AD-A080 837 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/2
COMMENTS ON KHACHIAN'S ALGORITHM FOR LINEAR PROGRAMMING.(U)
NOV 79 G B DANTZIG N00014-75-C-0267
UNCLASSIFIED SOL-79-22

| OF |

AD A080837

END
DATE FILMED
3-80
DDC

LEVEL II

Systems
Optimization
Laboratory

(12)

Department of Operations Research
Stanford University
Stanford, CA 94305

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

COMMENTS ON KHACHIAN'S ALGORITHM FOR LINEAR PROGRAMMING

by

George B. Dantzig

TECHNICAL REPORT SOL 79-22
November 1979

Research and reproduction of this report were partially supported by the Department of Energy Contract DE-AS03-76-SF00326, PA No. DE-AT-03-76ER72018; the National Science Foundation Grants MCS76-81259 A01, MCS78-20019 A01, ENG77-06761; Office of Naval Research Contract N00014-75-C-0267.

COMMENTS ON KHACHIAN'S ALGORITHM FOR LINEAR PROGRAMMING

by

George B. Dantzig*

Jack Edmonds once defined a "good" algorithm as one that can solve any problem within a class in a polynomial bounded time where each factor of each term of the polynomial is some parameter either expressing the size of the problem or the number of digits needed to store the data of the problem. The point of this little essay, is this: Solving in a polynomially bounded time says little.

A POLYNOMIAL BOUND CAN BE A BIG NUMBER

The *expected* number of steps to find a feasible solution to a linear program using Phase 1 of the Simplex Method, for moderately sized problems,is conjectured to be, of the order

$$\alpha \cdot m \quad \text{STEPS},$$

where m is the number of equations and α is typically 2 to 3 (or 4 to 6 for an optimal solution using both Phase I and II). Thousands of linear programs are solved each day using some variant of the simplex method—a value of $\alpha > 4$ is rarely seen. The effort to do each pivot step is of order m^2 but, because most coefficients of the matrix are usually zero, the work to do a pivot can be reduced to a fraction of m^2.

Even for problems involving 1000 equations and 3000 variables , α appears to be small. It is conjectured for n large relative to m, that the expected value of α will grow slowly, say

$$\exp(\alpha) < \log_2(2 + \frac{n}{m}) \quad .$$

Khachian seeks a solution to a system of strict inequalities were $\bar{n}$ is the number of variables and $\bar{m}$ the number of strict linear inequality constraints, [1]. The relations between (m, n) and $(\bar{m}, \bar{n})$ are as follows:

$$\bar{n} = n - m \quad , \quad \bar{m} = n$$

or

$$m = \bar{m} - \bar{n} \quad , \quad n = \bar{m}$$

*In response to numerous inquires about the Khachian algorithm.

Peter Gács and Laszlo Lovász state in [2] that the effort involved to solve a problem by Khachian's algorithm, is of the order $6L(\bar{n}+1)^2$ steps. Richard Stone has pointed out that the correct value is $4L(\bar{n}+1)^2$. For convenience below I have rounded this down to $4L\bar{n}^2$ steps. The work of each step (which is not a pivot operation) is of order $\bar{n}^2$; it can be thought of as comparable to a simplex iteration. Moreover, the devices that have been so successful in reducing the work of each iteration of the Simplex Method appear applicable to reducing the work of each step of the Khachian method. If coefficients range in value by 10^5 and are given to 5 significant decimal figures, then L is about $30\bar{m}\,\bar{n}$. This gives as an upper bound for the number of steps for the Khachian algorithm:

$$120\bar{m}\,\bar{n}^3 \quad \text{STEPS.}$$

As polynomials go, this is of surprisingly low order—a truly remarkable theoretical result and the beginning of a promising and exciting new area of research. It shows the essentially algebraic character of linear inequality systems in spite of the fact that the number of extreme points can be exponential in $\bar{m}$ and $\bar{n}$.

Whether or not the algorithm is practical depends on the size of this bound for moderately large linear programming problems of interest. The example discussed below is typical.

At Stanford we have developed a dynamic linear programming model, called PILOT, for Energy/Economic planning. It is used by the Electric Power Research Institute in their planning for the Utility Industry and by the Department of Energy for their long range studies. Case studies (scenarios) usually require 5 to 10 minutes of CPU time on the IBM 370-168 if started with a good guess. From a "cold start" it can take up to 1 hour of CPU time. Because the problem is dynamic in structure, the number of pivot steps seems to run a little higher on the average — about 3.5 m for each phase of the Simplex Method.

