European Journal of Operational Research 69 (1993) 121–130
North-Holland

Theory and Methodology

A cutting-plane approach to the edge-weighted maximal clique problem

G. Dijkhuizen and U. Faigle
University of Twente, Enschede, Netherlands

Received August 1990

Abstract: We investigate the computational performance of a cutting-plane algorithm for the problem of determining a maximal subclique in an edge-weighted complete graph. Our numerical results are contrasted with reports on closely related problems for which cutting-plane approaches perform well in instances of moderate size. Somewhat surprisingly, we find that our approach already in the case of $n = 15$ or $n = 25$ nodes in the underlying graph typically neither produces an integral solution nor yields a good approximation to the true optimal objective function value. This result seems to shed some doubt on the universal applicability of cuttingplane approaches as an efficient means to solve linear (0, 1)-programming problems of moderate size.

Keywords: Integer linear programming; Cutting-plane; Clique polytope; Partition polytope

1. Introduction

It is a generally observed phenomenon that linear programming problems are 'easy' to solve in practice even for rather large problem instances and even with theoretically inefficient methods like the simplex algorithm. The use of such methods, however, requires that a description of the set of feasible solutions via a system of linear inequality restrictions be available.

Many optimization problems arising in practice can only abstractly be formulated as linear programming problems over polyhedra $\mathbb{P}$ in the sense that no linear description of $\mathbb{P}$ is known. Such a well-known example is the traveling salesman problem (TSP) (cf. Lawler et al., 1985). The fundamental idea for a cutting-plane algorithm to solve the linear optimization problem over $\mathbb{P}$ is as follows: determine a set F of inequalities that are satisfied by the vectors in $\mathbb{P}$ and run a linear programming algorithm relative to the set F of restrictions. This idea was first successfully implemented by Dantzig et al. (1954) for a TSP with $n = 49$ cities. They note: "A surprising empirical observation is the use of only a trivial number of the many possible restraints to solve any particular problem." The class F chosen by Dantzig et al. comprised the so-called 2-matching and subtour elimination constraints, which have the property that the feasible solutions to the original problem are exactly the (0, 1)-vectors satisfying all constraints in F. Adding the class of 'comb inequalities' to F, Grötschel (1980) then solves a

Correspondence to: Dr. U. Faigle, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands.

TSP with $n = 120$ noting that already the original class of constraints yielded a very good approximation to the true optimal objective function value. Because of the good approximation for the objective function, one can try and combine a cutting-plane approach with a branch-and-bound procedure for the discrete (0, 1)-programming problem. This way, e.g., Padberg and Rinaldi (1987) successfully solve a TSP with $n = 532$.

One might expect that it is relatively straightforward to improve a cutting-plane algorithm relative to the polyhedron $\mathbb{P}$ by looking for new classes of valid inequalities and adding these to the old set of constraints. There is a practical difficulty: not only is it often quite involved to find a new class F' of constraints but it appears to be also intractable to efficiently solve the associated *separation problem*: detect some member of F' that is violated by a given vector x. Thus existing implementations of cutting-plane algorithms typically fully exploit only a relatively small class of easy to check valid inequalities while disregarding other classes completely or checking them at most heuristically.

Nevertheless, when the cutting-plane approach was applied to other discrete linear (0, 1)-programming problems, the results obtained seemed to suggest that the observation of Dantzig et al. (1954) quoted above was a general phenomenon not restricted to the TSP alone: for problem instances of moderate size, a cutting-plane algorithm exploiting a rather basic set of inequalities should produce either an integral and hence optimal solution for the original problem or at least a very good approximation to the true optimal value (see, e.g., Marcotorchino and Michaud, 1980; Reinelt, 1985; Wakabayashi, 1986; Faigle et al., 1987).

It is the purpose of this note to report about a computational study on a problem where the natural cutting-plane approach performs very poorly already in very small problem instances and, doing so, seems to shed doubt on the universal applicability of cutting-plane algorithms to linear (0, 1)-programming problems.

The problem we investigate is motivated by a facility location problem considered in Späth (1985): find a subclique of maximal weight in an edge-weighted complete graph. In spite of extensive use of cutting-planes our algorithm typically fails to produce (0, 1)-solutions already for graphs with $n = 15$ nodes. Moreover, we illustrate with the data from Späth (1985) for $n = 25$ that also the objective function value computed by the cutting-plane algorithm may be far from the true optimum (see Section 5 below).

We find our results even more surprising in the light of the seemingly close relationship between our *max-clique problem* and the *partition problem* studied in Wakabayashi (1986) and Faigle et al. (1987), where an edge-weighted complete graph is to be optimally partitioned into subcliques. In fact, many of the basic inequalities used in cutting-plane algorithms for the partition problem (e.g., the so-called triangle inequalities and partition inequalities in Section 2) are valid and facet inducing not only for the polytope associated with the partition polytope but also for the clique polytope.

