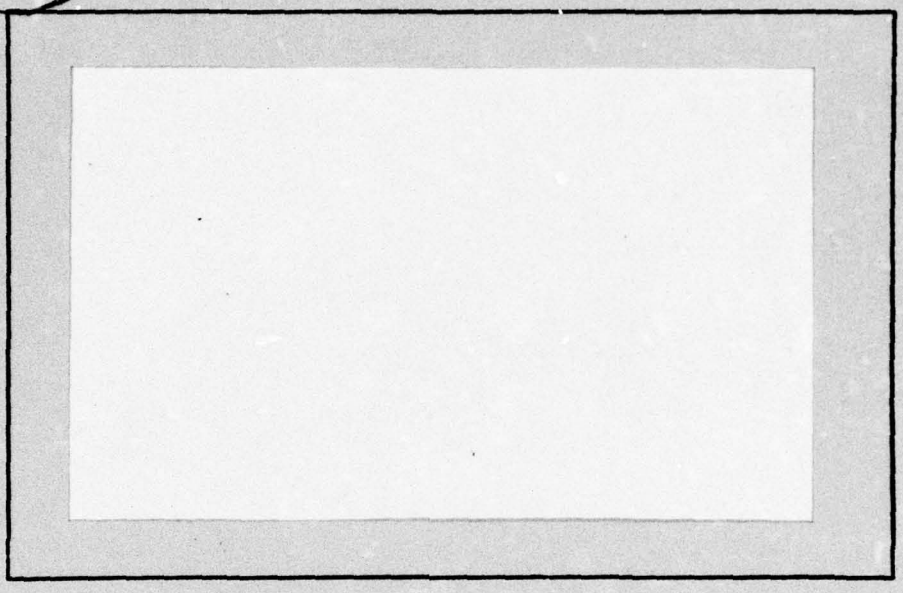
AD-A032 163 CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENC--ETC F/G 9/2
BRACKETING DISCRETE PROBLEMS BY TWO PROBLEMS OF LINEAR OPTIMIZA--ETC(U)
JUL 76 R JEROSLOW N00014-75-C-0621
UNCLASSIFIED RR-395 NL

| OF |
AD
A032163

END

DATE
FILMED
1-77

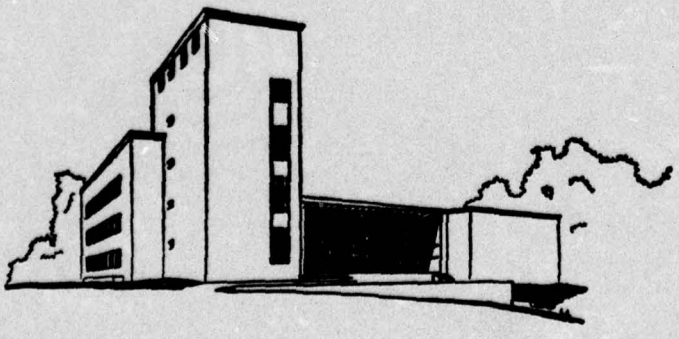


Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



Management Sciences Research Report No. 395

BRACKETING DISCRETE PROBLEMS BY TWO PROBLEMS OF LINEAR OPTIMIZATION

by Robert Jeroslow

July, 1976

This report was prepared as part of the activities of the Management Sciences Research Group, Carnegie-Mellon University, under Grant #GP 37510 X1 of the National Science Foundation and Contract N00014-75-C-0621 NR047-048 with the U.S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

Management Science Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

BRACKETING DISCRETE PROBLEMS BY TWO PROBLEMS OF LINEAR OPTIMIZATION

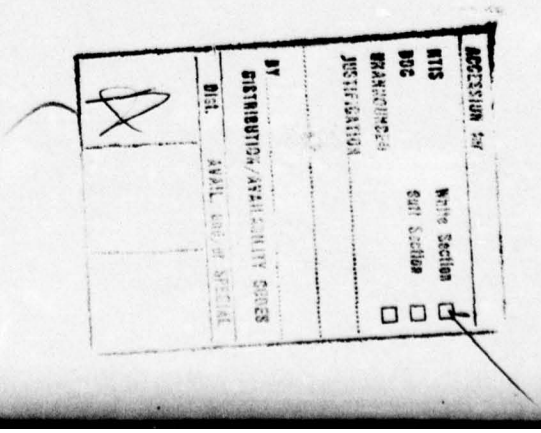
by Robert Jeroslow

We show that various graph-theoretic, logical, and integer optimization problems on discrete structures possess a computational complexity which is closely related to that of two problems of linear optimization.

