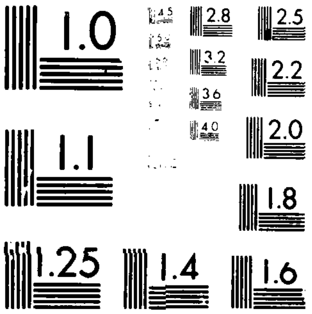
AD-A084 682   TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES   F/G 12/2
COMPLEXITY AND COMPUTABILITY OF SOLUTIONS TO LINEAR PROGRAMMING--ETC(U)
FEB 80   A CHARNES, W W COOPER, S DUFFUAA   N00014-75-C-0569
UNCLASSIFIED   CCS-367   NL

1 OF 1
AD
A084682

END
DATE
FILMED
6-80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

Research Report CCS 367

COMPLEXITY AND COMPUTABILITY OF SOLUTIONS TO LINEAR PROGRAMMING SYSTEMS

by

A. Charnes,* W.W. Cooper,**
S. Duffuaa,* M. Kress*

February 1980

*The University of Texas at Austin
**Harvard University Graduate School of Business Administration

This research was partly supported by Project NR047-071, ONR Contract N00014-80-C-0242, and Project NR047-021, ONR Contract N00014-75-C-0569, with the Center for Cybernetic Studies, The University of Texas at Austin. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

COMPLEXITY AND COMPUTABILITY OF SOLUTIONS TO LINEAR PROGRAMMING SYSTEMS

by

A. Charnes,* W.W. Cooper,**
S. Duffuaa,* M. Kress*

Abstract

Complexity, computability and solution of linear programming systems are re-examined in the light of Khachian's new notion of (approximate) solution. Algorithms, basic theorems and alternate representations are reviewed. It is shown that the Klee-Minty example has never been exponential for (exact) adjacent extreme point algorithms and that the Balinski-Gomory (exact) algorithm is polynomial where (approximate) ellipsoidal "centered-cut-off" algorithms (Levin, Shor, Khachian, Gacs-Lovasz) are exponential. Both the Klee-Minty and the new J. Clausen example are shown to be trivial (explicitly solvable) interval programming problems. A new notion of computable (approximate) solution is proposed together with an a priori regularization for linear programming systems. New polyhedral "constraint contraction" algorithms are proposed for approximate solution and the relevance of interval programming for good starts or exact solution is brought forth.

INTRODUCTION

The interest aroused by L.S. Khachian's announcement in the Soviet Doklady of "A Polynomial Algorithm for Linear Programming" and the misrepresentations (particularly by journalists) of Khachian's results, have induced the following re-examination of "informational complexity" and its relationship to computability and solution of linear programming systems. Contrary to journalistic pronouncements, Khachian's result is an existence theorem for determining consistency or inconsistency of a system of linear inequalities of a prescribed form. The algorithm and proof of its geometric convergence is due to Levin sixteen years earlier. Thus, Khachian's new contribution is an existence-theorem-backed notion of "approximate" solution.

As is made clear in the following section on linear programming's basic theorems and alternate representations, both complexity theory and computational effectiveness are vitally affected by the different geometries of "equivalent" linear programming systems. The classic Klee-Minty example for exponentiality of the simplex method (in their paper, "How Good is the Simplex Method?" [16]), is generalized and re-examined. It is shown that the dual constraint set for the generalization contains a single extreme point (which can be computed in n pivots from an artificial start) and, therefore, has never been exponential for "exact" solution by adjacent extreme point algorithms.

The classic Balinski-Gomory [2] polynomial bound algorithm for distribution (or so-called "transportation") models is shown to be polynomial where approximate solution by the ellipsoidal "centered-cut-off" algorithms of Levin, Shor, Khachian and Gacs-Lovasz is exponential. A new example

We wish to thank Drs. L. Seiford, J. Godfrey and A. Schinnar for supplying us with copies of the Russian and Gacs-Lovasz works herein cited.

by J. Clausen for simplex method exponentiality without the Klee-Minty dual side defect is examined. Both it and a generalized K-M example are shown to be instantly soluble as cases falling under Ben-Israel and Charnes' explicit solution of full row rank interval programming problems [3].

With these examples of complexity and solution in mind, attention is turned to computability of solutions. A new notion of a computable (approximate) solution is proposed, together with an a priori regularization for linear programming systems. To overcome the cumbersome storage and updating disadvantages cf the ellipsoidal algorithms as well as to provide an empirically more rapid method, two new classes of polyhedral "constraint-contraction" algorithms are proposed for approximate solution. For exact solution of problems with exponential numbers of extreme points, it is suggested that interval programming methods such as the Ben-Israel and Charnes solution for a good start and the "knapsack-pivot" algorithm of Charnes, Granot and Phillips [7] for completion of solution have been overlooked and merit serious computational study.

1. BASIC REPRESENTATIONS AND THEOREMS IN LINEAR PROGRAMMING

Solution of a linear programming problem implies securing both an optimal member of the primal constraint set ("optimal solution") and an optimal member of the dual problem constraint set. The purposes of analysis by a linear programming model are served only when, with a desired primal, the dual evaluators (optimal dual solution) are provided. Since all the major exact (extreme point) algorithm computer codes deliver an optimal dual pair, either of the dual problems may be chosen for"primal" computation. Experience has shown that there may be tremendous differences between the speed of computation from primal and dual problem sides.

When we speak of a "linear programming system," we refer to a system involving both primal and dual problems.

Re notation, we employ capital letters to denote matrices, lower case letters to denote column vectors (except that a capital letter with a subscript, e.g, P_o, may also denote a column vector). By A_j is meant the j^{th} column of A and ${}_iA$ is the i^{th} row of A. c^T is the transpose of the column vector c.

The major representations or formats for linear programming systems we shall designate as "Tucker's form" (T), the "equalities form" (E), and the interval programming form (I) as follows:

$$(T) \qquad \begin{array}{l} \max\ c^T x \\ \qquad Ax \le b \\ \qquad\ x \ge 0 \end{array} \qquad\qquad \begin{array}{l} \min\ w^T b \\ \qquad w^T A \ge c^T \\ \qquad\quad x \ge 0 \end{array}$$

$$(E) \qquad \begin{array}{l} \max\ c^T \lambda \\ \qquad P\ \lambda = P_o \\ \qquad\ \ \lambda \ge 0 \end{array} \qquad\qquad \begin{array}{l} \min\ u^T P_o \\ \qquad u^T P \ \ge\ c^T \end{array}$$

$$(I) \qquad \begin{array}{l} \max\ c^T x \\ \quad b^- \le Ax \le b^+ \end{array} \qquad\qquad \begin{array}{l} \min\ w^{+T} b^+ + w^{-T} b^- \\ \qquad (w^+ - w^-)^T A = c^T \\ \qquad\qquad w^+,\ w^- \ge\ 0 \end{array}$$

Note that the latter LP can be written equivalently as

$$\begin{array}{l} \min\ |u|^T \left(\frac{b^+ + b^-}{2}\right) + u^T \left(\frac{b^+ - b^-}{2}\right) \\ \qquad\qquad u^T A = c^T \\ \quad \text{where } |u|^T \equiv (\ldots, |u_i|, \ldots) \end{array}$$

Next, let $X = \{x : Ax \leq b,\ x \geq 0\}$, $W = \{w : w^T A \geq c^T,\ w \geq 0\}$,

$$\Lambda \equiv \{\lambda : P\lambda = P_o,\ \lambda \geq 0\},\ U \equiv \{u : u^T P \geq c^T\}$$

We shall require the following theorems.

LIEP Theorem (Charnes 1950): $\lambda \neq 0$ is an extreme point of Λ iff $\{P_j : \hat{\lambda}_j > 0\}$ is linearly independent.