We discuss the theoretical background of our cutting-plane algorithm briefly in Section 2 and Section 3 restricting ourselves to those classes of facet inducing constraints that actually enter the algorithm. The algorithm itself is sketched in Section 4.

Let us finally make it clear that our primary concern in this study is not to solve the max-clique problem efficiently but to investigate the performance of a cutting-plane approach to the max-clique problem. We have, therefore, been quite generous in letting our algorithm search for possibly violated cutting-planes on the one hand and have, on the other hand, not tried to combine the cutting-plane algorithm with a branch-and-bound procedure.

2. The clique polytope and related polytopes

The basic combinatorial model for the problem we consider can be formulated with the help of the complete (undirected) graph $K_n = (V, E)$ with $|V| = n$ nodes and $|E| = \binom{n}{2}$ edges as follows:

Let $\mathcal{F} \subseteq 2^E$ be a family of subsets of edges of K_n and $w : E \to \mathbb{R}$ a weight function. w extends to a *linear* weight function $w : \mathcal{F} \to \mathbb{R}$ via

$$w(F) = \sum_{e \in F} w_e \quad (F \in \mathcal{F}).$$

Interpreting each member $F \in \mathcal{F}$ as a (0, 1)-incidence vector in $\mathbb{R}^E$, $\mathscr{F}$ is exactly the set of vertices of its convex hull

$$\mathbb{P}(\mathscr{F}) = \{x \in \mathbb{R}^E \mid x = \sum \lambda_F \cdot F, \lambda_F \geq 0, \sum \lambda_F = 1\}.$$

Hence, the optimization problem

$$\max\{w(F) \mid F \in \mathscr{F}\}$$

is equivalent to the linear programming problem

$$\max\ w \cdot x \quad \text{s.t. } x \in \mathscr{P}(\mathscr{F})$$

over the polytope $\mathbb{P}(\mathscr{F})$.

Let $2 \leq b \leq n$ be an integer and choose $\mathscr{F}$ to consist of exactly the edge sets of the non-trivial complete subgraphs K' of K_n with at most b nodes. In this case, we call

$$\mathbb{P}_n(b) = \mathbb{P}(\mathscr{F})$$

the *(b-restricted) clique polytope* of K_n. It can be shown that the dimension of $\mathbb{P}_n(b)$ satisfies

$$\dim \mathbb{P}_n(b) = \begin{cases} \binom{n}{2} - 1 & \text{if } b = 2, \\ \binom{n}{2} & \text{if } 3 \leq b \leq n. \end{cases}$$

Moreover, we remark that the diameter of $\mathbb{P}_n(b)$ equals 1 (Faigle, 1987). This means that each member of $\mathscr{F}$ can be reached in one simplex iteration from any other member. The problem, of course, lies in not knowing how to carry out such an iteration optimally relative to a weight function w.

If b is fixed, optimizing over $\mathbb{P}_n(b)$ is polynomial because K_n has only a polynomial number of subgraphs with at most b nodes. In general, the linear optimization problem over $\mathbb{P}_n(b)$ is NP-complete since it apparently generalizes the problem of finding a maximal clique in a (not necessarily complete) graph G on n nodes (see Section 4 below). The latter is known to be NP-complete (cf. Garey and Johnson,. 1979).

Linear optimization problems over closely related combinatorial polytopes have been investigated. The *b-restricted partition polytope* $\overline{\mathbb{P}}_n(b)$ of Faigle, Schrader and Suletzki (1987) arises from families $\mathscr{F}$ of edge sets F that can be obtained as follows: partition the node set

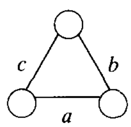
$$V = B_1 \dot{\cup} \cdots \dot{\cup} B_i \dot{\cup} \cdots$$

into pairwise disjoint blocks B_i such that $|B_i| \leq b$ $(i = 1, \dots)$ and let F consist of all those edges with both endpoints in the same block. The *clique partitioning polytope* of Wakabayashi (1986) (see also Grötschel and Wakabayashi, 1989) corresponds to the polytope $\overline{\mathbb{P}}_n(n)$ in our notation.

The *linear ordering polytope* $\vec{\mathbb{P}}_n$ is derived from the complete *directed* graph $\vec{K}_n = (V, \vec{E}_n)$ and hence is represented in $n(n-1)$-dimensional space. The vertices of $\vec{\mathbb{P}}_n$ correspond to those subsets $F \subseteq \vec{E}_n$ of cardinality $|F| = \binom{n}{2}$ that contain no directed cycle (Marcotorchino and Michaud, 1980; Grötschel, Jünger and Reinelt, 1984; Jünger, 1985, and Reinelt, 1985).