The first problem considered is that of parametric linear programming, as used toward error analysis for errors in the criterion function alone. We show that the problem, of determining whether a certain compact formula is accurate to the first-order effects of these errors, has a polynomial time algorithm only if a large class of combinatorial problems (including the tautology problem) does (Theorem 1).

The second problem considered is that of ordinary linear programming, which is not known to have a polynomial-time algorithm [5]. We show that, if the tautology problem has a polynomial-time algorithm, then there also is one for linear programming.

Notation and terminology is from [4]. In particular, P resp. NP denotes that class of languages that is recognizable in polynomial time by a deterministic resp. nondeterministic Turing machine.



Abstract

We prove two results: (1) If a certain restricted problem of first-order error analysis in linear programming (specified below), for errors in only the criterion function, has a polynomial-time algorithm, then so does the tautology problem; (2) If the tautology problem is decidable in polynomial time, then linear programs can be solved optimally in polynomial time.

Key Words:

1) Linear programming
2) Integer programming
3) Computational complexity

The Main Results

Let a sentential form a be given in disjunctive normal form

$$a = \bigvee_{h=1}^{t} a^h \tag{1}$$

where each a^h is a conjunction of literals

$$a = \bigwedge_{p=1}^{t(h)} d(h,\rho)\, x^{h,\rho} \tag{2}$$

In (2), d(h,p) is either a blank or a negation sign "¬." We assume that not both $x^{h,p}$ and $\neg\, x^{h,p}$ occur in a^h. We view that a is written on a single line (no subscripting or superscripting) from left-to-right, so that (1) abbreviates a linear string $a^1 \vee a^2 \vee \ldots \vee a^t$; and similarly with (2). Also, $x^{h,p}$ abbreviates the symbol "x(α)" in which α is the index of the literal occurring in the h-th disjunct of (1), in the p-th position of this disjunct (2), and α is written in binary. Let the indices actually occurring for literals in (1) be $\{ i_1, \ldots, i_r \}$ (possibly literal x(11) occurs, but x(10) occurs nowhere).

We now associate to a^h in (2) a linear inequality system

$$A^h x \geq b^h \tag{3}$$

which is, by definition

$$
\begin{array}{rl}
 & -x(i_1) \geq -1 \\
 & \vdots \\
(4) \qquad & -x(i_r) \geq -1 \\
\delta'(h,1)\, x^{h,1} & \delta(h,1) \\
\vdots \quad \vdots & \vdots \\
\delta'(h,t(h))\, x^{h,t(h)} & \delta(h,t(h))
\end{array}
$$

In (4), the symbols $\delta(h,p)$ are given by

$$
(5) \qquad \delta(h,p) \text{ is } \begin{cases} \text{" } \geq 1\text{" , if } d(h,p) \text{ is a blank} \\ \text{" } \geq 0\text{" , if } d(h,p) \text{ is a negation sign;} \end{cases}
$$

and

$$
(6) \qquad \delta'(h,p) \text{ is } \begin{cases} \text{blank, if } d(h,p) \text{ is a blank} \\ \text{" - ", if } d(h,p) \text{ is a negation sign.} \end{cases}
$$

For use in Turing machines, inequality systems like (3) are viewed as written on one line, with inequalities separated by commas.

Example: Suppose that

$$
(7) \qquad \mathcal{a} \text{ is } \Big(x(10) \wedge x(101) \wedge \neg\, x(1)\Big) \vee \Big(x(1) \wedge \neg\, x(100)\Big)
$$

Then $t = 2$, $t(1) = 3$, $t(2) = 2$, and

$\{ i_1, \ldots, i_r \} = \{ 1, 10, 100, 101 \}$, so that $r = 4$.

