UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

MASTER COPY - FOR REPRODUCTION PURPOSES

DTIC FILE COPY

(2)

REPORT DOCUMENTATION PAGE

AD-A202 502

DTIC ELECTE 06 1988 D

1b. RESTRICTIVE MARKINGS	
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) LIDS-P-1818	5. MONITORING ORGANIZATION REPORT NUMBER(S) ARO 24635.96-MA-UIR
6a. NAME OF PERFORMING ORGANIZATION M.I.T. / 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION U. S. Army Research Office
6c. ADDRESS (City, State, and ZIP Code) Laboratory for Information and Decision Systems Cambridge, MA 02139	7b. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U. S. Army Research Office / 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAL03-86-K-0171
8c. ADDRESS (City, State, and ZIP Code) P. O. Box 12211 Research Triangle Park, NC 27709-2211	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
A VERY SIMPLE POLYNOMIAL-TIME ALGORITHM FOR LINEAR PROGRAMMING

12. PERSONAL AUTHOR(S)
Paul Tseng

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ____ TO ____	September 1988	10

16. SUPPLEMENTARY NOTATION
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	This document proposes

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
~~In this note we propose~~ a polynomial-time algorithm for linear programming. This algorithm augments the objective by alogarithmic penalty function and then solves a sequence of quadratic approximations of this program. This algorithm has a complexity of $O(m^{1/2}L)$ iterations and $O(m^{3.5}L)$ arithmetic operations, where m is a number of variables and L is the size of the problem encoding in binary. This algorithm does not require knowledge of the optimal value and generates a sequence of primal(Dual) feasible solutions that converges to an optimal primal (dual) solution (the latter property provides a particularly simple stopping criterion). Moreover, this algorithm is simple and intuitive, both in the description and in the analysis - in contrast to existing polynomial-time algorithms. It works exclusively in the original space.

... logarithmic penalty function ...

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ARO 24635.96-MA-UIR

September 24, 1988 LIDS-P-1818

A Very Simple Polynomial-Time Algorithm for Linear Programming[1]

by

Paul Tseng[2]

Abstract

In this note we propose a polynomial-time algorithm for linear programming. This algorithm augments the objective by a logarithmic penalty function and then solves a sequence of quadratic approximations of this program. This algorithm has a complexity of $O(m^{1/2}\cdot L)$ iterations and $O(m^{3.5}\cdot L)$ arithmetic operations, where m is the number of variables and L is the size of the problem encoding in binary. This algorithm does not require knowledge of the optimal value and generates a sequence of primal (dual) feasible solutions that converges to an optimal primal (dual) solution (the latter property provides a particularly simple stopping criterion). Moreover, this algorithm is simple and intuitive, both in the description and in the analysis – in contrast to existing polynomial-time algorithms. It works exclusively in the original space.

KEY WORDS: linear program, Karmarkar method, quadratic approximation, logarithmic penalty functions

---

[1] This research is partially supported by the U.S. Army Research Office, contract DAAL03-86-K-0171 (Center for Intelligent Control Systems), and by the National Science Foundation, grant NSF-ECS-8519058.

[2] Center for Intelligent Control Systems, Room 35-205, Massachusetts Institute of Technology, Cambridge, MA 02139.

Acknowledgement: Thanks are due to Tom Luo and Kok-Choon Tan for their many helpful comments.

1. Introduction

Consider linear programming problems of the form

$$\begin{array}{ll} \text{Minimize} & \langle c,x\rangle \\ \text{subject to} & Ax = b,\ x \geq 0, \end{array} \qquad (P)$$

where c is an m-vector, A is an n×m matrix, b is an n-vector, and $\langle\cdot,\cdot\rangle$ denotes the usual Euclidean inner product. In our notation, all vectors are column vectors and superscript T denotes the transpose. We will denote by $\Re^m$ ($\Re^n$) the m-dimensional (n-dimensional) Euclidean space.

For any vector x in $\Re^m$, we will denote by x_j the jth component of x. For any positive vector x in $\Re^m$, we will denote by D_x the m×m positive diagonal matrix whose jth diagonal entry is the jth component of x. Let X denote the relative interior of the feasible set for (P), i.e.

$$X = \{\ x \in \Re^m \mid Ax = b,\ x > 0\ \}.$$

We will also denote by e the vector in $\Re^m$ all of whose components are 1's. "Log" will denote the natural log and $\|\cdot\|_1$, $\|\cdot\|_2$ will denote, respectively the L_1-norm and the L_2-norm. We make the following standing assumption about (P):

Assumption A:

(a) Both X and $\{\ u \in \Re^n \mid A^T u < c\ \}$ are nonempty.

