***Available online a*t www.pelagiaresearchlibrary.com

Pelagia Research Library

Advances in Applied Science Research, 2012, 3 (5):3367-3373

ISSN: 0976-8610
CODEN (USA): AASRFC

On Some Polynomial-time Algorithms for Solving Linear Programming Problems

B. O. Adejo and H. S. Adaji

Department of Mathematical Sciences, Kogi State University, Anyigba

ABSTRACT

In this article we survey some Polynomial-time Algorithms for Solving Linear Programming Problems namely: the ellipsoid method, Karmarkar's algorithm and the affine scaling algorithm. Finally, we considered a test problem which we solved with the methods where applicable and conclusions drawn from the results so obtained.

INTRODUCTION

An algorithm A is said to be a polynomial-time algorithm for a problem P, if the number of steps (i. e., iterations) required to solve P on applying A is bounded by a polynomial function $O(m,n,L)$ of dimension and input length of the problem.

The Ellipsoid method is a specific algorithm developed by soviet mathematicians: Shor [5], Yudin and Nemirovskii [7]. Khachian [4] adapted the ellipsoid method to derive the first polynomial-time algorithm for linear programming. Although the algorithm is theoretically better than the simplex algorithm, which has an exponential running time in the worst case, it is very slow practically and not competitive with the simplex method. Nevertheless, it is a very important theoretical tool for developing polynomial-time algorithms for a large class of convex optimization problems.

Narendra K. Karmarkar is an Indian mathematician; renowned for developing the Karmarkar's algorithm. He is listed as an ISI highly cited researcher. Karmarkar's algorithm [3] solves linear programming problems in polynomial time. Karmarkar [9] used a potential function in his analyses. Affine scaling algorithm was proposed by many researchers independently (e. g. Barnes [1], Vanderbei, Meketon and Freedman [6]). Later it was discovered that the same primal affine scaling algorithm was proposed by Dikin [2] in the 1960s.

This paper is structured as follows:- In Section 1, a brief introduction is given, Section 2 we discuss the ellipsoid method, in Section 3, we present Karmarkar's algorithm, in Section 4, we present the affine scaling algorithm, in Section 5, a test problem was solved, while in Section 6, we present our concluding remarks.

2. THE ELLIPSOID METHOD

This section describes the ellipsoid method for determining the feasibility or otherwise of a system of linear inequalities; and outline arguments that establish that the method can be made polynomial. We present the interpretation by Gacs and Lovasz of the Khachian's arguments.

Suppose we wish to find $n-vector\ x$ satisfying

$$A^T x \leq b \qquad \text{... (2.1)}$$

where A^T is an $m \; x \; n$ matrix and b is an m – vector. The columns of A corresponding to the outward normals to the constraints, are denoted by $a_1, \; a_2, \ldots \; a_n$ and the components of b are denoted by $\beta_1, \; \beta_2, \ldots \; \beta_m$. Thus (2.1) can be restated as

$a_i^{\;T} x \leq \beta_i, \; i = 1, \; 2, \; \ldots \; m$

we assume throughout that n is greater than one.

The ellipsoid method constructs a sequence of ellipsoids $E_0, \; E_1, \; . \; \; . \; . \; , \; E_k, \; . \; . \; .$ each of which contains a point satisfying (2.1); if one exists. On the $(k+1)st$ iteration, the method checks whether the centre x_k of the current ellipsoid E_k satisfies the constraints (2.1). If so, the method stops. If not, some constraints violated by x_k, say

$a_i^{\;T} x_k > \beta_i$ for some $i = 1 \leq i \leq m$ ... (2.2)

are chosen and the ellipsoid of minimum volume that contains the half-ellipsoid

$\left\{ x \in E_k \middle| a_i^{\;T} x \leq a^T x_k \right\}$ ... (2.3)

is constructed. The new ellipsoid and its centre are denoted by $E_{k+1} \; and \; x_{k+1}$ respectively, and the above iterative step is repeated.