The vertices of the polytope defined above can easily be interpreted as the feasible solutions for certain integer linear programming problems. The idea thereby is to formulate combinatorial properties of (0, 1)-vectors via linear inequalities that have to be also satisfied by convex combinations of those (0, 1)-vectors.

Say that three edges $a, b, c \in E_n$ form a *triangle* tr(a, b, c) in K_n if they are pairwise incident:

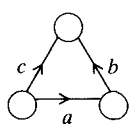


with each tr(a, b, c), we associate the *triangle inequality*:

$$x_a + x_b - x_c \leq 1. \tag{2.1}$$

Note that (2.1) expresses a combinatorial closure concept for (0, 1)-vectors: components corresponding to two sides of a triangle can only be '1' if also the third side yields '1'.

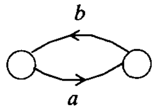
Three directed edges $a, b, c, \in \vec{E}_n$ form an *acyclic triangle* atr(a, b, c) in $\vec{K}_n$ if they give rise to the following configuration:



With each atr(a, b, c), we associate the *acyclic triangle inequality*:

$$x_a + x_b - x_c \leq 1. \tag{2.1'}$$

A *directed 2-cycle* in $\vec{K}_n$ is a pair 2-c(a, b) of oppositely directed edges, $a, b \in \vec{E}_n$ with the same endpoints:



We consider for each 2-$c(a, b)$ the linear restriction

$$x_a + x_b = 1. \tag{2.1.1'}$$

Combinatorially, (2.1.1′) precludes directed 2-cycles in a (0, 1)-solution and, furthermore, guarantees an orientation between every pair of nodes, i.e., a tournament. (2.1′) bars directed cycles from that tournament. Hence it is straightforward to verify the following facts.

Proposition 1. *A* (0, 1)*-vector*

$$x = (\ldots, x_a, \ldots) \in \mathbb{R}^{\vec{E}_n}$$

is a vertex of $\vec{\mathbb{P}}_n$ *if and only if x satisfies all linear restrictions of type* (2.1′) *and* (2.1.1′). □

Proposition 2. *A* (0, 1)*-vector*

$$x = (\ldots, x_a, \ldots) \in \mathbb{R}^{E_n}$$

is a vertex of the partition polytope $\overline{\mathbb{P}}_n(b)$ *if and only if x satisfies all linear restrictions of type* (2.1) *and, additionally, for all* $v \in V$,

$$\sum_{v \in a} x_a \leq b - 1. \quad \square \tag{2.2}$$

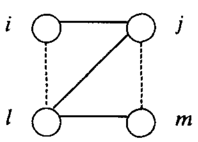
Note that (2.2) just ensures that the partition is indeed b-restricted.

Clearly, every vertex of the clique polytope $\mathbb{P}_n(b)$ is in particular a vertex of the partition polytope $\overline{\mathbb{P}}_n(b)$. Hence, every linear restriction that is valid for $\overline{\mathbb{P}}_n(b)$ is also valid for $\mathbb{P}_n(b)$. In order to single out the vertices of $\mathbb{P}_n(b)$ among those of $\overline{\mathbb{P}}_n(b)$, we introduce a new class of linear restrictions.

If $i, j, l, m \in V$ are four distinct nodes of K_n, then the associated *Z-inequality* $Z(i, j, l, m)$ is

$$x_{ij} + x_{jl} + x_{lm} - x_{il} - x_{jm} \leq 1 \tag{2.3}$$

(ij denotes here the edge with endpoints i and j etc.).



Proposition 3. *A non-zero* (0, 1)*-vector*

$$x = (\ldots, x_a, \ldots) \in \mathbb{R}^{E_n}$$

is a vertex of the clique polytope $\mathbb{P}_n(b)$ *if and only if x satisfies all linear restrictions of type* (2.1), (2.2) *and* (2.3).

Proof. It is straightforward to check that each vertex of $\mathbb{P}_n(b)$ satisfies (2.3).

Conversely, consider an arbitrary partition of V with at least two non-trivial blocks B_1, B_2. Let x be the associated (0, 1)-incidence vector. We claim that x violates some Z-inequality.

Indeed, we may choose distinct nodes $i, j \in B_1$ and $l, m \in B_2$ yielding

$$x_{ij} = x_{lm} = 1 \quad \text{and} \quad x_{jl} = x_{il} = x_{jm} = 0$$

and thus violating $Z(i, j, l, m)$. □

Because a linear restriction is valid for a (bounded) polytope if and only if it is valid for each of its vertices, the integer linear programming descriptions of the foregoing propositions give approximations for the polytopes $\vec{\mathbb{P}}_n$, $\overline{\mathbb{P}}_n(b)$ and $\mathbb{P}_n(b)$, where we also replace the integrality conditions $x_a \in \{0, 1\}$ by the *trivial inequalities*

$$0 \leq x_a \leq 1. \tag{2.4}$$

Rather than solving the integer linear program relative to a given weight function, one may try to find an approximative solution to the program by solving the corresponding *linear programming relaxation* implied by (2.4) with a standard linear programming algorithm.

