHT91] described an $O(|E| \log |V|)$ maximal planarization algorithm based on the Hopcroft-Tarjan planarity testing algorithm, and Kant [K92] generalized the maximal planarization algorithm of Jayakumar et al. [JTS89] for a special class of graphs, to an $O(|V|^2)$ planarization algorithm based on PQ-trees of Booth and Lueker [BL76].

For the weighted maximum planar subgraph problem Foulds et al. described some heuristics which deal with complete graphs [FR78, EFG82]. They applied their heuristics to determine good layouts of manufacturing facilities, whose modelling yields very dense graphs. Branch and bound algorithms have been proposed, but they only have a chance on small dense graphs [FR76]. Recently, Goldschmidt and Takvorian presented a two-phase heuristic for solving the unweighted maximum planar subgraph problem [GT92]. They also tried to find a triangulated planar subgraph, but if the density of the graph is not high enough, the heuristic fails.

In automatic graph drawing a given graph has to be layed-out in the plane, possibly according to a number of topological and aesthetic constraints. In [TBB88] Tamassia et al. describe their graphtheoretic approach. In the initial phase of the process, an unweighted maximum planar subgraph of the input graph is determined, which is then used as the basis for the layout of the original graph. The graphs occuring in such applications are relatively sparse, so that the above heuristic fails.

We attack the problem with a branch and cut technique. This approach gives us quite good and in many cases provably optimal solutions for sparse graphs and very dense graphs.

In section 2, we define the planar subgraph polytope $\mathcal{PLS}(G)$. Some basic facts about this polytope are given. Among others all the minimal nonplanar subgraphs of a graph G, which are exactly the subdivisions of K_5 and $K_{3,3}$ contained in G, turn out to define facets of $\mathcal{PLS}(G)$. In addition, some operations like lifting, edge splitting and edge contraction are examined. In section 3, we present the algorithm. The computational results are given in section 4.

2. The Planar Subgraph Polytope

Suppose a graph $G = (V, E)$ with edge weights w_e for all $e \in E$ is given. Let $\mathcal{P_G}$ be the set of all planar subgraphs of G. For each planar subgraph $P = (V', F) \in \mathcal{P_G}$, we define its incidence vector $\chi^P \in \mathrm{R}^E$ by setting $\chi^P_e = 1$ if $e \in F$ and $\chi^P_e = 0$ if $e \notin F$.

The polytope $\mathcal{PLS}(G)$ of G is defined as the convex hull over all incidence vectors of planar subgraphs of G

$$\mathcal{PLS}(G) := \mathrm{conv}\{\chi^P \in \mathrm{R}^E \mid P \in \mathcal{P}_G\}.$$

The problem of finding a planar subgraph P of G with weight $w(P)$ as large as possible can be written as the linear program

$$\max\{w^T x \mid x \in \mathcal{PLS}(G)\},$$

since the vertices of the polytope $\mathcal{PLS}(G)$ are exactly the incidence vectors of the planar subgraphs of G. In order to apply linear programming techniques to solve this LP one has to represent $\mathcal{PLS}(G)$ as the solution of an inequality system. In the following we give a partial description of the facial structure of $\mathcal{PLS}(G)$.

The set of all planar subgraphs $\mathcal{P}_G$ of G is an independence system, since every subgraph of a planar graph is planar.

Lemma 1 *The dimension of the $\mathcal{PLS}$-polytope of $G = (V, E)$ is $|E|$, so it is full dimensional. For all edges $e \in E$ the inequalities $x_e \geq 0$ and $x_e \leq 1$ define facets of $\mathcal{PLS}(G)$.*

Proof. The first part follows directly from the properties of independence systems. It is also easy to see that the inequality $x_e \leq 1$ is facet-defining by directly choosing the $|E| - 1$ edge sets $F_i = \{e \cup e_i\}$ for all $e_i \in E \setminus \{e\}$. Together with $F = \{e\}$ their incidence vectors give linear independent incidence vectors of planar subgraphs, which satisfy the inequality $x_e \leq 1$ with equality. □

A **minimal nonplanar graph** is a nonplanar graph for which the removal of an arbitrary edge yields a planar graph. Minimal nonplanar graphs are the circuits in the independence system $\mathcal{P}_G$.

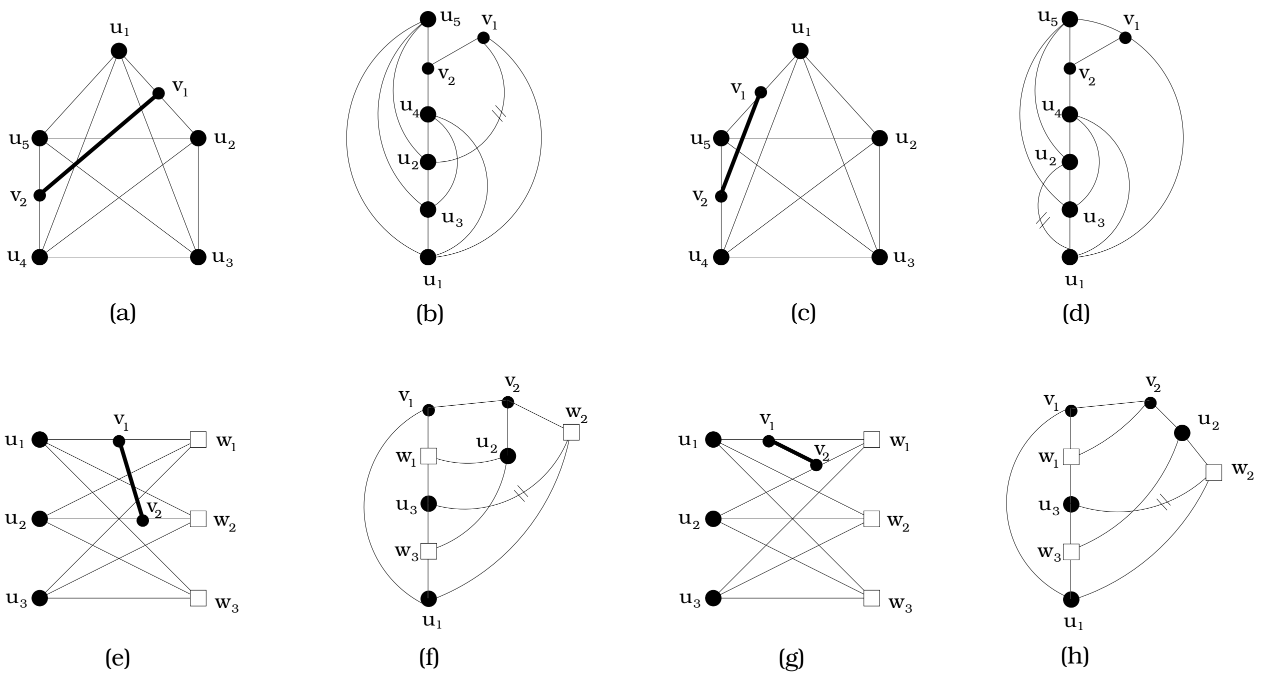

Figure 1

Since by Kuratowski's Theorem every nonplanar graph contains a subdivision of K_5 or $K_{3,3}$, one can easily observe that the minimal nonplanar graphs are exactly the subdivisions of K_5 or $K_{3,3}$. In the following we will examine their properties.