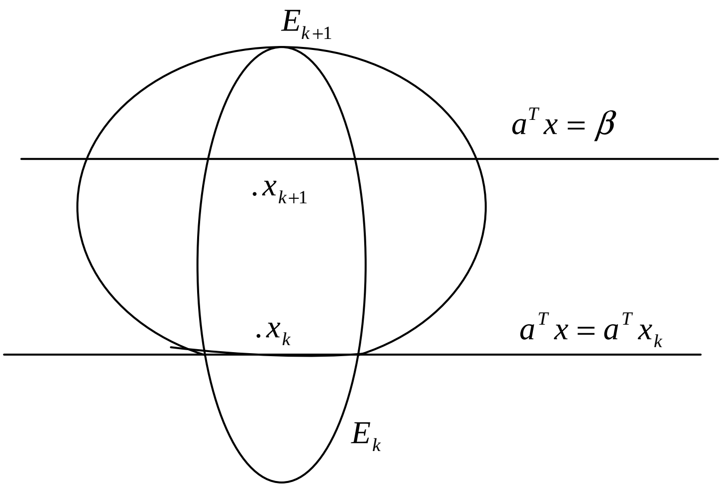

Fig 1(a). The ellipsoid method without deep cuts

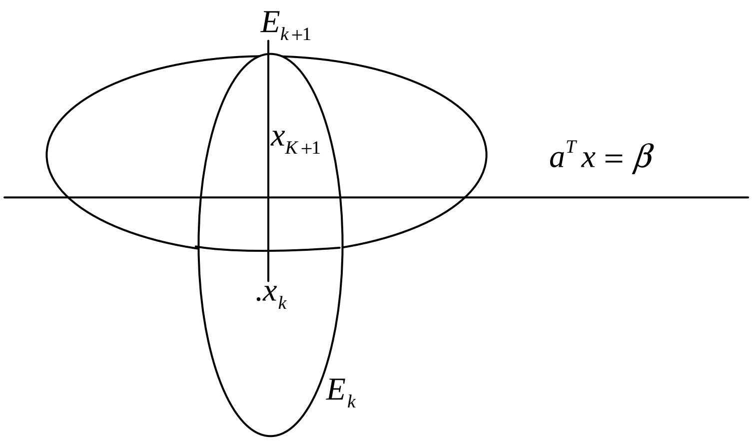

Fig 1(b). The ellipsoid method with deep cuts

Except for initialization, this gives a (possibly infinite) iterative algorithm for determining the feasibility of (2.1). Khachian [4] showed that one can determine whether (2.1) is feasible or not within a prespecified number of iterations by: (i) modifying this algorithm to account for finite precision arithmetic (ii) applying it to a suitable perturbation of the system (2.1), and (iii) choosing E_0 appropriately. System (2.1) is feasible if and only if termination occurs with a feasible solution of the perturbed system within a prescribed number of iterations (i.e $6n^2L$ iterations). Algebraically, we can represent the ellipsoid E_k as

$$E_k = \left\{ x \in R^n \middle| (x - x_k)^T B_k^{-1} (x - x_k) \le 1 \right\} \qquad \text{... (2.4)}$$

where x_k is its centre and B_k is a positive definite symmetric matrix. In terms of this representation the $(k+1)^{st}$ iterative step of the ellipsoid method is simply given by the formulae

$$x_{k+1} = x_k - \tau \left(\frac{B_k a}{\sqrt{a^T B_k a}} \right) \qquad \text{...(2.5)}$$

and

$$B_{k+1} = \delta \left[B_k - \frac{\sigma (B_k a)(B_k a)^T}{a^T B_k a} \right] \qquad \text{... (2.6)}$$

where

$$\tau = \frac{1}{n+1}, \sigma = \frac{2}{n+1} \ and \ \delta = \frac{n^2}{n^2 - 1} \qquad \text{... (2.7)}$$

E_{k+1} is determined by x_{k+1} *and* B_{k+1} as in (2.5) – (2.7), it is the ellipsoid of the minimum volume that contains the half-ellipsoid $\left\{ x \in E_k \middle| a^T x \le a^T x_k \right\}$

3 KARMARKAR'S ALGORITHM

Karmarkar's algorithm [9] considered a linear programming problem in canonical form as follows:

$$\begin{aligned} &\min imize\ c^T x \\ &subject\ to\ Ax = b \\ &\qquad\qquad e^T x = 1 \\ &\qquad\qquad x \ge 0 \end{aligned}$$

where $A \in Z^{m \times n}$, $e = (1, 1, 1, \ldots 1)^T$ … (3.1)

The LP problem (3.1) above in canonical form can be obtained from a standard form LP problem:

$$\begin{aligned} &\min imize\ c^T x \\ &Subject\ to\ Ax = b \\ &\qquad\qquad x \ge 0 \end{aligned} \qquad \text{… (3.2)}$$

The algorithm starts on the canonical form (3.1) and from the centre of the simplex $a_0 = \left(1/n \right) e$, generates a sequence of iterates $x^{(0)}, x^{(1)}, \ldots, x^{(x)}, \ldots$ in the following steps:-

In brief:

Step 1: Initialization

Set $x^{(0)} =$ the centre of the simplex $a_0 = \left(1/n \right) e$

Step 2: Compute the next point $x^{(k+1)} = \Phi\left(x^{(k)} \right)$

Step 3: Check for feasibility

Step 4: Check for optimality

GOTO Step 1

In details:

Step 1: The function $b = \Phi(a)$, where $b = x^{(k+1)}$ and $a = x^{(k)}$ in step 2 above is defined by the following sequence of operations:

(i) let $D = diag(a_1, a_2, ..., a_n)$ where a_i is the i^{th} entry, $i = 1, 2, ...n$

(ii) let $B = \begin{pmatrix} AD \\ e^T \end{pmatrix}$ i.e augment the matrix AD with a row of all 1's. This is to guarantee that $\ker B \subseteq \Sigma = \{x | \sum x_i = 1, x \geq 0\}$

(iii) Compute the orthogonal projection of DC into the null space of B:

$$C_p = \left[I - B^T (BB^T)^{-1} B\right] DC$$

(iv) Normalize C_p :

$$\hat{C} = \frac{C_p}{\|C_p\|}$$

i.e, $\hat{C}$, is the unit vector in the direction of C_p

(v) Let $b' = a_0 - \alpha r \hat{C}$

i.e, take a step of length αr in the direction of $\hat{C}$, where r is the radius of the largest inscribed sphere.

$$r = \frac{1}{\sqrt{n(n-1)}}$$

where $\alpha \in (0,1)$; settable $\alpha = 1/4$

(vi) Apply inverse projective transformation to b'

i. e, $b = \dfrac{Db'}{e^T Db'}$

Step 2: check for feasibility

Karmarkar [3] defined the 'potential function' by $f(x) = \sum_{i=1}^{n} \ln \dfrac{c^T x}{x_i}$

At each iteration, a certain improvement δ in $f(x)$ is expected. The value of the expected improvement δ depends on the choice of parameter α (e.g Karmarkar [9] choice of $\alpha = 1/4$, gave $\delta = 1/8$). If the expected improvement is not observed i. e, if $f\left(x^{(k+1)}\right) > f\left(x^{(k)}\right) - \delta$, then we stop and conclude that the minimum value of the objective function must be strictly positive (as the canonical form of the problem is obtained by transformation on the standard LP problem), then the original problem does not have a finite optimal (i. e: it is either infeasible or unbounded).

Step 3: Check for optimality:

The check for optimality is carried out periodically. It involves going from the current point to an extreme point (without increasing the value of the objective function and then testing the extreme point for optimality. This is done if the time spent since the last check is greater than the time required for checking.

The current iterate $x^{(k)}$ will be the optimal solution if

$$\frac{c^T x^{(k)}}{c^T x^{(0)}} \leq 2^{-q}$$

where q is an arbitrary positive integer.

4. **AFFINE SCALING ALGORITHM**:

The affine-scaling algorithm is a variant of Karmarkar's projective interior point algorithm. It was first introduced by Dikin [2], and later rediscovered by Barnes [1], Vanderbei, Meketon and Freedman [6] after the publication of Karmarkar's algorithm.

The affine-scaling algorithm has the following advantages over the original Karmarkar's algorithm.

(i) It starts on the LP problem in standard form and assumes that a point x^0 is known such that $Ax^0 = b,\ x^0 \geq 0.$

(ii) It generates a monotonic decreasing sequence of the objective function values and the minimum of the objective function need not to be known in advance

Affine scaling is obtained by the direct application of the sleepest scaling descent (SSD) algorithm to the LP problem in standard form (3.2). It is very attractive due to its simplicity and its excellent performance in practice. Its performance is noted to be quiet sensitive to the starting points and like in any interior point algorithm, the computation work of the algorithm is concentrated on the projection operation needed in each of the iterations. Now, we let

$$f = \left\{ x \in R^n \middle| Ax = b.\ x \geq 0 \right\} \qquad \ldots (4.1)$$

be the feasible region for the primal LP problem (3.2). Given a strictly interior point x_0 in f. The affine scaling algorithm creates an ellipsoid with the centre at x_0 in f and optimizes the objective function $c^T x$ over it. If $x_0 \in f$ is a strictly interior point, then, the Dikin ellipsoid

$$Ex_0^{-1} = \left\{ x \in R^n \middle| Ax = b,\ (x - x_0)^T x_0^{-1} (x - x_0) \leq 1 \right\}$$

where $x_0^{-1} = \left(\frac{1}{x_1}, \frac{1}{x_2}, \ldots . \frac{1}{x_n} \right)$ and x_0^{-2} is the diagonal matrix whose entries are $1/x_1^2, \ldots\ 1/x_n^2$ The ellipsoid Ex_0 is contained in f. In the affine-scaling algorithm we obtain the affine-scaling direction at x_0 as the solution of the following direction finding problem:

$$\begin{aligned} &\min imize\ c^T d \\ &subject\ to\ Ad = 0 \qquad \ldots (4.2) \\ &\qquad d^2 x_0^2\ d \leq 1 \end{aligned}$$