Duality States Theorem (Charnes & Cooper 1951): There are four mutually exclusive and collectively exhaustive (MECE) duality states for (T):

(1) $X = \emptyset, \quad W = \emptyset$

(2) $X = \emptyset, \quad \inf_W w^T b = -\infty$

(3) $\sup_X c^T x = \infty,\ W = \emptyset$

(4) $\exists\ x^* \in X,\ w^* \in W \ni \max_X c^T x = c^T x^* = w^{*T} b = \min_W w^T b$ with x^*, w^* extreme points.

This theorem is valid, of course, for the other two linear programming forms on replacing X and W by the corresponding primal and dual constraint sets, e.g., by Λ and U for the "equalities" (E) format.

Equivalent Inequality System: (ξ^*, η^*) is a pair of optimal solutions to the dual problems in (T) iff it is a solution to the linear inequality system:

$$\text{(TD)} \qquad \begin{array}{rrl} -c^T\xi & +\ \eta^T b & \leq 0 \\ A\xi & & \leq b \\ -I\ \xi & & \leq 0 \\ & -\eta^T A & \leq -c^T \\ & -\eta^T I & \leq 0 \end{array}$$

Explicit Solution Theorem (Ben-Israel and Charnes 1967):

If (I) has a finite optimum and A is of full row rank, then the set of all optimal solutions of (I) is given by:

$$x^* = A^{\#} z^* \ , \quad A^{\#} \text{ a right inverse of } A$$

where

$$z^*_i = \begin{cases} b^+_i \ , \text{ if } \ (c^T A^{\#})_i > 0 \\ b^-_i \ , \text{ if } (c^T A^{\#})_i < 0 \\ \theta_i \, b_i^+ + (1- \theta_i) \, b_i^- \ , \ \forall \ 0 \leq \theta_i \leq 1, \ \text{ if } (c^T A^{\#})_i = 0 \end{cases}$$

That "equivalent" linear programming problems may have substantially different constraint geometries may be seen in the following favorite example of K. Kortanek's. The primal problem (1.1) is a simple 2-node directed network problem. Its constraint set, Λ , has one extreme point. An equivalent dual problem (1.2) is of interval programming form. Its constraint set U is an infinite strip in 2-dimensions, i.e., has zero extreme points. Putting it back into equivalent (E) form, (1.3), its new constraint set has five extreme points.

$$(1.1) \quad \begin{array}{ll} \min & 2\lambda_1 + \lambda_2 \\ \text{s.t.} & -\lambda_1 + \lambda_2 = -1 \\ & \lambda_1 - \lambda_2 = 1 \\ & \lambda_1 \ , \lambda_2 \geq 0 \end{array} \qquad \begin{array}{l} \max \ -u_1 + u_2 \\ -u_1 + u_2 \leq 2 \\ u_1 - u_2 \leq 1 \\ \\ \end{array}$$

$$(1.2) \quad \begin{array}{l} \max \ -u_1 + u_2 \\ -1 \leq -u_1 + u_2 \leq 2 \end{array}$$

$$\begin{array}{lllllllll}
 & \max & -v_1^+ + v_1^- + v_2^+ - v_2^- & & & \\
 & \text{s.t.} & -v_1^+ + v_1^- + v_2^+ - v_2^- & + s_1 & & = 2 \\
(1.3) & & v_1^+ - v_1^- - v_2^+ + v_2^- & & + s_2 & = 1 \\
 & & v_j^+ , v_j^- , s_j \geq 0, \ j = 1,2 & & &
\end{array}$$

The non-equivalence geometrically of different equivalent representations of dual constraint sets was first pointed out and studied by Charnes, Cooper and Thompson in [11]. Currently, U. Eckhardt's work has yielded the most significant results. It and references to the other major work are to be found in [12].

2. KHACHIAN'S SOLUTION AND LEVIN'S ELLIPSOIDAL BISECTION METHOD

Consider the system of $m \geq 2$ linear inequalities in $n \geq 2$ real variables

$$(2.1) \qquad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \ , \quad i=1,\ldots,m$$

or, compactly,

$$(2.2) \qquad a_i^T x \leq b_i$$

When the a_{ij} and b_i are integers, the number of binary digits required to store this data is

$$(2.3) \qquad L \equiv 1 + \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} \log_2 (|a_{ij}| + 1) + \sum_{i=1}^{m} \log_2 (|b_i| + 1) + \log_2 (mn) \right\}$$

We shall call the convex non-differentiable function

$$(2.4) \qquad \theta(x) \equiv \max_i \theta_i(x) \ , \text{ where } \theta_i(x) \equiv a_i^T x - b_i \ ,$$

the "residual" at the point x of the system (2.1). Note that $\theta(x) \leq 0$ for some x iff the system is consistent. Alternately, x is a member of the constraint set iff it is a zero of the convex function

$$(2.5) \qquad \tilde{\theta}(x) \equiv \max [0, \theta(x)]$$

Thereby, finding a solution to the system (2.1) is equivalent to the convex programming problem of minimizing $\theta(x)$ as well as to finding a zero of $\tilde{\theta}(x)$.

N.Z. Shor of The Institute of Cybernetics of The Ukrainian Academy of Sciences, whose published research on algorithms for such problems, for instance, [19], [20], [21], [22], [23], extends continuously from earlier than 1968 to the present, is cited in Khachian's 1979 Doklady announcement [15] only for Shor's 1970 paper as the source of Khachian's idea for an algorithm. However, Shor's 1977 paper plainly states that the idea of and first proof of geometric convergence of a class of algorithms which includes Khachian's is due to A.Y. Levin in the 1965 Doklady volume [17].

Levin's idea is to start with the center of an ellipsoid which contains at least one solution to (2.1) if consistent. If this center is not a solution, the ellipsoid is bisected by the most violated hyperplane and the half-ellipsoid which contains a solution is enveloped by a new ellipsoid of minimal volume. The center of this new ellipsoid is taken as a new start and the process is repeated. This continues until the volume of the attained ellipsoid is as small as desired. Shor calls such algorithms "(ellipsoidal) centered cut-off" algorithms.

Khachian's existence theorem is implicit in the following two lemmas plus geometric convergence of the algorithm.

<u>Lemma 1</u>: If the system (2.1) with input L is consistent, then there exists a solution x^o in the Euclidean ball

$$S \equiv \{ x : ||x|| \leq 2^L \} .$$

<u>Lemma 2</u>: If the system (2.1) with input L is inconsistent, then for any $x \varepsilon R^n$ the residual $\theta(x) \geq 2 \cdot 2^{-L}$.

In $16n^2L$ steps Khachian's algorithm delivers an ellipsoid center with minimum algorithm residual $\leq 2^{-L}$ if the system is consistent. If not, the residual remains greater than $2 \cdot 2^{-L}$. The final center defines what we call Khachian's (new notion of) solution.

Since Khachian's announcement contained no proofs, Gacs and Lovasz in [13] established Lemma 1 using the LIEP theorem and determinantal bounds in Cramer's rule solutions of linear equations. They established the other properties but for the following new ellipsoidal algorithm which requires only $6n^2L$ steps and no Gram-Schmidt orthogonalization at each step.

Gacs and Lovasz define a sequence $x^o, x^1, \ldots \varepsilon R^n$ and a sequence of symmetric positive definite matrices $A^o, A^1, \ldots$ recursively as follows $x^o = 0$, $A^o = 2^L I$. Suppose that (x^k, A^k) is defined. If x^k is a solution of (2.1), stop. If not, pick any inequality in (2.1) which is violated, say,

$$a_i^T x^k > b_i$$

and set

$$x^{k+1} = x^k + \frac{1}{n+1} \frac{A^k a_i}{\sqrt{a_i^T A^k a_i}}$$

(2.6)

$$A^{k+1} = \frac{n^2}{n^2-1} \left\{ A^i - \frac{2}{n+1} \frac{(A^k a_i) \cdot (A^k a_i)^T}{a_i^T A^k a_i} \right\}$$

Note that $(A^k a_i) \cdot (A^k a_i)^T$ is a symmetric nxn matrix as the matrix product of an nx1 vector and its 1xn transpose. Gacs and Lovasz note further that all computations, e.g. square root, are supposed performed to an accuracy of exp(-10nL).

