MIT/LCS/TM-154

ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL OPTIMIZATION PROBLEMS

Richard M. Karp
Christos H. Papadimitriou

February 1980

ON LINEAR CHARACTERIZATIONS OF COMBINATORIAL OPTIMIZATION PROBLEMS

Richard M. Karp*
Computer Science Division
Department of Electircal Engineering
and Computer Science
University of California
Berkeley, California 94720

Christos H. Papadimitriou**
Laboratory for Computer Science
M.I.T.
Cambridge, Mass. 02139

ABSTRACT:

We show that there can be no computationally tractable description by linear inequalities of the polyhedron associated with any NP-complete combinatorial optimization problem unless *NP*=co-*NP* -- a very unlikely event. We also use the recent result by Khacian to present even stronger evidence that NP-complete combinatorial optimization problems cannot have efficient generators of violated inequalities.

*Research supported by National Science Foundation Grant MCS 77-09906

**Research supported by National Science Foundation Grant MCS 76-01193 and a Miller Fellowship

Key Words: Combinatorial Optimization problems, linear programming, convex polytopes, *P*, *NP*, and *co-NP*, Khacian;s algorithm.

1. INTRODUCTION

It is well known that several important combinatorial optimization problems can be formulated as the maximization of a linear functional over a polytope with 0-1 vertices. Examples include matching [5], the knapsack problem [1], the traveling-salesman problem [4], vertex packing and set packing [17],[18], the three-dimensional matching problem and many others.

There has been a very large body of literature aimed at the characterization of such convex polytopes by linear inequalities. The motivation apparently has been that such a characterization would bring a combinatorial optimization problem within the scope of linear programming methods, and thus might yield an efficient algorithm for its solution. This approach has worked in some cases, most notably the matching problem of Edmonds [6].

Unfortunately, there is now strong evidence that many combinatorial optimization problems are not amenable to efficient algorithms because they are NP-complete [14],[8]. This means that there is no polynomial-time algorithm that solves these problems unless $P=NP$, a very unlikely event. P is the class of problems for which efficient algorithms exist. NP is the class of sets that admit "good characterizations" in the sense of Edmonds. NP contains (appropriately stylized versions of) all "reasonable" combinatorial optimization problems, i.e., those for which feasibility checking and cost evaluation can be done efficiently. All combinatorial optimization problems mentioned so far are in NP; and all but matching are NP-complete.

Despite the evidence that NP-complete problems are intractable, research on the description by linear inequalities of the convex polytopes associated with NP-complete problems has continued - besides the above references we mention [3],[9],[10],[11],[16] and [21]. The main motivation has been the development of empirically reasonably efficient algorithms by applying the simplex method to a heuristically generated subset of the inequalities describing

the polytope. In order for such an algorithm to be guaranteed to terminate at the optimum, a complete description of the polytope by inequalities must be available. If only a partial description is used, then certain objective functions will force the simplex algorithm to terminate at an infeasible point. Unfortunately, so far, despite much intensive research effort, there has been no satisfactory description by linear inequalities of any convex polytope corresponding to an NP-complete combinatorial optimization problem. Note that, since an exponential number of inequalities might be required, such a description would not directly imply *P=NP* via the recently discovered polynomial-time algorithm for linear programming [15].

In this note we point out that no satisfactory description by linear inequalities of the polytope corresponding to an NP-complete combinatorial optimization problem is possible, unless *NP=co-NP*. The class *co-NP* consists of those sets whose complements are in *NP*. The hypothesis that *NP=co-NP* is weaker than *P=NP*, but is generally considered almost as improbable. For example, *NP=co-NP* would imply that there is a "good" characterization of non-Hamiltonian graphs, and that there is a short proof of every contradiction in the propositional calculus.

We must, of course, define what we mean by a "satisfactory" characterization. For characterization to be satisfactory we only require that the set of inequalities comprising the description be in *NP*. In other words, if an inequality is presented to us, we should be able somehow to produce a short argument proving that it is in the set of inequalities describing the polyhedron. All such descriptions discussed in the literature are in *NP*; in fact, all but the comb inequalities of [3] and their generalizations given in [10] are in *P*.