Lemma 2 *Given a minimal nonplanar subgraph $G[F]$ of a nonplanar graph $G = (V, E)$, $F \subseteq E$, an arbitrary edge $f \in F$ and an edge $e \in E \setminus F$ with one endnode not in $V(F)$. Then the graph $G[F']$ induced by $F' = F \setminus \{f\} \cup \{e\}$ is planar.*

Theorem 1 *Given a minimal nonplanar subgraph $G[F]$ of a nonplanar graph $G = (V, E)$ and an edge $e \in E \setminus F$. Then there exists an edge $f \in F$ such that the graph $G[F']$ induced by $F' = F \setminus \{f\} \cup \{e\}$ is planar.*

Proof. Consider first a minimal nonplanar subgraph which is a subdivision of K_5. We denote the nodes which are no subdivision nodes by $u_1, \ldots, u_5$. Let us assume that we add the edge (v_1, v_2) with endnodes v_1 on the path from u_1 to u_2 and v_2 on the path from u_4 to u_5 (see Figure 1(a)). An embedding of this graph is given in Figure 1(b). Clearly, removing one edge on the path from u_2 to v_1 will lead to a planar graph. If the edge (v_1, v_2) joins two adjacent paths, for example v_1 lies now on the path from u_1 to u_5 (see Figure 1(c)), Then again the deletion of one edge on the path from u_1 to u_2 gives a planar graph (see Figure 1(d)). The case in which not both nodes v_1 and v_2 are subdivision nodes, can be obtained from the above via a suitable contraction. All other cases are symmetric to the above.

Now consider a graph which is a subdivision of $K_{3,3}$. Again denote the nodes which are no subdivision nodes by $u_1, .., u_3, w_1, .., w_3$. Let us assume that we add the edge (v_1, v_2) with endnodes v_1 on the path from u_1 to w_1 and v_1 on the path from u_2 to w_2 (see Figure 1(e)). By considering the embedding of the graph given in Figure 1(f) it is obvious that the graph will be planar, if one edge on the path from w_2 to u_3 is deleted. The case in which the edge (v_1, v_2) joins two adjacent paths is treated in Figure 1(g)-(h). □

This theorem leads to the main theoretical result of this paper.

Theorem 2 *For all minimal nonplanar subgraphs $G' = (V', F)$ of $G = (V, E)$ the inequality $x(F) \leq |F| - 1$ defines a facet of $\mathcal{PLS}(G)$.*

Proof. Take the $|F|$ edge sets $F \setminus \{f\}$ for all edges $f \in F$. Consider an edge $e \in E \setminus F$, add it to F and delete an edge $f \in F$ such that the subgraph induced by $F' = F \setminus \{f\} \cup \{e\}$ is planar, which is possible due to Theorem 1. Do this for all edges $e \in E \setminus F$. All of the $|E|$ edge sets satisfy the inequality $x(F) \leq |F| - 1$ at equality and the graphs induced by them are planar. Consider the corresponding matrix A whose rows are the incidence vectors of these edge sets. If in the rows of A the edges in F appear before the remaining ones, it is obvious that A has rank $|E|$. Thus these $|E|$ planar graphs are all linearly independent and the theorem follows. $\square$

Corollary 1 *Let K_5 (resp. $K_{3,3}$) be contained in $G = (V, E)$. Then $x(K_5) \leq 9$ (resp. $x(K_{3,3}) \leq 8$) defines a facet of $\mathcal{PLS}(G)$.*

The facet-defining property of K_5 and $K_{3,3}$ is not very astonishing. There is one other class of inequalities, which plays an important role in the theory of planar graphs, that is Euler's formula for the relationships of vertices, edges and faces in a plane connected graph.

Lemma 3 **(Euler inequalities)** *For $G = (V, E)$ and $V' \subseteq V$ let $E' := E[V']$ and $G' = (V', E')$. Then the inequality $x(E') \leq 3|V'| - 6$ is valid for $\mathcal{PLS}(G)$. If G' is bipartite, the inequality intensifies to $x(E') \leq 2|V'| - 4$.*

Moreover, if the graph $G = (V, E)$ is dense, the above inequality may yield a facet, like it is the case for $G = K_n$, resp. $G = K_{m,n}$.

Theorem 3 *For the complete graph on n nodes $G = K_n$ the inequality $x(E) \leq 3|V| - 6$ defines a facet for $\mathcal{PLS}(G)$ for $n \geq 5$. If G is the complete bipartite graph $K_{m,n}$, then the inequality $x(E) \leq 2|V| - 4$ defines a facet for $\mathcal{PLS}(G)$ for $m, n \geq 3$.*

Proof. We show the theorem for the complete bipartite graph $K_{m,n}$. The proof for complete graphs is similar, but not as complicated and much shorter. Let us assume that $G = (U, W, E)$ with $|U| = m \geq n = |W| \geq 3$. For notational convenience we denote $x(E) \leq 2(|U| + |W|) - 4$ by $a^T x \leq a_0$. Suppose now that $c^T x \leq c_0$ is a valid inequality for $\mathcal{PLS}(G)$ satisfying $\{x \in \mathcal{PLS}(G) \mid a^T x = a_0\} \subseteq \{x \in \mathcal{PLS}(G) \mid c^T x = c_0\}$. We show that for some $\alpha \geq 0$ we have $c^T = \alpha a^T$ and $c_0 = \alpha a_0$. Consider the plane graph P shown in Figure 2(a), where $U = \{u_1, \ldots, u_m\}$ and $W = \{w_1, \ldots, w_n\}$. P has exactly $(3n - 2) + (n - 2) + 2(m - n) = 2(m + n) - 4$ edges, thus its incidence vector χ^P satisfies $a^T \chi^P = a_0$ and hence also $c^T \chi^P = c_0$.