This simple idea appears to often produce surprisingly good results in the case of the linear ordering problem (see, e.g., Marcotorchino and Michaud, 1980 or Reinelt, 1985) and the partition problem (see, e.g., Wakabayashi, 1986 or Faigle et al., 1987). Not only seems the LP-relaxation to produce a very good approximation to the optimal objective function value of the integer LP but, in many cases, actually to yield a solution vector with integer components, which in view of Propositions 1 and 2 must then be a true optimal solution for the integer LP.

When we tried to apply the same idea to the clique problem on the basis of Proposition 3, we were not so lucky. In other words, the LP-relaxation via the linear constraints (2.4), (2.1), (2.2) and (2.3) catches the structure of $\mathbb{P}_n(b)$ rather poorly.

3. Facets of the clique polytope

In order to obtain a better approximation of $\mathbb{P}_n(b)$ via the solution set of a suitable system of linear inequalities, we will now list some classes of valid inequalities for $\mathbb{P}_n(b)$. The validity of these inequalities is easily verified by checking their validity on the vertices of $\mathbb{P}_n(b)$:

$$\sum_{e \in E_n} x_e \geq 1, \tag{3.0.0}$$

$$\sum_{e \in E_n} x_e \leq \binom{b}{2}. \tag{3.0.1}$$

Let $\{i_1, \ldots, i_k\}$ be a set of k distinct nodes of K_n and define the edge sets

$$P = \{i_1 i_2, i_2 i_3, \ldots, i_{k-1} i_k\},$$

$$\bar{P} = \{i_1 i_3, i_2 i_4, \ldots, i_{k-3} i_{k-1}, i_{k-2} i_k\}.$$

Then we obtain the *path inequality* associated with $\{i_1, \ldots, i_k\}$ via

$$x(P) - x(\bar{P}) \leq \left\lfloor \frac{k+2}{4} \right\rfloor, \tag{3.1}$$

where, as usual, $x(P)$ is the sum of the variables associated with members of P. Obviously, (3.1) generalizes the triangle- and Z-inequalities ($k = 3$ and $k = 4$). We call a path inequality with $k = 5$ also a W-inequality.

In a similar fashion, we associate with a 6-element subset $\{i_1, i_2, i_3, i_4, i_5, i_6\}$ of nodes the edge sets

$$S = \{i_1 i_2, \ldots, i_5 i_6\},$$

$$\bar{S} = \{i_1 i_6, i_1 i_3, i_2 i_4, i_3 i_5, i_4 i_6\}$$

in order to obtain an *S-inequality* via

$$x(S) - x(\bar{S}) \leq 1. \tag{3.2}$$

Note that path inequalities for $k \geq 4$ and S-inequalities are generally not valid for the partition polytope $\bar{\mathbb{P}}_n(b)$. The next class of inequalities is also valid for $\bar{\mathbb{P}}_n(b)$.

Let $T \subseteq V$, $|T| \geq 2$, be a subset of nodes of K_n and $i \in V/T$. Denoting by $E(T)$ the edges of K_n with both endpoints in T, we define the associated *partition inequality* via

$$\sum_{t \in T} x_{it} - \sum_{e \in E(T)} x_e \leq 1. \tag{3.3}$$

In order to evaluate how closely a given inequality $\alpha^{\mathrm{T}} x \leq a$ approximates the polyhedron $\mathbb{P}_n(b)$, assume that it is valid for $\mathbb{P}_n(b)$ and consider the induced face

$$F(\alpha, a) = \{x \in \mathbb{P}_n(b) : \alpha^{\mathrm{T}} x = a\}.$$

Recall that $F(\alpha, a)$ is a *facet* of $\mathbb{P}_n(b)$ if

$$\dim F(\alpha, a) = \dim \mathbb{P}_n(b) - 1.$$

Facet inducing inequalities are unique (up to scaling) and are not implied by other valid inequalities of $\mathbb{P}_n(b)$. In that sense, each facet inducing inequality may be viewed as a best possible linear approximation to $\mathbb{P}_n(b)$.