(b) A has full row rank.

Assumption A (b) is made only to simplify the analysis and can be removed without affecting either the algorithm or the convergence results. Note that Assumption A (a) implies (cf. [3], Corollary 29.1.5) that the set of optimal solutions for (P) is nonempty and bounded. For any $\varepsilon > 0$, consider the following approximation of (P):

$$\begin{array}{ll} \text{Minimize} & f_\varepsilon(x) \\ \text{subject to} & Ax = b, \end{array} \qquad (P_\varepsilon)$$

where we define $f_\varepsilon:(0,\infty)^m \rightarrow \Re$ to be the penalized function:

$$f_\varepsilon(x) = \langle c,x\rangle - \varepsilon\textstyle\sum_j \log(x_j). \tag{1.1}$$

Note that

$$\nabla f_\varepsilon(x) = c - \varepsilon \begin{bmatrix} 1/x_1 \\ \vdots \\ 1/x_m \end{bmatrix}, \qquad \nabla^2 f_\varepsilon(x) = \varepsilon \begin{bmatrix} 1/(x_1)^2 & \cdots & 0 \\ & \vdots & \\ 0 & \cdots & 1/(x_m)^2 \end{bmatrix}. \tag{1.2}$$

The literature on Karmarkar's algorithm [1] is too vast to survey and we will not do so here. Our approach to solving (P), which is similar to that taken in [5]-[6], [11], is to solve, approximately, a sequence of problems $\{(P_{\varepsilon^r})\}$, where $\{\varepsilon^r\}$ is a sequence of geometrically decreasing scalars. The approximate solution of (P_{ε^r}), denoted by x^r, is obtained by solving the quadratic approximation of (P_{ε^r}) around x^{r-1}. The novel feature of our algorithm is its simplicity, both in the description and in the analysis. And yet it has an excellent complexity. Also, our algorithm is unusual in that it is a primal affine-scaling algorithm that generates feasible primal <u>and</u> dual solutions.

This note proceeds as follows: in §2 we show that, given an approximately-optimal primal dual pair of (P_ε), an approximately-optimal primal dual pair of $(P_{\alpha\varepsilon})$, for some $\alpha \in (0,1)$, can be obtained by solving a quadratic approximation of (P_ε). In §3 and §4 we present our algorithm and analyze its convergence. In §5 we discuss the initialization of our algorithm. Finally, in §6 we give our conclusion and discuss extensions.

2. Technical Preliminaries

Fix any $\varepsilon > 0$ and consider the problem (P_ε)

$$\begin{array}{ll} \text{Minimize} & \langle c,x\rangle - \varepsilon\sum_j \log(x_j) \\ \text{subject to} & Ax = b, \end{array}$$

and let $\bar{x}$ be any element of X and let $\bar{u}$ be any element of $\Re^n$. We replace the objective function by its quadratic approximation around $x = \bar{x}$. This gives (cf. (1.2))

$$\begin{array}{ll} \text{Minimize} & \langle c-\varepsilon(\bar{D})^{-1}e,z\rangle + \varepsilon\langle z,(\bar{D})^{-2}z\rangle/2 \\ \text{subject to} & Az = 0, \end{array}$$

where we denote $\bar{D} = D_{\bar{x}}$. The Karush-Kuhn-Tucker point for this problem, say (z,u), satisfies

$$c - \varepsilon(\bar{D})^{-1}e + \varepsilon(\bar{D})^{-2}z - A^Tu = 0, \tag{2.1a}$$
$$Az = 0. \tag{2.1b}$$

Let $d = (\bar{D})^{-1}z$. Solving for d gives

$$\varepsilon d = [I - (A\bar{D})^T(A\bar{D}^2A^T)^{-1}A\bar{D}]\bar{r}, \tag{2.2}$$

where we denote

$$\bar{r} = -\bar{D}c + \varepsilon e + (A\bar{D})^T\bar{u}.$$

Note that, since the orthogonal projection is a nonexpansive mapping (with respect to the L_2-norm), we have from (2.2)