For linear programming the x vector is the (ξ, η) pair of dual problem variables in the system (T). We exhibit the simplex method and Lemke's Dual Method schematically (Figure 1) in (ξ,η) space by projection from their (E) form variables. When consistent, i.e., case (iv) of the Duality States Theorem, the constraint set consists with probability one of a single point, i.e., the dual LP problems which have a unique optimal pair of solutions form an everywhere dense open set in the space of consistent data.

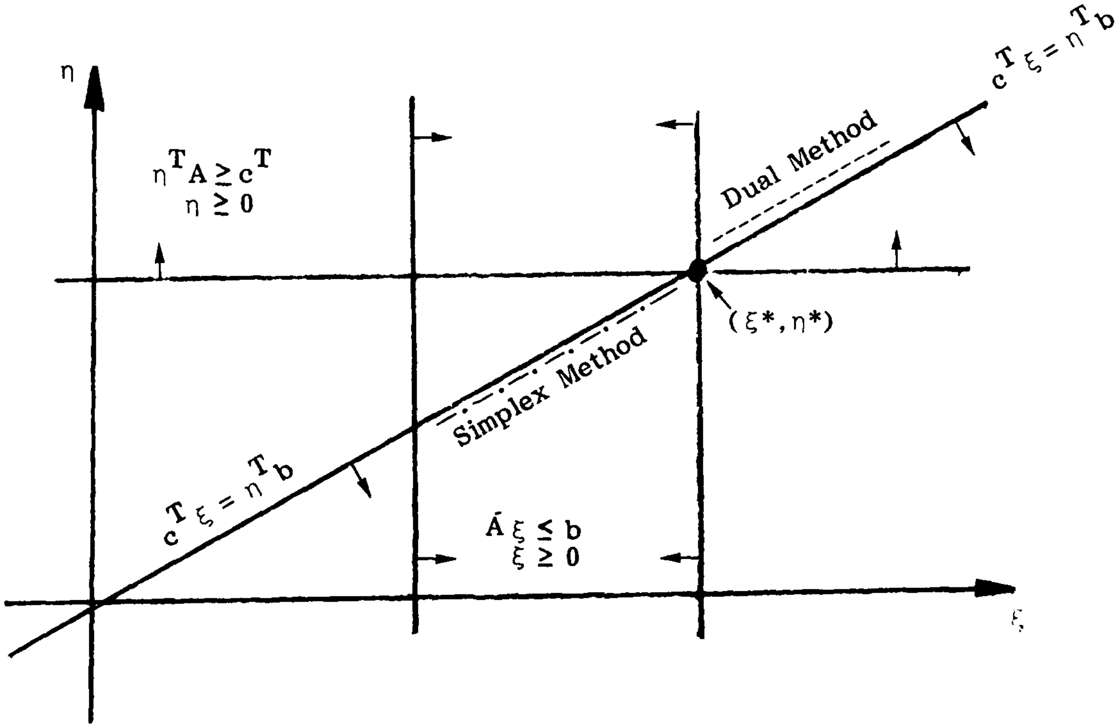


Figure 1

3. COMPLEXITY OF A GENERALIZED KLEE-MINTY[1] EXAMPLE

We do not accept the re-definition of complexity (or "informational complexity" as the Soviets call it) which Khachian implicitly makes in [15] as the function of "L" required to bound the number of iterations of an algorithm. Most inequality systems or linear programming problems are and have been given in terms of coefficients which are not necessarily integers. Further, all the classic results on complexity, e.g., Klee and Minty [16] or Balinski and Gomory [2], are given "in Jack Edmond's sense" as "the number of pivots or iterations required...(as a)...function of the two parameters...(number of equations plus inequalities and number of variables)...that specify the size of the program" (see p. 159 of Klee and Minty). We shall use Edmond's sense.

We generalize the Klee-Minty example by replacing their "ε" by arbitrary "ε_i's" where $0<\varepsilon_i<1/2$, all i. Their problem is then:

$$\max \quad x_n$$

s.t.

$$
\begin{array}{lr}
-x_1 & \leqslant 0 \\
x_1 & \leqslant 1 \\
\varepsilon_1 x_1 - x_2 & \leqslant 0 \\
\varepsilon_1 x_1 + x_2 & \leqslant 1 \\
= \; = \; = \; = \; = \; = \; = \; = \; = \; = & \\
\varepsilon_{n-1} x_{n-1} - x_n & \leqslant 0 \\
\varepsilon_{n-1} x_{n-1} + x_n & \leqslant 1
\end{array}
\tag{3.1}
$$

Its (unstudied) dual is:

[1] We wish to thank George Minty for his friendly help in providing a copy of the published paper.

$$
\begin{array}{lr}
\min & w_1^+ \qquad + w_2^+ \qquad + w_3^+ \qquad\qquad\qquad + w_n^+ \\
\text{s.t.} & \\
& -w_1^- + w_1^+ + \varepsilon_1 w_2^- + \varepsilon_1 w_2^+ \qquad\qquad\qquad = 0 \\
& - w_2^- + w_2^+ + \varepsilon_2 w_2^- + \varepsilon_2 w_2^+ \qquad\qquad = 0 \\
& \text{--- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---} \\
(3.2) & \text{--- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---} \\
& -w_{n-1}^- + w_{n-1}^+ + \varepsilon_{n-1} w_n^- + \varepsilon_{n-1} w_n^+ = 0 \\
& - w_n^- + w_n^+ = 1 \\
& w_j^-, w_j^+ \geq 0 , \quad j = 1, \ldots n
\end{array}
$$

We choose to analyze and solve this problem from the dual side with the modified simplex method of Charnes and Lemke (1952 [10], [5]). Notice from the last equation that every member of the dual constraint set must have $w_n^+ > 0$ (e.g., $w_n^+ \geq 1$). Therefore, it must have $w_{n-1}^- > 0$, and successively $w_{n-2}^-, w_{n-3}^-, \ldots, w_1^- > 0$. But the coefficient vectors of $w_1^-, \ldots, w_{n-1}^-, w_n^+$ form a linearly independent set which, further, is maximal since there are n of them. By the LIEP Theorem we have an extreme point and by maximality (in *every* solution) it is the only one. It is *the* optimum solution to (3.2) since we are in case (iv) of the Duality States Theorem.

By the theory of the modified simplex method the optimal primal solution is $x^{*T} = d^T B^{-1}$, where B is the matrix of coefficient vectors of $w_1^-, \ldots, w_{n-1}^-, w_n^+$ and d^T is the vector of their coefficients in the functional. Here $x_1^* = x_2^* = \ldots = x_{n-1}^* = 0$, $x_n^* = 1$. Thus, we have established

Theorem 1: The dual constraint set of the generalized Klee-Minty problem is a cone issuing from a single extreme point. The optimal primal solution is available by inverting the matrix of the basis designated by this extreme point.

The number of operations to invert B and obtain x^* is clearly $0(n^3)$. Thereby, if we are permitted knowledge of the dual extreme point, we have shown polynomiality of the simplex method for this example class by solving the dual problem.

If we are required to be ignorant of it, we next show that in n steps from a standard artificial basis, we achieve this extreme point, and automatically, of course, the corresponding primal optimum.

In no way does our result invalidate the soundness of the idea behind Klee and Minty's construction of a tilted, perturbed hypercube, which, as the intersection of 2n half-spaces, has 2^n extreme points. Thereby, if one must start "at the bottom," an adjacent extreme point algorithm on the primal side must take $0(2^n)$ steps.

To get on with our simplex computation, we consider the dual "regularized" problem ([4], [5]) in the form:

$$\begin{array}{lll} & \min & M^2\lambda_o + Me^T\nu + c^T\lambda \\ (3.4) & \text{s.t.} & P_o\lambda_o + E\nu + P\lambda = P_o \\ & & \lambda_o, \nu, \lambda \geq 0 \end{array}$$

Here the λ_j, $j \neq 0$, are the dual variables w_k^-, w_k^+ of (3.3), c^T the vector of functional coefficients, P the structural matrix, P_o its right-hand side vector and E the nx(n-1) matrix of the first n-1 unit (artificial) vectors, M is the "high" or non-Archimedean penalty "cost." Oddly, here, P_o is the n^{th} unit vector.

