COMBINATORICA 1(2) (1981) 169—197

THE ELLIPSOID METHOD AND ITS CONSEQUENCES IN COMBINATORIAL OPTIMIZATION

M. GRÖTSCHEL

Universität Bonn, Inst. für Ökonometrie und Operations Research
D—5300 Bonn, F.R.G.

L. LOVÁSZ

A. József University, Bolyai Institute, Szeged, Hungary H—6720

and

A. SCHRIJVER *

Inst. Actuariaat en Econometrie,
University of Amsterdam, Amsterdam, The Netherlands

Received 20 February 1980

L. G. Khachiyan recently published a polynomial algorithm to check feasibility of a system of linear inequalities. The method is an adaptation of an algorithm proposed by Shor for non-linear optimization problems. In this paper we show that the method also yields interesting results in combinatorial optimization. Thus it yields polynomial algorithms for vertex packing in perfect graphs; for the matching and matroid intersection problems; for optimum covering of directed cuts of a digraph; for the minimum value of a submodular set function; and for other important combinatorial problems. On the negative side, it yields a proof that weighted fractional chromatic number is NP-hard.

0. Introduction

A typical problem in combinatorial optimization is the following. Given a finite set S of vectors in $\mathbf{R}^n$ and a linear objective function c^Tx, find

$$\max\{c^Tx \mid x\in S\}. \tag{1}$$

Generally S is large (say exponential in n) but highly structured. For example, S may consist of all incidence vectors of perfect matchings in a graph. We are interested in finding the value of (1) by an algorithm whose running time is polynomial in n. Therefore, enumerating the elements of S is not a satisfactory solution.

The following approach was proposed by Edmonds [1965], Ford and Fulkerson [1962] and Hoffman [1960], and is the classical approach in combinatorial optimization. Let P denote the convex hull of S. Then clearly

$$\max\{c^Tx \mid x\in S\} = \max\{c^Tx \mid x\in P\}. \tag{2}$$

The right hand side here is a linear programming problem: maximize a linear objective function on a polytope. Of course, to be able to apply the methods of linear programming, we have to represent P as the set of solutions of a system of linear

* Research by the third author was supported by the Netherlands Organisation for the Advancement of Pure Research (Z.W.O.).

AMS subject classification (1980): 90 C XX, 05 C XX; 90 C 25, 90 C 10

inequalities. Such a representation, of course, always exists, but our ability to find the necessary inequalities depends on the structure of S. However, in many cases these inequalities (the facets of P) can be described. There are some beautiful theorems of this kind, e.g. Edmonds' [1965] description of the matching polytope. In these cases, the methods of linear programming can be applied to solve (1). However, until about a year ago there were two main obstacles in carrying out the above program even for nice sets S like the set of perfect matchings. First, no algorithm to solve linear programming with polynomial running time in the worst case was known. Second, the number of inequalities describing S is typically large (exponential in n) and hence even to formulate the linear program takes exponential space and time. Indeed, the well-known efficient combinatorial algorithms, like Edmonds' matching algorithm [1965] or Lucchesi's algorithm to find optimum coverings for directed cuts [1976] are based on different — ad hoc — ideas.

A recent algorithm to solve linear programs due to L. G. Khachiyan [1979], based on a method of Shor [1970], removes both difficulties. Its running time is polynomial; also, it is very insensitive to the number of constraints in the following sense: we do not need to list the faces in advance, but only need a subroutine which recognizes feasibility of a vector and if it is infeasible then computes a hyperplane separating it from P. Searching for such a hyperplane is another combinatorial optimization problem which is often much easier to solve. So this method, the *Ellipsoid Method,* reduces one combinatorial optimization problem to a second one. One might try to use this again to further transform the problem; but — interestingly enough — the method applied the second time leads back to the original problem. (However, sometimes after a simple transformation of this second problem the repeated application of the Ellipsoid Method may further simplify the problem.)

The main purpose of this paper is to exploit this equivalence between problems. After formulating the optimization problem in Chapter 1 exactly, we survey the Ellipsoid Method in Chapter 2. In Chapter 3 we prove the equivalence of the optimization and the separation problem, and their equivalence with other optimization problems. So we show that optimum dual solutions can be obtained by the method (since the dual problem has, generally in combinatorial problems, exponentially many variables, the method cannot be applied to the dual directly). Chapter 4 contains applications to the matching, matroid intersection, and branching problems, while in Chapter 5 we show how to apply the method to minimize a submodular set function and, as an application, to give algorithmic versions of some results of Edmonds and Giles [1977] and Frank [1979]. These include an algorithm to find optimum covering of directed cuts in a graph, solved first by Lucchesi [1976].

It is interesting to point out that these applications rely on the deep theorems characterizing facets of the corresponding polytope. This is in quite a contrast to previously known algorithms, which typically do not use these characterizations but quite often give them as a by-product.

The efficiency of the algorithms we give is polynomial but it seems much worse than those algorithms developed before. Even if we assume that this efficiency can be improved with more work, we do not consider it the purpose of our work to compete with the special-purpose algorithms. The main point is that the ellipsoid method proves the polynomial solvability of a large number of different combinatorial optimization problems at once, and hereby points out directions for the search for practically feasible polynomial algorithms.

Chapter 6 contains an algorithm to find maximum independent sets in perfect graphs. The algorithm makes use of a number $\vartheta(G)$ introduced by one of the authors as an estimation for the Shannon capacity of a graph (Lovász [1979]). Finally, in Chapter 7 we note that the vertex-packing problem of a graph is in a sense equivalent to the fractional chromatic number problem, and comment on the phenomenon that this latter problem is an example of a problem in NP which is NP-hard but (as for now) not known to be NP-complete.

1. Optimization on convex bodies: formulation of the problems and the results

Let K be a non-empty convex compact set in $\mathbf{R}^n$. We formulate the following two algorithmic problems in connection with K.

(1) *Strong optimization problem:* given a vector $c \in \mathbf{R}^n$, find a vector x in K which maximizes $c^T x$ on K.

(2) *Strong separation problem:* given a vector $y \in \mathbf{R}^n$, decide if $y \in K$, and if not, find a hyperplane which separates y from K; more exactly, find a vector $c \in \mathbf{R}^n$ such that $c^T y > \max \{c^T x \mid x \in K\}$.

Examples. Let K be the set of solutions of a system of linear inequalities

$$a_i^T x \leqq b_i \quad (i = 1, \ldots, m) \tag{3}$$

($a_i \in \mathbf{R}^n$, $b_i \in \mathbf{R}$). Then the strong separation problem can be solved trivially: we substitute $x = y$ in the constraints. If each of them is satisfied, $y \in K$. If constraint $a_i^T x \leqq b_i$ is violated, it yields a separating hyperplane. On the other hand, the optimization problem on K is just the linear programming problem.

As a second example, let K be given as the convex hull of a set $\{v_1, \ldots, v_m\}$ of points in $\mathbf{R}^n$. Then the optimization problem is easily solved by evaluating the objective function at each of the given points and selecting the maximum. On the other hand, to solve the separation problem we have to find a vector c in $\mathbf{R}^n$ such that

$$c^T y > c^T v_i \quad (i = 1, \ldots, m) \tag{4}$$

So this problem requires finding a feasible solution to a system of linear inequalities; this is again essentially the same as linear programming.

Note that the convex hull of $\{v_1, \ldots, v_m\}$ is, of course, a polytope and so it can be described as the set of solutions of a system of linear inequalities as well. But the number of these inequalities may be very large compared to m and n, and so their determination and the checking is too long. This illustrates that the solvability of the optimization and separation problems depends on the way K is given and not only on K.

We do not want to make any a priori arithmetical assumption on K. Thus it may well be that the vector in K maximizing $c^T x$ has irrational coordinates. In this case the formulation of the problem is not correct, since it is not clear how to state the answer. Therefore we have to formulate two weaker and more complicated, but more correct problems.

(5) *(Weak) optimization problem:* given a vector $c\in\mathbf{Q}^n$ and a number $\varepsilon>0$, find a vector $y\in\mathbf{Q}^n$ such that $d(y, K)\leqq\varepsilon$ and y almost maximizes c^Tx on K, i.e. for every $x\in K$, $c^Tx\leqq c^Ty+\varepsilon$. (Here $d(y, K)$ denotes the euclidean distance of y from K.)

(6) *(Weak) separation problem:* given a vector $y\in\mathbf{Q}^n$ and a number $\varepsilon>0$, conclude with one of the following: (i) asserting that $d(y, K)\leqq\varepsilon$; (ii) finding a vector $c\in\mathbf{Q}^n$ such that $\|c\|\geqq 1$ and for every $x\in K$, $c^Tx\leqq c^Ty+\varepsilon$.

We shall always assume that we are given a point a_0 and $0<r\leqq R$ such that

(7) $$S(a_0, r)\subseteqq K\subseteqq S(a_0, R),$$

where $S(a_0, r)$ denotes the euclidean ball of radius r about a_0. The second inclusion here simply means that K is bounded, where a bound is known explicitly; this is quite natural to assume both in theoretical and in (possible) practical applications. The first assumption, namely that K contains an explicit ball, is much less natural and we make it for purely technical reasons. What it really means is that K is full-dimensional, or at least we are given the affine subspace it spans and also that we are given a ball in this subspace contained in K. At the end of Chapter 3 we shall show that some assumption like this must be made.

So we define a *convex body* as a quintuple (K, n, a_0, r, R) such that $n\geqq 2$, K is a convex set in $\mathbf{R}^n$, $a_0\in K$, $0<r\leqq R$ and (7) is satisfied.

Let $\mathcal{K}$ be a class of compact convex bodies. We assume that each $K\in\mathcal{K}$ has some encoding. An *input* of the optimization problem for $\mathcal{K}$ is then the code of some member K of $\mathcal{K}$, a vector $c\in\mathbf{Q}^n$, and a number $\varepsilon>0$. Inputs of the other problems are defined similarly. The *length* of the input is defined in the (usual) binary encoding. Thus the length of the input is at least $n+|\log r|+|\log R|+|\log \varepsilon|$. An algorithm to solve the optimization problem for the class $\mathcal{K}$ is called *polynomial* if its running time is bounded by some polynomial of the size of the input.

The fact that the running time must be polynomial in $|\log\varepsilon|$ is crucial: it means that running the algorithm for $\varepsilon=1/2, 1/4, \ldots$ we get a sequence of approximations which converge exponentially fast in the running time. Other approximation algorithms for linear programming (Motzkin and Schoenberg [1954]) have only polynomial convergence speed. This exponential convergence rate enables Khachiyan to obtain an exact optimum in polynomial time (essentially by rounding) and us to give the combinatorial applications in this paper.

2. The ellipsoid method

Let us first describe the simple geometric idea behind the method. We start with a convex body K in $\mathbf{R}^n$, included in a ball $S(a_0, R)=E_0$, and a linear objective function c^Tx. In the k-th step there will be an ellipsoid E_k, which includes the set K_k of those points x of K for which c^Tx is at least as large as the best found so far. We look at the centre x_k of E_k. If x_k is not an element of K, then we take a hyperplane through x_k which avoids K. This hyperplane H cuts E_k into two halves; we pick that one which includes K_k and include it in a new ellipsoid E_{k+1}, which is essentially the ellipsoid of least volume containing this half of E_k, except for an allowance for rounding errors. Geometrically, this smallest ellipsoid can be described as follows. Let $F=E_k \cap H$, and let y be the point where a hyperplane parallel to H touches our half of E_k. Then the centre of this smallest ellipsoid divides the segment $x_k y$ in ratio $1:n$, the ellipsoid intersects H in F, and touches E_k in y. E_{k+1} then arises by blowing up and rounding. If $x_k \in K$, then we cut with the hyperplane $c^T x = c^T x_k$ similarly. The volumes of the ellipsoids E_k will tend to 0 exponentially and this guarantees that those centres x_k which are in K will tend to an optimum solution exponentially fast.

In what follows, let $\|x\|$ denote the euclidean norm of the vector x and let $\|A\|$ denote the norm of the matrix A, i.e.

$$\|A\| = \max \{\|Ax\| : \|x\| = 1\}.$$

For symmetric matrices, $\|A\|$ is the maximum absolute value of the eigenvalues of A, and also $\max \{|x^T Ax| : \|x\|=1\}$.