A second result concerns the generation of violated inequalities. Given a combinatorial optimization problem and a point $x \in R^n$, such a generator either determines that x lies in the convex polytope

associated with the problem, or else produces a linear inequality violated by x but satisfied by all points in the polytope. We prove that, if there exists a polynomial-time generator of violated inequalities for an NP-complete combinatorial optimization problem, then $P=NP$. The proof depends in an interesting way on Khacian's polynomial-time linear programming algorithm [15].

Our results strongly suggest that one cannot attain a usable complete characterization by inequalities of the polytopes that correspond to hard combinatorial optimization problems.

2. COMBINATORIAL OPTIMIZATION PROBLEMS AND FACIAL DESCRIPTIONS

A common type of combinatorial optimization problem is the following:

$$\begin{aligned} &\text{maximize } c\cdot x \\ &\text{subject to } x\in S \end{aligned} \qquad (1)$$

where[†] $S\subseteq Z^n$ is the set of feasible solutions and $c\in Z^n$. It is well known that an equivalent formulation of (1) is the following:

$$\begin{aligned} &\text{maximize } c\cdot x \\ &\text{subject to } x\in CH(S) \end{aligned} \qquad (2)$$

where CH(V) denotes the convex hull of the point set V.

Taking a view toward algorithmic issues, we shall carefully distinguish between problems and instances. A problem will generally have an infinite number of instances, each of which is similar in form to (1).

Definition 1 A combinatorial optimization problem (or, briefly, c.o.p) C is specified by;

(i) a set $L\subseteq\{0,1\}^*$;

(ii) a function n from L into Z^+;

(iii) for each $z\in L$, a set $S(z)\subseteq (Z^+)^{n(z)}$ such that each of the following three languages is recognizable in polynomial time:

[†]We denote the integers by Z, the nonnegative integers by Z^+, and the rationals by R.

L, $\{\langle z,y\rangle \mid |y|=n(z)\}$ and $\{\langle z,x\rangle \mid x \in S(z)\}$.

Definitions An instance of C is a pair $\langle z,c\rangle$ where $z \in L$ and $c \in Z^{n(z)}$.

The instance $\langle z,c\rangle$ corresponds to:

$$\min \sum_{j=1}^{n(z)} c_j x_j \qquad (3)$$

subject to $x = (x_1, x_2, \ldots, x_{n(z)}) \in S(z)$.

Weighted matching, set covering, integer programming, the traveling-salesman problem and a plethora of other problems can be expressed as combinatorial optimization problems. In one formulation of the undirected traveling-salesman problem, for example, z is the binary representation of a positive integer n, n(z) is $\binom{n}{2}$, the number of edges in K_n, the complete graph on n vertices, and S(z) is the set of characteristic vectors of the Hamiltonian circuits of K_n.

Given a c.o.p C, define D(C), the decision problem for C as

$$D(C) = \{\langle z,c,k\rangle \mid \langle z,c\rangle \text{ is an instance of C, } k \in Z \text{ and } \exists x \in S(z) \text{ such that } c \cdot x \geq k\}.$$

If D(C) is NP-complete, then C is called an NP-complete combinatorial optimization problem.

A facial description of a c.o.p. C is a set F(C) such that:

(i) each element of F(C) is of the form

$\langle z,f,g\rangle$ where
$z \in L$, $f \in Z^{n(z)}$ and $g \in Z$

(ii) for each $z \in L$, and for all $x \in R^{n(z)}$, the following are equivalent.

a) $x \in CH(S(z))$

b) for each triple $\langle z,f,g\rangle \in F(C)$, $f \cdot x \leq g$.

Thus F(C) gives a description by linear inequalities of CH(S(z)), for each $z \in L$.

The facial description F(C) is called a small facial description if there is a polynomial $p(\cdot)$ such that, for every $\langle z,f,g\rangle \in F(C)$, each component of f,g has absolute value $\leq 2^{p(|z|+n(z))}$. The existence of a small facial description implies, in particular, that, for every $z \in L$, CH(S(z)) is a convex polyhedron (i.e., it is the intersection of a finite number of half-spaces).