Proposition 4. *The following valid inequalities are facet inducing for* $\mathbb{P}_n(b)$:

(0)	*each trivial inequality*	$x_a \geq 0$;
(1)	*the inequality* (3.0.0)	*for* $3 \leq b \leq n$;
(2)	*the inequality* (3.0.1)	*for* $3 \leq b \leq n-2$;
(3)	*each trivial inequality*	*for* $4 \leq b \leq n$;
(4)	*each Z-inequality*	*for* $4 \leq b \leq n$;
(5)	*each W-inequality*	*for* $4 \leq b \leq n$, $n \geq 5$;
(6)	*each S-inequality*	*for* $4 \leq b \leq n$, $n \geq 6$;
(7)	*each partition inequality*	*for* $4 \leq b \leq n$.

We will sketch a proof for the fact that Z-inequalities are facet inducing. The analogous property for the other inequalities can be derived by similar arguments (cf. Dijkhuizen (1989) for details).

Proof of Proposition 4.4. Without loss of generality, we consider the Z-inequality

$$x_{13} + x_{32} + x_{24} - x_{12} - x_{34} \leq 1$$

on the subset $\{1, 2, 3, 4\}$ of nodes of K_n. If suffices to show that each (incidence vector of an) edge of K_n can be written as a linear combination of (incidence vectors of the edge sets of) cliques in K_n comprising at most 4 nodes and satisfying the given Z-inequality with equality. Then the induced face will have dimension at least $\binom{n}{2} - 1$. Because obviously not every subclique of K_n yields equality, the induced face must be a facet.

Consider first the case $n = 4$. Then it is readily verified that each edge is a linear combination of C_{13}, C_{23}, C_{24}, C_{123}, and C_{1234}, where, for example, C_{234} denotes the (incidence vector of) the edge set of the clique on the subset {2, 3, 4} of nodes.

If $n > 4$, we let K_{n-1} be the complete graph on $\{1, 2, \ldots, n-1\}$ and assume by induction that each edge of K_{n-1} can be obtained as a linear combination of cliques on at most 4 nodes for which equality holds. Then we have, for instance,

$$C_{1n} = C_{23} + C_{13n} + C_{24n} - C_{234n},$$

showing that also the remaining edges are representable in the desired manner. □

Let us remark in passing that the trivial inequalities $x_e \leq 1$ are implied by the set of triangle inequalities and hence are not facet inducing. Moreover, it can be shown that each path inequality with $k \leq 6$ is implied by the sets of trivial and path inequalities given in Proposition 4.

The classes listed in Proposition 4 are far from exhausting all facets of the polytope $\mathbb{P}_n(b)$ (a sample of techniques leading to some additional classes of facets can be found in Dijkhuizen, 1989). We restrict ourselves to these classes for practical reasons: we feel that these classes constitute a maximum of what an implementation of a cutting plane algorithm for the b-restricted clique problem can, at the current state of the art, realistically exploit.

The difficulty comes from the *separation problem* relative to a class J of linear inequalities: decide if a given vector $x \in \mathbb{R}^E$ violates one of the inequalities in J. Each cutting plane algorithm that fully exploits J must be able to solve the corresponding separation problem.

Already for the case where J is the class of partition inequalities no separation algorithm with polynomial running time is known. Exhaustive search through J is not practical since J has exponential size. Hence we are unable to fully exploit these inequalities. As in Wakabayashi (1986), we contend ourselves with a heuristic search for violated partition inequalities.

4. A cutting-plane algorithm

The *max-clique problem* relative to the weight vector $w \in \mathbb{R}^E$ and the complete graph $K_n = (V, E)$ on n nodes is

$$\text{(CP)} \quad \max \ w \cdot x \quad \text{s.t.} \ x \in \mathbb{P}_n(b),$$

where $2 \leq b \leq n$ is a given size restriction on the subcliques of K_n.

We will attempt to solve the linear programming relaxation (LP) below instead of (CP). It follows from Proposition 3 that each optimal solution x^* of (LP) is, in particular, an optimal solution of (CP) whenever x^* has integer components.

We now set
F_0 := class of trivial inequalities,
F_1 := the inequalities (3.0.0) and (3.0.1),
F_T := class of triangle inequalities,
F_Z := class of Z-inequalities,
F_W := class of W-inequalities,
F_S := class of S-inequalities,
F_P := class of partition inequalities,
$F := F_0 \cup F_1 \cup F_T \cup F_Z \cup F_W \cup F_S \cup F_P$
and specify the linear programming relaxation of (CP):

$$\text{(LP)} \quad \max \ w \cdot x \quad \text{s.t.} \ x \text{ satisfies all inequalities in } F.$$

Our *cutting-plane algorithm* can now be sketched as follows:

(0) initialize the linear program

$$\max \ w \cdot x \quad \text{s.t.} \ 1 \leq \sum_{e \in E} x_e \leq \binom{b}{2}, \quad 0 \leq x_e \leq 1; \qquad (\overline{\text{LP}})$$

(1) determine an optimal solution x^* for $(\overline{\text{LP}})$;

(2) if x^* is the edge-incidence vector of a clique in K_n, then output the optimal solution x^* of (CP) and STOP;

(3) search for inequalities in F that are violated by x^*;

(4) if search in (3) is successful, update $(\overline{\text{LP}})$ and go to (1);

(5) STOP; (no solution for (CP) was found).