We turn to the exact formulation of the procedure. Let $K \subset \mathbf{R}^n$ be a compact convex set, $S(a_0, r) \subseteq K \subseteq S(a_0, R)$, $c^T x$ a linear objective function and $\varepsilon > 0$. Without loss of generality, assume that $\varepsilon < r$, $\|c\| \geqq 1$, and $n \geqq 2$. Assume that there is a subroutine SEP to solve the (weak) separation problem for K. This means that given a vector $y \in \mathbf{Q}^n$ and $\delta > 0$, SEP either concludes that $y \in S(K, \delta)$ or yields a vector d such that

$$\max \{d^T x \mid x \in K\} \leqq d^T y + \delta. \tag{1}$$

To solve the weak optimization problem on K we run the following algorithm. Let

$$N = 4n^2 \left\lceil \log \frac{2R^2\|c\|}{r\varepsilon} \right\rceil, \tag{2}$$

$$\delta = \frac{R^2 4^{-N}}{300n}, \tag{3}$$

and

$$p = 5N. \tag{4}$$

We now define a sequence $x_0, x_1, \ldots$ of vectors and a sequence $A_0, A_1, \ldots$ of positive definite matrices as follows. Let $x_0 = a_0$ and $A_0 = R^2 I$. Assuming that x_k, A_k are defined, we run the subroutine SEP with $y = x_k$ and δ. If it concludes that $x_k \in S(K, \delta)$ we say that k is a *feasible index*, and set $a = c$. If SEP yields a vector $d \in \mathbf{R}^n$ such that $\|d\| \geqq 1$ and

$$\max \{d^T x : x \in K\} \leqq d^T x_k + \delta, \tag{5}$$

then we call k an *infeasible index* and let $a = -d$. Next define

$$b_k = A_k a / \sqrt{a^T A_k a}, \tag{6}$$

$$x_k^* = x_k + \frac{1}{n+1} b_k, \tag{7}$$

$$A_k^* = \frac{2n^2+3}{2n^2} \left(A_k - \frac{2}{n+1} b_k b_k^T \right). \tag{8}$$

(With $\frac{n^2}{n^2-1}$ instead of $\frac{2n^2+3}{2n^2}$, we would get the smallest ellipsoid including the appropriate half of E_k; see Gács and Lovász [1979]. Here we take this larger factor because of rounding errors.) Further let

$$x_{k+1} \approx x_k^*, \quad \text{and} \quad A_{k+1} \approx A_k^*, \tag{9}$$

where the sign $\approx$ means that the left hand side is obtained by rounding the right hand side to p binary digits behind the decimal point, taking care that A_{k+1} is symmetric.

The sequence (x_k), k feasible, will give good approximations for the optimum solution of our problem. To prove this, we shall need some lemmas, which will also illuminate the geometric background of the algorithm.

First we introduce some further notation. Let

$$E_k = \{x \in \mathbf{R}^n \mid (x-x_k)^T A_k^{-1}(x-x_k) \leqq 1\}, \tag{10}$$

and

$$E_k^* = \{x \in \mathbf{R}^n \mid (x-x_k^*)^T A_k^{*-1}(x-x_k^*) \leqq 1\}. \tag{11}$$

(2.1) Lemma. *The matrices $A_0, A_1, \dots, A_N$ are positive definite. Moreover,*

$$\|x_k\| \leqq \|a_0\| + R2^k, \quad \|A_k\| \leqq R^2 2^k, \quad \textit{and} \quad \|A_k^{-1}\| \leqq R^{-2}4^k. \tag{12}$$

Proof. By induction on k. For $k=0$ all the statements are obvious. Assume that they are true for k. Then note first that

$$(A_k^*)^{-1} = \frac{2n^2}{2n^2+3}\left(A_k^{-1} + \frac{2}{n-1}\cdot\frac{aa^T}{a^T A_k a}\right), \tag{13}$$

as it is easy to verify by computation, and hence A_k^* is positive definite. Using this it follows easily that

$$\|A_k^*\| = \frac{2n^2+3}{2n^2}\left\|A_k - \frac{2}{n+1} b_k b_k^T\right\| \leqq \frac{2n^2+3}{2n^2}\|A_k\| \leqq \left(1+\frac{3}{2n^2}\right) R^2 2^k, \tag{14}$$

and so

$$\|A_{k+1}\| \leqq \|A_k^*\| + \|A_{k+1} - A_k^*\| \leqq \left(1+\frac{3}{2n^2}\right) R^2 2^k + n2^{-p} \leqq R^2 2^{k+1}. \tag{15}$$

Further,

$$\|b_k\| = \frac{\|A_k a\|}{\sqrt{a^T A_k a}} = \sqrt{\frac{a^T A_k^2 a}{a^T A_k a}} \leqq \sqrt{\|A_k\|} \leqq R2^k, \tag{16}$$

and so

$$\begin{aligned} \|x_{k+1}\| &\leqq \|x_k\| + \frac{1}{n+1}\|b_k\| + \|x_{k+1} - x_k^*\| \leqq \|a_0\| + R2^k + \frac{1}{n+1}R2^k + \sqrt{n}\,2^{-p} \\ &\leqq \|a_0\| + R2^{k+1}. \end{aligned} \tag{17}$$

Further, by (13),

$$\|(A_k^*)^{-1}\| \leqq \frac{2n^2}{2n^2+3}\left(\|A_k^{-1}\| + \frac{2}{n-1}\cdot\frac{\|a\|^2}{a^TA_ka}\right) \tag{18}$$

$$\leqq \frac{2n^2}{2n^2+3}\left(\|A_k^{-1}\| + \frac{2}{n-1}\|A_k^{-1}\|\right) < \frac{n+1}{n-1}\|A_k^{-1}\|.$$

Let λ_0 denote the least eigenvalue of A_{k+1} and let v be a corresponding eigenvector, $\|v\|=1$. Then

$$\lambda_0 = v^TA_{k+1}v = v^TA_k^*v+v^T(A_{k+1}-A_k^*)v \geqq \|(A_k^*)^{-1}\|^{-1}-\|A_{k+1}-A_k^*\| \tag{19}$$

$$\geqq \frac{n-1}{n+1}\|A_k^{-1}\|^{-1}-n2^{-p} \geqq \frac{n-1}{n+1}R^24^{-k}-n2^{-p} > R^24^{-(k+1)}.$$

This proves that A_{k+1} is positive definite and also that

$$\|A_{k+1}^{-1}\| = 1/\lambda_0 \leqq R^{-2}4^{k+1}. \quad \blacksquare \tag{20}$$

(2.2) Lemma. *Let μ denote the n-dimensional volume. Then*

$$\frac{\mu(E_{k+1})}{\mu(E_k)} < e^{-1/5n}. \tag{21}$$

Proof. The volume of E_k^* can be calculated from the volume of E_k; this is easy if one notices that it suffices to consider the case $A_k=I$. Rounding errors can be estimated similarly as in the previous proofs. $\blacksquare$

Set

$$\zeta_k = \max\{c^Tx_j | 0 \leqq j < k, j \textit{ feasible}\}, \tag{22}$$

and

$$K_k = K\cap\{x | c^Tx \geqq \zeta_k\}. \tag{23}$$

(2.3) Lemma. $E_k \supseteqq K_k$, *for* $k=0, 1, \ldots, N$.

Proof. By induction on k. For $k=0$ the assertion is obvious. Let $x \in K_{k+1}$. Then

$$x \in K_k \subseteqq E_k, \tag{24}$$

and also

$$a_k^Tx \geqq a_k^Tx_k - \delta, \tag{25}$$

where a_k equals the auxiliary vector a used in step k (if k is a feasible index we do not even have the δ here). Write

$$x = x_k+y+tb_k, \tag{26}$$

where $a_k^Ty=0$ (since b_k and a_k are not perpendicular because of the positive definiteness of A_k, such a decomposition of x always exists). By (24),

$$1 \geqq (y+tb_k)^TA_k^{-1}(y+tb_k) = y^TA_k^{-1}y+t^2b_k^TA_k^{-1}b_k = y^TA_k^{-1}y+t^2. \tag{27}$$

Hence $t \leqq 1$. On the other hand, (25) yields

$$-\delta \leqq ta_k^T b_k = t\sqrt{a_k^T A_k a_k}. \tag{28}$$

Now we have

$$(x-x_{k+1})^T A_{k+1}^{-1}(x-x_{k+1}) \leqq (x-x_k^*)A_k^{*-1}(x-x_k^*)+R_1, \tag{29}$$

where the remainder term R_1 can be estimated easily by similar methods to those in the proof of Lemma (2.1), and it turns out that $R_1 < 1/12n^2$. For the main term we have by (13), (26), (27), (28) and (12):

$$\begin{aligned}(x-x_k^*)^T A_k^{*-1}(x-x_k^*) &= \frac{2n^2}{2n^2+3}\left[\left(t-\frac{1}{n+1}\right)b_k+y\right]^T\cdot \\ &\quad\cdot\left(A_k^{-1}+\frac{2}{n-1}\cdot\frac{a_k a_k^T}{a_k^T A_k a_k}\right)\left[\left(t-\frac{1}{n+1}\right)b_k+y\right] \\ &= \frac{2n^2}{2n^2+3}\left(\left(t-\frac{1}{n+1}\right)^2+y^T A_k^{-1}y+\frac{2}{n-1}\left(t-\frac{1}{n+1}\right)^2\right) \\ &\leqq \frac{2n^2}{2n^2+3}\left(\frac{n^2}{n^2-1}-\frac{2t(1-t)}{n-1}\right) \\ &\leqq \frac{2n^4}{2n^4+n^2-3}+\frac{4\delta}{(n-1)\sqrt{a_k^T A_k a_k}}\frac{2n}{2n^4+n^2-3}+\frac{4\delta}{n-1}\|A_k^{-1}\| \\ &\leqq \frac{2n^4}{2n^4+n^2-3}+\frac{4\delta R^{-2}4^N}{n-1} \leqq 1-\frac{1}{12n^2}.\end{aligned} \tag{30}$$

Hence $(x-x_{k+1})^T A_{k+1}^{-1}(x-x_{k+1}) \leqq 1$, and so $x \in E_{k+1}$. ∎

Now we are able to prove the main theorem in this section.

(2.4) Theorem. *Let j be a feasible index for which*

$$c^T x_j = \max\{c^T x_k \mid 0 \leqq k < N,\ k \text{ feasible}\}. \tag{31}$$

Then $c^T x_j \geqq \max\{c^T x \mid x \in K\} - \varepsilon$.

Proof. Let us observe first that Lemmas (2.2) and (2.3) imply that

$$\mu(K_N) \leqq \mu(E_N) \leqq e^{-N/4n}\mu(E_0) = e^{-N/4n}R^n V_n, \tag{32}$$

where V_n is the volume of the n-dimensional unit ball. On the other hand, let

$$\zeta = \max\{c^T x \mid x \in K\} \tag{33}$$

and $y \in K$ such that $c^T y = \zeta$. Consider the cone whose base is the $(n-1)$-dimensional ball of radius r and centre x_0 in the hyperplane $c^T x = c^T x_0$ and whose vertex is

y. The piece of this cone in the half-space $c^T x \geqq c^T x_j$ is contained in K_N. The volume of this piece is

$$\frac{V_{n-1} \cdot r^{n-1} \cdot (\zeta - c^T x_0)}{n\|c\|} \left(\frac{\zeta - c^T x_j}{\zeta - c^T x_0}\right)^n \leqq \mu(K_N) \leqq e^{-N/4n} R^n V_n. \tag{34}$$

Hence

$$\zeta - c^T x_j \leqq e^{-N/4n^2} R \left(\frac{\zeta - c^T x_0}{r}\right)^{\frac{n-1}{n}} \left(\frac{nV_n}{V_{n-1}}\right)^{1/n} \|c\|^{1/n}. \tag{35}$$

We still need an upper bound on ζ. Since

$$|\zeta - c^T x_0| = |c^T (y - x_0)| \leqq \|c\| \cdot \|y - x_0\| \leqq R\|c\|, \tag{36}$$

we finally have

$$\zeta - c^T x_j < 2e^{-N/4n^2} \frac{R^2}{r} \|c\| \leqq \varepsilon. \quad \blacksquare \tag{37}$$

3. Equivalence of optimization and other problems

First we prove the equivalence of the (weak) separation problem and the (weak) optimization problem, for any given K. More exactly, this means the following.

