SOME COMBINATORIAL APPLICATIONS OF THE NEW LINEAR PROGRAMMING ALGORITHM

L. LOVÁSZ
Mathematical Institute
A. Jozsef University
Szeged

A. SCHRIJVER
Amsterdam

INTRODUCTION

One of the greatest successes in combinatorial optimization in recent years has been the algorithm of Khachiyan (1979), which solves the linear programming problem in polynomial time. The existence of such an algorithm has been a major unsolved problem in the theory of computational complexity, and has been an obstable in the way of the classification of various combinatorial problems with respect to the P-NP scheme, for example in scheduling theory. There are a large number of problems which have been reduced to linear programs; the polynomial solvability of the linear programming problem immediately implies the polynomial solvability of these problems.

It turns out, however, that there is a really wide class of combinatorial problems which cannot be reduced to linear programming and yet the method of the new linear programming algorithm can be applied to solve them, or more precisely, to reduce them to simpler combinatorial problems. It has to be noted that the basic idea of the new method, as Khachiyan remarks in his paper, is due to Shor (1970), who applied it in non-linear optimization. Since the applications of the method given in this paper are somewhere between linear and non-linear programming (or some of them are, in fact, non-linear), it is more apt to call this method the Shor-Khachiyan method or - rhyming with the classical simplex method - the Ellipsoid Method.

The method will be outlined in Section 1. Here we also describe the general approach to combinatorial optimization which makes the application of the method possible.

The main result formulated in Section 2 is that every submodular set-function can be minimized in polynomial time. This result, combined with the rather trivial Greedy

Algorithm and some more applications of the Ellipsoid Method, implies the polynomial solvability of a number of combinatorial optimization problems such as the matching, matroid intersection, optimum branching, optimum covering of directed cuts, etc.

In the third section we very briefly mention some other applications and conclude with some remarks on the prospects of this method.

1. THE ELLIPSOID METHOD AND POLYHEDRAL COMBINATORICS

Let $P \subseteq R^n$ be a polyhedron, determined by the inequalities

$$a_1^T x \leqq b_1, \ \ldots, \ a_m^T x \leqq b_m,$$

and let $c^T x$ be a linear objective function that we want to maximize over P. For the sake of simplicity assume that P is bounded and full-dimensional; say P is contained in the ball $S(0,R)$ of radius R about the origin, and it contains somewhere a ball of radius r.

We define a sequence of points $x_0, x_1, \ldots$ and a sequence of ellipsoids $E_0, E_1, \ldots$ such that x_i is the centre of E_i. Let $x_0 = 0$ and $E_0 = S(0,R)$. Assume that x_k and E_k are defined. Then let us check whether or not $x_k \in P$.

Case 1. If $x_k \in P$, then consider the half-ellipsoid which is the intersection of E_k with the halfspace $c^T x \geqq c^T x_k$, and include it in an ellipsoid E_{k+1} with least possible volume. Let x_{k+1} be the centre of E_{k+1}.

Case 2. If $x_k \notin P$ then let $a_i^T x \leqq b_i$ be a constraint which is violated by x_k. Include the intersection of E_k with the halfspace $a_i^T x \leqq b_i$ in an ellipsoid E_{k+1} with least possible volume. Let x_{k+1} be the centre of E_{k+1}.

The following can be proved : if we look at those values of k for which $x_k \in P$ (call these briefly feasible values), then

$$\max\{c^T x_k : 1 \leqq k \leqq p, \ k \ \text{feasible}\} \to \max\{c^T x : x \in P\}, \ (p \to \infty), \qquad \ldots \quad (1)$$

and the convergence is exponentially fast. In a sense this fact may be considered as the polynomial solution of the optimization problem. If we want to get the precise solution, we have to know and use something of the arithmetical nature of the vertices of P (this is quite natural : if a vertex might have, say, irrational coordinates then it would be impossible to describe the answer). In combinatorial applications, for example, we quite often will know that the vertices of P are lattice points. The exact optimum then can be obtained simply by rounding.

The idea of the proof of the fast convergence in (1) is quite simple, although precise details are tedious. We look at the piece of P where the value of the objective function is not smaller than the maximum value at feasible points x_k found so far, and prove by induction on k that this piece is included in the current ellipsoid. It is easy to see that the volume of E_k drops exponentially fast with k, and hence the volume of the piece of P where the objective function is not smaller than the current record also must drop exponentially fast. Hence the difference between the current record and the true optimum of the objective function must also decrease exponentially fast.