$$\|d\|_2 \leq \|\bar{r}\|_2/\varepsilon. \tag{2.3}$$

Let

$$x = \bar{x}+z, \tag{2.4}$$

$D = D_x$, and $\Delta = D_d$. Then $D = \bar{D} + \Delta\bar{D}$ and hence

$$\begin{aligned} -Dc + \varepsilon e + (AD)^Tu &= [-\bar{D}c + \varepsilon e + (A\bar{D})^Tu] + \Delta[-\bar{D}c + (A\bar{D})^Tu] \\ &= \varepsilon d + \Delta[-\bar{D}c + (A\bar{D})^Tu] \\ &= \Delta[\varepsilon e - \bar{D}c + (A\bar{D})^Tu] \end{aligned}$$

$$= \varepsilon\Delta d,$$

where the second and the fourth equality follow from (2.1a). This implies that (with d_j denoting the jth component of d)

$$\begin{aligned} \|-Dc + \varepsilon e + (AD)^T u\|_2 &= \varepsilon\|\Delta d\|_2 \\ &\le \varepsilon\|\Delta d\|_1 \\ &= \varepsilon \textstyle\sum_j (d_j)^2 \\ &= \varepsilon(\|d\|_2)^2 \\ &\le (\|\bar{r}\|_2)^2/\varepsilon, \end{aligned} \tag{2.5}$$

where the first inequality follows from properties of the L_1-norm and the L_2-norm and the second inequality follows from (2.3).

Consider any $\beta \in (0,1)$ and any scalar α satisfying

$$(\beta^2+m^{1/2})/(\beta+m^{1/2}) \le \alpha < 1. \tag{2.6}$$

Let $\varepsilon' = \alpha\varepsilon$ and $r' = -Dc + \varepsilon' e + (AD)^T u$. Then

$$\begin{aligned} \|r'\|_2/\varepsilon' &= \|-Dc + \alpha\varepsilon e + (AD)^T u\|_2/(\alpha\varepsilon) \\ &\le \|-Dc + \varepsilon e + (AD)^T u\|_2/(\alpha\varepsilon) + (1-\alpha)\cdot m^{1/2}/\alpha \\ &\le (\|\bar{r}\|_2/\varepsilon)^2/\alpha + (1/\alpha - 1)\cdot m^{1/2}, \end{aligned}$$

where the first inequality follows from the triangle inequality and the second inequality follows from (2.5). Hence, by (2.6), if $\|\bar{r}\|_2/\varepsilon \le \beta$, then $\|r'\|_2/\varepsilon' \le \beta$. Furthermore, by (2.3), we have that $\|d\|_2 \le \beta < 1$. Hence $e+d > 0$ and (cf. (2.4)) $x > 0$. Also, by (2.1b) and (2.4), $Ax = A(\bar{x}+z) = b$.

For any $\varepsilon > 0$, let $\rho_\varepsilon:(0,\infty)^m \times \Re^n \to [0,\infty)$ denote the function

$$\rho_\varepsilon(y,p) = \|-D_y c + \varepsilon e + (AD_y)^T p\|_2/\varepsilon. \tag{2.7}$$

We have then just proved the following important lemma:

Lemma 1 For any $\varepsilon > 0$, any $\beta \in (0,1)$ and any $(\bar{x},\bar{u}) \in X \times \Re^n$ such that $\rho_\varepsilon(\bar{x},\bar{u}) \leq \beta$, we have

$$(x,u) \in X \times \Re^n, \quad \rho_{\alpha\varepsilon}(x,u) \leq \beta,$$

where $\alpha = (\beta^2 + m^{1/2})/(\beta + m^{1/2})$ and (x,u) is defined as in (2.1a), (2.1b), (2.4).

3. The Homotopy Algorithm

Choose $\alpha = (1/4 + m^{1/2})/(1/2 + m^{1/2})$. Lemma 1 and (2.1a)-(2.1b), (2.4) motivate the following variant of Karmarkar's algorithm, parameterized by two positive scalars $\gamma \leq \varepsilon$:

Homotopy Algorithm

Step 0: Choose any $(x^1,u^1) \in X \times \Re^n$ such that $\rho_\varepsilon(x^1,u^1) \leq 1/2$. Let $\varepsilon^1 = \varepsilon$.

Step r: Compute (z^{r+1},u^{r+1}) to be a solution of

$$\begin{bmatrix} \varepsilon^r (D_{x^r})^{-2} & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z \\ u \end{bmatrix} = \begin{bmatrix} \varepsilon^r (D_{x^r})^{-1} e - c \\ 0 \end{bmatrix}.$$

Set $x^{r+1} = x^r + z^{r+1}$, $\varepsilon^{r+1} = \alpha\varepsilon^r$.

If $\varepsilon^{r+1} \leq \gamma$, terminate.