The c.o.p.'s that occur in practice invariably have small facial descriptions. We describe two especially common classes of such problems.

(1) Zero-one problems This is the case where every vector in S(z) is a 0-1 vector;

(2) Problems of integer programming type In this case the input z specifies an integer $m \times n$ matrix A and an integer n-vector b, and

$$S(z) = \{x \mid Ax \leq b,\ x \geq 0,\ x \text{ integer}\}$$

Lemma 1 Every zero-one problem or problem of integer programming type has a small facial description.

Proof Any convex polyhedron Q in R^n can be expressed in terms of a finite set V of vertices and a finite set W of extreme rays; Q is just the set of vectors of the form x_1+x_2, where x_1 is a convex combination of vertices and x_2 is a positive combination of extreme rays. Let S be the unique minimum-dimensional affine subspace of R^n containing Q, and let the dimension of S be d. Then $S-v_1=\{x-v_1 \mid x \in S\}$ is a linear subspace of dimension d. Let B be a set of n-d unit vectors, none of which lie in $S-v_1$. Then Q can be described by a finite number of linear enequalities, each of the form $f \cdot x \leq g$, where $f \cdot x=g$ is the equation of a supporting hyperplane of Q. It follows that f and g are determined by some selection process of the following type:

select $\ell+1$ vertices $v_0,v_1,\ldots v_\ell$, where $0 \leq \ell \leq n-1$, and $n-1-\ell$ vectors $h_1,h_2,\ldots h_{n-1-\ell}$ from $W \cup B$, such that

$\{v_1-v_0,v_2-v_0,\ldots,v_\ell-v_0,h_1,h_2,\ldots,h_{n-1-\ell}\}$ is linearly independent.

Then f and g are determined, up to a constant multiple, by:

$g=f\cdot v_0$

$f\cdot(v_i-v_0)=0 \qquad i=1,2,\ldots,\ell$

$f\cdot h_j=0 \qquad j=1,2,\ldots,n-1-\ell$

We first show that, in the two cases of interest, the vertices and extreme rays of CH(S(z)) are integer vectors whose coefficients are small. In the zero-one case this is especially simple: the vertices are zero-one vectors, and there are no extreme rays. In the case where z is the binary encoding of an integer matrix A and a vector b, and $S(z)=\{x \mid Ax\le b, \text{ x integer}\}$, each extreme ray is a row of A; hence each of its coefficients is of absolute value $\le 2^{|z|}$. As for the vertices, we can rely on a result which was independently discovered recently by many people, including [2], Sieveking, S.A. Cook, and [13]: there is a polynomial $q(\cdot)$ such that every component of every vertex of $\{x \mid Ax\le b, \text{ x integer}\}$ is of absolute value $2^{q(s)}$, where $s=\sum_x \lceil \log(1+|x|)\rceil$. Here x ranges over all entries of A and b; thus, $s\sim|z|$.

Now we are ready to show that all coefficients of f and g are suitably small. Recall that f satisfies $W\cdot f=0$, where W is a $(n-1)\times n$ matrix of rank n-1; each row of W is either of the form v_i-v_0 or of the form h_j. Without loss of generality, assume that the first n-1 columns of W are linearly independent, and write

$W = C\vdots d$, where C is a non-singular $(n-1)\times(n-1)$ matrix, and d is a column vector. Then f is determined by

$$\begin{matrix} f_1 \\ \vdots \\ f_{n-1} \end{matrix} + C^{-1} d\, f_n = 0$$

By Cramer's rule,

$$(C^{-1})_{ij} = \frac{(-1)^{i+j}\,\Delta_{ji}}{|C|}$$

where Δ_{ji} is the j-1 minor of C. Hence, we can take f to be the following integer vector (or its negative).

$$f_i = \sum_{j=1}^{n-1} (-1)^{i+j} \Delta_{ji} d_j \quad i=1,2,\ldots,n-1 \quad f_n = |C|.$$

It follows that each component of f or g is $\leq (2nx)^n$, where x is the largest absolute value of an entry in a vertex or extreme ray. And the result that, for a suitable polynomial p, each coefficient is $\leq 2^{p(|z|+n(z))}$ now follows from the bounds derived earlier on the coefficients of vertices and extreme rays.□

3. THE COMPUTATIONAL COMPLEXITY OF SMALL FACIAL DESCRIPTIONS

The following theorem is our main result.