The problem (4.2) is a convex programming problem with one convex quadratic constraint and it is solved by obtaining the Karush-Kuhn-Tucker conditions and then solving the associated linear system of equations. Let $d*$ be the solution to the direction-finding problem (4.2), then, $x* = x + \alpha\ d*,$ where α is chosen so that the new iterate $x*$ remains inside the feasible region f and ensures some improvement in the objective function $c^T x.$

The algorithm can be stated formally as follows:

$Let\ x^0 > 0$ that satisfies $Ax^0 = b$ given:

In general, if x^k is known, define

$D_k = diag\left(x_1^{\ k}, x_2^{\ k}, \ldots, x_n^{\ k} \right)$ and compute $X^{k+1} > 0$ by the formula

$$x^{k+1} = x^k - \frac{RD^2{}_k (C - A\lambda_k)}{\left\| D_k \left(C - A^T \lambda_K \right) \right\|} \qquad \ldots (4.3)$$

where $\lambda_k = AD_k^2 C \left(AD_k^2 A^T \right)^{-1}$

5. **Numerical Example:**

We solve the following LP problem by both Karmarkar and Affine scaling methods:

$$\begin{aligned} &\min imize \quad z = x_1 + x_2 \\ &subject\,to\; 2x_1 + x_2 \geq 4 \\ &\qquad\qquad x_1 + 7x_2 \geq 7 \\ &\qquad\qquad\; x_1,\, x_2 \geq 0 \end{aligned} \qquad \ldots (5.1)$$

Applying surplus variables to the LP problem (5.1), the LP problem becomes

$$\begin{aligned} &\min imize \quad z = x_1 + x_2 \\ &subject\,to\; 2x_1 + x_2 - x_3 = 4 \\ &\qquad\qquad x_1 + 7x_2 \quad - x_4 = 7 \\ &\qquad\qquad\; x_1,\, x_2, x_3,\, x_4 \geq 0 \end{aligned}$$

i.e

$$\min imize \quad z = x_1 + x_2$$

$$subject\,to \begin{pmatrix} 2 & 1 & -1 & 0 \\ 1 & 7 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \end{pmatrix} \qquad \ldots (5.2)$$

$$x_1,\, x_2,\, x_3,\, x_4 \geq$$

where $A = \begin{pmatrix} 2 & 1 & -1 & 0 \\ 1 & 7 & 0 & -1 \end{pmatrix}$

6. Concluding Remarks

Using the following starting point $x^0 = (2,\, 2,\, 2,\, 9)$ that satisfies the constraints $Ax^0 = b$ for both the affine scaling and Karmarkar's methods, for the affine scaling method, the objective function was obtained as follows:

Iteration	Values of the objective function
1	3.7
2	3.4
3	3.3
4	3.1

but up to the 4th iteration, convergence was yet to be achieved. However, the values of the objective function kept decreasing (i.e improving). For the same problem, Karmarkar at two iterations gave the following values for the objective function

Iteration	Values of the objective function
1	1.00
2	0.45

calculation of iterations in the above two methods were very tedious and cumbersome. However, the simplex method gave the most exact value of the objective function $z = \frac{32}{13} \approx 2.38.$

The above thus confirms the simplex method as the best solution method so far followed by the affine scaling method and the Karmarkar's method in that order, for the calculation of LP problems of smaller sizes.

REFERENCES

[1] Barnes, E. R: *Mathematical Programming*, 36, (**1996**) pp.174 – 182

[2] Dikin, I. I.: 'Iterative Solution of Problems of Linear and Quadratic Programming', Soviet Mathematics Doklady, 8, (**1967**) pp. 675 – 679

[3] Karmarkar, N.: 'A New Polynomial-time Algorithm for Linear Programming', Combinatoria; 4, (**1984**) pp. 373–95

[4] Khachian H. G: 'A New Polynomial-time Algorithm for Linear Programming', (Russian). Koklady Adademii Nauk SSSR, 244, (**1979**) pp. 1093 – 1096

[5] Shor, N. Z.: 'The Rate of Convergence of the Generalized Gradient Descent Method', Kibernetika 4(3), (1968), pp. 98 – 99 (Translated in Cybernetics, 4 (3), (**1968**), pp. 79 – 80)

[6] Vanderbei, R. J., Meketon, M. S. and Freedman B. A (**1986**). 'A Modification Karmarkar's Linear Programming Algorithm', Algorithmica 1, pp. 395 - 407

[7] Yudin, D. B. and Nemirovskii, A. S: 'Informational Complexity and Effective Methods of Solution for Convex Extremal Problems' Ekonomika i Matematicheskie Melody 12, (**1976**), pp. 357–369 (translated in Matekon: Translations of Russian and East European Math. Economics, 13 (**1976**), pp. 25 - 45