In our implementation of the cutting-plane algorithm, we carried out step (1) with the software package MPSX/370 on an IBM 9370 computer.

We execute step (3) in the following manner: we start exhaustive search through F_T until we have found MAXCUT violated triangle inequalities. Then we search through F_Z until 3 * MAXCUT violated Z-inequalities are obtained.

Only if our current solution x^* satisfies all inequalities in $F_T \cup F_Z$, we search through $F_W \cup F_S$ for at most 5 * MAXCUT violated inequalities, where the parameter MAXCUT is chosen in advance dependent on the problem size n.

We try to detect violated partition inequalities only if all inequalities in $F_T \cup F_Z \cup F_W \cup F_Z$ hold for x^*. Our search through F_P is not exhaustive but based on a heuristic. Thus, if the algorithm halts in step (5), we have no guarantee that F was completely exploited. We do exploit, however, all path inequalities and all S-inequalities.

The update of $(\overline{\text{LP}})$ in step (4) always retains the inequalities of $F_0 \cup F_1$. Among the other inequalities in the set of restrictions of the current $(\overline{LP})$, those are removed that are not binding for x^*. Then all inequalities found in (3) are adjoined to yield the updated set of restrictions for $(\overline{\text{LP}})$.

Without going into details, let us mention that the subroutines we employ in step (3) have running time complexity $O(n^4)$ for $F_T \cup F_Z$ and $O(n^6)$ for $F_W \cup F_S$ while the heuristic for F_P runs in time $O(n^3)$. This indicates that the running time of our cutting-plane algorithm quickly increases with n. For the examples we describe in Section 5, it ranges between a few minutes (for $n = 10$) to several hours ($n = 25$).

We want to make it clear, however, that our objective in this study was not to implement a time-efficient cutting-plane algorithm for the max-clique problem but to test the feasibility of trying to solve (CP) via a cutting-plane approach. The only allowance for real-world time restrictions consisted in replacing exhaustive search through the class F_P by a fast heuristic and to limit the number of iterations (i.e., executions of step (1)) in the algorithm to 100 (or 200 for the problem of Späth, 1985).

5. Computational results

In this section, we report about some of the computational results we obtained for three types of test problems: the 'classical' max-clique problem, problems with positive and negative weights on the edges and problems with only non-negative weights.

There is a philosophical problem about how test examples should be chosen: on the basis of random generated data or of 'real-world' data. If the widely held belief that random problems are 'easier' is true, we have given our algorithm a fair chance. For the classical max-clique problems, the weight function of course is predetermined while we have generated random graphs with various edge densities. For the problems on K_n, the edge weights are uniformly drawn within a given range. A critical case are the data for the location problem of Späth (1985): the original data were randomly generated; since we took them over identically we must consider them as 'real-world' data.

The parameter MAXCUT was typically set

$$\text{MAXCUT} = \begin{cases} 50 & \text{for } n = 10, \\ 75 & \text{for } n = 15, \\ 100 & \text{for } n = 20. \end{cases}$$

Table 1

n	Density	Problems		Total iterations	Total TZ	Total WS	Total P	Average facets
		Given	Solved					
10	25%	10	9	42	21	20	1	50
	50%	10	6	255	170	81	4	54
	75%	10	8	284	211	70	3	70
15	10%	10	8	62	27	34	1	71
	25%	10	0	314	187	129	1	72
	75%	3	0	300	294	6	0	112
20	10%	10	6	128	72	56	0	78
	25%	5	0	500	379	121	0	100

5.1. The classical max-clique problem

In the classical max-clique problem, we are given a graph $G=(V, E')$ on n nodes and are to find a subclique of G of maximal cardinality. We can model this problem on $\mathbb{P}_n(n)$ by interpreting E' as a subset of the edge set of the complete graphs $K_n=(V, E)$ and choosing the weight

$$w_e = \begin{cases} 1 & \text{for } e \in E', \\ -M & \text{for } e \in E \setminus E', \end{cases}$$

where M is 'large' (e.g. $M = \binom{n}{2}$).