(3.1) Theorem. *Let $\mathscr{K}$ be a class of convex bodies. There is a polynomial algorithm to solve the separation problem for the members of $\mathscr{K}$, if and only if there is a polynomial algorithm to solve the optimization problem for the members of $\mathscr{K}$.*

A class $\mathscr{K}$ such that there exists a polynomial algorithm to solve the optimization problem (or the separation problem) for members of $\mathscr{K}$ will be called *solvable*.

Proof. (I) The "only if" part. In view of the results of Chapter 2, the only thing to check is that the algorithm described there is polynomial-bounded. This follows since by assumption, the subroutine SEP is polynomial, hence the number of digits in the entries of a is polynomial and so the computation of x_{k+1} and A_{k+1} requires only a polynomial number of steps. All other numbers occurring have only a polynomial number of digits, by Lemma (2.1). The number of iterations is also polynomial. Hence the algorithm runs in polynomial time.

(II) The "if" part. Without loss of generality assume that $a_0 = 0$. Let K^* be the polar of K, i.e.,

$$K^* = \{u \mid u^T x \leqq 1 \text{ for each } x \in K\}. \tag{1}$$

It is well-known that K^* is a convex body, $(K^*)^* = K$, and

$$S(0, 1/R) \subseteq K^* \subseteq S(0, 1/r). \tag{2}$$

If $\mathscr{K}$ is a class of convex bodies with $a_0 = 0$, let $\mathscr{K}^* = \{K^* \mid K \in \mathscr{K}\}$.

(3.2) Lemma. *The separation problem for a class $\mathcal{K}^*$ of convex bodies with $a_0=0$ is polynomially solvable iff the optimization problem is polynomially solvable for the class $\mathcal{K}$.*

Since $(\mathcal{K}^*)^*=\mathcal{K}$, this lemma immediately implies the "if" part of the theorem: if the optimization problem is polynomially solvable for $\mathcal{K}$ then the separation problem is polynomially solvable for $\mathcal{K}^*$. But then by part (I), the optimization problem is polynomially solvable for $\mathcal{K}^*$ and so using the lemma again, it follows that the separation problem is polynomially solvable for $\mathcal{K}$.

Proof of the Lemma. (I) The "if" part. Let $K^*\in\mathcal{K}^*$, $v\in\mathbf{R}^n$ and $\varepsilon>0$. Using the optimization subroutine for K, with objective function v and error εr, we get a vector $z\in\mathbf{R}^n$ such that $d(z, K)\leqq\varepsilon r$, and

$$v^T z \geqq \max\{v^T x \mid x\in K\}-\varepsilon r. \tag{3}$$

Now if $v^T z\leqq 1$ then $v^T x\leqq 1+\varepsilon r$ and hence $v_0=\frac{1}{1+\varepsilon r}v\in K^*$. Therefore $\|v_0\|\leqq 1/r$, whence $d(v, K^*)\leqq\|v-v_0\|\leqq\varepsilon$.

On the other hand, if $v^T z>1$ then z is a solution of the separation problem for K^*. In fact, let $z_0\in K$ such that $\|z-z_0\|\leqq\varepsilon r$. Then for every $u\in K^*$, $\|u\|\leqq 1/r$, and so

$$z^T u = (z-z_0)^T u+z_0^T u \leqq \|u\|\cdot\|z-z_0\|+1 \leqq \varepsilon+z^T v, \tag{4}$$

which proves that z is a solution of the separation problem for K^*.

(II) The "only if" part follows by the "if" part of the Theorem (which we already know). ∎

Let $\mathcal{K}$ and $\mathcal{L}$ be two classes of convex bodies. Define

$$\mathcal{K}\wedge\mathcal{L} = \{K\cap L \mid K\in\mathcal{K},\ L\in\mathcal{L},\ \dim K=\dim L,\ a_0(K)=a_0(L)\}. \tag{5}$$

(3.3) Corollary. *If $\mathcal{K}$ and $\mathcal{L}$ are solvable then so is $\mathcal{K}\wedge\mathcal{L}$.*

Proof. The separation problem for $\mathcal{K}\wedge\mathcal{L}$ goes trivially back to the separation problems for $\mathcal{K}$ and $\mathcal{L}$. ∎

(3.4) Corollary. *Let $\mathcal{K}$ be a class of convex bodies with $a_0=0$. Then $\mathcal{K}$ is solvable iff $\mathcal{K}^*$ is solvable.*

The proof is trivial by Lemma (3.2).

Let $\mathbf{R}_+^n$ be the non-negative orthant in $\mathbf{R}^n$. Next we study convex bodies K such that there are $\varrho>0$, $R>0$ with

$$\mathbf{R}_+^n\cap S(0,\varrho)\subseteq K\subseteq \mathbf{R}_+^n\cap S(0,R), \tag{6}$$

and if $x\in K$, $0\leqq y\leqq x$, then $y\in K$.
The *anti-blocker* of K is defined by

$$A(K) = \{y\in\mathbf{R}_+^n \mid y^T x\leqq 1 \text{ for every } x\in K\}. \tag{7}$$

Moreover, $A(\mathcal{K})=\{A(K)\mid K\in\mathcal{K}\}$.

(3.5) Corollary. *Let $\mathcal{K}$ be a class of convex bodies satisfying* (6). *Then $\mathcal{K}$ is solvable iff $A(\mathcal{K})$ is solvable.*

The proof is the same as that of Lemma (3.2).

Next we want to show that without the assumption that K contains a ball, or even without the explicit knowledge of this ball, there is no algorithm at all to solve the optimization problem. More exactly, we consider a class of full-dimensional convex bodies in $\mathbf{R}^2$, and assume that for each of these bodies there is an "oracle" or "black box", which, if we plug in a point $y \in \mathbf{R}^2$, tells us whether or not y is contained in the body and if not, prints out a separating hyperplane in the sense of the strong separation problem. We may require that the output is always rational but may allow arbitrary real vectors as input. Now we are given one of the black boxes together with the information that $a_0=0$ is contained in the body described by this black box, and that it is contained in some disc about 0. We are also given a linear objective function $c^T x$ which we would like to maximize in the sense of the weak optimization problem. Note that if, in addition, we would be given an $r>0$ and the information that the body contains the disc of radius r about 0, then this problem could be solved: this is just the contents of Theorem (3.1). However, we are going to show by an "adversary" argument that there is no algorithm at all to solve the weak optimization problem if no ball contained in the bodies is explicitely known.

Let $L(t, \varphi)$ denote the segment of length t which ends in 0 and forms an angle of φ with the positive half of the x-axis. Let $K(t, \varphi, r)$ denote the neighborhood of $L(t, \varphi)$ of radius r. We assume that $0<t\leqq 1$, $0<r\leqq 1$, and $0\leqq\varphi\leqq 90°$, $r, t, \varphi \in \mathbf{Q}$. Our adversary designs one or more black boxes for every t, φ, and r; he does this so — and generously tells this to us — that if $y \notin K(t, \varphi, r)$ then the machine in the box checks whether or not $y \in K(1, \varphi, r)$ and if the answer is "no" the separating line it constructs will also separate y from $K(1, \varphi, r)$. Otherwise, the separating algorithm in the box may be arbitrary.

Assume that we are given a box and the information that it is one of the boxes constructed above. Then we know that the convex body described by the box is contained in the disc of radius 2 about 0 and that it contains 0 in its interior. Suppose that we have an algorithm to solve the weak optimization problem with (say) the objective function $x+y$. By a usual argument, this algorithm must also work if our adversary is allowed to exchange one black box for the other during the run of the algorithm, provided this second black box would have given the same answers to the previous questions as the one used so far. So if we describe a strategy for the adversary to switch boxes so that after an arbitrary number of steps there are still two black boxes which would have given the same answers as obtained previously but for which the maxima of $x+y$ over the corresponding convex bodies are essentially different (by more than 1/2, say), then the counterstrategy of the adversary is succesful and our algorithm is wrong.

Now the strategy of our adversary is the following: he always gives us a box for some $K(1, \varphi, r)$ and such that all the previously checked points are outside $K(1, \varphi, r)$ (except for 0, if we had happened to ask this superfluous question). It is easy to see that he can do this: if we ask a point outside the current $K(1, \varphi, r)$ he does not have to interfere, and if we ask a point inside, he can construct a $K(1, \psi, s)$ which is contained inside $K(1, \varphi, r)$ but avoids y, and can replace the current black box by a black box for $K(1, \psi, s)$.

It is clear that at each step, not only the current $K(1, \varphi, r)$ but also $K(t, \varphi, r)$ is compatible with all the previous answers, for every $0\leqq t\leqq 1$, but the maxima of

$x+y$ over $K(1, \varphi, r)$ and $K\left(\frac{1}{4}, \varphi, r\right)$ differ by more than 1/2. This completes the proof.

Let us remark that the same argument would show that the weak optimization problem is not solvable algorithmically, using a strong separation oracle, for the class of non-full-dimensional bounded convex bodies.

Finally we show that for polytopes many of the results are even nicer. By a *rational polytope* we mean a quadruple $(P;\ n, a_0, T)$ where P is a full-dimensional polytope (in $\mathbf{R}^n$), $a_0 \in \operatorname{Int} P$, and every component of a_0 as well as of every vertex of P is a rational number with numerator and denominator not exceeding T in absolute value. (This definition is much in the spirit of our previous discussion: the vertices of P must be rational in order to be able to explicitly present them and explicit bounds must be known for their complexity.)

(3.6) Theorem. *Let $(P;\ n, a_0, T)$ be a rational polytope. Then $S(a_0, r) \subseteq P \subseteq S(a_0, R)$, where $R=2nT$ and $r=(2T)^{-n^2-n}$. Furthermore, every facet of P can be written as $a^T x \leqq b$, where a ($\neq 0$) is an integral vector, b is an integer, and the entries of a as well as b are less than $T'=(nT)^n$.*

Thus every rational polytope can be viewed as a convex body, with r and R as above. A certain converse of this assertion holds as well.

(3.7) Theorem. *Let $P \subseteq \mathbf{R}^n$ be a polytope, $a_0 \in \operatorname{Int} P$, and assume that every component of a_0 is a rational number with numerator and denominator less than T in absolute value. Also assume that every facet of P can be written as $a^T x \leqq b$, where a ($\neq 0$) is an integral vector, b is an integer and the entries of a as well as b are less than T in absolute value. Then $(P;\ n, a_0, T')$ is a rational polytope where $T'=(nT)^n$.*

The *proof* of these two theorems is rather straightforward arithmetic and is omitted (cf. Lemmas 1—2 in Gács and Lovász [1981]).

(3.8) Theorem. *Let $\mathcal{K}$ be a class of rational polytopes. Suppose that $\mathcal{K}$ is solvable. Then the strong optimization problem and the strong separation problem are solvable for $\mathcal{K}$ in time polynomial in n,* log T, *and* log S, *where S is the maximum of the absolute values of the numerators and denominators occurring in c (in y, respectively).*

Proof. Let $(P;\ n, a_0, T) \in \mathcal{K}$, $c \in \mathbf{Z}^n$, $Q=2T^{2n+1}$ and

$$d = Q^n c + (1, Q, \ldots, Q^{n-1})^T. \tag{10}$$

We prove that $\max\{d^T x \mid x \in P\}$ is attained at a unique vertex of P and that this vertex maximizes $c^T x$ as well.

For let x_0 be a vertex of P maximizing $d^T x$ and let x_1 be another vertex. Write

$$x_0 - x_1 = \frac{1}{\alpha} z, \tag{11}$$

where $0 < \alpha < T^{2n}$ is an integer and $z=(z_1, \ldots, z_n)^T$ is an integral vector with $|z_i| < 2T^{2n+1} = Q$. Then

$$0 \leqq d^T(x_0 - x_1) = \frac{1}{\alpha}\left\{Q^n c^T z + \sum_{j=1}^{n} Q^{j-1} z_j\right\}. \tag{12}$$

Here $c^Tz \geqq 0$, since it is an integer and if $c^Tz \leqq -1$, then the first term in the bracket is larger in absolute value than the second. Hence

$$\frac{1}{\alpha} c^T z = c^T(x_0 - x_1) \geqq 0 \tag{13}$$

for every vertex x_1, i.e., x_0 indeed maximizes the objective function c^Tx over P. Also note that the second term in (12) is non-zero since $z \neq 0$. Hence

$$d^T(x_0 - x_1) \geqq \frac{1}{\alpha} \geqq \frac{1}{T^{2n}} \tag{14}$$

and so x_0 is the unique vertex of P maximizing the objective function d^Tx.