In this case,

(8) a^1 is $x(10) \wedge x(101) \wedge \neg x(1)$

and (3), (4) becomes

$$
\begin{aligned}
-x(1) &\geq -1 \\
-x(10) &\geq -1 \\
-x(100) &\geq -1 \\
-x(101) &\geq -1 \\
x(10) &\geq 1 \\
x(101) &\geq 1 \\
-x(1) &\geq 0 \ .
\end{aligned}
\tag{9}
$$

A vector $x = \left(x(i_1),\ldots,x(i_r) \right)$ can be viewed as a (truth) valuation when its components are only zeroes and ones; we view zero as "false" and one as "true."

Lemma 1:

$$
\left\{ x \geq 0 \mid A^h x \geq b^h \right\} = \operatorname{conv} \left\{ x \;\middle|\; \begin{array}{l} x \text{ is a valuation} \\ \text{that makes } a^h \text{ true} \end{array} \right\} \tag{10}
$$

Proof: Clearly, if x is a valuation that makes a^h true, then $A^h x \geq b^h$; $x \geq 0$ is also evident. Hence the direction $\supseteq$ in (10) is evident.

For the converse, suppose $x \geq 0$ and $A^h x \geq b^h$. If x has only zero or one as components, clearly x makes a^h true. Otherwise, the proof is by induction on f, the number of fractional components of x.

If $f \geq 1$, let x(j) be a fractional component of x. From (4) and $x \geq 0$, we have $0 < x(j) < 1$. Since $A^h x \geq b^h$, x(j) cannot be among the

variables $x^{h,p}$ for $p = 1,\ldots,t(h)$. Let $x^{(0)}$ be x with x(j) changed to '0', and let $x^{(1)}$ be x with x(j) changed to '1.' Then we have a convex combination

$$x = x(j)\, x^{(1)} + \Big(1 - x(j) \Big)\, x^{(0)} . \tag{11}$$

Since $x^{(1)}$ and $x^{(0)}$ have one fewer fractional component, by induction

$$x^{(0)},\ x^{(1)} \in \operatorname{conv} \left\{ x \;\middle|\; \begin{array}{l} x \text{ is a valuation} \\ \text{that makes } \mathcal{a}^h \text{ true} \end{array} \right\} \tag{12}$$

Then from (11), x is in the r.h.s of (10).

Q.E.D.

In what follows, we put $c = (c(i_1),\ldots,c(i_r))$.

Lemma 2: Let U be any open set about $(0,\ldots,0) \in R^r$.

Then we have

$$\min \left\{ cx \;\middle|\; \begin{array}{l} \text{for at least one } h = 1,\ldots,t \\ \text{we have } A^h x \geq b^h,\ x \geq 0 \end{array} \right\} \tag{13}$$

$$\leq \Big(\sum_{j=1}^{r} \min \{ 0,\ c(i_j) \} \Big) + \frac{1}{2} \max \{ |c_j| \;\big|\; j=1,\ldots,r \}$$

holds for all $c \in U$, if and only if $\mathcal{a}$ in (1) is a tautology.

Proof: We have

$$(14)\qquad \min\left\{ cx \;\middle|\; \begin{array}{l}\text{for at least one } h = 1,\ldots,t \\ \text{we have } A^h x \geq b^h,\ x \geq 0\end{array} \right\}$$

$$= \min_{h=1,\ldots,t} \min \{ cx \mid x \geq 0,\ A^h x \geq b^h \}$$

$$= \min_{h=1,\ldots,t} \min \left\{ cx \;\middle|\; \begin{array}{l}x \text{ is a valuation} \\ \text{that makes } a^h \text{ true}\end{array} \right\}$$

$$= \min \left\{ cx \;\middle|\; \begin{array}{l}x \text{ is a valuation} \\ \text{that makes } a \text{ true}\end{array} \right\}$$

The second equality in (14) follows by Lemma 1, and the fact that the extreme values of a linear form on a polytope occur at extreme points of the polytope.