Let us construct a new graph P_2^i by deleting the edge (w_{i+1}, u_{i+1}) and adding the edge (w_i, u_{i+2}). P_2^i is still planar and satisfies $a^T \chi^{P_2^i} = a_0$ and hence $c^T \chi^{P_2^i} = c_0$. This implies $0 = c_0 - c_0 = c^T \chi^P - c^T \chi^{P_2^i} = c_{w_{i+1} u_{i+1}} - c_{w_i u_{i+2}}$, thus (1) $c_{w_2 u_2} = c_{w_1 u_3}$. In general, we construct P_h^i from P by adding the edges $(w_i, u_{i+2}), \ldots, (w_i, u_{i+h})$ and deleting the edges $(w_{i+1}, u_{i+1}), \ldots, (w_{i+h-1}, u_{i+h-1})$ for $i = 1, \ldots, n-2$ and $h = 2, \ldots, n-i$. The graphs P_h^i are obviously still planar and satisfy $c^T \chi^{P_h^i} = c_0$ with equality (Figure 2(b) shows P_3^1). Subtraction yields $c^T \chi^{P_{h-1}^i} - c^T \chi^{P_h^i} = ((c_{w_i u_{i+2}} + \cdots + c_{w_i u_{i+h-1}}) - (c_{w_{i+1} u_{i+1}} + \cdots + c_{w_{i+h-2} u_{i+h-2}})) - ((c_{w_i u_{i+2}} + \cdots + c_{w_i u_{i+h}}) - (c_{w_{i+1} u_{i+1}} + \cdots + c_{w_{i+h-1} u_{i+h-1}})) = -c_{w_i u_{i+h}} + c_{w_{i+h-1} u_{i+h-1}}$. Together with (1) we have (2) $c_{w_i u_{i+h}} = c_{w_{i+h-1} u_{i+h-1}}$ for $i = 1, \ldots, n-2$ and $h = 2, \ldots, n-i$. Symmetrically, we get the same for u_i, that is $c_{u_i w_{i+h}} = c_{w_{i+h-1} u_{i+h-1}}$ for $i = 2, \ldots, n-2$ and $h = 2, \ldots, n-i$.

Next let us construct $F_k^n = F_k$ similar as P_h^i from P by deleting the edges $(u_{n-1}, w_{n-1}), \ldots, (u_{n-k+1}, w_{n-k+1})$ and adding the edges $(w_{n-2}, u_n), \ldots, (w_{n-k}, u_n)$ for $k = 2, \ldots, n-1$. Subtraction

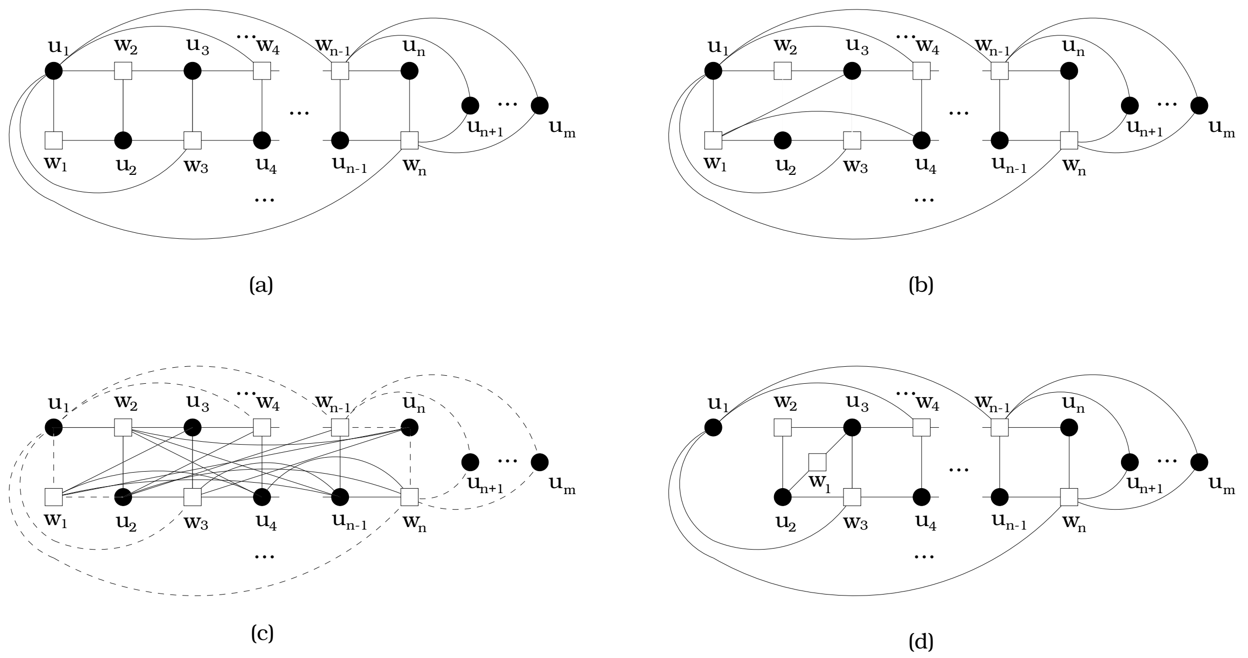

Figure 2

of F_{k-1} from F_k yields $c_{w_{n-k}u_n} = c_{u_{n-k+1}w_{n-k+1}}$ for $k = 3, \ldots, n-1$. Together with (2) we have $c_{w_{n-1}u_{n-1}} = c_{w_{n-h}u_n} = c_{u_{n-h+1}w_{n-h+1}} = c_{w_{(n-1)-(h-2)}u_{(n-1)-(h-2)}} = c_{w_{n-1-j}u_{n-1-j}}$ for $j = 1, \ldots, n-3$.

The planarity of the graph H_i^w arising from P by adding the edge (w_i, u_{i+2}) and deleting edge (u_{i+1}, w_{i+2}) for $i = 1, \ldots, n-2$ is evident by consideration of Figure 2(a). The same holds for H_i^u arising from P by adding the edge (u_i, w_{i+2}) and deleting (w_i, u_{i+1}) for $i = 2, \ldots, n-2$. We also preserve planarity by adding (w_1, u_3) and deleting (u_1, w_2) (If $n = 3$ the nodes u_{n+h} for $h = 1, \ldots, m-n$ have to be embedded into a different face). This way we get the c-values for the deleted edges.