Now use the hypothesized polynomial algorithm to find a vector $y \in \mathbf{R}^n$ such that

$$d(y, P) \leqq \varepsilon = \frac{1}{10} \|d\|^{-1} n^{-1/2} T^{-2n-3} \tag{15}$$

and $d^Ty \geqq d^Tx_0 - \varepsilon$. We claim that

$$\|y - x_0\| \leqq \frac{1}{2T^2}. \tag{16}$$

For let y_0 be the point of P next to y. Represent y_0 as a convex combination of $n+1$ vertices of P, one of which is x_0:

$$y_0 = \sum_{i=0}^{n} \lambda_i x_i, \quad \lambda_i \geqq 0, \quad \sum_{i=0}^{n} \lambda_i = 1. \tag{17}$$

Then by (14)

$$d^Ty = d^T(y - y_0) + d^Ty_0 \leqq \varepsilon\|d\| + \sum_{i=0}^{n} \lambda_i d^T x_i \leqq \varepsilon\|d\| + d^Tx_0 - \frac{1-\lambda_0}{T^{2n}}. \tag{18}$$

Hence

$$\frac{1-\lambda_0}{T^{2n}} \leqq \varepsilon(\|d\|+1) \leqq 2\varepsilon\|d\| \tag{19}$$

and

$$\|y - x_0\| \leqq \|y - y_0\| + \|y_0 - x_0\| \leqq \varepsilon + (1-\lambda_0)\left\| \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_0} x_i - x_0 \right\| \tag{20}$$

$$\leqq \varepsilon + (1-\lambda_0) 2\sqrt{n}T \leqq \varepsilon + 4\varepsilon\|d\|\sqrt{n}T^{2n+1} \leqq 1/2T^2.$$

Now it is rather clear how to conclude: round each entry of y to the next rational number with denominator less than T; the resulting vector is x_0. The rounding can be done by using the technique of continued fractions. We leave the details to the reader.

The separation algorithm can be obtained by applying the previous algorithm to P^* (assuming that $a_0 = 0$, possibly after translation). ∎

If the strong separation problem concludes that $y \in P$ then it is nice to have a "proof" of that, i.e., a representation of y as a convex combination of vertices of P. This problem can also be solved.

(3.9) Theorem. *Let $\mathcal{K}$ be a solvable class of rational polytopes. Then there exists an algorithm which, given $(P;\, n, a_0, T)\in\mathcal{K}$ and a rational vector $y\in P$, yields vertices $x_0, x_1, \dots, x_n$ of P and coefficients $\lambda_0, \lambda_1, \dots, \lambda_n \geqq 0$ such that $\lambda_0+\lambda_1+\dots+\lambda_n=1$* and $\lambda_0 x_0+\lambda_1 x_1+\dots+\lambda_n x_n=y$, *in time polynomial in n,* $\log T$ *and* $\log S$, *where S is the maximum absolute value of numerators and denominators of components of y.*

Proof. We construct a sequence $x_0, x_1, \dots, x_n$ of vertices, $y_0, y_1, \dots, y_n$ of points and $F_1, F_2, \dots, F_n$ of facets of P as follows. Let x_0 be any vertex of P, and let $y_0=y$. Assume that x_i, y_i and F_i are defined for $i\leqq j$. Let y_{j+1} be the last point of P on the semi-line from x_j through y_j, let

$$y'_{j+1} = y_{j+1}+\varepsilon(y_j-x_j) \tag{21}$$

where $0<\varepsilon<(nT)^{-3n^2}$. Let F_{j+1} be a facet separating y'_{j+1} from P, and let x_{j+1} be a vertex of $F_1\cap\dots\cap F_{j+1}$. It is straightforward to prove by induction that $x_i, y_i\in F_j$ for $j\geqq i$, $y\in\operatorname{conv}(x_0, \dots, x_i, y_i)$, and $\dim(F_1\cap\dots\cap F_j)=n-j$. Hence $x_n=y_n$ and so y is contained in the convex hull of $x_0, \dots, x_n$.

The procedure described above is easy to follow with computation. The vertex of $F_1\cap\dots\cap F_j$ can be obtained as follows. Let $a_i^T x\leqq b_i$ be the inequality corresponding to facet F_i; then maximize the objective function $\left(\sum_{i=1}^{j} a_i\right)^T x$. We leave the details to the reader. ∎

The "dual" form of this theorem will also play an important role in the sequel. It shows that if we consider optimization on P as a linear program, an optimal dual basic solution can be found in polynomial time, if the class is solvable.

(3.10) Theorem. *Let $\mathcal{K}$ be a solvable class of rational polytopes. Then there exists a polynomial-bounded algorithm which, given $(P;\, n, a_0, T)\in\mathcal{K}$, $c\in\mathbb{Q}^n$ provides facets $a_i^T x\leqq b_i$ $(i=1, \dots, n)$ and rationals $\lambda_i\geqq 0$ $(i=1, \dots, n)$ such that $\sum_{i=1}^{n}\lambda_i a_i=c$ and $\sum_{i=1}^{n}\lambda_i b_i=\max\{c^T x \mid x\in P\}$.*

The *proof* is easy by considering $\mathcal{K}^*$.

4. Matroid intersection, branchings and matchings

We now apply the methods described in the previous chapters to a number of combinatorial problems. As said in the introduction our main aim is to show the *existence* of polynomial algorithms for certain combinatorial problems, and these algorithms are not meant as substitutes for the algorithms developed for these problems before (see Lawler [1976] for a survey). However, in the next chapters we shall show the existence of polynomial algorithms also for certain problems which were not yet solved in this sense. The algorithms found there, though polynomial, in general do not seem to have the highest possible rate of efficiency, and the challenge remains to find better algorithms.

First we apply the ellipsoid method to matroid intersection (cf. Edmonds [1970, 1979], Lawler [1970]). Note that given a matroid (V, r), the corresponding *matroid polytope* is the convex hull of the characteristic vectors of independent sets. The idea is very simple: given an integral "weight" function w on V, the trivial "greedy algorithm" finds an independent set V' maximizing $\sum_{v \in V'} w(v)$. That is, it finds a vertex x of the corresponding matroid polytope maximizing the objective function $w^T x$, in time bounded by a polynomial in $|V|$ and $\log \|w\|$. So the class of matroid polytopes is solvable. Therefore, by Corollary (3.3) also intersections of matroid polytopes are solvable. Since the intersection of two matroid polytopes has integer vertices again, this provides us with a polynomial algorithm for matroid intersection. (In fact, we obtain a polynomial algorithm for common "fractional" independent sets for any number of matroids.) Obviously, we may replace "matroid" by "polymatroid". In Chapter 5 we shall extend this algorithm to a more general class of polytopes, and we shall show there how to obtain optimal integral *dual* solutions.

In this application, and in the following examples we leave it to the reader to check that without loss of generality we may transform the polytopes in question to full-dimensional polytopes, and to find a vector a_0 and a number T such that (i) each numerator and denominator occurring in the components of the vertices of the polytope, and in those of a_0, do not exceed T in absolute value, (ii) a_0 is an internal point of the polytope, and (iii) $\log T$ is bounded by a polynomial in the size of the original combinatorial problem (in most cases we have $T=1$).

Also the second application is illustrative for the use of the method. Let $G=(V, E)$ be a complete graph, with $|V|$ even, and let P be the perfect matching polytope in $\mathbf{Q}^E$, i.e., P is the convex hull of the characteristic vectors of perfect matchings in G. So the strong optimization problem for this polytope is equivalent to the problem of finding a maximum weighted perfect matching. Edmonds [1965] showed that P consists of all vectors x in $\mathbf{Q}^E$ such that

$$\begin{array}{lll} \text{(i)} & x(e) \geqq 0 & (e \in E) \\ \text{(ii)} & \sum_{e \ni v} x(e) = 1 & (v \in V), \\ \text{(iii)} & \sum_{e \in E'} x(e) \geqq 1 & (E' \text{ odd cut}). \end{array}$$

Here a set E' of edges is an *odd cut* if E' is the set of edges with exactly one endpoint in V', where V' is some subset of V of odd cardinality. From this characterization one can derive a good algorithm for the strong separation problem for P as follows. Given a vector x in $\mathbf{Q}^E$ one easily checks in polynomial time conditions (i) and (ii) above. In case of violation one finds a hyperplane separating x from P. To check condition (iii) it suffices to have a polynomial method finding an odd cut E' minimizing $\sum_{e \in E'} x(e)$: if this minimum is not less than 1 we may conclude that x is in P, and otherwise E' yields a separating hyperplane. Now Padberg and Rao [1979] showed that such a method can be derived easily from Ford—Fulkerson's min-cut algorithm.

So the class of perfect matching polytopes is solvable, and hence there exists a polynomial algorithm finding maximum weighted perfect matchings (cf. Edmonds [1965]). One similarly derives a polynomial algorithm finding maximum weighted "b-matchings" (cf. Padberg and Rao [1979]).

A third application shows the existence of a polynomial algorithm for finding optimum branchings in a directed graph (cf. Chu and Liu [1965], Edmonds [1967]). Let $D=(V, A)$ be a digraph, and let r be some fixed vertex of D, called the *root*. A *branching* is a set A' of arrows of D making up a rooted directed spanning tree, with root r. A *rooted cut* is a set A' of arrows with $A'=\delta^-(V')$ for some non-empty set V' of vertices not containing r, where $\delta^-(V')$ denotes the set of arrows entering V'. It follows from Edmonds' branching theorem [1973] that the convex hull of (the incidence vectors of) the sets of arrows containing a branching as a subset (i.e., the sets intersecting each rooted cut), is a polytope P in $\mathbf{R}^A$ defined by the following linear inequalities:

(1)
$$\text{(i)} \quad 0 \leqq x(a) \leqq 1 \quad (a \in A),$$
$$\text{(ii)} \quad \sum_{a \in A'} x(a) \geqq 1 \quad (A' \text{ rooted cut}).$$

So there exists an algorithm which, given a digraph $D=(V, A)$, a root r, and a non-negative integral weight function w defined on A, determines a branching of minimum weight, in time polynomially bounded by $|V|$ and $\log \|w\|$, if and only if the strong optimization problem is solvable for the class of polytopes P arising in this way. By Theorem (3.1) and (3.8) it is enough to show that the strong separation problem is solvable. Indeed, if $x \in \mathbf{R}^A$ one easily checks condition (i) above and one finds a separating hyperplane in case of violation. To check condition (ii), we can find a rooted cut A' minimizing $\sum_{a \in A'} x(a)$ in time polynomially bounded by $|V|$ and $\log T$ (where T is the maximum of the numerators and denominators occurring in x), namely by applying Ford—Fulkerson's max flow-min cut algorithm to the corresponding network with capacity function x, source r and sink s, for each $s \neq r$. If the minimum is not less than 1 we conclude $x \in P$, and otherwise A' determines a separating hyperplane. (Again, see Chapter 5 for a more general approach.)