Starting with the artificial basis of P_o and the n-1 unit vectors of E, $V_1, \ldots, V_{n-1}$, whose functional coefficients are respectively M^2, M, ..., M, and using the basis entry criterion of maximum "$z_j - c_j$", the initial (abbreviated) simplex tableau is:

$\vec{c}$ / Basis	M^2	0	0	0		0	0	0	1	1	1		1	1	1
	P_o	P_1^-	P_2^-	P_3^-		P_{n-2}^-	P_{n-1}^-	P_n^-	P_1^+	P_2^+	P_3^+		P_{n-2}^+	P_{n-1}^+	P_n^+
V_1	0	-1	ε_1						1	ε_1					
V_2	0		-1	ε_2						1	ε_2				
				-1							1				
V_{n-3}						ε_{n-3}							ε_{n-3}		
V_{n-2}	0					-1	ε_{n-2}						1	ε_{n-2}	
V_{n-1}	0						-1	ε_{n-1}						1	ε_{n-1}
P_o	1							-1							1
$\vec{z} - \vec{c}$	0	-M	$-M\alpha_1$	$-M\alpha_2$		$-M\alpha_{n-3}$	$-M\alpha_{n-2}$	$-M^2+\varepsilon_{n-1}M$	M	$M\alpha_1$	$M\alpha_2$		$M\alpha_{n-3}$	$M\alpha_{n-2}$	$M^2+\varepsilon_{n-1}M$

where $\alpha_j = 1+\varepsilon_j$

Figure 2

Here P_j^-, P_j^+ are the respective column vectors of the variables w_j^-, w_j^+, and we only show the column vector for "λ_o", e.g., P_o in the tableau.

Only w_n^-, w_n^+ have M^2 in their "z_j-c_j", and since

(3.5)
$$\begin{aligned} P_n^+ &= P_o + c_{n-1} V_{n-1} \\ P_n^- &= -P_o + c_{n-1} V_{n-1} \end{aligned}$$

we have

(3.6)
$$\begin{aligned} z(w_n^+) - c_n^+ &= M^2 + 0(M) \\ z(w_n^-) - c_n^- &= -M^2 + 0(M) \end{aligned}$$

Thus, P_n^+ enters the basis. V_{n-1} leaves since $0/\varepsilon_{n-1} < 1/1$ (corresponding to V_{n-1} and P_o).

Since $\varepsilon_{n-1} V_{n-1} = P_n^+ - P_o$, $P_n^- = -2P_o P_n^+$. Thus while P_o, P_n^+ are in the basis

(3.7) $z(w_n^-) - c_n^- = -2M^2 + 0(M)$

and P_n^- will not be a candidate to enter.

Next, the only "M^2" entry possibilities are

(3.8)
$$\begin{aligned} P_{n-1}^+ &= \varepsilon_{n-2} V_{n-2} + \varepsilon_{n-1}^{-1} P_n^+ - \varepsilon_{n-1}^{-1} P_o \\ P_{n-1}^- &= \varepsilon_{n-2} V_{n-2} - \varepsilon_{n-1}^{-1} P_n^+ + \varepsilon_{n-1}^{-1} P_o \end{aligned}$$

Thus, P_{n-1}^- enters and V_{n-2} leaves the basis.

Thereafter, while P_{n-1}^-, P_n^+, P_o are in the basis

(3.9)
$$\begin{aligned} P_{n-1}^+ &= P_{n-1}^- + 2\,\varepsilon_{n-1}^{-1} P_n^+ - 2\,\varepsilon_{n-1}^{-1} P_o \\ z(w_{n-1}^+) - c_{n-1}^+ &= -2\,\varepsilon_{n-1}^{-1} M^2 + 0(M) \end{aligned}$$

and P_{n-1}^+ will not enter the basis.

By induction, for $1 \leq k \leq n-2$, when $P^-_{n-k}, \ldots, P^-_{n-1}, P^+_n, P_o$ are in the basis, P^-_{n-k-1} enters and V_{n-k-1} leaves. Further, P^+_{n-k-1} will not enter while $P^-_{n-k-1}, \ldots, P^-_{n-1}, P^+_n, P_o$ are in the basis.

For $k = n-2$, V_1 is removed and $P^-_2, \ldots, P^-_{n-1}, P^+_n, P_o$ is the basis. Now

$$P^-_1 = -\varepsilon_1^{-1} P^-_2 - \ldots - (\varepsilon_1\varepsilon_2\ldots\varepsilon_{n-2})^{-1} P^-_{n-1} - (\varepsilon_1\ldots\varepsilon_{n-1})^{-1} P^+_n + (\varepsilon_1\ldots\varepsilon_{n-1})^{-1} P_o$$

(3.10)

$$P^+_1 = \varepsilon_1^{-1} P^-_2 + \ldots + (\varepsilon_1\ldots\varepsilon_{n-1})^{-1} P^{-1}_{n-1} + (\varepsilon_1\ldots\varepsilon_{n-1})^{-1} P^+_n - (\varepsilon_1\ldots\varepsilon_{n-1})^{-1} P_o$$

Thus, P^-_1 enters and P_o leaves. In n pivots the unique optimal extreme point has been attained together with (via the modified simplex machinery) the optimal x^* as the dual evaluators for this extreme point.

We have established

Theorem 2: The generalized Klee-Minty problem is computable in n steps by the (modified) simplex method applied to the dual problem from a standard artificial basis start.

Thereby, the Klee-Minty problem has never been an example for exponentiality of the simplex method.

As for approximate or Khachian solution, if we take $c^+_n = 2^{2^n}$, $L = 0(2^n)$ and the ellipsoidal algorithms are exponential.

Our above argument is still valid, however, since only order comparisons, not exact values, are needed for the "$z_j - c_j$" in the simplex (or dual) algorithms. Further, Charnes and Cooper have shown explicitly in [5] how simplex or dual method calculations can be made using even non-Archimedean "M"'s by extending the simplex tableau by two rows containing respectively the coefficients of M^2 and M and never requiring numerical values for M^2 or M in calculation.

The classic result of Balinski and Gomory [2] for polynomiality (in exact solution) of an extreme point method for the important class of distribution models in linear programming,[2] is that via their "mutual primal-dual algorithm" only$\left(\sum_i a_i\right)\left(\min\ \{m,n\}\right)$steps are required to solve:

$$(3.11)\quad \begin{aligned} &\min \sum_{i,j} c_{ij}\,x_{ij} \\ &\text{s.t.} \sum_j x_{ij} = a_i\ ,\ i=1,\ldots,m \\ &\qquad \sum_i x_{ij} = b_j\ ,\ j=1,\ldots,n \end{aligned}$$

$$\text{where}\qquad x_{ij} \geq 0$$

and the a_i ,b_j are integers.

Since, as before, only order comparisons are needed, not numerical values, the Balinski-Gomory method is polynomial where the Khachian approach is exponential in examples of (3.11) with some $c_{ij} = 0(2^{2^n})$. It should be noted, however, that the Balinski-Gomory algorithm requires a start with a particular kind of primal and dual variable choice.

4. INTERVAL PROGRAMMING, KLEE-MINTY AND J. CLAUSEN EXAMPLES

At a recent meeting in Denmark, Dr. Jens Clausen of the Datalogisk Institute presented a new example for exponentiality of the simplex method which we observe to be free of the dual side defect of the Klee and Minty type. We shall show that both it and the generalized

[2] We wish to thank Ralph Gomory for his friendly help in locating their paper.

Klee-Minty example are instantly soluble by the Ben-Israel and Charnes theorem in interval programming, i.e., only one matrix inverse need be computed.