So far we have shown equality of all coefficients c_e of edges $e_i^h = (u_i, w_{i+h})$ for $i = 2, \ldots, n-1$ and $h = 0, \ldots, n-i$, the edges $f_i^h = (w_i, u_{i+h})$ for $i = 1, \ldots, n-2$ and $h = 0, \ldots, n-i$, where $i + h > 2$, and $e = (u_1, w_2)$ (see Figure 2(c), the values of the solid drawn edges are known). Equality of the c-values for the edges $e_1^j = (u_1, w_j)$ for $j = 3, \ldots, n-2$ is obtained by replacing e_1^j with (w_{j-2}, u_{j+1}), which keeps planarity. The values of (u_1, w_{n-1}) and (u_1, w_n) are obtained by replacing them with (w_{n-2}, u_m) and (w_{n-3}, u_m), respectively.

For the case $m = n$ we have shown equality of the coefficients of almost all edges but (u_1, w_1), (w_1, u_2), (w_{n-1}, u_n) and (u_n, w_n). By interchanging nodes u_2 with u_3 in P (see Figure 2(a)) we obtain again a planar graph with equal weight and get $c_{w_1u_2} + c_{u_3w_4} = c_{w_1u_3} + c_{u_2w_4}$, where all of them but $c_{w_1u_2}$ is known to be equal. The c-values of the remaining edges can be obtained by the following construction. Delete edge (u_1, w_1) from P and add (w_1, u_3). We can observe that this graph is still planar (see Figure 2(d)) and satisfies $a^T x = a_0$ with equality, hence $c^T x = c_0$, which implies $c_{u_1w_1} = c_{w_1u_3}$. By the same construction in the rightmost rectangle we get $c_{u_nw_n} = c_{w_{n-2}u_n}$ and $c_{w_{n-1}u_n} = c_{w_{n-2}u_n}$ (in the last case we have to embed u_n within the face determined by $\{u_1, w_{n-2}, u_{n-1}, w_n\}$).

In case $m > n$ we need to show the equality of the c-values for all edges (u_{n+h}, w_i) for $h = 1, \ldots, m-n$ and $i = 1, \ldots, n$. We embed the node u_{n+h} (for fixed h) into the first rectangle determined by $\{u_1, w_2, u_2, w_1\}$. Therefore we have to delete the edges (u_{n+h}, w_{n-1}) and (u_{n+h}, w_n) from P and to add (w_1, u_{n+h}) and (w_2, u_{n+h}). Let R_i denote the graphs yielded by doing this for each of the rectangles $\{u_i, w_{i+1}, u_{i+1}, w_i\}$ for $i = 1, \ldots, n-1$. By subtraction of R^{i-1} from R^i we get the equalities $c_{w_1 u_{n+h}} = c_{w_3 u_{n+h}} = \cdots = c_{w_n u_{n+h}}$ and $c_{w_2 u_{n+h}} = c_{w_4 u_{n+h}} = \cdots = c_{w_{n-1} u_{n+h}}$ in case n is odd. Now take the embedding of graph R^1, delete edge (u_2, w_2) from it and add edge (u_{n+h}, w_3). This implies $c_{w_3 u_{n+h}} = c_{u_2 w_2}$. From R^2 we get by the same construction $c_{w_4 u_{n+h}} = c_{u_3 w_3}$ and so we have shown the equality of the c-values for all edges in $G = K_{m,n}$. Thus setting $\alpha = c_0/a_0$ proves the theorem. □

The facet-defining property also holds if we delete one arbitrary edge of K_n, which is not the case for $K_{m,n}$. For the graph $K_{3,4}$ with one deleted edge the inequality just yields a face of dimension $|E| - 2$. The proof of Corollary 2 stays essentially the same as that for K_n.

Corollary 2 *For the complete graph $G = (V, E)$ on $n >= 6$ nodes where one arbitrary edge e is removed, the inequality $x(E) \leq 3|V| - 6$ is facet-defining for $\mathcal{PLS}(G)$.*

One may think that the same must also hold for complete graphs, where two edges are removed. But in generally this is not the case. For the graph K_6 the above inequality is not facet-defining for any pair of deleted edges.

The inequalities considered up to this point have all coefficients $c_e = 1$. This is not the case for the following inequality.

Theorem 4 *Let $G = (V, E)$ be a subdivision of K_5 on the nodes $u_1, \ldots, u_5, v_1, v_2$ extended by the edge (v_1, v_2), where v_1 and v_2 denote the subdivision nodes. Further assume $(u_i, v_1), (u_j, v_1) \in E$ and $(u_k, v_2), (u_l, v_2) \in E$ with $i \neq j$, $k \neq l$, $j \neq k$ and $j \neq l$.*

If $i = k$ we define $U = \{(u_h, u_i), (u_h, u_j), (u_h, u_l) \text{ for } h \neq i, j, l\} \cup \{(u_j, v_1), (u_l, v_2)\}$. The definition of U for the case $i = l$ is symmetric. If i, j, k and l are pairwise distinct, we define $U = \{(u_i, v_1), (u_j, v_1), (u_k, v_2), (u_l, v_2)\}$.

Let $c_e = 2$ for all edges $e \in U$ and $c_e = 1$ for the remaining edges $e \notin U$. Then the inequality $2x(U) + x(E \setminus U) \leq c(E) - 2$ is facet-defining for $\mathcal{PLS}(G)$.