In fact this branching algorithm is one instance of a more general procedure. Let $\mathscr{E}$ be a *clutter*, i.e., a finite collection of finite sets no two of which are contained in each other. The *blocker* $B(\mathscr{E})$ of $\mathscr{E}$ is the collection of all minimal sets intersecting every set in $\mathscr{E}$ (minimal with respect to inclusion). E.g., if $\mathscr{E}$ is the collection of branchings in a digraph, then $B(\mathscr{E})$ is the collection of minimal rooted cuts. One easily checks that $B(B(\mathscr{E}))=\mathscr{E}$ for every clutter $\mathscr{E}$. Sometimes an even stronger duality relation may hold. Let $V=\bigcup\mathscr{E}$, and let P be the convex hull of the characteristic vectors (in $\mathbf{R}^V$) of all subsets of V containing some set in $\mathscr{E}$. Clearly, each vector x in P satisfies:

(2)
$$\text{(i)} \quad 0 \leqq x(v) \leqq 1 \quad (v \in V),$$
$$\text{(ii)} \quad \sum_{v \in V'} x(v) \geqq 1 \quad (V' \in B(\mathscr{E})),$$

as these inequalities hold for incidence vectors of sets in $\mathscr{E}$. In case P is completely determined by these linear inequalities, $\mathscr{E}$ (or the hypergraph $(V, \mathscr{E})$) is said to have the $\mathbf{Q}_+$-*max flow-min cut-property* or the $\mathbf{Q}_+$-*MFMC-property* (cf. Seymour [1977]). Thus the clutter of all branchings in a digraph has the $\mathbf{Q}_+$-MFMC-property, as we saw above. Fulkerson [1970] showed the interesting fact that a clutter $\mathscr{E}$ has the $\mathbf{Q}_+$-MFMC-property if and only if its blocker $B(\mathscr{E})$ has the $\mathbf{Q}_+$-MFMC-property.

Now one easily extends the derivation of a polynomial algorithm for branchings from such an algorithm for rooted cuts as described above, to the following theorem.

(4.1) Theorem. *Let $\mathscr{C}$ be a class of clutters with the $\mathbf{Q}_+$-MFMC-property, such that there exists an algorithm which finds, given $\mathscr{E}\in\mathscr{C}$ and $w\in\mathbf{Z}_+^V$ (where $V=\bigcup\mathscr{E}$), a set V' in $\mathscr{E}$ minimizing $\sum_{v\in V'} w(v)$, in time bounded by a polynomial in $|V|$ and $\log\|w\|$. Then the same is true for the class of blockers of clutters in $\mathscr{C}$.*

One should be careful with how the clutter $\mathscr{E}$ is given. Perhaps formally the most proper way to formulate the theorem is as follows: there exists an algorithm A and a polynomial $f(x)$ such that given a "minimization algorithm" for some clutter $\mathscr{E}$ with the $\mathbf{Q}_+$-MFMC-property, with "time bound" $g(\log\|w\|)$, and given some vector $u\in\mathbf{Z}_+^V$ $(V=\bigcup\mathscr{E})$, A finds a set V' in $B(\mathscr{E})$ minimizing $\sum_{v\in V'} u(v)$ in time bounded by $f(|V|\cdot\log\|u\|\cdot g(|V|\log\|u\|))$.

Among the other instances of Theorem (4.1) are the following. Let $D=(V, A)$ be a digraph. A *directed cut* is a set A' of arrows of D such that $A'=\delta^-(V')$ for some nonempty proper subset V' of V with $\delta^+(V')=\emptyset$ (as usual, $\delta^-(V')$ and $\delta^+(V')$ denote the sets of arrows entering and leaving V', respectively). A *covering* is a set of arrows intersecting each directed cut, i.e., a set of arrows whose contraction makes the digraph strongly connected. Let $\mathscr{E}$ be the clutter of all minimal directed cuts. If follows from the Lucchesi—Younger theorem [1978] that $\mathscr{E}$ has the $\mathbf{Q}_+$-MFMC-property, and an easy adaptation of Ford—Fulkerson's max flow-min cut algorithm yields a polynomial algorithm for finding minimum weighted directed cuts, given some nonnegative weight function on the arrows. (To this end we could add for each arrow also the reversed arrow with infinite capacity). Hence, by Theorem (4.1) there exists a polynomial algorithm for finding minimum weighted coverings in a digraph (such algorithms were found earlier by Lucchesi [1976], Karzanov [1979] and Frank [1981]).

In fact we do not need to call upon Ford—Fulkerson algorithm for finding minimum weighted cuts; such an algorithm can be derived also from Theorem (4.1). Indeed, let $D=(V, A)$ be a digraph and let r and s be two specified vertices. Let $\mathscr{E}$ be the clutter of all directed $r-s$-paths (considered as sets of arrows). So the blocker $B(\mathscr{E})$ of $\mathscr{E}$ consists of all minimal $r-s$-cuts. It follows from the max flow-min cut theorem that $\mathscr{E}$ has the $\mathbf{Q}_+$-MFMC-property. There exists an (easy) polynomial algorithm for finding shortest paths (Dijkstra [1959]), hence there exists a polynomial algorithm for finding minimum weighted cuts.

Theorem (4.1) also applies to T-cuts and T-joins. Let $G=(V, E)$ be an undirected graph, and let T be a set of vertices of G of even size. A set E' of edges of G is called a *T-cut* if there exists a set V' of vertices with $|V'\cap T|$ odd such that E' is the set of edges of G intersecting V' in exactly one vertex. A *T-join* is a set E' of edges with the property that T coincides with the set of vertices of odd valency in the graph (V, E'). One easily checks that the clutter $\mathscr{E}$ of all minimal T-cuts has as blocker the clutter of all minimal T-joins, and Edmonds and Johnson [1970] showed that $\mathscr{E}$ has the $\mathbf{Q}_+$-MFMC-property. Padberg and Rao [1979] adapted the Ford—Fulkerson minimum cut algorithm to obtain a polynomial algorithm to find minimum weighted T-cuts, given a nonnegative weight function on the edges (cf. also Chapter 5). Hence there exists a polynomial algorithm for finding minimum weighted T-joins, which was demonstrated earlier by Edmonds and Johnson [1970]. As special cases we may derive a polynomial algorithm for the Chinese postman problem (take T to be the set of vertices of odd valencey in G), and again a polynomial algorithm for finding minimum weighted perfect matchings (take $T=V$, and add a large constant to all weights; it is easy to derive conversely from a polynomial algorithm for minimum weighted perfect matchings, a polynomial algorithm for minimum weighted T-joins).

From Theorem (3.10) we know that if $\mathscr{C}$ is a class of clutters with the properties as described in Theorem (4.1), then there exists an algorithm to find optimal dual solutions, that is, given $\mathscr{E}\in\mathscr{C}$ and $w\in\mathbf{Z}_+^V$ (where $V=\bigcup\mathscr{E}$), to find sets $E_1, \dots, E_t$ in $B(\mathscr{E})$, with $t\leqq|V|$, and nonnegative numbers $\lambda_1, \dots, \lambda_t$ such that

$$(3)\qquad \begin{aligned} &\text{(i)} \quad \lambda_1\chi_{E_1}+\dots+\lambda_t\chi_{E_t} \leqq w, \\ &\text{(ii)} \quad \lambda_1+\dots+\lambda_t = \min_{V'\in\mathscr{E}} \sum_{v\in V'} w(v) \end{aligned}$$

(where χ_E denotes the characteristic vector in $\mathbf{R}^V$ of E), in time polynomially bounded by $|V|$ and $\log\|w\|$. So in the special cases discussed above this provides us with polynomial algorithm to find optimal fractional packings of branchings, rooted cuts, coverings, directed cuts, $r-s$-cuts, $r-s$-paths (i.e., optimum fractional $r-s$-flow), T-joins, T-cuts. Similarly, polynomial algorithms for finding optimum fractional two-commodity flow, and for fractional packings of two-commodity cuts may be derived (cf. Hu [1963, 1973]). Moreover, a recent theorem of Okamura and Seymour [1979] implies the existence of a polynomial algorithm for finding optimum fractional multicommodity flows in planar undirected graphs, provided that all sources and all sinks are on the boundary of the infinite face.

It is not necessarily true that if $\mathscr{E}$ has the $\mathbf{Q}_+$-MFMC-property, we can take the $\lambda_1, \dots, \lambda_t$ in the dual solution to be integers. If this is the case for each $w\in\mathbf{Z}_+^V$ then $\mathscr{E}$ is said to have the *$\mathbf{Z}_+$-MFMC-property*; and if for each such w we can take the $\lambda_1, \dots, \lambda_t$ to be half-integers, $\mathscr{E}$ has the *$\frac{1}{2}\mathbf{Z}_+$-MFMC-property*. E.g., the clutters of branchings, rooted cuts, coverings, $r-s$-cuts, $r-s$-paths all have the $\mathbf{Z}_+$-MFMC-property (proved by Fulkerson [1974], Edmonds [1973], Lucchesi and Younger [1978], Ford and Fulkerson [1956], and Fulkerson [1968], respectively), and the clutters of T-joins, two-commodity cuts, two-commodity paths, and multicommodity cuts in planar graphs (with commodities on the boundary), have the $\frac{1}{2}\mathbf{Z}_+$-MFMC-property (proved by Edmonds and Johnson [1970], Hu [1963], Seymour [1978], and Okamura and Seymour [1979], respectively). Edmonds and Giles [1977] posed the problem whether the clutter of directed cuts has the $\mathbf{Z}_+$-MFMC-property, but this was recently disproved by Schrijver [1980].

We were not able to derive from the ellipsoid method in general a polynomial algorithm for optimum (half-) integer dual solutions, if such solutions exist. However, in the case of optimum packings of (rooted, directed, $r-s$-, T-) cuts we can find by Theorem (3.10) an optimum fractional solution in which the number of cuts with nonzero coefficient is at most $|V|$. Hence, by well-known techniques (cf. Edmonds and Giles [1977], Frank [1979], Lovász [1975]) we can make these cuts *laminar* (i.e., non-crossing) in polynomial time, and we can find (half-)integer coefficients for the new collection of cuts, again in polynomial time, thus yielding an optimum (half-) integer packing of cuts.

We do not know whether the class $\mathscr{C}$ of *all* clutters with the $\mathbf{Q}_+$-MFMC-property is polynomially solvable in the sense of Theorem (4.1) (in which case Theorem (4.1) would become trivial). In Chapter 6 we shall see this indeed is the case for its anti-blocking analogue.

In this chapter, as well as in the next chapters we see that the existence of polynomial algorithms can be derived from the ellipsoid method for many problems for which such algorithms have been designed before. However, we were not able to derive such an algorithm for the following two problems, for which (complicated) polynomial algorithms are known: the problem of finding a maximum weighted independent set of vertices in a $K_{1,3}$-free graph (Minty [1980]), and that of finding a maximum collection of independent lines in a projective space (Lovász [1980]). A main obstacle to derive such algorithms from the ellipsoid method is that so far no characterizations in terms of facets of the corresponding convex-hull polytopes have been found.

However, if $G=(V,E)$ is a $K_{1,3}$-free graph, let P be the convex hull of the characteristic vectors of independent sets in G. So Minty showed that the optimization problem for the class of these polytopes is solvable, and hence, by Theorems (3.5) and (3.8), also the strong separation problem is solvable. This amounts to the following. Given a weight function w on V, one can find in polynomial time a maximum weighted "fractional clique", i.e., a function $x\colon V\to\mathbf{Q}_+$ such that $\sum_{v\in V'} x(x)\leqq 1$ for each independent subset V' of V, with $\sum_{v\in V} w(v)x(v)$ as large as possible. Similarly one can find a minimum fractional colouring (cf. Chapter 7).

So although a theoretical description of the facets of P has not yet been found, these facets can be identified by the separation algorithm. Perhaps a description of the facets yields a direct, not too complicated algorithm for the separation problem, and then one may derive in turn by the ellipsoid method a polynomial algorithm for the independent set problem in $K_{1,3}$-free graphs.

The problem of finding a maximum *weighted* collection of independent lines in a projective space is still open. Again, it might be possible to derive such an algorithm from a characterization in terms of facets of the corresponding convex-hull polytope, but such a characterization has not been found so far.

5. Submodular functions and directed graphs

In this chapter we show the existence of a polynomial algorithm finding the minimum value of a submodular set function, and we derive polynomial algorithms for the optimization problems introduced by Edmonds and Giles [1977] and by Frank [1979].

Let X be a finite set, let $\mathscr{F}$ be a collection of subsets of X closed under union and intersection, and let f be an integer-valued submodular function defined on $\mathscr{F}$, that is, let $f\colon \mathscr{F}\to\mathbf{Z}$ be such that

$$f(X')+f(X'')\geqq f(X'\cap X'')+f(X'\cup X'') \tag{1}$$

for $X', X''\in\mathscr{F}$. Examples of submodular functions are the rank functions of matroids, and the function f defined on all sets V' of vertices of a capacitated digraph by: $f(V')$ is the sum of the capacities of the arrows leaving V'.