Consider first the Klee-Minty type. It may be written in <u>pseudo</u> interval programming form as:

$$\min \quad x_n$$

$$(4.1) \qquad \begin{array}{rcll} 0 & \le & x_1 & \le 1 \\ z\varepsilon_1 x_1 & \le & \varepsilon_1 x_1 + x_2 & \le 1 \\ \text{— —} & & \text{— — — — — — — — —} & \text{—} \\ \text{— —} & & \text{— — — — — — — — —} & \text{—} \\ 2\varepsilon_{n-1} x_{n-1} & \le & \varepsilon_{n-1} x_{n-1} + x_n & \le 1 \end{array}$$

Next we replace the left-side vector by any consistent final vector b^-, e.g., $b_i^-=0$, $i=1,\ldots,n$, so that we have an interval programming problem.

The point of this is that, referring to the Ben-Israel and Charnes theorem, only the <u>signs</u> of the $c^T A_i^{-1}$ and <u>not</u> the <u>values</u> of b_i^- and b_i^+ determine optimal solution. Here

$$(4.2) \qquad A = \begin{bmatrix} 1 & & & \bigcirc \\ \varepsilon_1 & 1 & & \\ & \ddots & \ddots & \\ \bigcirc & & \varepsilon_{n-1} & 1 \end{bmatrix}$$

and we can write $A^{-1} = E^{n-1} E^{n-2} \ldots E^1$ where E^k is the elementary matrix which adds $-\varepsilon_k$ times the k^{th} row of A to the $(k+1)^{th}$ row of A. From the form of A, E^k affects only the k^{th} column of A.

We can now easily calculate $c^T A^{-1}$. Since $c^T = (0,\ldots,0,1)$,

$$(4.3) \qquad \begin{array}{rl} c^T E^{n-1} & = (0,\ldots 0,-\varepsilon_{n-1},1) \\ c^T E^{n-1} E^{n-2} & = (0,\ldots 0,-\varepsilon_{n-2},-\varepsilon_{n-1},1) \\ \text{— — — — —} & \text{— — — — —} \\ c^T E^{n-1} \ldots . E^1 & = (-\varepsilon_1,-\varepsilon_2,\ldots.,-\varepsilon_{n-1},1) \end{array}$$

Therefore, $z_i^* = b_i^- = 0$, $i=1,\dots,n-1$, $z_n^* = b_n^+ = 1$. From $x^* = A^{-1}z^*$ we get $x_i^+ = 0$, $i=1,\dots,n-1$, $x_n^* = 1$, which already satisfies (4.1). It is optimal for (4.1) since $b_i^- = 0$ yields a less restricted constraint set than in (4.1).

Let us next consider Dr. Jens Clausen's example which was presented at the Danish-Polish Operations Research Societies meeting at Rolighed, Denmark in May 1979:

$$
\begin{array}{lll}
 & \max & \displaystyle\sum_{j=1}^{n} c_j x_j \\
(4.4) & \text{s.t.} & \displaystyle\sum_{j=1}^{n} a_{ij} x_j \le 5^{i-1},\ i=1,\dots n \\
 & & x_j \ge 0
\end{array}
$$

where, dropping a factor of $4^{-n}\cdot 5$, $c_j = (4/5)^j$, and $A \equiv (a_{ij})$ is a lower triangular matrix with all $a_{ii} = 1$ and $a_{ij} = 2\cdot(5/4)^{i-j}$ for $i > j$. *Schematically*,

$$
\begin{array}{llllllll}
\max & \left(\frac{4}{5}\right)x_1 + & \left(\frac{4}{5}\right)^2 x_2 + & \left(\frac{4}{5}\right)^3 x_3 + & \cdots\cdots & & +\left(\frac{4}{5}\right)^n x_n & \\
\text{s.t.} & & & & & & & \\
 & x_1 & & & & & & \leqslant 5^0 \\
 & 2\left(\frac{5}{4}\right)x_1 + & x_2 & & & & & \leqslant 5^1 \\
(4.5) & 2\left(\frac{5}{4}\right)^2 x_1 + & 2\left(\frac{5}{4}\right)x_2 + & x_3 & & & & \leqslant 5^2 \\
 & \vdots & \vdots & \ddots & \ddots & & & \vdots \\
 & 2\left(\frac{5}{4}\right)^{n-1} x_1 + & 2\left(\frac{5}{4}\right)^{n-2} x_2 + & 2\left(\frac{5}{4}\right)^{n-3} x_3 + & \cdots & + 2\left(\frac{5}{4}\right) x_{n-1} & + x_n & \leqslant 5^{n-1}
\end{array}
$$

with $x_1,\dots,x_n \geqslant 0$

Since $A \geq 0$ and $x \geqslant 0$, we can choose to "have" $b_i^- = \sum_{j=1}^{i-1} a_{ij}x_j$ in an interval programming "envelope." Thereby we would have $x_i^* = 0$ whenever $c^T A_i^{-1} < 0$. We next ignore the $x \geqslant 0$ conditions to consider the less restricted interval programming problem:

$$(4.6) \qquad \begin{array}{ll} \max & c^T x \\ & b^- \leqslant Ax \leqslant b^+ \end{array}$$

where b^- is implicitly designated as above and $b_i^+ = 5^{i-1}$, $i = 1, \dots, n$

Since A is lower triangular, again E^k differs from the identity matrix only in the k^{th} column and we can apply the same product construction as before but with different entries in the E^k to obtain $c^T A^{-1}$. Thus, defining $c^T(1) \equiv c^T E^{n-1}$, $c^T(2) \equiv c^T(1)E^{n-2}, \dots, c^T(n-1) \equiv c^T(n-2)E^1$,

we have $\qquad c^T A_i^{-1} = (c_1, \dots, c_{i-1}, c_i(i-1), \dots, c_{n-1}(1), c_n)$.

Thereby, $\qquad c^T(1) = (c_1, \dots, c_{n-2}, c_{n-1} - a_{n,n-1}\, c_n, c_n)$

$$c^T(2) = (c_1, \dots, c_{n-3}, c_{n-2} - a_{n-1,n-2}\, c_{n-1}(1), c_{n-1}(1), c_n)$$

etc.

For Clausen's example we obtain

$$(4.7) \qquad \begin{array}{lcl} c^T A_n^{-1} & = c_n & = \left(\frac{4}{5}\right)^n \\ c^T A_{n-1}^{-1} & = c_{n-1}(1) & = -\left(\frac{4}{5}\right)^{n-1} \\ c^T A_{n-2}^{-1} & = c_{n-2}(2) & = \left(\frac{4}{5}\right)^{n-2} \end{array}$$

and

$$c^T A_i^{-1} = c_i(n-i) = (-1)^{n-i}\left(\frac{4}{5}\right)^i, \quad i = 1, \dots, n$$

Thus, $z_n^* = b_n^+$, $z_{n-1}^* = b_{n-1}^-$, $z_{n-2}^* = b_{n-2}^+$, etc., i.e., we alternate b_i^+ and b_i^- starting from b_n^+ and ending with b_1^+ or b_1^- according as n is odd or even. Since $x_i^* = 0$ whenever $z_i^* = b_i^-$, the remaining components of x^*, say $\bar{x}*$, are obtained as

$$\bar{x}* = D^{-1}d$$

where D is A with those i^{th} rows and columns removed for which $x_i^* = 0$ and d is the similar contraction of the b^+ vector.

The new device we have employed above to obtain an approximating (class of) interval programming problem(s) for which we have instant solution of the Minty-Klee and Clausen examples is really a method for obtaining a "good" start in an interval programming approach to solution of linear programming problems. Such a start would be the first step in using an interval programming algorithm like the "knapsack pivot" algorithm of Charnes, F. Granot and F. Phillips [7] or the SUBOPT algorithm of Ben-Israel and Robers [18], or the "primal" algorithm of Charnes, D. Granot and F. Granot [6]. We shall develop this new method in other publications.