If a is a tautology, all valuations x make a true, hence a minimum for cx is attained by setting

$$(15)\qquad x(i_j) = \begin{cases} 0, & \text{if } c(i_j) \geq 0 \\ 1, & \text{if } c(i_j) < 0 \end{cases}$$

The value of this optimum (15) is $\Sigma_j \min \{ 0, c(i_j) \}$, and (3) holds.

If a is not a tautology, let x^o be a valuation which makes a false. Suppose that U contains a ball of radius $2\delta > 0$ about $(0,\ldots,0)$. Define c by

$$(16)\qquad c(i_j) = \begin{cases} -\delta, & \text{if } x^o(i_j) = 1 \\ \delta, & \text{if } x^o(i_j) = 0 \end{cases}$$

Then for any valuation x we have

$$cx \geq cx^{o} + k\,\delta \tag{17}$$

where k is the number of components in which x differs from x^{o}. Since any valuation x that makes $\mathcal{a}$ true must differ from x^{o} in at least one component, by (14) we have

$$\min \left\{ cx \;\middle|\; \begin{array}{l} \text{for at least one } h = 1,\ldots,t \\ \text{we have } A^{h}x \geq b^{h},\ x \geq 0 \end{array} \right\} \geq cx^{0} + \delta \tag{18}$$

$$= \left(\sum_{j=1} \min \{0, c(i_j)\} \right) + \max \{ |c_j| \mid j=1,\ldots,r \}.$$

From (18) and the fact that $|c_j| = \delta > 0$ for $j = 1,\ldots,r$ we see that (13) fails.

Q.E.D.

Lemma 3: The disjunctive program

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & A^{h}x \geq b^{h},\ x \geq 0 \text{ for some } h = 1,\ldots,t \end{array} \tag{19}$$

and the linear program

$$\begin{array}{ll} \text{minimize} & cx \\ \text{subject to} & A^{h}x^{h} - b^{h}x_{h} \geq 0\ ,\quad h = 1,\ldots,t \\ & \sum_{h=1}^{t} x_{h} = 1 \\ x & - \sum_{h=1}^{t} x_{h} = 0 \\ & x,\ x^{h},\ x_{h} \geq 0 \quad \text{for } h = 1,\ldots,t \end{array} \tag{20}$$

have the same optimal value for all vectors c.

Proof: The linear program (20) for the disjunctive program (19) is from [1], where it was shown that the values of (19) and (20) are equal when all systems $\{ x \geq 0 \mid A^h x \geq b^h \}$ are consistent. This is the case here, since no literal and its negation both occur in a^h .

Q.E.D.

Lemma 4: A three-tape Turing machine can convert the string a in (1) into the string for the constraints of (20) in running time $0(\ell h(a)^5)$, where $\ell h(a)$ denotes the length of a.

Proof: Let the input be on Tape 1.

Then on Tape 2 a complete list of the distinct variable indices $\{ i_1, \ldots, i_r \}$ can be produced in time $0(\ell h(a)^2)$ by scanning Tape 1.

A block Z of zeroes having length equal to $\ell h(i_r) + 2$ can be maintained on Tape 3, toward obtaining distinct indices of variables in the systems $A^h x^h - b^h x_h \geq 0$.

When a^h is next scanned and rewritten on Tape 1, all its variable indices are changed by concatenating to the right a block of zeroes that is (h - 1) copies of Z, as exhibited on Tape 3; hence this block never has length exceeding $(t - 1)\ell h(a) \leq \ell h(a)^2$. Each time a variable index is thus changed at most $\ell h(a)$ symbols must be moved a distance $\ell h(a)^2$, hence at most $0(\ell h(a)^3)$ is required each time. Since these changes are performed at most $\ell h(a)$ times, the re-indexing requires at most $0(\ell h(a)^4)$ time.