For the PILOT Model:

$$m = 1000 \quad , \qquad n = 3000$$

or

$$\bar{n} = 2000 \quad , \qquad \bar{m} = 3000 \quad .$$

The comparative number of steps are:

$$\text{KHACHIAN:} \quad 120\bar{m}\bar{n}^3 \quad \text{STEPS} = 120 \times 3000 \times 2000^3 = 2.88 \times 10^{15} \quad \text{STEPS}$$
$$\text{SIMPLEX:} \quad 3.5m \quad \text{STEPS} = 3.5 \times 10^3 \quad \text{STEPS} \quad .$$

For a 1000 equation system of the sparsity of PILOT, each step takes about 1/2 second on the IBM 370-168 or at the rate of 6.3×10^7 steps per year. Thus the estimated and actual times to solve this problem are

$$\text{KHACHIAN:} \quad 2.88 \times 10^{15} \quad \text{STEPS} \doteq 50,000,000 \quad \text{years}$$

$$\text{SIMPLEX:} \quad 3.5 \times 10^3 \quad \text{STEPS} \doteq 30 \quad \text{minutes} \quad .$$

This comparison is, of course, unfair. The upper bound given by Khachian is not tight and a tight bound for the worst case could turn out to be a much smaller number. Walter Murray has looked into the way that the ellipsoids adjust and contract in volume. Each ellipsoid appears to be a slight perturbation of its predecessor. Because of this, Murray believes that the expected number of steps (for problems of the same dimensions and size of the coefficients) will not be much different from that of the worst case [4].

There is a great deal of interest in the Khachian algorithm. One can expect a reexamination of similar previously proposed algorithms that converge in the limit to a feasible solution. Restated in terms of integer coefficients, a finiteness proof with a polynomial bound might also be obtained perhaps some will turn out to have practical upper bound and expected value estimates. For a discussion of this possibility see [4].

It has been suggested by N. Zadeh, Alan Hoffman and others, that the Simplex Algorithm be reexamined to see if it too has polynomial bound under the assumptions that the class of linear programs considered has:

(1) integer coefficients, and

(2) the maximum of the absolute value of coefficients is less than a constant independent of m and n.

Examples have been constructed by V. Klee and G. Minty [3], P. Wolfe, and N. Zadeh such that the number of steps grows exponentially but these do not satisfy (1) and (2) above.

In summary, the existence of an algorithm with a polynomial bound of $120\overline{m}\,\overline{n}^3$ steps, each step requiring about $\overline{n}^2$ operations, is an important theoretical result. Unfortunately a polynomial bound does not imply a "good algorithm". To qualify as good, the bound must not be too high for practical problems of interest such as those routinely solved by the Simplex Method. The effect of the Khachian result will be to intensify the research to find an algorithm with a more practical bound like $2(\overline{m}-\overline{n})$ steps, each step requiring about $(\overline{m}-\overline{n})^2$ operations (the empirically observed rough average for finding a feasible solution using the Simplex Method).

References

[1] L.G. Khachian, Doklady Akademii Nauk USSR, 1979, Vol.. 244, No. 5, pages 1093-1096.

[2] P. Gács and L. Lovász, "Khachian's Algorithm for Linear Programming," Computer Science Department, STAN-CS-79-750, Stanford University.

[3] V. Klee and G.J. Minty, "How Good is the Simplex Algorithm," *Inequalities III*, Academic Press, New York, 1972, pp. 159–175.

[4] Walter Murray, Ellipsoidal Algorithms for Linear Programming, Working Paper 79-1, Systems Optimization Laboratory Department of Operations Research, Stanford University, November 1979.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: SOL-79-22

2. GOVT ACCESSION NO.:

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): Comments on Khachian's Algorithm for Linear Programming

5. TYPE OF REPORT & PERIOD COVERED: Technical Report

6. PERFORMING ORG. REPORT NUMBER: SOL 79-22

7. AUTHOR(s): George B. DANTZIG

8. CONTRACT OR GRANT NUMBER(s): N00014-75C-0267

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Operations Research - SOL
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-143

11. CONTROLLING OFFICE NAME AND ADDRESS:
Operations Research Program -- ONR
Department of the Navy
800 N. Quincy St., Arlington, VA 22217

12. REPORT DATE: Nov[redacted]979

13. NUMBER OF PAGES: 4

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report):

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):
This document has been approved for public release and sale; its distribution is 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):
KHACHIAN
POLYNOMIAL BOUND
LINEAR PROGRAMMING
SIMPLEX METHOD

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

Khachian's polynomial bound for finding a feasible solution to a relaxed linear program is an important theoretical result. Unfortunately, a polynomial bound does not imply a good algorithm because such a bound could be too large for problems of practical interest. For example, using the formulae in the original paper, practical problems with 3000 non-negative variables and 1000 equations (which are solved under one-half hour on IBM 370-168 using the simplex method) would involve over 10^{15} iterations and would take 50,000,000 years to solve using Khachian's method.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4/08 765