For our test problems we obtained the results given in Table 1. Table 1 reports a problem as 'solved' only if the corresponding problem (CP) has been solved, i.e., only if the sequence of cutting plane relaxations $(\overline{LP})$ produces an integral optimal solution. Furthermore, an account of the total number of iterations of type $(\overline{LP})$ for each class of test problems is given, from which the average per test problem can easily be recovered. The total number of iterations is subdivided according to the types of violated constraints involved in the update for the next iteration $(\overline{LP})$. For example, for $n=10$ and density 50% of the test problems, 6 out of 10 problems were solved. The attempt to solve the 10 problems led to 225 LP's of type $(\overline{LP})$, 170 updates involved only triangle or Z-inequalities, 81 updates used W- or S-inequalities, and 4 updates included partition inequalities. The last column of Table 1 yields a measure for the average size of $(\overline{LP})$ during the algorithm.

In the cases ($n=15$, 75%) and ($n=20$, 25%) we stopped the test run after having observed that our algorithm had produced no solution within the set limit of 100 iterations on the first examples.

5.2. Positive and negative weights

We have generated random weights w_e in the range

$$-100 \le w_e \le +100$$

for 5 test examples of size $n=10$ and have applied the cutting plane algorithm to the parameter values $b=3$, 4, 10 in each of the five examples. In 12 of the resulting 15 test runs, the algorithm was able to find the optimal solution for the corresponding max-clique problem (CP).

The observed performance for examples of size $n=15$ was drastically worse. In this case, we considered random weights in the range

$$-500 \le w_e \le +500$$

for 5 test examples and tried to solve the max-clique problem for $b=3$, 4, 5. Only in 2 of the resulting 15 test runs was the algorithm successful!

5.3. Nonnegative weights

We consider 10 test examples of size $n=10$ and 10 test examples of size $n=15$. The weight range is

$$0 \le w_e \le 1000$$

and the results are given in Table 2. Notice again the drop in the performance of the algorithm as the problem size increases from $n=10$ to $n=15$. We also tried to run the algorithm for $n=15$ and

Table 2

n	b	Problems solved	Total iterations	Total TZ	Total WS	Total P	Average facets
10	3	7	126	70	48	8	30
	4	9	159	117	41	1	52
	5	8	151	108	39	4	78
	6	9	89	79	10	0	102
	7	9	72	66	6	0	109
	8	8	41	39	2	0	119
	9	10	29	28	1	0	105
15	3	4	116	64	52	0	47
	4	3	482	246	234	2	80
	5	5	781	580	201	0	78

$b \geq 6$. In each case, however, our limit of 100 iterations was reached and hence no result was obtained.

5.4. *A facility location problem*

In Späth (1985) a certain facility location problem is modelled as follows: in the complete graph $K_n = (V, E)$ with nonnegative edge weights d_e, $e \in E$, find a clique on b nodes of smallest possible edge weight.

If M is a strict upper bound on the d_e's we can formulate an equivalent model in our context. We work with the positive edge weights

$$w_e = M - d_e \quad (e \in E)$$

and try to solve the problem

$$\begin{array}{ll} \max & w \cdot x \\ \text{s.t.} & x \in \mathbb{P}_n(b). \end{array}$$

Späth (1985) gives a set of data for $n = 25$ (and $M = 1000$) to which he applies a simple exchange heuristic and thus achieves optimal solutions at least for the cases $b = 3, 4, 5, 6$ in Table 3.

When we used Späth's data in our algorithm, we were not so much interested in solving the associated integer programming problems (in fact, none was solved). We wanted to know how closely the objective function value of our LP-relaxation approximates the value of the true optimum. To achieve better performance, we permitted 200 iterations in each run of our algorithm. The results were as in Table 3.

Analyzing our computations results, it is apparent that a pure cutting-plane approach to the max-clique problem is not advisable even for small instances (enumeration of all cliques may be competitive in running time and offers the guarantee of finding the true optimum).

An alternative might be to embed a cutting-plane algorithm as a subroutine in a branch-and-bound procedure. For this to be successful, the cutting-plane subroutine should be time efficient.

A considerable speed-up could be achieved by basing the algorithm on the subset

$$F' = F_0 \cup F_1 \cup F_T \cup F_Z$$

of linear restrictions instead of F. It is interesting to note that F_P seems to hardly play any role for the computation in our test problems (the same phenomenon is observed in Wakabayashi (1986), where only 7 examples out of a total of 24 with $12 \leq n \leq 137$ make use of partition inequalities). Our results also suggest that the triangle and Z-inequalities are more important than the W- and S-inequalities.

On the other hand, Table 3 suggests that a cutting-plane algorithm may vastly overestimate the value of the true optimum and thus not lend itself to an efficient branch-and-bound procedure. From a practical point of view, we feel that the (edge-) weighted max-clique is currently beyond efficient exact solution methods even for very moderate problem instances. A cutting-plane approach seems to do little to improve that picture.