Details of this argument can be found in [14].

Now we come to the combinatorial part of the paper. A general setting in which combinatorial optimization problems can be treated is the following. Given a finite subset $S \subseteq R^n$ and a linear objective function $c^T x$, find

$$\max\{c^T x : x \in S\}.$$

Since a linear function always assumes its optimum at a vertex of a polytope, we have

$$\max\{c^T x : x \in S\} = \max\{c^T x : x \in \mathrm{conv}(S)\}.$$

Since $\mathrm{conv}(S)$ is a polytope, the left hand side is equivalent to a linear program. So if we can write up this program explicitly and then solve it by the Simplex or Ellipsoid Method, we are done.

This program, initiated among others by Edmonds, Fulkerson and Hoffman, has motivated the considerable amount of research that has concerned the facial structure of combinatorially defined polytopes. But even for classes of polytopes for which the facets are well known, this approach could not lead to satisfactory (polynomial) algorithms. The reason for this is that (with a very few exceptions) the number of facets of $\mathrm{conv}(S)$, i.e. the minimum number of inequalities needed to describe $\mathrm{conv}(S)$, is exponentially large in the size of the original combinatorial structure. So even to write up the linear program which we have to solve in order to find the desired optimum, takes exponential time. (This is not surprising if we think of the fact that S in itself is usually exponentially large in the size of the original combinatorial structure; if not, we can determine the optimum by evaluating the objective function at every element of S.)

It may be the most important feature of the Ellipsoid Method that it overcomes this difficulty in many cases. To see how, note that the inequalities defining P enter the algorithm only at one point : when we have to decide if $x_k \in P$ and if not, we need an inequality $a_i^T x \leq b_i$ which is violated by x_k. One way to do this is to substitute x_k into each of the inequalities $a_i^T x \leq b_i$; but since in combinatorial optimization problems these inequalities are usually very structured, we may well have a subroutine which checks $x_k \in P$, and picks a violated inequality if $x_k \notin P$, without explicitly writing up all inequalities defining P. So the Ellipsoid Method reduces the original combinatorial problem to another one, which (as we shall illustrate in the next section) is quite different from the original and often easier to solve.

One way to formulate this relation between combinatorial problems is in terms of anti-blocking polyhedra (see Fulkerson [12]). Let A be non-negative $m \times m$ matrix and b a non-negative n-vector. Let

$$P = \{x \in R_+^m : Ax \leq b\}. \qquad \ldots \quad (2)$$

Then the anti-blocker of the polytope P is the polytope

$$P^* = \{y \in R_+^m : x^T y \leq 1 \quad \text{for every} \quad x \in P\}.$$

Theorem 1.1. Let K be a class of polytopes of type (2) such that there exists an algorithm to maximize an arbitrary linear objective function over polytopes in K in time polynomial in nm and the number of decimal digits in the coefficients of the objective function. Then there exists such an algorithm for the class of anti-blockers of polytopes in K.

2. MINIMIZING SUBMODULAR FUNCTIONS AND APPLICATIONS

Let f be a function defined on the subsets of a set S. The function f will be called <u>submodular</u> if

$$f(X \cap Y) + f(X \cup Y) \leqq f(X) + f(Y)$$

holds for every pair of subsets of S. The main result which we state in this section is the following.

<u>Theorem 2.1</u>. Let f be a submodular function on the subsets of a finite set S. Assume that we have a subroutine to compute $f(X)$ in time less than T. Then the subset of S minimizing f can be determined in time polynomial in T and $|S|$.

As a first application we show how to find the value of a maximum flow between a,b of a network G. For $X \subseteq V(G)$, denote by $\delta(X)$ the capacity of the cut determined by X. It is a simple well-known fact that $\delta(X)$ is a submodular function. By the Max-flow-min-cut Theorem, the value of a maximum flow is

$$\min\{\delta(X) : a \in X \quad V(G) - b\}.$$

This minimum can be determined in polynomial time by minimizing the submodular function $\delta(X \cup \{a\})$ over the subsets of $V(G)-a-b$. It is, of course, also important to find an optimum flow, and this does not follow directly from the Ellipsoid Method. But if we have a polynomial algorithm to find the value of an optimum flow for every network, then there are several quite easy procedures to find a maximum flow. We leave this to the reader.