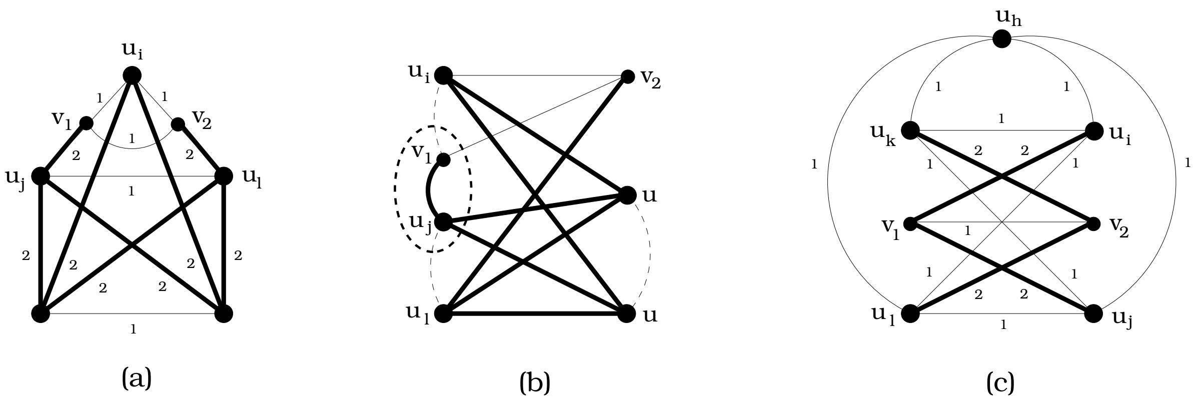

Figure 3

Proof. First consider the case $(i = k)$ (see Figure 3(a)). Each subdivision of $K_{3,3}$ contained in G must exactly have one node of $\{v_1, v_2\}$ as subdivision node, say v_1. Such a subdivision must be of the form shown in Figure 3(b). The edge set is $F_{v_1} = U \cup \{(v_1, v_2), (u_i, v_2)\}$. Symmetrically, if v_2 is the subdivision node we obtain the edge set $F_{v_2} = U \cup \{(v_1, v_2), (u_i, v_1)\}$. Whenever an edge $e \notin U$

is deleted from G, either a subdivision of K_5 is left or one of the above subdivisions of $K_{3,3}$ is still contained in the remaining graph. Thus validity is shown. The common edge set of all the minimal nonplanar graphs is exactly U. Thus $G \backslash \{e\}$ for any $e \in U$ is planar. By deleting the edge (v_1, v_2) in G we obtain exactly the subdivision of K_5. Together with (v_1, v_2) we can delete any other edge to get a planar graph, in particular we can choose any additional edge $e \in E \backslash (U \cup \{(v_1, v_2)\})$. It is also obvious that the removal of the edges (u_i, v_1) and (u_i, v_2) from G results in a planar graph. Thus we have found $|E|$ incidence vectors of planar subgraphs, which all satisfy the inequality $2x(U) + x(E \setminus U) \leq c(E) - 2$ with equality. The linear independence of these vectors can easily be verified.

Let us consider the second case, that is $i \neq k, l$ and $j \neq k, l$. An embedding of the graph is shown in Figure 3(c). After the deletion of the edge (u_i, u_k) there is still a path u_i, u_h, u_k between u_i and u_k, which preserves the presence of a subdivision of $K_{3,3}$. This is the same for the edges (u_i, u_l), (u_j, u_k) and (u_j, u_l). If (v_1, v_2) is deleted, the resulting graph is a subdivision of K_5. Hence we have to delete either one edge in U or at least two of the edges not in U to get a planar graph. Thus validity is shown. Together with the edge (v_1, v_2) we can delete any other edge in $E \backslash U$ in order to get a planar graph. It is also obvious that the deletion of the edges (u_k, u_j) and (u_i, u_l) also yields a planar graph. Hence we have again $|E|$ incidence vectors of planar graph, which are linear independent and satisfy the inequality $2x(U) + x(E \setminus U) \leq c(E) - 2$ with equality. □

In the following we will see that some operations like "edge splitting" and "edge contraction" keep the facet-defining properties.

Definition 1 Let $cx \leq c_0$ be an inequality defined in $\mathbb{R}^E$ and f be an edge in E. We say that the inequality $c^* x^* \leq c_0^*$ defined in $\mathbb{R}^{E^*}$ is obtained by **splitting the edge f** (h times) in the following sense. The edge $f = (u, w)$ is replaced by a path $P = (u = v_0, e_0, v_1, \ldots, e_h, v_{h+1} = w)$ and the weights are given by $c_0^* = c_0 + hc_f$, $c_{e_i}^* = c_f$ for $0 \leq i \leq h$, and $c_e^* = c_e$ for each e not contained in P. We also define the inverse operation, the (edge) **contraction on a path P** where we replace the path $P = (v_0, e_0, v_1, \ldots, e_h, v_{h+1})$ by the edge $f = (v_0, v_{h+1})$ if $\deg(v_i) = 2$ for $1 \leq i \leq h$ and $c_{e_i} = c'$ for $0 \leq i \leq h$. In this case $c_0^* = c_0 - hc'$, $c_f^* = c'$ and $c_e^* = c_e$ for each e not contained in P.

Note that for every facet-defining inequality $cx \leq c_0$ the weights c_e for all the edges on a path $P = (v_0, e_0, v_1, \ldots, e_h, v_{h+1})$ with $\deg(v_i) = 2$ for $1 \leq i \leq h$ are equal, because the removal of one edge on the path destroys exactly the same subdivisions of K_5 or $K_{3,3}$ as the removal of an arbitrary other edge on P does. This applies also to the edge $f = (v_0, v_{h+1})$. Hence we have the following lemma.

Lemma 4 *Let $cx \leq c_0$ be facet-defining for $\mathcal{PLS}(G)$. Then the inequality $c^* x^* \leq c_0^*$ obtained from $cx \leq c_0$ by splitting an edge f or contracting a path $P = (v_0, e_0, v_1, \ldots, e_h, v_{h+1})$ with $\deg(v_i) = 2$ for $1 \leq i \leq h$ is facet-defining for $\mathcal{PLS}(G')$, where G' denotes the graph obtained by the above substitution.*

Consider an inequality which is facet-defining for $\mathcal{PLS}(G)$. By adding an edge to G, which is incident to at most one node in G, the planarity or nonplanarity of G is not affected. The sequential lifting theorem for independence systems together with the above remark gives us the following lemma. We call the set of edges which have non-zero coefficients in the inequality $c^T x \leq c_0$ the support of the inequality.

Lemma 5 **(Zero Lifting)** *Let $G = (V, E)$ be a graph, $U \subseteq E$ and $c^T x \leq c_0$ a facet-inducing inequality for $\mathcal{PLS}(G[E \setminus U])$. Choose any $e \in U$ which has at most one endnode incident to the support of $c^T x \leq c_0$. Then $c^T x \leq c_0$ defines a facet of $\mathcal{PLS}(G[E \setminus U \cup \{e\}])$.*

Corollary 3 (Euler inequalities) *Let (V', F) be a clique or a complete bipartite subgraph contained in G. Then the Euler inequalities $x(F) \leq 3|V'| - 6$ or $x(F) \leq 2|V'| - 4$, respectively, are facet-defining for G.*

In the following section we will describe how the above theoretical results can help us to create good separation routines in order to get good upper bounds.