(5.1) Theorem. *There exists an algorithm to find X' in $\mathcal{F}$ minimizing $f(X')$, in time polynomially bounded by $|X|$ and $\log B$, where B is some (previously known) upper bound for $|f(X')|$ $(X' \in \mathcal{F})$.*

So as special cases we can decide in polynomial time, given a matroid (X, r) and a weight function w on X, whether w is in the corresponding matroid polytope (i.e., whether $\sum_{x \in X'} w(x) \leqq r(X')$ for each subset X' of X), and we can derive a polynomial algorithm for finding minimum capacitated cuts in networks.

We need to make some requirements on the way $\mathcal{F}$ and f are given. First we should know an upper bound B for $|f(X')|$ $(X' \in \mathcal{F})$. Secondly, we must know in advance the sets $\cap\mathcal{F}$ and $\cup\mathcal{F}$, as well as for which pairs x_1, x_2 in X there exists $X' \in \mathcal{F}$ with $x_1 \in X'$ and $x_2 \notin X'$. This makes it possible to decide whether a given subset X' of X is in $\mathcal{F}$, and it allows us to assume without loss of generality that $\cap\mathcal{F} = \emptyset$ and $\cup\mathcal{F} = X$. Finally, given X' in $\mathcal{F}$ we must be able to find $f(X')$. It is enough to know that $f(X')$ can be calculated in time polynomial in $|X|$ and $\log B$, or that some oracle gives the answer. Most of the special-case submodular functions fulfil these requirements.

Proof. We shall reduce the minimization problem for submodular functions to the strong separation problem for polymatroid polytopes. Since the class of polymatroid polytopes is solvable (as optimization can be done by the greedy algorithm — see Edmonds [1970]), this will solve the problem.

Since we know an upper bound B for $|f(X')|$ we can find the minimum value of f by applying binary search. So it suffices to have a polynomial algorithm finding an X' in $\mathcal{F}$ with $f(X') < K$, or deciding that no such X' exists, for any given K. Since adding a constant to the values of f does not violate submodularity we can take $K=0$. Now if $f(\emptyset) < 0$ we can take $X' = \emptyset$. Hence we may assume that $f(\emptyset) = 0$.

Let g be the function defined on $\mathcal{F}$ by

$$g(X') = f(X') + 2B|X'|, \tag{2}$$

for $X' \in \mathcal{F}$. So g is nonnegative, integral, monotone and submodular. Moreover, $f(X') < 0$ if and only if $g(X') < 2B|X'|$. Next define for each subset X' of X the set

$$\overline{X'} = \cap \{X'' \in \mathcal{F} \mid X' \subseteqq X''\}. \tag{3}$$

(Note that $\overline{X'}$ can be determined in polynomial time.) Let $h(X') = g(\overline{X'})$ for each subset X' of X. One easily checks that h again is nonnegative, integral, monotone and submodular. Moreover, the problem of the existence of an X' in $\mathcal{F}$ with $g(X') < 2B|X'|$ is equivalent to that of the existence of a subset X' of X with $h(X') < 2B|X'|$. But the latter problem is just a special case of the strong separation problem for the vector $2B$ times the incidence vector of X and the polymatroid polytope corresponding to h:

$$\{v \in \mathbf{R}_+^X \mid \sum_{x \in X'} v(x) \leqq h(X') \text{ for all } X' \subseteqq X\} \tag{4}$$

(by Theorem (3.8) the separation algorithm yields facets as separating hyperplanes, i.e., subsets X' of X violating the inequality). As the optimization problem is solvable for the class of these polytopes we are finished. ∎

We apply the algorithm for finding the minimum value of a submodular function to theorems of Edmonds and Giles and of Frank.

Let $D=(V, A)$ be a digraph, and let $\mathcal{F}$ be a collection of subsets of V such that if $V', V'' \in \mathcal{F}$ and $V' \cap V'' \neq \emptyset$ and $V' \cup V'' \neq V$ then $V' \cap V'' \in \mathcal{F}$ and $V' \cup V'' \in \mathcal{F}$. Let f be an integer-valued function defined on $\mathcal{F}$ such that for all $V', V'' \in \mathcal{F}$ with $V' \cap V'' \neq \emptyset$ and $V' \cup V'' \neq V$ we have

$$f(V')+f(V'') \geqq f(V' \cap V'')+f(V' \cup V''). \tag{5}$$

Denote by $\delta^+(V')$ and $\delta^-(V')$ the sets of arrows leaving (entering, respectively) the set V' of vertices. Let vectors $b, c, d \in \mathbf{Z}^A$ be given, and consider the linear programming maximization problem

$$\textit{maximize} \sum_{a \in A} c(a)x(a), \tag{6}$$

where $x \in \mathbf{R}^A$ such that

$$\begin{aligned} &\text{(i)} \quad d(a) \leqq x(a) \leqq b(a) && (a \in A), \\ &\text{(ii)} \quad \sum_{a \in \delta^+(V')} x(a) - \sum_{a \in \delta^-(V')} x(a) \leqq f(V') && (V' \in \mathcal{F}). \end{aligned} \tag{7}$$

Edmonds and Giles showed that this problem has an integer optimum solution; this is equivalent to the fact that the polytope defined by the linear inequalities (7) has integral vertices. Edmonds and Giles also showed that the dual minimization problem can be solved with integral coefficients.

As special cases of Edmonds and Giles' result one has Ford and Fulkerson's max flow-min cut-theorem, the Lucchesi—Younger theorem on packing directed cuts and minimum coverings, Edmonds' (poly-)matroid intersection theorem, and theorems of Frank [1981] on orientations of undirected graphs. Moreover, one may derive the theorem due to Frank that if f is an integral submodular function defined on a collection $\mathcal{F}$ and g is an integral supermodular function on $\mathcal{F}$ (i.e., $-g$ is submodular) such that $g(X') \leqq f(X')$ for all X' in $\mathcal{F}$, then there exists an integral *modular* function h on $\mathcal{F}$ (i.e., both sub- and supermodular) such that $g(X') \leqq h(X') \leqq f(X')$ for all X' in $\mathcal{F}$.

We shall give an algorithm which solves the maximization problem (6) in time polynomially bounded by $|V|$ and $\log B$, where B is some (previously known) upper bound on $|f(X')|$ $(X' \in \mathcal{F})$, $\|b\|$, $\|c\|$ and $\|d\|$. We must know in advance for each pair of vertices v_1, v_2 of D whether $v_1 \in V'$ and $v_2 \notin V'$ for some $V' \in \mathcal{F}$ (this makes it possible to decide whether $V' \in \mathcal{F}$). Moreover we must have a subroutine calculating $f(V')$ if $V' \in \mathcal{F}$, in time polynomially bounded by $|V|$ and $\log B$.

First of all, we may suppose that $d(a) < b(a)$ for each arrow a, since if $d(a) > b(a)$ the polytope (7) is empty, and if $d(a) = b(a)$ we can remove the arrow a from the digraph and replace $f(V')$ by $f(V') \pm d(a)$ if $a \in \delta^{\mp}(V')$. We may even assume that the polytope (7) is full-dimensional and that we know an interior point x whose components have numerators and denominators not larger than a polynomial in $|V|$ and B. Otherwise we can extend the digraph D with one new vertex v_0 and with new arrows (v_0, v) for each "old" vertex v of D. Define $d(a) = -nB$, $b(a)=0$ and $c(a)=2n^nB^{n+4}$ for the new arrows a. One easily checks that the corresponding new polytope is full-dimensional, and one easily finds an x as required.

Moreover, the solutions of the original optimization problem correspond exactly to those solutions x of the new problem with $x(a)=0$ for each new arrow a.

Assuming the polytope (7) to be full-dimensional, by Theorem (3.8) it is enough to show that the strong separation problem is solvable. Let $x\in\mathbf{Q}^A$. One easily checks in polynomial time whether condition (i) is fulfilled. In case of violation we find a separating hyperplane. To check condition (ii) it suffices to find a set V' in $\mathcal{F}$ minimizing

(8) $$g(V') := f(V') - \sum_{a\in\delta^+(V')} x(a) + \sum_{a\in\delta^-(V')} x(a)$$

in time polynomial in $\log B$ and $\log T$, where T is the maximum of the numerators and denominators occurring in x. Note that g is submodular, hence we can appeal to the algorithm finding the minimum value of a submodular function. To this end we have to multiply the values by a factor to make the function integral (this factor is bounded above by $T^{|V|^2}$), and we have to apply the algorithm for each v_1, v_2 in V with $v_1\neq v_2$, to the function restricted to $\{V'\in\mathcal{F} \mid v_1\in V', v_2\notin V'\}$, since this collection is closed under union and intersection. Note that these requisites do not affect the polynomial boundedness of the required time.

So we proved that the class of "Edmonds—Giles" polytopes is solvable. Hence, by Theorem (3.10) we can find an optimum solution for the dual linear programming problem. In general, this solution will be fractional, but one can make this solution integral by making the collection of sets in $\mathcal{F}$ with non-zero dual coefficient laminar, by the well-known techniques (see Edmonds and Giles [1977]), in polynomial time. Now the (possibly fractional) coefficients can be replaced by integer coefficients, and these coefficients can be found by solving a linear program of polynomial size.

This can be used to show the following. Let f be a submodular function defined on the subsets of the set V, and define g by

(9) $$g(V') = \min \sum_{i=1}^{t} f(V_i)$$

for $V'\subseteq V$, where the minimum ranges over all partitions of V' into nonempty classes $V_1, \ldots, V_t$ $(t\geq 0)$. Then g is the largest submodular function with $g\leq f$ and $g(\emptyset)=0$. Now a partition attaining the minimum in (9) can be found in polynomial time, since it can be translated straightforwardly into an optimal integral dual solution for an Edmonds—Giles linear programming problem: Let v_0 be a new point. Construct the graph G on vertex set $V\cup\{v_0\}$ by connecting every point of V to v_0 by an arc. Take $d(a)=0, b(a)=1, c(a)=1$ for every arc a.

We leave it to the reader to derive by similar methods a polynomial algorithm for finding the solution to the following optimization problem, designed by Frank [1979]. Let $D=(V, A)$ be a digraph, and let $\mathcal{F}$ be a collection of subsets of V such that if $V', V''\in\mathcal{F}$ and $V'\cap V''\neq\emptyset$ then $V'\cap V''\in\mathcal{F}$ and $V'\cup V''\in\mathcal{F}$. Let f be a nonnegative integral function defined on $\mathcal{F}$ such that

(9) $$f(V')+f(V'') \leq f(V'\cap V'')+f(V'\cup V'')$$

if $V', V''\in\mathcal{F}$ and $V'\cap V''\neq\emptyset$. Let $b, c, d\in\mathbf{Z}_+^A$. Consider the linear programming problem

(10) $$\textit{minimize} \sum_{a\in A} c(a)x(a),$$

where $x \in \mathbf{R}^A$ such that

(11) (i) $d(a) \leqq x(a) \leqq b(a) \quad (a \in A)$,

(ii) $\sum_{a \in \delta^+(V')} x(a) \geqq f(V') \quad (V' \in \mathcal{F})$.

Frank showed that this problem, and its dual, have integer solutions. As special cases one may derive again Ford and Fulkerson's max flow-min cut theorem and Edmonds polymatroid intersection theorem, and also Fulkerson's theorem on minimum weighted branchings [1974].

We finally remark that the algorithm for finding a set V' in $\mathcal{F}$ minimizing $f(V')$, where f is a submodular function defined on $\mathcal{F}$, can be modified to a polynomial algorithm for finding a set V' in $\mathcal{F}$ of odd size minimizing $f(V')$. This extends Padberg and Rao's algorithm [1979] to find minimum odd cuts. More generally, let $\mathcal{G} \subseteq \mathcal{F}$ be such that if $V' \in \mathcal{G}$ and $V'' \in \mathcal{F} \setminus \mathcal{G}$ then $V' \cap V'' \in \mathcal{G}$ or $V' \cup V'' \in \mathcal{G}$. (E.g., $\mathcal{G}$ is the collection of sets in $\mathcal{F}$ intersecting V_0 in a number of elements not divisible by k, for some fixed subset V_0 of V and some natural number k.) Then there exists a polynomial algorithm to find V' in $\mathcal{G}$ minimizing $f(V')$ (by this we mean: $f(V') = \min \{f(V'') \mid V'' \in \mathcal{G}\}$). This algorithm needs, besides the prerequisites for $\mathcal{F}$ and f as above, a polynomial subroutine deciding whether a given set V' is in $\mathcal{G}$. Without loss of generality we may assume that $\emptyset \notin \mathcal{G}$ and $V \notin \mathcal{G}$.