5. COMPLEXITY, POLYNOMIALITY AND COMPUTABILITY

What is the relationship of complexity and polynomiality to computability of solutions? The Balinski and Gomory polynomial algorithm for distribution problems was never used; its bound on number of iterations was two orders of magnitude worse than common computational experience on real problems. Their 12-iteration example [2] was solved immediately by the VAM "start" (see p. 58 [5]).

An essential requirement in complexity proofs is the capacity to argue by mathematical induction. To achieve this, Balinski and Gomory discovered a separability characteristic in the distribution model constraint set. They were unable to achieve polynomiality for their "mutual primal-dual" algorithm for general linear programming because of the lack of separability, hence inductibility. Complexity research thus focuses attention only on algorithmic forms which are "inductible."

In the literature there seems to be a common misapprehension that complexity (and polynomiality) is an efficient research directive to better methods of solving "worst" case problems. Yet, as A. Vazsonyi observes, for over a century the most efficient method of computation of various functions has been through *non*-convergent (hence *a priori* "hyper-exponentially complex") asymptotic series. Of course, one can then *change the definition* of solution, just as some authors have redefined convergence to include non-convergence.

Again, Gomory's brilliant finitely convergent integer programming algorithms frequently converge too slowly for practical use. Often, the evidently exponential "branch and bound" method is employed effectively. Focusing on polynomiality, R. Jeroslow [14] published two exponential examples for branch and bound. No information regarding structure (e.g., asymmetries or branch exclusions) such as is accessible where "BAB" is effectively used is provided for these examples. Hence, we think, no computer specialist would ever contemplate use of BAB for such problems. Incidentally, both examples (for all n) require at most 2 iterations of the "Page-Cut" interval programming based algorithm of Armstrong, Charnes and Phillips [1].

Typically, complexity research ignores important characteristics of real computation. Rather than "worst" case solution, we would argue that it struggles primarily with *"imposed problem ignorance."* Khachian deals with Turing machines, not computers. Actual operations such as storage and retrieval, list processing, Gram-Schmidt orthogonalizations, updating and control of round-off error are dismissed as inconsequential since they are polynomial in complexity.

But polynomial algorithms such as Khachian's can be totally impractical for computation of realistically sized problems. Even for small examples, N.Z. Shor's every other paper mentions the slowness of convergence to approximate solutions. The world largest routinely solved linear programming problem (on UNIVAC 1108's by the U.S. Treasury Department's Computer Group) involves 50,000 constraints (excluding non-negativity) in 62.5 million variables. This distribution (assignment) model is solved with software developed by Klingman, Glover, Stutz, Karney, Barr and students [8] over nearly ten years of systematic experimentation and processing innovation exploring the whole spectrum of network model algorithms. The algorithm is the simplex algorithm in [5] provided with a good start (but worse than VAM, which, though polynomial, takes too much time).

Re computability, clearly, if one can get a good start for an adjacent extreme point method, it doesn't matter how many extreme points there are in the constraint set. Since most real models are compounds of already identified types of model structures, one should use such knowledge to develop classes of good starts via simple approximating structures. One might try approximate solution methods, too, when the observed structures are not so transparent. Then, if exact solution is required, one could "purify" the approximate solutions to extreme point solutions by methods such as the "Purification Algorithm" of Charnes, Kortanek and Raike [9], [9a].

Our previous demonstrations have indicated the power of interval programming methods for obtaining good starts as well as their promise for efficient solution of types of problems which do not fit network structures very well. Since interval programming methods like the "knapsack pivot" algorithm largely employ the highly developed computer processing software modules of the simplex method, we suggest computational research in interval programming is likely to be highly productive.

6. CONSTRAINT CONTRACTION ALGORITHMS

Since the work of Charnes, Cooper and Drews over twenty years ago (see Chapter XVIII § 6 of [5], especially pp. 693-695), little research has been done on computation of LP problems via the inequality system (TD). Their research was aborted by Drews' leave from ESSO to complete his Ph.D. at Berkeley. The current work, however, features ellipsoidal constraint set envelopments. For, Shor and associates have been trying to develop algorithms for convex non-differentiable programs which employ analytic function approximants, of which the simplest are quadratic. Ellipsoids, however, furnish bad approximations to polyhedral constraint sets in n dimensions. This may substantially increase computation time. For, although the constraint sets here are mostly a single (unknown) point, the bisection process involves at each step closest envelopment of a partly polyhedral set.

Our idea is to start with the "smallest" homothetic expansion of the constraint set which contains our initial starting point. "Balls" of size 2^L are <u>not</u> usually required; only $\max_i |b_i|$. Thereafter, each step concludes with a homothetic contraction and a new starting point. We present two types of algorithms, exterior (reflection) algorithms and interior (centering) algorithms. In both we stop when a sufficiently close "solution" (and constraint approximation) has been reached.

To give some idea of relative speeds, the following examples of inequality systems (not even LP) have been computed (by hand).

Consider the system

$$(6.1)\qquad \begin{aligned} 2^2 - 1 \le x_1 \le 2^3 - 1 \\ 2^2 - 1 \le x_2 \le 2^4 - 1 \end{aligned}$$

or, equivalently,

$$(6.2)\qquad \begin{aligned} x_1 \quad &\le 2^3 - 1 \\ -x_1 \quad &\le -2^2 + 1 \\ x_2 &\le 2^4 - 1 \\ -x_2 &\le -2^3 + 1 \end{aligned}$$

Here L = 19. Khachian's algorithm requires more than 40 iterations, Gacs-Lovasz more than 18, whereas our "curtailed reflection" algorithm requires only 2, from the same start of (0,0).

Replace this system by the inconsistent one

$$(6.3)\qquad \begin{aligned} x_1 \quad &\le 2^2 - 1 \\ -x_1 \quad &\le -2^3 + 1 \\ x_2 &\le 2^4 - 1 \\ -x_2 &\le -2^3 + 1 \end{aligned}$$

Then Khachian requires 1146 and Gacs-Lovasz 456 iterations to establish inconsistency.

Evidently, the ellipsoidal algorithms would greatly benefit if they could always deal with an equivalent consistent system. We, therefore, present the following "regularization" of the general linear programming system (T) according to the precepts and results of Charnes (1952) in [4].

$$\begin{array}{llll} & \max\ c^T\xi - M\xi_o & & \min\ \eta^T b + \eta_o U \\ \text{s.t.} & A\xi + b\xi_o \le b & & \eta^T A + \eta_o e^T \ \ge c^T \\ (6.4) & e^T\xi \quad \le U & & \eta^T b \quad \ge -M \\ & \xi,\xi_o \ge 0 & & \eta,\eta_o \ge 0 \end{array}$$

where $M \equiv \|c\| \cdot 2^{L+1}$, $U \equiv \sqrt{n}\,(2^L+1)$

Both primal and dual constraint sets are consistent ($\xi = 0$, $\xi_o = 1$; $\eta = 0$, $\eta_o = \max_j c_j$) so that case (iv) of the Duality States theorem holds.

To demonstrate consistency of the original system (T) to whatever degree of approximate solution is desired, one need merely obtain a member of the (TD) constraint set with $\xi_o \le 1$. We thus define a "Curtailed Solution of Degree (α,β)" to be a member of the constraint set of the new (TD) wherein the coefficients in (T) have been curtailed to precision of $2^{-\alpha}$ and magnitude of 2^β. We suggest that this notion of solution may be useful in matters of approximate solution.

Exterior (Reflection) Algorithms

The reflection of a point on the "wrong" side of a constraining hyperplane to the "right" side of it yields a point at lesser distance to the constraint set (see Chapter XVIII of [5]). The amount of distance reduction depends chiefly on the "thickness" (diameter) of the set and the pre-reflection distance to it. In our C^2 (constraint contraction) algorithms a balance is sought between rapid contraction of set size and maintenance of closeness to the constraint set.

When the vectors a^T_i in (1.2) are taken so that $a^T_i a_i = 1$, $|\theta_i(x)|$ is the distance from x to the i^{th} hyperplane. Addition of the same constant to all b_i results in an expansion or contraction of the constraint set so that the distance from the center of the set (e.g., the generally unique member of (TD), see figure 1) to each hyperplane is changed by the same amount for all hyperplanes, i.e., it generates a homothetic transformation of the set about the center.