Theorem 1 Let C be a c.o.p. and let F(C) be a small facial description. If $F(C) \in NP$ then $D(C) \in co\text{-}NP$.

Proof Assuming as given a nondeterministic polynomial-time algorithm for recognizing the triples $\langle z,f,g\rangle \in F(C)$, we give a nondeterministic polynomial-time algorithm for recognizing the complement of D(C).

Algorithm B

(i) If the input is not of the form $\langle z,c,k\rangle$, where $z \in L$, $c \in Z^{n(z)}$ and $k \in Z$, then accept the input and halt.

(ii) Generate nondeterministically a $n \times n$ matrix $F=(f_{ij})$ of integers having absolute value $\leq 2^{p(|z|+n(z))}$ and a n-vector g of integers having absolute value $\leq 2^{p(|z|+n(z))}$

(iii) Apply the nondeterministic polynomial-time recognition algorithm for F(C) to verify that each triple $\langle z,(f_{i1},f_{i2},\ldots,f_{in}),g_i\rangle$ is in F(C);

(iv) Verify that F is nonsingular and (in polynomial time) solve the system, $y^T F=c$;

(v) Verify that $y \geq 0$ and $y^T b > k$.

Algorithm B clearly runs in polynomial time. To prove that it accepts the complement of D(C), we note the equivalence of the following statements:

(i) $\langle z,c,k\rangle \notin D(C)$;

(ii) the program

max $c \cdot x$

subject to $x \in CH(S(z))$

has optimal value $< k$;

(iii) the program

max $c \cdot x$

(I) subject to $f \cdot x \leq g$ $\langle z,f,g \rangle \in F(C)$

has optimal value $< k$;

(iv) the dual of program (I) has optimal value $< k$;

(v) the dual of program (I) has a basic feasible solution of value $< k$;

(vi) there exists a $n(z) \times n(z)$ matrix (f_{ij}) and a $n(z)$-vector g such that

$$\langle z,(f_{i1},f_{i2},\ldots,f_{in(z)}),g_i \rangle \in F(C),\ i=1,2,\ldots,n(z)$$

and the system

$$y^T F = c$$

has a unique nonnegative solution y such that

$y^T g < k$. □

The following Corollary constitutes our evidence that computationally tractable facial descriptions for NP-complets combinatorial optimization problems are unlikely to exist.

Corollary 1 If $F(C)$ is a small facial description of a NP-complete c.o.p., and $F(C) \in NP$, then $NP = co\text{-}NP$.

Proof The NP-complete language $D(C)$ is in $co\text{-}NP$. Since the complement of every language in NP is reducible to the complement of $D(C)$, it follows that $co\text{-}NP \subseteq NP$, and $NP = co\text{-}(co\text{-}NP) \subseteq co\text{-}NP$ □.

Our approach can also prove a slightly different kind of result. Let $F(C)$ be a collection of valid inequalities for C i.e., each element of $F(C)$ is a triple $\langle z,f,g \rangle$, such that $f \cdot x \leq g$ holds for every $x \in S(z)$. Suppose that $F(C) \in NP$. Call an instance $\langle z,d \rangle$ of C bad for F if the optimum solution for the instance is not optimum for

max $c \cdot x$

subject to $f \cdot x \leq g$ $\langle z,f,g \rangle \in F(C)$

Let I be a subset of the set of instances of C such that $I \in P$ but $\{\langle z,c,k \rangle \mid \langle z,c \rangle \in I$ and $\langle z,c,k \rangle \in D(C)\}$ is NP-complete.

Claim 1 If $NP \neq co\text{-}NP$ then I contains infinitely many instances that are bad for F.