A more essential application is the following. Let G be a digraph. A <u>directed cut</u> is the set of edges connecting $X \subset V(G)$ to $V(G) - X$, provided $X \neq \phi$ or $V(G)$, and there is no edge connecting $V(G) - X$ to X. A theorem of Lucchesi and Younger asserts that the maximum number of edge-disjoint directed cuts is equal to the minimum number of edges covering all directed cuts. More recently Lucchesi [19], Karzanov [15] and Frank [10] gave polynomial algorithms to find this number. To show that such an algorithm can be based on the Ellipsoid Method, let us assign a variable x_j to every edge j and consider the polytope

$$0 \leq x_j \leq 1, \qquad \text{(for every edge } j)$$

$$\sum_{j \in D} x_j \geq 1 \qquad \text{(for every directed cut } D).$$

It easily follows from the Lucchesi-Younger Theorem that the vertices of this polytope are 0-1 vectors corresponding to those sets of edges which cover all directed cuts. So to determine the minimum number of edges covering all directed cuts we have to minimize the linear objective function Σx_j over the polytope P. In order to apply the Ellipsoid Method, we only need a polynomial subroutine which checks $x \in P$ and if the answer is in the negative, it finds a constraint which is violated. The first set of constraints is easily checked one by one, so we may assume that $x \geq 0$ and that we are only interested in checking whether or not all the directed cut-constraints are satisfied. This is clearly equivalent to the following problem :

Given a digraph and a weight on every edge, find a directed cut with minimum weight.

This can be reduced to the problem of minimizing a submodular function as follows. For every edge e of G, add a new edge which connects the endpoints of e in the reverse order, and let its weight be N, where $N > \Sigma x_j$. Define $\delta(X)$ as the sum of weights of edges connecting X to $V(G) - X$ in this new graph. Then $\delta(X)$ is submodular, and moreover, a non-empty set minimizing it necessarily determines a directed cut. So it suffices to minimize the submodular set-function $\delta(X)$ over the non-empty subsets of $V(G)$, which as we have seen, can be done in polynomial time.

By similar methods we can find algorithms which go with the minimax theorems in Edmonds-Giles [6] and Frank [9]. (These results generalize several minimax results, like polymatroid intersection (Edmonds [3,5], Lawler [17]), optimum branching (Edmonds [2]), packing of rooted cuts (Fulkerson [12]), max-flow-min-cut (Ford-Fulkerson [8]), packing of directed cuts (Lucchesi and Younger [20]), etc.)

Let us conclude with the discussion of the matching problem. This needs an extension of the submodular function minimization problem. The following result generalizes a recent algorithm of Padberg and Rao [21].

Theorem 2.2. Let f be a submodular set-function defined on the subsets of a finite set S. Assume that we have a subroutine to compute $f(X)$ in time at most T. Let H be a collection of subsets of S with the property that $X \in H$, $Y \notin H$, $X \cap Y \notin H$ implies that $X \cup Y \in H$. Also assume that we have a subroutine to check $X \in H$ in time at most T'. Then there is an algorithm to find

$$\min\{f(X) : X \in H\}$$

in time polynomial in $|S|$, T and T'.

An example of a collection H of subsets with the given property is the collection of all subsets with odd cardinality.

Let G be a graph with an even number of vertices and let us assign a non-negative weight w_j to each edge. We want to find a perfect matching with maximum total weight. By Edmonds [1] the convex hull of perfect matchings of G is given by the inequalities

$$x_j \geqq 0 \qquad \text{(for every edge } j\text{)}$$

$$\sum_{j \ni v} x_j = 1 \qquad \text{(for every vertex } v\text{)}$$

$$\sum_{j \in C} x_j \geqq 1 \qquad \text{(for every odd cut } C\text{)},$$

where an odd cut means the set of edges connecting X to $V(G) - X$ for some $X \subseteq V(G)$, $|X|$ odd. To be able to apply the Ellipsoid Method we have to be able to check if a vector x satisfies these inequalities. The first two kinds of constraints are easily checked one by one. To deal with the third kind, we need a subroutine to find an odd cut C for which $\Sigma_{j \in C} x_j$ is minimum. But this means to minimize a submodular function over the odd-size subsets of $V(G)$. By Theorem 2.2, this can be done in polynomial time.

3. CONCLUDING REMARKS

1. In the combinatorial applications discussed above we were only concerned with the speed of our algorithms up to polynomiality. As a matter of fact the running times, as far as we could estimate, are rather poor. For those applications where polynomial algorithms have been known before these are much faster than ours. The main point has been to solve all these problems using one technique, the Ellipsoid Method.