In addition, when a^h is scanned for the second time, the upper

bounds represented by $-x(i_1) \geq -1,\ldots,-x(i_r) \geq 1$, are to be added (properly indexed). Writing these bounds requires at most $0(\ell h(\mathcal{A})^3)$ time and space, with at most $\ell h(\mathcal{A})$ symbols to be moved $0(\ell h(\mathcal{A})^3)$ spaces at most $\ell h(\mathcal{A})$ times. Thus the overall effort here is at most $0(\ell h(\mathcal{A})^5)$.

Writing the last constraints of (20) requires at most $0(\ell h(\mathcal{A})^3)$ time.

Q.E.D.

Theorem 1: Fix integers $p,q > 0$.

Suppose that there is a Turing machine which, given an inequality system

$$Ay \geq b \tag{21}$$

together with a list J of indices of the variables y, determines, in time $f(\ell h(Ay \geq b, J))$, whether or not

$$\min \left\{ \sum_{j \in J} c_j y_j \mid Ay \geq b,\ y \geq 0 \right\} \leq \sum_{j \in J} \min \{0, c_j\} + \frac{1}{2} \max \{ |c_j| \mid j \in J \} \tag{22}$$

for all $c \in U$, where U is the open ball about the origin of radius p/q. Then the tautology problem is decidable in time $f(C\ell h(\mathcal{A})^5 + D)$, for some $C,D > 0$.

In particular, if the above determination can be accomplished in polynomial time, then P = NP.

Proof: By Lemma 4, in running time $0(\ell h(\mathcal{A})^5)$ the constraints (20) can be written. In time $0(\ell h(\mathcal{A})^2)$ the set J can be designated to be the indices of x in (20). By Lemmas 2 and 3, we see that (22) holds if and only if $\mathcal{A}$ is a tautology.

Q.E.D.

Since the set U in Theorem 1 can be only a neighborhood of the origin, there are two ways of construing Theorem 1. The first and most obvious one, is to take $U = R^r$ and view Theorem 1 as a result on parametric linear programming. But a second, viable alternative interpretation of Theorem 1, is as a result about linear programming with errors in the criterion function, for which we require to know whether a specified solution form is accurate to the first-order effects of these errors.

The same techniques used to establish Theorem 1 can provide similar results on projections of linear inequality systems.

For instance, it is not hard to show that the projection of the constraints (20) upon only the variables x is precisely $\text{conv}\left(\bigcup_{h=1}^{t} \{ x \geq 0 \mid A^h x \geq b^h \} \right)$. By Lemma 1, this projection is

$$\text{conv}\left(\bigcup_{h=1}^{t} \{ x \mid x \text{ is a valuation making } \mathcal{A}^h \text{ true} \} \right) = \text{conv}\{ x \mid x \text{ is a valuation making } \mathcal{A} \text{ true} \}$$

. Therefore this projection requires no other defining inequalities beyond

$$0 \leq x(i_j) \leq 1 , \quad j = 1,\ldots,r \tag{23}$$

if and only if $\mathcal{A}$ is a tautology.

Now it is easy to obtain the 2r inequalities (23) directly from (20), and therefore the question, as to whether or not the projection onto co-ordinates x involves any inequality beyond those of (23) (other than non-negative combinations of (23)), is at least as difficult as the tautology problem.

Incidentally, the Fourier-Dines-Motzkin elimination method can be used to project the system (20) onto co-ordinates x, by transferring the corresponding variables, as indeterminates, to the right-hand-side. Then the elimination method proceeds uniformly in the right-hand-side, with the indeterminates retained. It follows that the problem, of avoiding eliminations which will only create redundant information, is at least as difficult as the tautology problem.

We next show that ordinary linear programming, i.e., linear programming with exact data, is no more difficult than tasks in the polynomial class of the tautology problem.

Let CON denote the problem of determining, whether the linear inequality system

$$Ax = b,\ x \geq 0 \qquad (x = (x_1,\ldots,x_r)) \tag{24}$$

is consistent. As mentioned above, all data A,b in (24) are to be in binary integers, and different equalities are separated by commas and occur on one line. The matrix A is m by r, and b is m by 1.

Lemma 5: CON $\in$ NP

Proof: By [4, Theorem 1] it suffices to show that CON is accepted by a non-deterministic Turing machine which operates in polynomial time.