The algorithm is defined by induction on $|V|$. Suppose the algorithm has been defined for all such structures with smaller $|V|$. Find a set V' in $\mathcal{F}$ such that $\emptyset \neq V' \neq V$ which minimizes $f(V')$. This can be done by applying the polynomial algorithm described above to the function f restricted to the collection $\{V' \in \mathcal{F} \mid v_1 \in V', v_2 \notin V'\}$, for all v_1, v_2 in V. If $V' \in \mathcal{G}$ we are finished. If $V' \notin \mathcal{G}$ there will be a set V'' in $\mathcal{G}$ minimizing $f(V'')$ such that $V'' \subseteq V'$ or $V' \subseteq V''$. Indeed, if $V'' \in \mathcal{G}$ minimizes $f(V'')$ then either $V' \cap V'' \in \mathcal{G}$ or $V' \cup V'' \in \mathcal{G}$; in the former case we have

(12)
$$f(V' \cap V'') + f(V' \cup V'') \leqq f(V') + f(V''),$$
$$f(V' \cap V'') \geqq f(V''),$$
$$f(V' \cup V'') \geqq f(V'),$$

as V' and V'' minimize $f(V')$ and $f(V'')$ for $V' \in \mathcal{F}$ and $V'' \in \mathcal{G}$, respectively. Hence $f(V' \cap V'') = f(V'')$. If $V' \cup V'' \in \mathcal{G}$ we can exchange $\cup$ and $\cap$ in this reasoning. Now there exists an algorithm finding $V'' \in \mathcal{G}$ with $V'' \supseteq V'$ minimizing $f(V'')$, and an algorithm finding $V'' \in \mathcal{G}$ with $V'' \subseteq V'$ minimizing $f(V'')$, for these algorithms follow straightforwardly from the previously defined algorithms for sets of size $|V'|$ and $|V \setminus V'|$. We leave it to the reader to check that this gives us a polynomial algorithm.

6. Independent sets in perfect graphs

In the previous chapters we have applied the ellipsoid method to classes of polytopes. We now apply the method to a class of non-polytopal convex sets, in order to obtain a polynomial algorithm finding maximum (weighted) independent sets and minimum colorings of perfect graphs (cf. Lovász [1972]). Applying these algorithms to the complement, we get algorithms to find maximum weighted clique and minimum weighted covering by cliques.

Let $G=(V, E)$ be an undirected graph, and let $\alpha(G)$ denote the *independence number* of G, i.e., the maximum number of pairwise non-adjacent vertices. Let $\alpha^*(G)$ denote the *fractional independence number* of G, i.e., the maximum value of $\sum_{v\in V} c(v)$ where the $c(v)$ are nonnegative real numbers such that $\sum_{v\in C} c(v)\leqq 1$ for each clique C of G. So $\alpha(G)\leqq\alpha^*(G)$, and furthermore G is perfect if and only if $\alpha(G')=\alpha^*(G')$ for each induced subgraph G' of G. Since $\alpha^*(G)$ is the optimum of a linear programming problem, we could try to calculate $\alpha^*(G)$ by means of the ellipsoid method; but the size of this problem is not polynomially bounded as there can exist too many cliques C.

However, the following number $\vartheta(G)$ was introduced in Lovász [1979]. Suppose $V=\{1, ..., n\}$. Then $\vartheta(G)$ is the maximum value of $\sum_{i,j=1}^{n} b_{ij}$, where $B=(b_{ij})$ belongs to the following convex body of matrices

$$\text{(1)} \qquad \{B=(b_{ij})\,|\,B \text{ is positive semidefinite with trace at most } 1, \text{ and } b_{ij}=0 \text{ if } i \text{ and } j \text{ are adjacent vertices of } G\ (i\neq j)\}.$$

If B belongs to this class we shall say that B *represents* G. It was shown that $\alpha(G)\leqq\vartheta(G)\leqq\alpha^*(G)$. (In fact, $\vartheta(G)$ is an upper bound for the *Shannon capacity* of G.) We show that $\vartheta(G)$ can be calculated (approximated) by the ellipsoid method in time bounded by a polynomial in $|V|$. This allows us to find $\alpha(G)$ for graphs G with $\alpha(G)=\vartheta(G)$, in particular for perfect graphs.

In fact we exhibit an algorithm finding the maximum *weight* $\sum_{v\in A} w(v)$, where A is an independent set in a perfect graph, given some nonnegative integral weight function on V, in time polynomially bounded by $|V|$ and $\log \|w\|$. Obviously, this maximum weight is equal to $\alpha(G_w)$, where the graph G_w arises from G by replacing each vertex v of G by $w(v)$ pairwise non-adjacent new vertices and where two vertices of G_w are adjacent iff their originals in G are adjacent. Note that if G is perfect then also any G_w is perfect (cf. Lovász [1972]). Moreover, $\vartheta(G_w)$ is equal to the maximum value of

$$\text{(2)} \qquad \sum_{i,j=1}^{n} \sqrt{w_i w_j}\cdot b_{ij}$$

where $B=(b_{ij})$ represents G. This can be seen as follows. If $B=(b_{ij})$ represents G then, by replacing each entry b_{ij} by a matrix of size $w_i\times w_j$ with constant entries $(w_i w_j)^{-1/2} b_{ij}$, we obtain a matrix B' representing G_w, with

$$\text{(3)} \qquad \sum_{i,j=1}^{n} \sqrt{w_i w_j}\cdot b_{ij} = \sum_{i,j=1}^{n'} b'_{ij}.$$

Conversely, if B' represents G_w, then, by replacing the $w_i\times w_j$ submatrix induced by the copies of i and j, by the sum of its entries divided by $\sqrt{w_i w_j}$, we obtain a matrix B representing G, satisfying (3) again.

To approximate $\vartheta(G_w)$ up to an error of at most $\varepsilon>0$, we can replace $\sqrt{w_i w_j}$ of (2) by some rational number ω_{ij} with

$$(4) \qquad |\omega_{ij}-\sqrt{w_i w_j}| < \varepsilon/2n^2$$

(taking $\omega_{ij}=\omega_{ji}$), where the denominators of the ω_{ij} are at most $2n^2/\varepsilon$. Then $\vartheta(G_w)$ differs by at most $\frac{1}{2}\varepsilon$ from the maximum value of $\sum_{i,j}\omega_{ij}b_{ij}$ with $B=(b_{ij})$ representing G. So we need to approximate this last number with accuracy $\frac{1}{2}\varepsilon$, which can be done by the ellipsoid method.

To apply the ellipsoid method we replace the set (1) by a full-dimensional convex body, by forgetting the coordinates below the main diagonal, as well as the coordinates (i, j) for adjacent i and j. We end up with a full-dimensional convex body in the $\left(n+\binom{n}{2}-|E|\right)$-dimensional space. One easily finds an interior point a_0 in it, and radii r and R such that the convex body contains $S(a_0, r)$ and is contained in $S(a_0, R)$, and such that the logarithms of r and R and of the numerators and denominators occurring in a_0 are bounded (in absolute value) by a polynomial in n (fixed over all graphs G). So we may apply Theorem (3.1). We show that the separation problem is solvable for the class of convex bodies obtained in this way. Let b be some vector in the $\left(n+\binom{n}{2}-|E|\right)$-dimensional space. Extend this vector, in the obvious way, to a symmetric $n\times n$-matrix $B=(b_{ij})$ with $b_{ij}=0$ if i and j are adjacent vertices of G. If $\operatorname{Tr} B>1$, the separation problem is trivial. Suppose that $\operatorname{Tr} B\leqq 1$. Find a basis for the columns; without loss of generality assume that these are columns $1, \dots, k$. Then the principal submatrix $B'=(b_{ij})_{i=1,\dots,k}^{j=1,\dots,k}$ is non-singular and has rank $B'=\operatorname{rank} B$. It is easy to prove that B is positive semidefinite iff B' is positive definite, which in turn is equivalent to

$$(5) \qquad \det B_t = \det(b_{ij})_{i=1,\dots,t}^{j=1,\dots,t} > 0,$$

for $t=1, \dots, k$. Since these determinants can be calculated in polynomial time, thereby we have checked in polynomial time whether B belongs to the convex set (1). If, moreover, we find that B is not positive semidefinite then let t be the smallest index for which $\det B_t<0$. Let φ_i denote $(-1)^i$ times the (i, t)-th minor of B_t $(i=1, \dots, t)$, and $\varphi_i=0$ if $i>t$. Then

$$(6) \qquad \sum_{i,j=1}^{n}\varphi_i\varphi_j\beta_{ij} \geqq 0$$

for every positive semidefinite matrix (β_{ij}). By definition, and by simple computation,

$$(7) \qquad \sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_i\varphi_j b_{ij} = \det B_t\cdot\det B_{t-1} \leqq 0,$$

(if $t=1$, then $\det B_0=1$, by definition). So the matrix $(\varphi_i\varphi_j)_{i,j=1}^{n}$ is a solution of the separation problem.

Therefore, by Theorem (3.1), we can approximate the maximum value of $\Sigma\,\omega_{ij}b_{ij}$ for $B=(b_{ij})$ in (1) with accuracy $\frac{1}{2}\varepsilon$ (and hence $\vartheta(G_w)$ with accuracy ε), in time polynomially bounded by $|V|$, $|\log \varepsilon|$ and $\log T$ where T is the maximum among the denominators and numerators occurring in (ω_{ij}). If we know that $\alpha(G_w)=\vartheta(G_w)$ it follows that $\vartheta(G_w)$ is an integer, and we can take $\varepsilon=\frac{1}{2}$. In particular there exists an algorithm which calculates $\alpha(G_w)$ for perfect graphs G in time polynomially bounded by $|V|$ and $\log \|w\|$.

We can find an explicit maximum weighted independent set in a perfect graph as follows. Compare $\alpha(G_w)$ with $\alpha(G'_{w'})$, where G' and w' arise from G and w by removing vertex 1 from G and the corresponding component from w. If $\alpha(G'_{w'})=\alpha(G_w)$ we replace G by G' and w by w'; otherwise we leave G and w unchanged. Next we try to remove vertex 2 similarly, and so on. At the end we are left with a collection of vertices forming a maximum weighted independent set in G.

So given a perfect graph $G=(V, E)$ and a weight function w on V we can find an independent set V' maximizing $\sum_{v\in V'} w(v)$. This implies that the strong optimization problem is solvable for the class of convex hulls of the independent sets in perfect graphs. For perfect graphs $G=(V, E)$ this convex hull is given by the linear inequalities

$$\text{(8)} \qquad \begin{aligned} &\text{(i)} \quad x(v) \geqq 0 \qquad (v\in V),\\ &\text{(ii)} \quad \sum_{v\in C} x(v) \leqq 1 \quad (C \text{ clique}). \end{aligned}$$

This yields that also the strong separation problem is solvable for this class, but this is not interesting anymore, as it amounts to finding a maximum weighted clique in a perfect graph, i.e., a maximum weighted independent set in the complementary graph, which is perfect again.

However, by Theorem (3.10) we can find an optimal (fractional) dual solution for the corresponding linear programming problem. So, given $w\in \mathbf{Z}_+^V$, we can find cliques $C_1, \dots, C_t$ and positive real numbers $\lambda_1, \dots, \lambda_t$ $(t\leqq |V|)$ such that $\lambda_1+\dots$ $\dots+\lambda_t=\alpha(G_w)$ and

$$\text{(9)} \qquad \sum_{\substack{j\\ i\in C_j}} \lambda_j = w_i$$

for each vertex i, in polynomial time. But for perfect graphs $\alpha(G)$ is equal to the minimum number of cliques needed to cover V (i.e., to the chromatic number of the complementary graph), which means that there exist *integers* $\lambda_1, \dots, \lambda_t$ with the required properties. Indeed we can find such integers as follows.