3. The algorithm

We have designed a simple experimental version of a branch and cut algorithm using valid inequalities for $\mathcal{PLS}(G)$. The algorithm is similar to the algorithm for the linear ordering problem reported in Grötschel, Jünger and Reinelt [GJR84]. The implementation was not hard, since we could use much of the problem independent routines described in the recent paper of Jünger, Reinelt and Thienel [JRT92] on a branch and cut algorithm for the traveling salesman problem. In contrast to the algorithm described there, we neither used sparse graph techniques nor methods for fixing and setting variables by logical implication.

The cutting plane generation as well as the lower bound heuristic is based on a planarity testing algorithm. In order to implement a first version of the branch and cut algorithm, we added only a few lines to an already implemented version of the linear planarity testing algorithm of Hopcroft and Tarjan, which is very fast (see [M92]). Since it is the central part of the algorithm, we will briefly describe it in the following.

The planarity testing algorithm of Hopcroft and Tarjan

At the beginning we call a depth-first-search procedure in order to divide the edge set of the graph $G = (V, E)$ into back edges and tree edges. We start by identifying a cycle C. When this cycle is removed from G, the graph falls apart into several pieces. The algorithm is called recursively to embed each piece in the plane together with the original cycle. Then the embeddings of the pieces are combined, if possible, to give an embedding of the entire graph.

One may think of successively adding paths consisting of tree edges and one back edge at the end to a previously obtained partial embedding. For more details, see [M92] or [HT74]. In the following we describe some details of the branch and cut algorithm.

Cutting plane generation

The trivial inequalities are handled implicitly by the LP-solver via lower and upper bounds. At the beginning we also add the inequality $x(E) \leq 3|V| - 6$, if it is violated, resp. $x(E) \leq 2|V| - 4$ in case G is bipartite, if it is violated.

Let x be an LP-solution produced in the cutting plane procedure applied in some node of the enumeration tree. For $0 \leq \varepsilon \leq 1$ we define $E_\varepsilon = \{e \in E \mid x_e \geq 1 - \varepsilon\}$ and consider $G_\varepsilon = (V, E_\varepsilon)$. For the unweighted graph G_ε the linear planarity testing algorithm of Hopcroft and Tarjan is called. The algorithm stops if it finds an edge set F which is not planar. In case the inequality $x(F) \leq |F| - 1$ is violated, we add the inequality to the constraints of the current LP. We remove the back edge of the path, which proved the nonplanarity of F after it was added and proceed with the planarity testing algorithm.

This way we usually find several forbidden subgraphs of the graph G_ε in one run of the planarity testing algorithm. Of course, these forbidden subgraphs do not necessarily define facets of the $\mathcal{PLS}$-polytope. However, these subgraphs must contain subgraphs which define facets (see Theorem 2). We try to reduce them to facet-defining inequalities in the following way. Once an edge set F is found, where the inequality $x(F) \leq |F| - 1$ is violated, we successively delete one edge $f \in F$ from it, and start again the planarity testing algorithm. If $F \setminus \{f\}$ is planar, we add it again to F. In either case we choose a different edge $f \in F$. In at most $|F|$ steps we have reduced F to a set of edges, which induces a minimal nonplanar subgraph. So we have found an inequality $x(F) \leq |F|-1$ which is facet-defining for $\mathcal{PLS}(G)$ and still violated by the current $\mathcal{LP}$-solution.

Another possible (but not yet implemented) separation routine is a heuristic, which searches for violated Euler-inequalities (see Corollary 3).

Lower bound heuristic

After an LP has been solved, we try to exploit the solution to produce a feasible solution. Again, we apply the planarity testing algorithm. This way we produce lower bounds which are useful not only for fathoming nodes in the branch and cut tree but also for fixing variables due to their reduced costs during a cutting plane phase.

After discovering a forbidden substructure, the back edge of the last added path is removed, so that the remaining substructure becomes planar. Since different depth-first-search trees yield different paths and thus different lower bounds, in every call of the planarity testing algorithm the depth-first-search tree is changed.

We also implemented a simple random heuristic, where the edges are subsequently added to the graph, if they don't destroy planarity. Our experimental results confirm the results of Cimikowski, who reported that simple random heuristics lead to better results on random graphs than the above described method [C92].

It would be much better to use more powerful heuristics, because in a branch and cut algorithm it is important to get good lower bounds. In a future implementation, we will try the algorithm of Cai, Han and Tarjan [CHT91] or Kant [K92] which yield a maximal planar subgraph. We also plan to try out the deltahedron heuristic of Foulds and Robinson [FR78], the wheel expansion heuristic of Eades et al [EFG82] or simply a greedy heuristic. This should be one of the next steps to improve the quality of the feasible solutions produced in the course of the algorithm.

Branching

Branching takes place if the current solution is infeasible yet no cutting planes have been found. We choose a variable x_e with fractional value as close as possible to $\frac{1}{2}$ and among those one with maximum absolute objective function coefficient.

4. Computational experiments

For the implementation of the above algorithm we combined the above described adaptions of a previous PASCAL implementation of the planarity testing algorithm [M92] with an adaption of a C-implementation of the branch and cut frame used in Jünger, Reinelt and Thienel [JRT92]. Our computational experiments were carried out on a SUN SPARCstation 10 model 20.