Consider the non-deterministic Turing machine $\mathcal{J}$ which proceeds by placing T ("tight") or S ("possibly slack") next to each variable x_j in (24), j = 1,...,r. Having made such arbitrary choices there are 2^r possible outcomes), $\mathcal{J}$ next proceeds deterministically.

If there are more than m S's, $\mathcal{J}$ rejects the string. If there are m S's or fewer, let the vector of x's which have 'S' placed next to them be designated y, and let the corresponding columns of A be designated B. Then $\mathcal{J}$ attempts to solve By = b, which has at least as many constraints as variables, using any of the algorithms which require only a cubic order of arithmetic operations and order comparisons to be performed. (Rational numbers which occur in these computations are stored as pairs of binary integers). Since addition and multiplication of two binary integers can be accomplished in polynomial time, these algorithms are implementable in polynomial time.

If there are an infinite number of solutions to By = b, $\mathcal{J}$ rejects the string. If there is no solution to By = b, $\mathcal{J}$ rejects the string. If there is a unique solution y^o to By = b found by the algorithm, and also $y^o \geq 0$, $\mathcal{J}$ acepts the string. Otherwise, $\mathcal{J}$ rejects the string.

From standard results on linear inequalities, if (24) is consistent, then it has a solution with more than m of the x's at a positive level, with a corresponding matrix B such that By = b has a unique solution. Therefore $\mathcal{J}$, which clearly operates in polynomial time, determines whether or not (24) is consistent.

Q.E.D.

Let LP denote the problem of computing an optimum to the linear program

$$\begin{aligned} &\min \quad cx \\ (25) \qquad &\text{Subject to} \quad Ax = b \\ &\qquad\qquad\quad x \geq 0 \qquad \Big(x = (x_1,\ldots,x_r)\Big). \end{aligned}$$

The optimum is to be presented as r pairs of binary integers (p,q), which represent rational numbers p/q that are the co-ordinates of the optimum. If (25) is inconsistent, the symbols "INC" are to be reported as a solution; if (25) is consistent but has no optimum, then "UBD" is to be reported.

Note that LP is a computation problem, not a problem of language acceptance.

Theorem 2: Suppose that there is a Turing machine which determines if systems of the type (24) are consistent, and it operates in time $f(\ell h(Ax = b,\ x \geq 0))$.

Then for a suitable polynomial p(v), there is a Turing machine which computes LP in time not exceeding

$$(26) \quad (r+2)\, f\Big(6\,\ell h(cx,\ Ax = b,\ x \geq 0)\Big) + p\Big(h(cx,\ Ax = b,\ x \geq 0)\Big).$$

In particular, if either CON or the tautology problem is decided in polynomial time, there is also a polynomial time algorithm for computing LP.

Conversely, if LP is computable in polynomial time, then CON $\in$ P.

Proof: We describe a Turing machine $\mathcal{J}$.

First, $\mathcal{J}$ decides if $Ax = b,\ x \geq 0$ is consistent in time $f(\ell h(Ax = b, x \geq 0)) \leq f(6\ell h(cx, Ax = b, x \geq 0))$. If it is not consistent, $\mathcal{J}$ reports "INC" and halts.

Otherwise, $\mathcal{J}$ determines if the dual constraints $\theta A \leq c$ are consistent in time $f\left(\ell h(\theta^{(1)} A - \theta^{(2)} A + \lambda = c,\ \theta^{(1)} \geq 0,\ \theta^{(2)} \geq 0,\ \lambda \geq 0)\right)$ $\leq f(6\ell h(cx, Ax = b, x \geq 0))$. If these are inconsistent, $\mathcal{J}$ reports "UBD" and halts.