[Note: For simplicity we have fixed $\beta = 1/2$.] We gave the above algorithm the name "homotopy" because it solves (approximately) a sequence of problems $\{(P_{\varepsilon^r})\}$ that approaches (P) (see [2]).

4. Convergence Analysis

By Lemma 1, the homotopy algorithm generates, in at most $(\log(\gamma)-\log(\varepsilon))/\log(\alpha)$ steps, an $(x,u)\in X\times\Re^n$ satisfying

$$\|-D_x c + \gamma e + (AD_x)^T u\|_2 \le \gamma/2, \qquad (4.1)$$
$$Ax = b.$$

Since $\gamma > 0$, (4.1) implies that

$$0 < D_x c - (AD_x)^T u \le (3\gamma/2)\cdot e.$$

Since D_x is a positive diagonal matrix, multiplying both sides by $(D_x)^{-1}$ gives

$$0 < c - A^T u \le (3\gamma/2)\cdot(D_x)^{-1}e.$$

This in turn implies that, for each $j\in\{1,\ldots,m\}$,

$$\begin{aligned} 0 &< c_j - \langle A_j,u\rangle \le 3\gamma^{1/2}/2 && \text{if } x_j \ge \gamma^{1/2}, \\ 0 &< c_j - \langle A_j,u\rangle && \text{otherwise,} \end{aligned}$$

where c_j denotes the jth component of c and A_j denotes the jth column of A (note that an analogous argument shows that u^r is dual feasible for all r). Also, since

$$\begin{aligned} \log(1-\delta) &= -\delta - \delta^2/2 - \delta^3/3 - \delta^4/4 - \ldots \\ &\le -\delta(1 + (\delta/2) + (\delta/2)^2 + (\delta/2)^3 + \ldots) \\ &= -\delta/(1-\delta/2) = -1/(1/\delta-1/2), \end{aligned}$$

for any $\delta\in(0,1)$, we have that

$$\log(\alpha) = \log(1-(2+4m^{1/2})^{-1})$$

$\leq -1/(2+4m^{1/2}-1/2)$.

Hence we have just proved following:

Lemma 2 For any positive scalars $\gamma \leq \varepsilon$, the homotopy algorithm generates, in at most $(3/2+4m^{1/2})\cdot(\log(\varepsilon)-\log(\gamma))$ steps, a pair of optimal primal and dual solutions to a perturbed problem of (P), where the costs are perturbed by at most $3\gamma^{1/2}/2$ and the lower bounds are perturbed by at most $\gamma^{1/2}$.

Thus if we choose $\varepsilon = 2^{\Theta(L)}$ and $\gamma = 2^{-\Theta(L)}$, where L denotes the size of the problem encoding in binary (defined as in [1]), the homotopy algorithm would terminate in $O(m^{1/2}\cdot L)$ steps with an optimal primal dual solution pair to a perturbed problem of (P) and the size of the perturbation is $2^{-\Theta(L)}$. An optimal primal dual solution pair to (P) can then be recovered by using, say, the techniques described in [7] (also see [1]). Since the amount of computation per step is at most $O(m^3)$ arithmetic operations (not counting Step 0), the homotopy algorithm has a complexity of $O(m^{3.5}\cdot L)$ arithmetic operations. [We assume for the moment that Step 0 can be done very "fast". See §5 for justification.] This complexity can be reduced to $O(m^{3.31}L)$ by using Strassen's (impractical) matrix inversion method [10]. It may be possible to reduce the complexity to $O(m^3L)$ by using the rank-one update technique described in [1], [11].

5. Algorithm Initialization

In this section we show that, for ε sufficiently large, Step 0 of the homotopy algorithm (i.e. to generate a primal dual pair $(x,u)\in X\times\Re^n$ satisfying $\rho_\varepsilon(x,u) \leq 1/2$) can be done very "fast".

Suppose that (P) is in the canonical form considered by Karmarkar (see §5 of [1] for details on how to transform general linear programs into this canonical form). We claim that, for $\varepsilon = 2\|c\|_2$, a point $(x,u)\in X\times\Re^n$ satisfying $\rho_\varepsilon(x,u) \leq 1/2$ can be found immediately. To see this, note that in Karmarkar's canonical form, A and b have the form

$$A = \begin{bmatrix} A' \\ e^T \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ m \end{bmatrix},$$

where A' is some $(n-1)\times m$ matrix, and the point e is assumed to satisfy $Ae = b$. Let $x = e$ and $p = (0,\ldots,0,-1)^T$. Then

$$\begin{aligned} e + (AD_x)^T p &= e + A^T p \\ &= e - e = 0. \end{aligned} \tag{5.1}$$