6. Discussion

The poor performance of a cutting-plane approach to the max-clique problem appears even

Table 3

b	TZ	WS	P	Facets	Iterations			Späth
					50	100	200	
3	3	5	0	85	2945	–	–	2853
4	18	13	0	101	5757	–	–	5471
5	76	51	2	112	9339	9258	9224	8478
6	147	53	0	130	13572	13422	13309	12325
7	164	36	0	187	18379	18139	17758	16559
8	164	36	0	257	23730	23384	23015	21025
9	177	23	0	361	29932	29224	28718	26235
10	189	11	0	334	35798	35295	34705	31508
11	195	5	0	401	43185	41620	41116	37834

more striking when contrasted with the performance of similar approaches to seemingly closely related problems.

Wakabayashi (1986) has investigated the (edge-) weighted partition problem (cf. Section 2) and based a cutting-plane algorithm on the sets of triangle and partition inequalities (in addition to the trivial inequalities). In all of the 24 test problems considered with n up to 137, the algorithm is able to produce the optimal (integral) solution. Moreover, in 17 cases already the triangle inequalities suffice to obtain the optimal solution. Acceptable results are also achieved in Faigle, Schrader and Suletzki (1987) for the b-restricted partition problem with n up to 70, where the cutting-plane method is incorporated into a branch-and-bound algorithm.

For the linear ordering problem (cf. Section 2), Marcotorchino and Michaud (1980) suggest a cutting-plane algorithm based on the acyclic triangle and the directed 2-cycle inequalities alone. They report about computational experience with n up to 72 and note that the cutting-plane algorithm may produce a non-integral solution – but "ce phénomène est assez rare et ne semble pas se produire dans les problèmes pratiques".

A computational study of the linear ordering problems is also done in Reinelt (1985) on examples with n up to 60. The cutting-plane approach there includes in addition to triangle and directed 2-cycle inequalities also so-called 'Möbius ladders'. It is observed that in most of the cases the cutting-plane algorithm yields the optimal integral solution. Moreover, only in a minority of examples, Möbius ladders are actually invoked. It is further found that the objective value of the LP-relaxation using only triangle and 2-cycle inequalities, in all instances furnishes a very good approximation for the true optimal objective function value.

Trying to explain why our cutting-plane approach to the max-clique problem fails already for small problem instances, it seems to us that the success of cutting-plane methods in general is very much problem dependent and that the max-clique problem is, in this sense, 'untractable' even for modest problem instances.

Another explanation, of course, could be that we simply have failed to detect the 'appropriate' cutting-plane for our problem. It should be interesting to see if there are other classes of facet inducing inequalities for $\mathbb{P}_n(b)$ that are simple enough to be incorporated into a computationally feasible algorithm with superior performance characteristics.

References

Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1954), "Solution of a large scale traveling-salesman problem", *Operations Research* 2, 393–410.

Dijkhuizen, G. (1989), "Een polydrische benadering van het maximale clique probleem", Master's Thesis, Faculty of Applied Mathematics, University of Twente (in Dutch).

Faigle, U. (1987), "The clique polytope of a graph", Memorandum No. 663, Faculty of Applied Mathematics, University of Twente.

Faigle, U., Schrader, R., and Suletzki, R. (1987), "A cutting-plane algorithm for optimal graph partitioning", *Methods of Operations Research* 57, 109–116.

Garey, M.R., and Johnson, D.S. (1979), *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, CA.

Grötschel, M. (1980), "On the symmetric travelling salesman problem: Solution of a 120-city problem", *Mathematical Programming Studies* 12, 61–77.

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

Grötschel, M., and Wakabayashi, Y. (1989), "A cutting plane algorithm for a clustering problem", *Mathematical Programming B* 45, 59 –96.

Jünger, M. (1985), *Polyhedral Combinatorics and the Acyclic Subgraph Problem*, Heldermann-Verlag, Berlin.

Lawler, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., and Shmoys, D.B. (1985), *The Traveling Salesman Problem – A Guided Tour of Combinatorial Optimization*, John Wiley, Chichester.

Marcotorchino, J.F., and Michaud, P. (1980), "Optimization en analyse des données relationnelles", in: E. Diday, L. Lebart, J.P. Pagès and R. Tomassone (eds.), *Data Analysis and Informatics*, North Holland, Amsterdam, 655–670.

Padberg, M., and Rinaldi, G. (1987), "Optimization of a 532-city symmetric travelling salesman problem by branch and cut", *Operations Research Letters* 6, 1–7.

Reinelt, G. (1985), *The Linear Ordering Problem: Algorithms and Applications*, Heldermann-Verlag, Berlin.

Späth, H. (1985), "Heuristic determining cliques of given cadinality and with minimal cost within weighted complete graphs", *Zeitschrift für Operations Research* 29, 125–131.

Wakabayashi, Y. (1986), "Aggregation of binary relations: Algorithmic and polyhedral investigations", Doctoral Dissertation, University of Augsburg.