Each algorithm generates a sequence of points x^o, x^1, $x^2,\ldots,$ $x^k,\ldots$ and a sequence of right hand sides b, $b^1,\ldots,b^k$ ..., where $b^k = b + e\Delta^k$, where $|\Delta^k| \to 0$. The point x^k satisfies

$$(6.5) \qquad a_i^T\, x^k \le b^{k-1}\ , \qquad i=1,\ \ldots\ n$$

and Δ^k (hence b^k) is chosen so that x^k is outside the constraint set of

$$(6.6) \qquad a^T x \le b^k$$

We stop at x^N when $|\Delta^N|$ is sufficiently small.

The constraint contraction process proceeds as follows. Start with some x^o. If x^o satisfies (1.2), stop. If not, choose Δ^o so that $\max_i\,(a_i^T\, x^o - b_i - \Delta^o) = 0$, i.e., $\Delta^o = \max_i\,(a_i^T\, x^o - b_i)$. Set $\Delta^1 = \frac{3}{4}\,\Delta^o$. Then, x^o is outside $\{x{:}a^T x \le b'\}$. Recursively, having obtained x^k within $\{\,x{:}a^T\, x \le b^k\,\}$, then $\max_i\,(a_i^T x^k - b_j) = \rho^k\Delta^k$, where $0 < \rho^k \le 1$. Choose $\Delta^{k+1} = \frac{3}{4}\,\rho^k\Delta^k$. Then x^k is outside $\{x{:}a^T x \le b^{k+1}\}$ we apply our chosen reflection procedure to get x^{k+1} within the latter set.

The simplest procedure reflects in the most violated (variant:any violated) hyperplane, i.e., the hyperplane with normal a_r^T where $a_r^T\, x^k - b_r^k = \max_i\,(a_i^T\, x^k - b_i^k)$. Next set $v^{kT} \equiv -a_r^T$ and $z^1(k) = x^k + \alpha_1 v^k$, where α_1 is chosen so that $a_r^T z^1\,(k) - b_r^k = -(a_r^T\, x^k - b_r^k)$. Thereby

$$(6.7) \qquad \alpha_1 = -2(a_r^T\, x^k - b_r^k)$$

If $z^1(k)$ is in the "k^{th} set" $\{x : a^T x \le b^k\}$, set $z^1(k) = x^{k+1}$. If not, reflect $z^1(k)$ to obtain $z^2(k)$, etc. until some $z^m(k)$ is in and then set $z^m(k) = x^{k+1}$.

To get into the k^{th} set more rapidly, one might use "curtailed" reflection or "guided" reflection. In curtailed reflection one proceeds no farther than the most distant un-violated hyperplane s such that $a^T_s a_r < 0$. I.e., one takes s so that

$$(6.8) \qquad b^k_s - a^T_s x^k = \max_{I^+_r} \; (b^k_i - a^T_i x^k)$$

$$\text{where} \quad I^+_r = \{i : b^k_i - a^T_i x^k > 0 \quad \text{and} \quad a^T_i a_r < 0\}$$

For $z^1(k)$ not to violate s, one must have

$$(6.9) \qquad \alpha_1 \le \frac{b^k_s - a^T_s x^k}{-a^T_s a_r}$$

Thereby α_1 is chosen as $\min \left[-2(a^T_r x^k - b^k_r), \; \frac{b^k_s - a^T_s x^k}{-a^T_s a_r} \right]$.

In guided reflection one reflects x^k in hyperplane r, but moves in the direction of a^T_s. Thus

$$a^T_r(x^k + \alpha_1 a_s) - b^k_r = -(a^T_r x^k - b^k_r)$$

so

$$(6.10) \qquad \alpha_1 = \frac{-2(a^T_r x^k - b^k_r)}{a^T_r a_s}$$

Interior (Centering) Algorithms

In our interior "centering" algorithms we start each iteration, say k, with a point x^k on the boundary of the k^{th} set $\{x : a^T x \leq b^k\}$. We attempt to move to the center of the constraint set by moving out from the boundary in a direction v^k, to be defined. Thus, we consider $z^k \equiv x^k + \alpha v^k$ and try to find a best (reasonably computable) value for α. For the k^{th} set (and b^k) we define $d_i(x) = -\theta_i(x)$. Thus,

$$(6.11) \quad d_i(z^k) = b_i^k - a_i^T (x^k + \alpha v^k) = d_i(x^k) - \alpha a_i^T v^k$$

Define

$$I_k^o \equiv \{i : d_i(x^k) = 0\}, \quad V_k^+ \equiv \{i : a_i^T v^k > 0, \quad V_k^- \equiv \{i : a_i^T v^k < 0\}$$

In order that z^k remain within the k^{th} set, we must have $d_i(z^k) \geq 0$. By (6.11), this is true iff $\alpha a_i^T v^k \leq d_i(x^k)$. Letting r be such that

$$(6.12) \quad \frac{d_r(x^k)}{a_r^T v^k} = \min_{V_k^+} \frac{d_i(x^k)}{a_i^T v^k}$$

This reduces to

$$(6.13) \quad \alpha \leq \frac{d_r(x^k)}{a_r^T v^k}$$

for z^k to remain in the set.

We define the "push-off" direction from the boundary, v^k, as follows:

$$(6.14) \quad v^k = \begin{cases} \sum_{I_k^o} (-1) a_i, \text{ if } |I_n^o| \geq n \\ \sum_{I_k^o \cup C_k} (-1) a_i, \text{ otherwise, where } |I_k^o \cup c_k| = n \\ \quad \text{and } C_k \equiv \{i : |C_k| = n - |I_k^o| \text{ and } d_i(x^k) \leq d_j(x^k) \text{ for j and } C_k\} \end{cases}$$

We must have $0 < \alpha \leq d_r(x^k)/a_r^T v^k$. To "center" as well as possible we want to choose α in this interval so that we

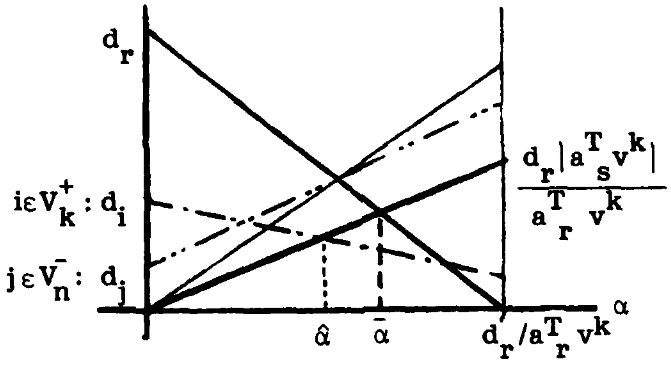
(6.15) $$\max_{\alpha} \min_{i} d_i(z^k)$$

Let $s \in I_k^o$ be such that

(6.16) $$-a_s^T v^k \leq -a_i^T v^k \text{ , } \forall i \in I_r^o$$

Then s corresponds to the line of minimum slope from the set I_r^o in the following plot:

(6.17)


d_r
$i \varepsilon V_k^+ : d_i$
$j \varepsilon V_n^- : d_j$
$\frac{d_r |a_s^T v^k|}{a_r^T v^k}$
α
$\hat{\alpha}$
$\bar{\alpha}$
$d_r/a_r^T v^k$

Evidently, it would be fairly complicated to exactly choose α to max min $d_i(z^k)$, e.g., $\hat{\alpha}$. Thus, we choose

(6.18) $$\alpha = \bar{\alpha} = d_r(x^k)/(|a_r^T v^k| + |a_s^T v^k|)$$

or, even more simply, if desired,

(6.19) $$\alpha = \tilde{\alpha} = \frac{1}{2} d_r(x^k)/a_r^T v^k$$

Then, $x^{k+1} = z^k$ and we choose Δ^{k+1} as the minimum $\Delta \geq 0$ such that $d_i'(z^k) \geq 0$, where $d_i'(z^k)$ means we use not b_i^k, but $b_i + \Delta$. The iteration is complete in taking $b^{k+1} = b_i + \Delta^{k+1}$.