Example Let F(TSP) be the ingenious partial characterization of the facets of the traveling-salesman polytope given in [10]. Call an instance of the traveling-salesman problem Euclidean if the cost vector can be realized as the L_2 distances of a finite set of points on the place. Then, since the Euclidean restriction of the traveling-salesman problem is NP-complete [19] we conclude that, unless $NP=co\text{-}NP$, there exist infinitely many bad Euclidean instances of the TSP. Similarly, since the Hamiltonian circuit problem is NP-complete, we can claim that, unless $NP=co\text{-}NP$, there exist infinitely many bad instances in which each component of c is 0 or 1.

4 THE COMPUTATIONAL COMPLEXITY OF GENERATORS

This section concerns the complexity of algorithms which generate violated inequalities. Given a c.o.p. C, a generator of violated inequalities is an algorithm G(C) which accepts as input pairs of the form $\langle z,p \rangle$, where $z \in L$ and $p \in R^{n(z)}$. The output of G(C) is a follows:

if $P \in CH(S(z))$ *then* 'O.K.'

else a pair (f,g) such that $f \in Z^{n(z)}$, $g \in Z$, $f \cdot p > g$ and, for all $x \in S(z)$, $f \cdot x \leq g$.

Associated naturally with any generator G(C) is the following facial description F(C):

$F_G(C) = \{\langle z,f,g \rangle \mid$ for some p, G(C) has input (z,p) and output (f,g)$\}$.

The generator G(C) is called a small generator if $F_G(C)$ is a small facial description.

Most attempts to solve a combinatorial optimization problem C by exploiting its facial structure require the use of a generator. Given an instance $\langle z,c \rangle$, the typical approach is to start with a collection of inequalities $Ax \leq b$ satisfied by all $x \in S(z)$, and solve

the linear program

$$\max c \cdot x$$
$$\text{subject to } Ax \leq b$$

If the optimal solution $x^{(1)}$ is not in $CH(S(z))$ then one or more valid inequalities are generated and adjoined to the linear program. The process of solving a linear program and then generating and adjoining further valid inequalities is repeated until an optimal solution $x^{(k)}$ is found which actually lies in $CH(S(z))$. In practice the method of generating inequalities is ad hoc, and may even entail human intervention in the computational process [9],[27].

Theorem 2 Let C be a c.o.p, and let G(C) be a small generator of violated inequalities for C. If G(C) runs in polynomial time, then $D(C) \in P$.

Corrollary 2 If C is NP-complete, then it has no small polynomial-time generator unless $P=NP$. □

To prove Theorem 2 we show that, using a small polynomial-time generator G(C), one can test in polynomial time whether $\langle z,c,k\rangle \in D(C)$. This is done by applying a variant of Khachian's algorithm for testing the feasibility of a system of linear inequalities [15],[7]. The algorithm is applied to the system

$$\begin{aligned} &c \cdot x \geq k \\ &f \cdot x \leq g \quad \langle z,f,g\rangle \in F_G(C), \end{aligned} \tag{4}$$

which is feasible if and only if $\langle z,c,k\rangle \in D(C)$.

The proof of Theorem 2 depends on a series of lemmas. The proofs of some of these Lemmas (namely 2,3, and 6) which are nearly identical to the proofs of similar lemmas in [7] will be omitted.

Let $p(\cdot)$ be a polynomial such that, for every $\langle z,f,g\rangle \in F_G(C)$, each component of f,g has absolute value $\leq 2^{p(|z|+n(z))}$. For a fixed z, let $t=(n(z)+1)^2 \cdot (p(|z|+n(z))+1)+\sum_j (\lceil \log_2 c_j \rceil+1)+(\lceil \log_2 k \rceil+1)$;

t is an upper bound on the number of binary digits needed to write down all affinely independent subsystems of the system (4).

Lemma 2 The system (4) is feasible if and only if the systems (5) and (5½) are both feasible, where

$$\begin{aligned} &c\cdot x > k - 2^{-t} \\ &f\cdot x < g + \epsilon/\|f\| \quad \langle z,f,g\rangle \in F_G(C) \qquad (5) \\ &-2^t \le x_j \le 2^t \quad j=1,2,\ldots,n(z), \end{aligned}$$

and (5½) is (5) with the first inequality replaced by $c\cdot x > k - \frac{1}{2}2^{-t}$. □

Here by $\|f\|$ we denote the L_2 norm of the vector f, and $\epsilon = 2^{-(2n(z)+2)t}$. Notice that $fx < g + \epsilon/\|f\|$ is satisfied by those points x that have Euclidean distance less than ϵ from some point x' satisfying $fx' \le g$.