Table 2. Triangulated graphs incremented by 10,20,... edges

#problem	#Nodes	#Edges	Solution	Sol[C92]	Guarantee	BC-nodes	#LPs	Time
g100.10	100	304	292	264	0.68	1	20	120
g100.20	100	314	290	294	1.36	1	21	120
g100.30	100	324	282	267	4.08	1	20	120
g100.40	100	334	285	282	3.06	1	17	120
g100.50	100	344	269	294	8.53	1	12	120
g100.60	100	354	264	258	10.20	1	17	120
g100.70	100	364	243	240	17.34	1	14	120
g100.80	100	374	243	231	17.34	1	14	120
g100.90	100	384	243	261	17.34	1	14	120
g100.100	100	394	247	294	15.98	1	17	120

Table 1. Results for the graphs in [GT92]

Problem	#Nodes	#Edges	Solution	Sol[GT92]	Guarantee	BC-nodes	#LPs	Time
g1	10	22	20	20	0.00	1	4	0
g2	45	85	82	82	0.00	1	17	7
g10.0	10	24	24	24	0.00	1	1	0
g10.1	10	25	24	24	0.00	1	1	0
g10.2	10	26	24	24	0.00	1	3	0
g10.3	10	27	24	24	0.00	1	1	0
g25.0	25	69	69	68	0.00	1	1	0
g25.1	25	70	69	69	0.00	1	2	0
g25.2	25	71	69	68	0.00	1	1	0
g25.3	25	72	69	68	0.00	1	1	0
g50.0	50	144	144	129	0.00	1	1	1
g50.1	50	145	144	138	0.00	1	1	1
g50.2	50	146	144	142	0.00	1	1	1
g50.3	50	147	144	—	0.00	1	1	0
g100.0	100	294	294	183	0.00	1	1	4
g100.1	100	295	294	215	0.00	1	1	4
g100.2	100	296	294	234	0.00	1	3	13
g100.3	100	297	294	—	0.00	1	1	4

We could find only a few papers where computational results are reported. Goldschmidt and Takvorian [GT92] presented some results for triangulated planar graphs of 10, 25, 50 and 100 vertices to which they added incrementally one, two and three edges. Additionally they gave results for two graphs which already occurred in Jayakumar et al. (labeled g1) [JTS89] and in Kant (labeled g2) [K92]. In all these cases our algorithm found and proved the optimal solution in a reasonable amount of time (see Table 1). The columns from left to right display the number of nodes, the number of edges, the value of the best solution found by our algorithm, the value of the best solution from Goldschmidt and Takvorian [GT92], the quality guarantee ((upperbound-lowerbound)/upperbound), the number of branch and cut nodes, the number of LPs and the CPU times in seconds (Fractions of seconds are not shown). Cimikowski [C92] considered problem instances in which triangulated planar graphs are augmented by 10, 20, ... edges. We tried our code on such instances (see Table 2). Here the limits of our currently simple approach becomes clear: In no case we could find optimum solutions within 120 seconds of CPU time. Elaborate (and time consuming) heuristics must be added to our implementation in order to make it competitive on harder problem instances.

We also solved a facility layout problem given by Foulds and Robinson [FR78]. With a heuristic Leung [L92] got a solution value of 1101, whereas we could find and prove the optimal solution value of 1105 in about 7 minutes.

In order to further explore the limits of our branch and cut algorithm, we tested it on a series of randomly generated graphs. At first we increased the density on graphs with 10 and 20 nodes. We did this for unweighted graphs (see Table 3) and for weighted graphs (see Table 4), where the weights were normal distributed with mean 100 and standard deviation $\sigma = 20$. We tried different random seeds, but the variance was not high so that the table gives the right impression. In all cases we stopped the computation after 120 seconds of CPU-time. Experience shows that the results do not improve much with a longer computation time. One can observe that the easiest problem instances are those on sparse graphs and very dense graphs. For weighted graphs the behaviour of our branch and cut algorithm is much worse than for unweighted graphs. We believe that this is due to the fact that we have not yet implemented any heuristic for lower bounds, which makes use of the edge weights.

Since in automatic graph drawing the graphs are relatively sparse, we ran a series of sparse graphs. We increased the number of nodes and defined the number of edges to be 1.5 times the number of nodes and 2 times $|V|$ (see Table 5) for the unweighted case.

Table 3. Increasing the density for unweighted graphs on 10 and 20 nodes

#Nodes	#Density	#Edges	Solution	Guarantee	BC-nodes	#LPs	Time
10	10	4	4	0.00	1	1	0
10	20	9	9	0.00	1	1	0
10	30	13	13	0.00	1	1	0
10	40	18	17	0.00	1	2	0
10	50	22	20	0.00	1	5	0
10	60	27	24	0.00	21	137	6
10	70	31	24	0.00	3	16	1
10	80	36	24	0.00	1	2	0
10	90	40	24	0.00	1	1	0
10	100	45	24	0.00	1	1	0
20	10	19	19	0.00	1	1	0
20	20	38	36	0.00	1	5	0
20	30	57	43	16.98	48	832	120
20	40	76	47	5.55	12	508	120
20	50	95	52	0.00	25	339	103
20	60	114	49	0.00	5	113	43
20	70	133	49	0.00	1	2	0
20	80	152	53	0.00	1	5	2
20	90	171	54	0.00	1	1	0
20	100	190	54	0.00	1	1	0

Table 4. Increasing the density for weighted graphs on 10 and 20 nodes

#Nodes	#Density	#Edges	Guarantee	BC-nodes	#LPs	Time
10	10	4	0.00	1	1	0
10	20	9	0.00	1	1	0
10	30	13	0.00	1	1	0
10	40	18	0.00	1	2	0
10	50	22	0.00	1	11	0
10	60	27	0.00	9	55	4
10	70	31	2.74	153	1145	120
10	80	36	0.00	3	16	2
10	90	40	1.89	159	905	120
10	100	45	1.64	118	816	120
20	10	19	0.00	1	1	0
20	20	38	3.43	85	819	120
20	30	57	12.89	14	495	120
20	40	76	10.79	9	354	120
20	50	95	8.17	6	262	120
20	60	114	9.49	5	209	120
20	70	133	3.30	3	173	120
20	80	152	5.04	2	143	120
20	90	171	4.79	3	131	120
20	100	190	4.47	3	158	120

Finally, we tested our implementation for a graph given by Tamassia, Di Battista and Batini in a paper about automatic graph drawing [TBB88] (see Figure 4(a)). In order to get the maximum planar subgraph the algorithm removed four of the 62 edges (24 seconds). For the embedding of the planar subgraph we used the program of Mutzel [M92]. The insertion of the previously removed edges causes nine crossings, which is much less than the number of crossings in Figure 4(a). The resulting embedding looks quite nice (see Figure 4(b)).