Otherwise, $\mathcal{J}$ will proceed to determine the consistency of systems of the form

$$
\begin{aligned}
&\theta^{(1)} A - \theta^{(2)} A + \lambda = c \\
(27)_p \qquad &(\theta^{(1)} - \theta^{(2)})\, b - cx = 0 \\
&Ax = b \\
&x,\ \theta^{(1)},\ \theta^{(2)},\ \lambda \geq 0 \\
&x_j = 0, \qquad j \in S_p
\end{aligned}
$$

for certain sets S_p. Each one of these consistency issues are determined in time not exceeding $f(6\ell h(cx, Ax = b, x \geq 0))$. Note that $(27)_p$ is consistency if and only if

$$
\begin{aligned}
&\theta A \leq c, \\
(27)'_p \qquad &\theta b = cx \\
&Ax = b \\
&x \geq 0 \\
&x_j = 0, \qquad j \in S_p
\end{aligned}
$$

is consistent. By standard results, $(27)'_p$ holds if and only if there is an optimum to (25) with $x_j = 0$ for $j \in S_p$.

Initially, $S_1 = \{1\}$. When $(27)_p$ is consistent, we put $S_{p+1} = S_p \cup \{ p + 1 \}$ if $p \leq r$; when $(27)_p$ is inconsistent, we put $S_{p+1} (S_p \cup \{ p + 1 \}) \setminus \{p\}$ if $p < r$ and $S_{p+1} = S_p - \{p\}$ if $p = r$.

The systems $(27)_p$ are examined for consistency in order $p = 1,\ldots,r$, and this requires time not exceeding $rf(6 \ell h(cx, Ax = b, x \geq 0) + 3 \ell h(cx, Ax = b, x \geq 0) + D$ for some constant $D > 0$.

By construction, there is an optimal solution to (25) with $x_j = 0$ for all $j \in S_{r+1}$, but no such optimum with $x_k = 0$ for any $k \notin S_{r+1}$. By standard results, the complement of S_{r+1} has indices corresponding to a subvector y of x, possessing columns B in A which are linearly independent. Therefore By = b has a unique solution that is computed in polynomial time, and satisfies $y \geq 0$. In this manner, a solution to (25) is computed in time bounded by (26).

Clearly, if f is a polynomial, so is (26); and if either CON or the tautology problem is decided in polynomial time, f will be a polynomial. This gives the "particular" remark. For the "conversely" remark, we recall simply that Phase I of the Simplex Algorithm determines consistency [3].

Q.E.D.

Carnegie-Mellon University
July 15, 1976

References

[1] E. Balas, "Disjunctive Programming: Properties of the Convex Hull of Feasible Points," MSRR no. 348, Carnegie-Mellon University, July 1974.

[2] S. Cook, "The Complexity of Theorem-Proving Procedures," Conference Record of Third ACM Symposium on Theory of Computing, 1970, pp. 151-158.

[3] G. Hadley, Linear Programming, Addison-Wesley, 1962.

[4] R.M. Karp, "Reducibility Among Combinatorial Problems," Department of Operations Research, University of California at Berkeley, April 1972.

[5] V. Klee and G.J. Minty, "How Good is the Simplex Algorithm?," in Inequalities III, O. Shisha, ed., Academic Press, N.Y., 1971.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: M.S.R.R. 395	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): BRACKETING DISCRETE PROBLEMS BY TWO PROBLEMS OF LINEAR OPTIMIZATION.	5. TYPE OF REPORT & PERIOD COVERED: July, 1976
	6. PERFORMING ORG. REPORT NUMBER: M.S.R.R. 395
7. AUTHOR(s): Robert Jeroslow	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0621, NSF-GP-37510
9. PERFORMING ORGANIZATION NAME AND ADDRESS: GSIA, Carnegie-Mellon University, Pittsburgh PA	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 047-048
11. CONTROLLING OFFICE NAME AND ADDRESS: Personnel and Training Research Programs, Office of Naval Research (Code 458), Arlington, Virginia 22217	12. REPORT DATE: July 1976
	13. NUMBER OF PAGES: 14
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear programming, Integer programming, Computational complexity.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We prove two results: (1) If a certain restricted problem of first-order error analysis in linear programming (specified below), for errors in only the criterion function, has a polynomial-time algorithm, then so does the tautology problem; (2) If the tautology problem is decidable in polynomial time, then linear programs can be solved optimally in polynomial time.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403426 LB