It is natural that such a general approach cannot compete with special-purpose algorithms. For those problems to which the Ellipsoid method seems to have yielded the first polynomial solutions, this fact should be a challenge to find better algorithms making better use of the specialities of the problem.

2. The Ellipsoid Method applies to convex bodies other than polyhedra, and this fact can also be utilized in combinatorics. An application of this kind is an algorithm to compute the independence number of a perfect graph in polynomial time [14]. More generally, one can compute a rather sharp upper bound for the independence number of an arbitrary graph (see [18]).

3. It may be interesting to point out that the majority of those combinatorial optimization problems which are known to be polynomially solvable, can be solved by a combination of the Ellipsoid Method and the Greedy Algorithm.

4. The Ellipsoid Method shows that the investigation into the facial structure of polytopes may be useful in designing algorithms in a more specific way than just "gaining insight". This is particularly significant if we consider that the description of facets of combinatorial polyhedra means a "good characterization" of the maximum value of linear objective functions. So this is a quite general situation where knowing a "good characterization" helps in designing a good algorithm. Thus the Ellipsoid Method may also contribute to the problem whether $NP \cap coNP = P$.

REFERENCES

1. J. Edmonds, Maximum matching and a polyhedron with 0,1-vertices, J. Res. Nat. Bur. Stan. B 69(1965), 125-130.
2. J. Edmonds, Optimum branchings, J. Res Nat. Bur. Stan. B 71(1967), 233-240.
3. J. Edmonds, Submodular functions, matroids, and certain polyhedra, in : Combinatorial Structures and their Appl. (Proc. Intern. Conf. Calgary, 1969; R. Guy, H. Hanani, N. Sauer, J. Schönheim, eds.), Gordon and Breach, N.Y. 1970, 69-87.
4. J. Edmonds, Edge-disjoint branchings, in : Combinatorial Algorithms (Courant Comp. Sci. Symp. Monterey, 1972; R. Rustin, ed.) Academic Press, N.Y. 1973, 91-96.
5. J. Edmonds, Matroid intersection, Annals of Disctrete Math., 4(1979), 39-49.
6. J. Edmonds and R. Giles, A min-max relation for submodular functions on graphs, Annals of Discrete Math., 1(1977), 185-204.
7. J. Edmonds and E.L. Johnson, Matching, Euler tours, and the Chinese postman, Math. Prog., 5 (1973), 88-124.

8. L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton Univ. Press, Princeton, N.J. 1962.

9. A. Frank, Kernel systems of directed graphs, Acta Sci. Math. Szeged 41(1979), 63-76.

10. A. Frank, How to make a digraph strongly connected? Combinatorica (submitted).

11. D.R. Fulkerson, Packing rooted directed cuts in weighted directed graphs, Math. Prog., 6(1974), 1-13.

12. D.R. Fulkerson, Anti-blocking polyhedra, J. Combinatorial Theory,B 12(1972), 50-71.

13. P. Gács and L. Lovász, Khachiyan's algorithm for linear programming, Math. Prog. Studies (submitted).

14. M. Grötschel, L. Lovász and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica (submitted).

15. A.V. Karzanov, On the minimal number of arcs of a digraph meeting all its directed cut sets. (To appear).

16. L.G. Khachiyan, A polynomial algorithm in linear programming, Dokl. Akad. Nauk SSSR 244(1979), 1093-1096.

17. E.L. Lawler, Optimal matroid intersections, in : Combinatorial Structures and their Applications (Proc. Intern. Conf. Calgary, 1969; R. Guy, H. Hanani, N. Sauer, J. Schönheim, eds.), Gordon and Breach, N.Y. 1970, 233-235.

18. L. Lovász, On the Shannon capacity of a graph, IEEE Trans. on Inf. Theory, 25(1979), 1-7.

19. C.L. Lucchesi, A minimax equality for directed graphs, Thesis, University of Waterloo, 1976.

20. C.L. Lucchesi and D.H. Younger, A minimax relation for directed graphs, J. London Math. Soc., 17(1978), 369-374.

21. M.W. Padberg and M.R. Rao, Minimum cut-sets and b-matchings, (to appear).

22. N.Z. Shor, Convergence rate of the gradient desent method with dilatation of the space, Kibernetika 2(1970), 80-85.