CONCLUSION

In the foregoing, we have attempted to show through key examples in the context of Khachian's and related work the inappropriateness and misguided character of "complexity" as a guide to worthwhile directions of research in computability of mathematical programming (especially LP) systems. We have tried to indicate better directions and have suggested explicitly methods which might improve computation in both approximate and exact solution. Let us conclude with two further examples of "computability" versus "complexity" approaches.

First, L.E. Woolsey observed that he always got extremely large times for geometric programming calculations whenever constraints were slack in the final solution. He, therefore, added additional variables and a corresponding small perturbation of the functional to insure that all constraints were satisfied as equalities. Result: great improvement in comoutational speed. Further result: R.J. Duffin and associates developed new, simpler proofs of geometric programming theory using the Woolsey perturbation and ordinary Lagrangean theory.

Finally, in the earliest days of linear programming computation oil company computer groups observed that they got excessive times in general LP calculations whenever there were zeros in the right hand side vector. Computability solution: Introduce a new variable and equation constraint, $x_{n+1} = 1$ and add multiples of it to the constraints with zeros to knock them out. Result: great improvement in computation time. Although this device is part of the folklore, many newcomers to computation are still unaware of it.

Oddly, there is a historical perspective for it. Gauss, in trying to improve computation of the usually (we would call) ill-conditioned systems of linear equations involved in geodetic or magnetic problems, introduced an additional variable and equation which would increase the value of the determinant (hence the condition of the system) over that of the old system. This helped. Gauss apparently hit upon such a device since he himself performed actual computation.

REFERENCES

1. R. Armstrong, A. Charnes, F. Phillips, "Page Cuts for Integer Interval Linear Programming," Discrete Appl. Math. 1, 1-14, (1979).

2. M.L. Balinski and R.E. Gomory, "A Primal Method for the Assignment and Transportation Problems," Management Science, 10, 578-594, (1964).

3. A. Ben-Israel and A. Charnes, "An Explicit Solution of a Class of Linear Programming Problems," Operations Research, 16, 1166-1175, (1968).

4. A. Charnes, "Optimality and Degeneracy in Linear Programming," Econometrica, 20, 160-170, (1952).

5. A. Charnes and W.W. Cooper, Management Models and Industrial Applications of Linear Programming, Volumes I and II, J. Wiley and Sons, (1961).

6. A. Charnes, D. and F. Granot, "A Primal Algorithm for Interval Linear Programming Problems," Linear Algebra and Its Applic., 17, 65-78, (1977).

7. A. Charnes, F. Granot and F. Phillips, "An Algorithm for Solving Interval Linear Programming Problems," Operations Research, 25, (July-August 1977).

8. A. Charnes, D. Karney, D. Klingman and J. Stutz, "Past, Present and Future of Large Scale Transshipment Computer Codes and Applications," Computers and Operations Research Journal, 2, 71-81, (1975).

9. A. Charnes and K. Kortanek, "An Opposite Sign Algorithm for 'Purification' to an Extreme Point Solution," ONR Research Report No. 84, Systems Research Group, Northwestern University, June 1963.

9a. A. Charnes, K. Kortanek and W. Raike, "Extreme Point Solutions in Mathematical Programming: An Opposite Sign Algorithm," Systems Research Memorandum No. 129, Systems Research Group, Northwestern University, June 1965.

10. A. Charnes and C.E. Lemke, "A Modified Simplex Method for Control of Roundoff Error in Linear Programming," Proc. ACM, Pittsburgh (May 2-3, 1952), 97-98, (1952).

11. A. Charnes, W.W. Cooper and G. Thompson, "Some Properties of Redundant Constraints and Extraneous Variables in Direct and Dual Linear Programming Problems," Operations Research, 10, 711-723, (1962).

12. U. Eckhardt, "Representation of Convex Sets," Extremal Methods and Systems Analysis, Ed. A.V. Fiacco and K.O. Kortanek, 374-384, Springer-Verlag, (1980).

13. P. Gacs and L. Lovasz, "Khachian's Algorithm for Linear Programming," Working Paper, Computer Science Department, Stanford University, (Fall 1979).

13. P. Gacs and L. Lovasz, "Khachian's Algorithm for Linear Programming," Working Paper, Computer Science Department, Stanford University, (Fall 1979).

14. R. Jeroslow, "Trivial Integer Programs Unsolvable by Branch and Bound," Math. Programming, 6, 105-109, (1974).

15. L.G. Khachian, "A Polynomial Algorithm in Linear Programming," Soviet Math. Doklady, 20, (1979).

16. V. Klee and G.J. Minty, "How Good is the Simplex Algorithm?", Inequalities III, Ed. by O. Shisha, Academic Press, (1971).

17. A. Yu. Levin, "An Algorithm for the Minimization of Convex Functions," Dokl. Akad. Nauk SSSR, 160, No. 6 (1965).

18. P.D. Robers and A. Ben-Israel, "A Suboptimization Method for Interval Linear Programming: A New Method for Linear Programming," Linear Algebra and Its Appl., 3, 383-405, (1970).

19. N.Z. Shor, "On the Rate of Convergence of the Generalized Gradient Methods," Kibernetika, No. 3, (1968).

20. N.Z. Shor and V.I. Biletskii, "The Method of Space Dilatation to Accelerate Convergence in Gully-type Problems," Optimal Decision Theory, 2, Kiev, (1969).

21. N.Z. Shor, "The Use of the Operation of Space Dilatation in the Minimization of Convex Functions," Kibernetika, No. 1, (1970).

22. N.Z. Shor, "Generalized Gradient Methods for the Minimization of Nonsmooth Functions and the Application in Mathematical Programming (survey)," Ekonom, Mat. Metody, 12, No. 2 (1976).

23. N.Z. Shor, "Cut-Off Method with Space Extension in Convex Programming Problems," Kibernetika, No. 1, 94-95, Jan.-Feb. 1977 (article submitted 28 October 1976).

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Center for Cybernetic Studies The University of Texas at Austin	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. AD-A084 682

3. REPORT TITLE

(6 Complexity and Computability of Solutions to Linear Programming Systems

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

(9 Research rept.

5. AUTHOR(S) *(First name, middle initial, last name)*

(10 A. Charnes, W.W. Cooper, S. Duffuaa, ~~and~~ M. Kress

6. REPORT DATE (11 February 1980 (12 39	7a. TOTAL NO. OF PAGES 36	7b. NO. OF REFS 24

8a. CONTRACT OR GRANT NO. 15 N00014-75-C-0569, N00014-80-C-0242	9a. ORIGINATOR'S REPORT NUMBER(S) (14 CCS-367
b. PROJECT NO. NR047-021	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Office of Naval Research (Code 434) Washington, DC

13. ABSTRACT

Complexity, computability and solution of linear programming systems are re-examined in the light of Khachian's new notion of (approximate) solution. Algorithms, basic theorems and alternate representations are reviewed. It is shown that the Klee-Minty example has never been exponential for (exact) adjacent extreme point algorithms and that the Balinski-Gomory (exact) algorithm is polynomial where (approximate) ellipsoidal "centered-cut-off" algorithms (Levin, Shor, Khachian, Gacs-Lovasz) are exponential. Both the Klee-Minty and the new J. Clausen example are shown to be trivial (explicitly solvable) interval programming problems. A new notion of computable (approximate) solution is proposed together with an a priori regularization for linear programming systems. New polyhedral "constraint contraction" algorithms are proposed for approximate solution and the relevance of interval programming for good starts or exact solution is brought forth.

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Complexity						
Computability						
Linear Programming Systems						
Constraint Contraction Algorithms						
Ellipsoidal Algorithms						
Nondifferentiable Convex Programming						

Unclassified

Security Classification