Lemma 3 If the system (5) is feasible, then the set T of feasible points has volume in $R^{n(z)}$ at least $\epsilon^{n(z)}$. □

Using the small generator G(C) for C, we shall test the feasibility of (5) --and (5½)-- by an iterative process. At the beginning of each iteration, the state of the computation is given by a pair (p,A), where $p \in R^{n(z)}$, A is an $n(z)\times n(z)$ positive definite matrix, and the feasible region T contained in the ellipsoid $E(p,A) = \{x: (z-p)^T A^{-1}(z-p) \le 1\}$. At the beginning of the computation $p = \vec{0}$ and $A = 2^t \cdot I_{n(z)}$.

At each iteration we identify, using G(C), either an inequality of (5½) violated by p, or a feasible point of (5). This is done as follows: We first check whether $c\cdot p \le k - \frac{1}{2}2^{-t}$; if so, we have found our inequality. Otherwise, we call G(C) with input (z,p). If the output of G(C) is "O.K.", then we are done, because we have identified a feasible point of (5). Also, if the output of G(C) is (f,g) and if so happens that $fp < g + \epsilon/\|f\|$, then we have found the inequality sought.

The tricky case is that in which G(C) returns (f,g), and $g + \epsilon/\|f\| \ge f\cdot p > g$. We cannot call G(C) again to obtain another inequality, since G(C) may keep returning (f,g). Our algorithm for identifying a violated inequality of (5½) proceeds in this

case as follows: We create a sequence of points $p=p_0,p_1,\ldots,p_m$, and a sequence $(f_0,g_0),\ldots,(f_{m-1},g_{m-1})$ of inequalities in $F_G(z)$, for some $m\le n(z)$. We let $H_j=\{x:f_jx=g_j\}$ $j=0,\ldots,m-1$.

The sequences $\{p_j\}$ and $\{(f_j,g_j)\}$ have the following properties

(a) The hyperplanes in $\{H_j\}$ are affinely independent, and $p_j\in \bigcap_{i<j} H_i$ for $j=0,\ldots,m$.

(b) $\|p_j-p\|\le\epsilon(2^{jt}-1)$ for $j=0,\ldots,m$.

(c) The distance of p_j from H_j is $\le 2^{jt}\epsilon$, for $j=0,\ldots,m-1$.

(d) The output of G(C) to input (z,p_m) is either ""O.K." or (f_m,g_m), such that $f_mp\ge g_m+\epsilon/\|f\|$.

Originally, with $m=0$ and $p_0=p$, the conditions (a-c) are vacuously satisfied. Inductively, once we have defined $p_0,\ldots p_j$, we call G(C) with input (z,p_j). If the output is "O.K.", or a (f_j,g_j) satisfying (d), we stop with $m=j$. Otherwise, we let H_j be the hyperplane corresponding to the output of G(C). We have to prove that $H_0,\ldots,H_j$ are all affinely independent.

Lemma 4 Let r be a rational point with denominators bounded by 2^t, and let H be a small hyperplane such that $x\notin H$. Then the distance from r to H is at least $2^{-(n(z)+1)t}$.

Proof This distance is $|fr-g|/\|f\|$, where $H=\{x:fx=g\}$. The numerator is a positive integer, whereas the denominator is at most $\|f\|2^{n(z)t}\le\sqrt{n}\cdot 2^{p(z)+n(z))}\cdot 2^{n(z)\cdot t}\le 2^{+(n(z)+1)t}$. □

Corollary Let r be a point at the intersection of affinely independent small hyperplanes $\{H_j\}_{j=1}^m$ and let H be a small hyperplane such that $H\cap\bigcap_{j\le m} H_j=\emptyset$. Then the distance from r to H is at least $2^{-(n(z)+1)t}$.