5. Final remarks

Our implementation of a branch and cut algorithm for finding maximum planar subgraphs is very simple in comparison with branch and cut algorithms for other combinatorial optimization problems such as the linear ordering problem [GJR84], or the traveling salesman problem [PR91,JRT92]. Nevertheless, we could use it to solve some problems occuring in the literature to optimality for the first time. This makes us confident that our planned refinements on some of which (and possibly

Table 5. Sparse graphs

#Nodes	#Edges	Solution	Guarantee	BC-nodes	#LPs	Time
10	15	14	0.00	1	2	0
20	30	28	0.00	1	5	0
30	45	42	0.00	1	31	3
40	60	55	3.50	116	864	120
50	75	68	5.55	49	596	120
60	90	79	8.14	23	409	120
70	105	93	7.92	13	295	120
80	120	103	10.43	5	210	120
90	135	112	13.84	6	184	120
100	150	127	12.41	2	124	120
10	20	19	0.00	1	2	0
20	40	37	0.00	1	5	0
30	60	52	7.14	49	759	120
40	80	66	13.15	17	465	120
50	100	79	16.84	7	285	120
60	120	90	21.05	4	182	120
70	140	106	21.48	3	151	120
80	160	117	23.53	2	137	120
90	180	132	23.69	2	98	120
100	200	139	27.60	1	84	120

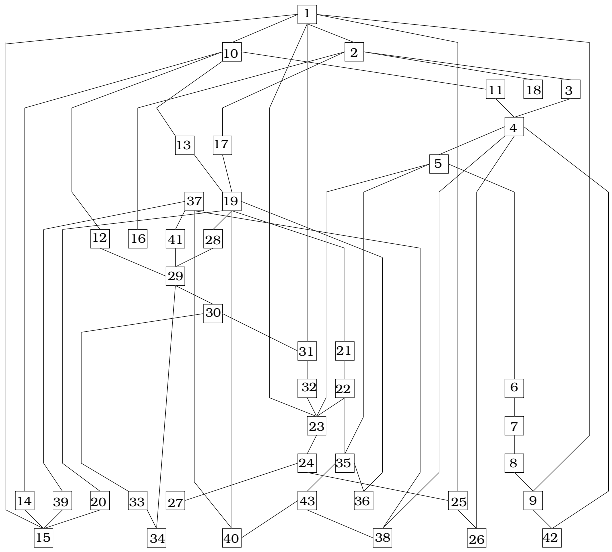

Figure 4(a)

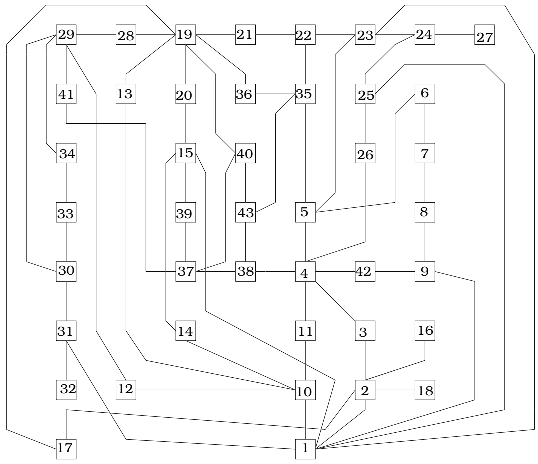

Figure 4(b)

others) we hope to be able to report in a further paper, will lead to a useful algorithm.

6. References

[BL76] Booth, K.S., and G.S. Lueker, "Testing for the consecutive ones property, interval graphs and graph planarity testing using PQ-tree algorithms", *J. of Computer and System Sciences* 13 (1976) 335–379

[CHT91] Cai, J., X. Han and R.E. Tarjan, "An O($m \log n$)-Time Algorithm for the Maximal Planar Subgraph Problem", to appear in *SIAM J. Comput.* (1992)

[C92] Cimikowski, R.J., "An Empirical Analysis of Graph Planarization Heuristics", unpublished manuscript, Computer Science Dept., Montana State Univ. (1992)

[EFG82] Eades P., L.R. Foulds and J.W. Giffin, "An efficient Heuristic for Identifying a Maximum Weight Planar Subgraph", *Combinatorial Mathematics IX, Lecture Notes in Mathematics No. 952*, Springer-Verlag, Berlin (1982)

[FR76] Foulds, L.R., and R.W. Robinson, "A Strategy for Solving the Plant Layout Problem", *Operational Research Quaterly* 27 (1976) 845–855

[FR78] Foulds, L.R., and R.W. Robinson, "Graph Theoretic Heuristics for the Plant Layout Problem", *Int. J. Production Research* 16 (1978) 27–37

[GJ79] Garey, M.R., and D.S. Johnson, "Computers and Intractibility: A Guide to the Theory of NP-completeness", Freeman & Co., San Francisco (1979)

[GT92] Goldschmidt, O., and A. Takvorian, "An efficient graph planarization two-phase heuristic", Technical Report ORP91-01, Dept. of Mech. Engr., Univ. Texas-Austin (1992)

[GJR84] Grötschel, M., M. Jünger, and G. Reinelt, "A cutting plane algorithm for the linear ordering problem", *Operations Research* 32 (1984) 1195–1220

[HT74] Hopcroft, J., and R.E. Tarjan, "Efficient planarity testing", *J. ACM* 21 (1974) 549–568

[JTS89] Jayakumar, R., K. Thulasiraman and M.N.S. Swamy, "O(n^2) Algorithms for Graph Planarization", *IEEE Trans. on Computer-aided Design* 8 (1989) 257–267

[JRT92] Jünger M., G. Reinelt, and S. Thienel, "Provably good solutions for the traveling salesman problem", Rep. No. 92.114, Angewandte Mathematik und Informatik, Universität zu Köln (1992)

[K92] Kant, G., "An O(n^2) Maximal Planarization Algorithm based on PQ-trees", Technical Report, RUU-CS-92-03, Dept. of Computer Science, Utrecht University (1992)

[L92] Leung, J., "A new graph-theoretic heuristic for facility layout", *Management Science* 38 (1992) 594–605

[LG78] Liu, P. C., and R.C. Geldmacher, "On the deletion of nonplanar edges of a graph", *Proc. 10th. S-E Conf. on Comb., Graph Theory, and Comp.*, Boca Raton, FL (1977) 727–738

[M92] Mutzel, P., "A fast linear time embedding algorithm based on the Hopcroft-Tarjan planarity test", Rep. No. 92.107, Angewandte Mathematik und Informatik, Universität zu Köln (1992)

[PR91] Padberg, M. W., and G. Rinaldi, "A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems", *SIAM Review* 33 (1991) 60–100

[TBB88] Tamassia, R., G. Di Battista, and C. Batini, "Automatic graph drawing and readability of diagrams", *IEEE Transactions on Systems, Man and Cybernetics* 18 (1988) 61–79