First, if $w\equiv 1$, each clique C_j with $\lambda_j>0$ intersects all maximum-sized independent sets. So we can remove clique C_1 from G, thus obtaining a graph G' with $\alpha(G')=\alpha(G)-1$, and we can repeat the procedure for G'. After $\alpha(G)$ repetitions we have found $\alpha(G)$ cliques covering V.

If w is arbitrary, let λ'_j be the lower integer part of λ_j, and let

$$\text{(10)} \qquad w'_i = w_i - \sum_{\substack{j\\ i\in C_j}} \lambda'_j .$$

Since $(\lambda_j-\lambda'_j)<1$ we know by (9) that $w'_i<t\leqq |V|$. Therefore, $G_{w'}$ has at most $|V|^2$ vertices, and as in the previous paragraph a covering with $\alpha(G_{w'})$ cliques can be found in time polynomially bounded by $|V|^2$. This covering, together with the

covering by $C_1, \ldots, C_t$ with coefficients $\lambda_1', \ldots, \lambda_t'$, yields an optimum integral dual solution as required.

We remark that the algorithm to find $\alpha(G)$ clearly works for all graphs G with $\alpha(G)=\vartheta(G)$ but that our method to find an explicit maximum independent set requires that $\alpha(G')=\vartheta(G')$ for induced subgraphs G' of G. This holds for all induced subgraphs if and only if G is perfect, as was shown by Lovász [1981].

7. Intractability of vertex-packing and fractional colouring

The ellipsoid method yields a certain "polynomial equivalence" of combinatorial problems, in the sense that there exists a polynomial algorithm for one problem iff such an algorithm exists for some other problem. We can use this principle also in the negative: if some problem is "hard" (e.g., NP-complete) then it follows that also certain other problems are hard.

We apply this to the problem of determining the independence number $\alpha(G)$ of a graph G, which is known to be NP-complete (cf. Garey and Johnson [1979]). More precisely, and more generally: given a graph $G=(V, E)$, a weight function $w\colon V\to \mathbf{Z}_+$ and a number K, the problem of deciding whether there exists an independent set V' of vertices such that $\sum_{v\in V'} w(v)\geqq K$ (i.e., whether $\alpha(G_w)\geqq K$) is NP-complete. To formulate this in terms of polytopes, let $P(G)$ be the convex hull of the characteristic vectors of independent sets in G. Then the strong optimization problem for the class of polytopes $P(G)$, is NP-complete.

Now consider the anti-blocker $A(P(G))$ of $P(G)$ (cf. Chapter 3). By Corollary (3.5) the strong optimization problem for the class of polytopes $A(P(G))$ is solvable iff it is solvable for the class of polytopes $P(G)$. This remains true if we restrict G to a subclass of the class of all graphs.

Now the strong optimization problem for $A(P(G))$ asks for a *maximum weighted fractional clique*, i.e., for a vector x in $\mathbf{R}_+^V$ such that $\sum_{v\in V'} x(v)\leqq 1$ for each independent set V', and such that $\sum_{v\in V} w(v)x(v)$ is as large as possible, given some weight function w in $\mathbf{Z}_+^V$. By linear programming duality this maximum is equal to the *weighted fractional chromatic number*, i.e., to the minimum value $\gamma_w^*(G)$ of $\lambda_1+\ldots+\lambda_t$, where $\lambda_1, \ldots, \lambda_t$ are positive numbers for which there exist independent sets $V_1, \ldots, V_t$ such that for every vertex v we have

$$\sum_{\substack{j\\ v\in V_j}} \lambda_j = w(v) \tag{1}$$

(we can take $t\leqq |V|$). Hence, given a class of graphs, there exists a polynomial algorithm determining $\alpha(G_w)$ for each graph G in this class and for each weight function w, iff such an algorithm exists determining the fractional chromatic number for each such G and w. In fact, the ellipsoid method shows that both problems are "Turing reducible" to each other (cf. Garey and Johnson [1979]). This implies that, since the former problem for the class of all graphs is NP-complete, the latter problem is both NP-hard and NP-easy, i.e., NP-equivalent.

In fact the problem of determining the fractional chromatic number belongs to the class NP, as in order to show in polynomial time that $\gamma_w^*(G)\leqq K$ we can bound the numerators and denominators of the λ_j by $\|w\| \cdot |V|^{|V|}$. So the fractional colouring problem is not only Turing reducible, but even polynomial reducible to the independence number problem, but we do not know the other way around.

Since the problem of determining $\alpha(G)$ is already NP-complete if we restrict G to planar cubic graphs the problem of determining $\gamma_w^*(G)$ remains to be NP-equivalent if G is restricted similarly. The problem of determining the fractional chromatic number $\gamma_w^*(G)$ and that of determining the *chromatic number* $\gamma(G)$ seem to be incomparable with respect to hardness. For cubic graphs G, $\gamma(G)$ can be determined easily in polynomial time, but the problem of determining $\gamma_w^*(G)$ is NP-equivalent. In contrast to this, for *line graphs* G of cubic graphs the problem of determining $\gamma(G)$ is NP-complete (Holyer [1979]), whereas $\gamma_w^*(G)$ can be determined in polynomial time (since $\alpha(G_w)$ can be determined in polynomial time by the matching algorithm).

Acknowledgement. We are indebted to J. Edmonds, B. Korte, R. Karp, R. Bland, A. Bachem, M. Padberg, D. Shmoys, and many other colleagues who participated in the discusssion and correspondence during which the idea of this paper arose and details took shape.

References

[1965] CHU YOENG-JIN and LIU TSENG-HUNG, On the shortest arborescence of a directed graph, *Scientia Sinica* 4 (1965) 1396—1400.

[1959] E. W. DIJKSTRA, A note on two problems in connexion with graphs, *Numer. Math.* 1 (1959) 269—271.

[1965] J. EDMONDS, Maximum matching and a polyhedron with 0, 1-vertices, *J. Res. Nat. Bur. Standards Sect. B* 69 (1965) 125—130.

[1967] J. EDMONDS, Optimum branchings, *J. Res. Nat. Bur. Standards Sect. B* 71 (1967) 233—240.

[1970] J. EDMONDS, Submodular functions, matroids, and certain polyhedra, in: *Combinatorial structures and their applications, Proc. Intern. Conf. Calgary, Alb.*, 1969 (R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds.), Gordon and Breach, New York, 1970, 69—87.

[1970] J. EDMONDS and E. L. JOHNSON, Matching: a well-solved class of integer linear programs, *ibid.* 89—92.

[1973] J. EDMONDS, Edge-disjoint branchings, in: *Combinatorial algorithms, Courant Comp. Sci. Symp.* Monterey, Ca., 1972 (R. Rustin, ed.), Acad. Press, New York, 1973, 91—96.

[1979] J. EDMONDS, Matroid intersection, *Annals of Discrete Math.* 4 (1979) 39—49.

[1977] J. EDMONDS and R. GILES, A min-max relation for submodular functions on graphs, *Annals of Discrete Math.* 1 (1977) 185—204.

[1973] J. EDMONDS and E. L. JOHNSON, Matching, Euler tours and the Chinese postman, *Math. Programming* 5 (1973) 88—124.

[1956] L. R. FORD and D. R. FULKERSON, Maximum flow through a network, *Canad. J. Math.* 8 (1956) 399—404.

[1962] L. R. FORD and D. R. FULKERSON, *Flows in networks*, Princeton Univ. Press, Princeton, N. J., 1962.

[1980] A. FRANK, On the orientations of graphs, *J. Combinatorial Theory (B)* 28 (1980) 251—260.

[1979] A. FRANK, Kernel systems of directed graphs, *Acta Sci. Math.* (Szeged) 41 (1979) 63—76.

[1981] A. FRANK, How to make a digraph strongly connected, *Combinatorica* 1 (2) (1981) 145—153.

[1968] D. R. FULKERSON, Networks, frames, and blocking systems, in: *Mathematics of the decision sciences*, Part I (G. B. Dantzig and A. F. Veinott, eds.), Amer. Math. Soc., Providence, R. I., 1968, 303—334.

[1970] D. R. FULKERSON, Blocking polyhedra, in: *Graph theory and its applications, Proc. adv. Seminar Madison*, Wis., 1969 (B. Harris, ed.), Acad. Press, New York, 1970, 93—112.

[1974] D. R. FULKERSON, Packing rooted directed cuts in a weighted directed graph, *Math. Programming* 6 (1974) 1—13.

[1981] P. Gács and L. Lovász, Khachiyan's algorithm for linear programming, *Math. Programming Studies* 14 (1981) 61—68.

[1979] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, 1979.

[1960] A. J. Hoffman, Some recent applications of the theory of linear inequalities to extremal combinatorial analysis, in: *Combinatorial analysis, Proc. 10th Symp. on Appl. Math. Columbia Univ.*, 1958 (R. E. Bellman and M. Hall, Jr, eds.), Amer. Math. Soc., Providence, R. I., 1960, 113—127.

[1979] I. Holyer, The NP-completeness of edge-colouring, *SIAM J. Comp.*, to appear.

[1963] T. C. Hu, Multicommodity network flows, *Operations Res.* 11 (1963) 344—360.

[1973] T. C. Hu, Two-commodity cut-packing problem, *Discrete Math.* 4 (1973) 108—109.

[1979] A. V. Karzanov, On the minimal number of arcs of a digraph meeting all its directed cutsets, *to appear.*

[1979] L. G. Khachiyan, A polynomial algorithm in linear programming, *Doklady Akademii Nauk SSSR* 244 (1979) 1093—1096 (English translation: *Soviet Math. Dokl.* 20, 191—194).

[1970] E. L. Lawler, Optimal matroid intersections, in: *Combinatorial structures and their applications, Proc. Intern. Conf. Calgary, Alb.*, 1969 (R. Guy, H. Hanani, N. Sauer, and J. Schönheim, eds.), Gordon and Breach, New York, 1970, 233—235.

[1976] E. L. Lawler, *Combinatorial optimization: networks and matroids*, Holt, Rinehart and Winston, New York, 1976.

[1972] L. Lovász, Normal hypergraphs and the perfect graph conjecture, *Discrete Math.* 2 (1972) 253—267.

[1975] L. Lovász, 2-Matchings and 2-covers of hypergraphs, *Acta Math. Acad. Sci. Hungar.* 26 (1975) 433—444.

[1978] L. Lovász, The matroid matching problem, *Proc. Conf. Algebraic Methods in Graph Theory* (Szeged, 1978), to appear.

[1979] L. Lovász, On the Shannon capacity of a graph, *IEEE Trans. on Information Theory* 25 (1979) 1—7.

[1981] L. Lovász, Perfect graphs, in: *More selected topics in graph theory* (L. W. Beineke and R. J. Wilson, eds), to appear

[1976] C. L. Lucchesi, A minimax equality for directed graphs, *Doctoral Thesis*, Univ. Waterloo, Waterloo, Ont., 1976.

[1978] C. L. Lucchesi and D. H. Younger, A minimax relation for directed graphs, *J. London Math. Soc. (2)* 17 (1978) 369—374.

[1980] G. J. Minty, On maximal independent sets of vertices in claw-free graphs, *J. Combinatorial Theory (B)*, 28 (1980) 284—304.

[1954] T. S. Motzkin and I. J. Schoenberg, The relaxation method for linear inequalities, *Canad. J. Math.* 6 (1954) 393—404.

[1979] H. Okamura and P. D. Seymour, Multicommodity flows in planar graphs, *J. Combinatorial Theory (B)*, to appear.

[1979] M. W. Padberg and M. R. Rao, Minimum cut-sets and b-matchings, *to appear.*

[1980] A. Schrijver, A counterexample to a conjecture of Edmonds and Giles, *Discrete Math.* 32 (1980) 213—214.

[1977] P. D. Seymour, The matroids with the max-flow min-cut property, *J. Combinatorial Theory (B)* 23 (1977) 189—222.

[1978] P. D. Seymour, A two-commodity cut theorem, *Discrete Math.* 23 (1978) 177—181.

[1970] N. Z. Shor, Convergence rate of the gradient descent method with dilatation of the space, *Kibernetika* 2 (1970) 80—85 (English translation: *Cybernetics* 6 (1970) 102—108).