Hence, by (2.7), $x = e$, and the triangle inequality,

$$\begin{aligned} \rho_\varepsilon(x,\varepsilon p) &= \|-D_x c + \varepsilon e + \varepsilon(AD_x)^T p\|_2/\varepsilon \\ &\le \|c\|_2/\varepsilon + \|e + (AD_x)^T p\|_2 \\ &= 1/2. \end{aligned}$$

Alternatively, we can solve the following problem

$$\begin{aligned} &\text{Maximize } \textstyle\sum_j \log(x_j) \\ &\text{subject to } \; Ax = b, \end{aligned} \tag{5.2}$$

whose Karush-Kuhn-Tucker point (x,p) can be seen to satisfy (5.1) (such a point exists if the feasible set for (P) is bounded). Then, for $\varepsilon = 2\|D_x c\|_2$, we also have $\rho_\varepsilon(x,\varepsilon p) \le 1/2$. The (unique) primal solution of (5.2) is sometimes called the analytic center [8] of the convex polyhedron $\{ x \mid Ax = b, x \ge 0 \}$. Polynomial-time algorithms for solving (5.2) are described in [8] and [9] (note that (5.2) does not have to be solved exactly).

6. Conclusion and Extensions

In this note we have proposed a very simple polynomial-time algorithm for linear programming. This algorithm solves a sequence of approximations to the original problem, each augmented by a logarithmic penalty function, and, unlike many other Karmarkar-type algorithms, uses no space transformation of any kind. This algorithm uses a number of steps (namely $O(m^{1/2}L)$) that is the lowest amongst interior point methods for linear programming. Because the primal and dual solutions obtained at each step are respectively primal and dual feasible, determining termination is particularly simple: terminate if the difference in their costs is below a prespecified tolerance.

There are many directions in which our result can be extended. Can the complexity be decreased further? Can our algorithm extend to quadratic programming or to problems with upper bound on the variables? Does there exist a general class of convex functions $h_\varepsilon:X\rightarrow X$, where X is a convex set in $\Re^m$, such that $h_\varepsilon(x) \rightarrow h_0(x)$ pointwise as $\varepsilon \downarrow 0$ and, given an approximate minimum of h_ε, an approximate minimum of $h_{\alpha\varepsilon}$ (α is a sufficiently small scalar in $(0,1)$) can be obtained very "quickly"?

References

[1] Karmarkar, N., "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica*, 4 (1984), 373-395.

[2] Garcia, C. B. and Zangwill, W. I., *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Inc., N. J. (1981).

[3] Rockafellar, R. T., *Convex Analysis*, Princeton University Press, Princeton, N.J. (1970).

[4] Bertsekas, D. P., *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press (1982).

[5] Gonzaga, C. C., "An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ Operations," Memo. UCB/ERL M87/10, Electronics Research Laboratory, University of California, Berkeley, CA (March 1987).

[6] Monteiro, R. C., and Adler, I., "An $O(n^3L)$ Primal-Dual Interior Point Algorithm for Linear Programming," Technical Report, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA (May 1987).

[7] Grötschel, Lovász, and Schrijver, A., "The Ellipsoid Method and its Consequences in Combinatorial Optimization," *Combinatorica*, 1 (1981), 169-197.

[8] Sonnevend, G., "An 'Analytical Center' for Polyhedrons and New Classes of Global algorithms for Linear (Smooth, Convex) Programming," Preprint, Dept. of Numerical Analysis, Institute of Mathematics Eotvos University, 1088, Budapest, Muzeum Korut, 6-8 (1985).

[9] Freund, R. M., "Projective Transformations for Interior Point Methods, Part II: Analysis of an Algorithm for finding the Weighted Center of a Polyhedral System," OR 180-88, Operations Research Center, M.I.T., MA (June 1988).

[10] Strassen, V., "Gaussian Elimination is not Optimal," *Numerische Mathematik*, 13 (1969), 354-356.

[11] Gonzaga, C. C., "An Algorithm for Solving Linear Programming in $O(n^3L)$ Operations," Technical Report UCB/ERL M87/10, Electronics Research Laboratory, University of California, Berkeley, CA (March 1987).

[12] Gill, P. E., Murray, W., Saunders, M. A., Tomlin, J. A., and Wright, M. H., "On Projected Newton Methods for Linear Programming and Equivalence to Karmarkar's Projective Method," *Mathematical Programming*, 36 (1986), 183-209.