Proof The flat $\bigcap_{j\le m} H_j$ has at least one rational point r' with denominators at most 2^t, and the distances from r and r' to H are equal. Apply Lemma 4. □

Therefore, if H_j were affinely dependent on $H_0,\ldots,H_{j-1}$, then the distance from p_j to H_j would have been at least $2^{-(n(z)+1)t}$, and therefore the distance from p to H_j at least $2^{-(n(z)+1)t}-\epsilon(2^{jt}-1)\geq\epsilon$, a contradiction.

Once we have established that $\bigcap_{i\leq j} H_i \neq\emptyset$, we define p_{j+1} by projecting p_j onto $\bigcap_{i\leq j} H_i$. We have to show that (a),(b) and (c) hold for p_{j+1}, in order to conclude the inductive step.

(a) is immediate from the discussion above. To show (b) and(c) we nee the following Lemma:

Lemma 5 Let $H_0,\ldots,H_{j+1}$ be affinely independent small hyperplanes, let $r\in\bigcap_{i\leq j} H_i$, and let the distance from r to H_{j+1} be δ. Then the distance r to $\bigcap_{i\leq j+1} H_i$ is at most $(2^t-1)\delta$.

Proof Without loss of generality $H_{j+1}=\{x:x_1=0\}$, and $r=(\delta,r_2,\ldots,r_{n(z)})$. Also without loss of generality, suppose that the columns 2,3,...,j+1 of the matrix whose rows are $f_1,\ldots,f_j$ (where $H_i=\{x:f_i x=g_i\}$, i=1,...,j) are independent. The point r is thus obtained by setting $x_k=r_k$ for k=j+2,...,n(z), $x_1=\delta$ and solving for the remaining x's. Similarly, one point r' in $\bigcap_{i\leq j+1} H_j$ is obtained by setting $x_n=r_k$ for k=j+2,...,n(z),$x_1=0$ and solving for the remaining x's. It is easy to see that $\|r-r'\|\leq\delta 2^{p(|z|+n(z))}\cdot n^2(z).b$, where b is the largest in absolute value element of the inverse of the non-singular matrix of the columns 2 through j+1 of the matrix whose rows are $f_1,\ldots f_j$. Thus $\|r-r'\|\leq(2^t-1)\delta$. □

(c) follows immediately from the Lemma. Also for (b), the distance of p_j from p_{j+1} is at most $(2^t-1)2^{jt}\epsilon$, and thus the distance from p to p_{j+1} is at most $(2^t-1)2^{jt}\epsilon+(2^{jt}-1)\epsilon=(2^{(j+1)t}-1)\epsilon$, and (b) is proved.

It follows that this scheme will, after at most $n(z)$ projections, produce an inequality (f,g) for which our original point p, the center of $E(p,A)$, satisfies $fp \geq g + \epsilon/\|f\|$; or it will produce a point p_j satisfying $f \cdot p_j < g + \epsilon/\|f\|$ for all $\langle z,f,g \rangle \in F_G(C)$; furthermore, this point p_j satisfies $\|p_j - p\| \leq \epsilon(2^{n(z)t} - 1)$, and hence $cp_j > k - 2^{-t}$, given that $c \cdot p > k - \frac{1}{2}2^{-t}$.

To summarize, given the center p of the ellipsoid $E(p,A)$, we can in polynomial time either determine a feasible point, or isolate a violated inequality of (5). Then using this violated inequality a new pair p',A' is computed. The ellipsoid $E_{(p',A')}$ has smaller volume than $E_{(p,A)}$, but does include those points in $E_{(p,A)}$ which satisfy the inequality violated at p. In particular, $T \subseteq E_{(p',A')}$. Following the proof in [7], the following convergence result is obtained.

Lemma 6 There is a constant c and a polynomial π such that, if (5) is feasible, then a feasible solution will be found within $\pi(n,t)$ iterations. This is time even if intermediate results are kept at only cnt bits of precision.□

Our variant of Khachian's algorithm tests feasibility of (5), and hence membership of $\langle z,c,k \rangle$ in $D(C)$, in polynomial time. This completes the proof of Theorem 2.

Our discussion of Khachian's algorithm involves a refinement which promises to be useful in other contexts. In discussing Khachian's method for testing the feasibility of a system $Ax \leq b$ of m inequalities in n variables, references [15] and [7] introduce a parameter L representing the sum of the lengths in binary of all the coefficients of the system. The bound on the number of iterations is stated in terms of L. However, the analyses remain valid if the parameter is instead taken to be an upper bound t on the sum of the lengths in binary of the coefficients of any $(n+1) \times (n+1)$ subsystem of the original system. This observation greatly improves the time bound in cases where $m >> n$.

Finally, we note that there is a positive way of looking at Theorem 2. Consider the class of combinatorial optimization problems that are known to be in P -- such as the minimum

spanning tree problem, matching, max flow, matroid intersection and parity, and many others. Khacian's algorithm and Theorem 2 may be considered as a unifying algorithm which solves all of them, using generators of violated inequalities for each. It is therefore an interesting problem to develop polynomial-time generators of violated inequalities for these classical combinatorial optimization problems. (This was recently done for matching by Padberg and Rao).

REFERENCES

[1] Balas, E., "Facets of the knapsack polytope", Mathematical Programming 8, pp 146-164, 1975.

[2] Borosh, I, L.B. Treybig, "Bounds on positive integral solutions to linear Diophantine equations", Proceedings of the American Mathematical Society 55, 299-304, 1976.

[3] Chvátal, V., "Edmonds polytopes and weakly Hamiltonian graphs", Mathematical Programming 5, 29-40, 1973.

[4] Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson, "Solutions of a large-scale traveling-salesman problem", Operations Research 2, 393-410, 1954

[5] Edmonds, J., "Maximum matching and a polyhedron of 0-1 vertices", Journal of Research of the National Bureau of Standards, B, 69, 125-130, 1965.

[6] Edmonds, J. "Paths, trees and flowers", Canadian Journal of Mathematics 17, 449-467, 1965.

[7] Gács, P. and L. Lovász, "Khacian's Algorithm for Linear Programming", Computer Science Department Report Stanford Unversity, 1979.

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

[9] Grötschel, M. Polyedrische Characterisierungen Kombinatorischer Optimierungsprobleme, Verlag Anton Hain, 1977.

[10] Grötschel, M. and M.W. Padberg, "On the Symmetric Traveling-Salesman Problem I: Inequalities", Mathematical Programming 16, No. 3, 265-280, 1979.

[11] Grötschel, M. and M.W. Padberg, "On the Symmetric Traveling-Salesman Problem II: Lifting Theorems and Facets", Mathematical Programming 16, No. 3, 281-302, 1979.

[12] Hong, S. and M. Padberg, "On the solution of traveling salesman problems", 9th International Symposium on Mathematical Programming, Budapest, 1976.

[13] Kannan, R, and C.L. Monma, "On the computational complexity of integer programming problems", Report 7780-OR, Institut für Okonometrie und Operations Research, Bonn, 1977.

References continued

[14] Karp, R.M., "Reducibility among combinatorial problems", in Complexity of Computer Computations, R.E. Miller and J.W. Thatcher (eds.) Plenum, 1972.

[15] Khacian, L.G., "A Polynomial Algorithm for Linear Programming", Doklady Akademic Nank SSSR, Vol. 244, No. 5, 1093-1096, 1979.

[16] Maurras, J.F., "Some results on the convex hull of the Hamiltonian cycles of symmetric complete graphs" in Combinatorial Programming: methods and applications, B. Roy (ed.), Reidel, 1975.

[17] Nemhauser, G.L. and Trotter L.E., "Properties of vertex packing and independence systems polyhedra", Mathematical Programming 6, 48-61, 1974.

[18] Padberg, M., "On the facial structure of set packing polyhedra", Mathematical Programming 5, 199-215, 1973.

[19] Papadimitriou, C.H. "The Euclidean traveling salesman problem is NP-complete", Theoretical Computer Science 4, 237-244, 1977.

[20] Papadimitriou, C.H. and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall, to appear, 1980.

[21] Wolsey, L.A., "Facets of a linear inequality in 0-1 variables", Mathematical Programming 8, 165-178, 1975.