

SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK 14853

TECHNICAL REPORT NO. 707

July 1986

POLYNOMIAL ALGORITHMS FOR
LINEAR PROGRAMMING

By

Michael J. Todd*

*Research supported in part by NSF Grant ECS-8602534 and the Mathematical Sciences Institute through a grant from the Army Research Office.

TECHNICAL REPORT 707

INSERT

Published in Advances in Optimization and Control (H.A. Eiselt and G. Pederzoli, eds.) (1988). Berlin: Springer-Verlag, 49-66.

ABSTRACT

This paper contrasts the recent polynomial algorithms for linear programming of Khachian and Karmarkar. We show that each requires the solution of a weighted least-squares subproblem at every iteration. By comparing these subproblems we obtain further insights into the two methods.

1. Introduction

In this paper we contrast the ellipsoid method for linear inequalities, which was shown by Khachian [1979,1980] to provide a polynomial algorithm for linear programming, with the new projective algorithm for linear programming of Karmarkar [1984a,b]. Note that the ellipsoid method was developed for convex, not necessarily differentiable, optimization by Yudin and Nemirovsky [1977]--see the survey article of Bland, Goldfarb and Todd [1981].

Many authors have noted that these two algorithms use many similar concepts--ideas from nonlinear programming, geometric motivation, and infinite iterative schemes that can be truncated after a polynomial number of steps when applied to rational data, with an exact optimal solution then available by rounding. However, the details of the two algorithms seem very different. We will show that in fact the heart of each iteration of either algorithm is the solution of a weighted least-squares subproblem, and that these subproblems are very closely related. This viewpoint allows further insights into the two methods, in particular suggesting reasons for the very slow convergence of the ellipsoid method compared to the apparently very fast convergence of the projective algorithm.

The weighted least-squares subproblems have other important features. Both the ellipsoid and the projective algorithms appear at first sight not to provide solutions to the dual linear programming problem, but a closer examination shows that dual solutions are indeed generated during the course of the methods, essentially from the least-squares problems. Naturally, optimal dual solutions can be generated from optimal primal solutions at termination; however, approximate dual solutions at

intermediate stages are very useful in guaranteeing the quality of the current solution or in certifying infeasibility.

Section 2 describes the ellipsoid method from the viewpoint of Burrell and Todd [1985], showing how the weighted least-squares subproblems arise. Section 3 then outlines Karmarkar's algorithm, demonstrating that computation of the search direction once again requires the solution of a least-squares subproblem. In section 4 we contrast the two methods by comparing these subproblems and how they change from one iteration to the next.

In the rest of this introductory section we discuss the two methods informally. As we have noted above, for real data both algorithms are infinite iterative methods, for which it is possible to establish linear convergence rates. Such behavior is usually regarded as unacceptably slow in nonlinear programming, where under suitable conditions (quasi-) Newton methods achieve superlinear rates; however, this linear convergence rate is from the very first iteration, rather than asymptotic, and the convergence ratio can be bounded as a function of the dimensions alone, independent of the data.

Using this ratio enables one to establish polynomial bounds on the number of bit operations required to solve a linear programming problem. Suppose a problem has n variables, a number of constraints of the same order of magnitude, and input size L (i.e., L is the number of bits to specify the data of the problem). Then the ellipsoid method requires about $O(n^4 L^2)$ bit operations and Karmarkar's algorithm about $O(n^{3.5} L^2)$. Note that Karmarkar [1984a,b] quotes $O(n^6 L^2)$ for the ellipsoid method, from Khachian [1979], whereas the revised figure appears in Khachian [1980].

Also, the basic algorithm in Karmarkar [1984a,b] requires about $O(n^4 L^2)$ operations--this is the version we shall describe in section 3--whereas the $n^{.5}$ factor is removed by a modification that reduces the linear algebra necessary at each step. Thus there is only a slight advantage to Karmarkar's algorithm from the standpoint of theoretical bounds.

In practical performance, the differences becomes marked. The ellipsoid method appears to require a number of iterations close to its worst-case bound $O(n^2 L)$. On the other hand, a number of studies have established that many variants of Karmarkar's algorithm only take a number of iterations between 20 and 50 to get a very accurate solution, and this number appears to grow very slowly, if at all, with n . It is not our aim to discuss the computational significance of Karmarkar's algorithm, but at least for certain very large sparse problems exceptional results have been obtained. Besides the conference presentations of Karmarkar and Karmarkar and Sinha [1985], the best computational results have been given by Gill, Murray, Saunders, Tomlin and Wright [1985] and Adler, Karmarkar, Resende and Veiga [1986]. The latter paper reports times ranging from three times slower to eight times faster than the MINOS code on a set of medium to large problems.

In contrast to the overwhelming computational superiority of the projective algorithm over the ellipsoid method, its theoretical implications are far more limited. Grötschel, Lovász and Schrijver [1981,1986] and others have used the ellipsoid method to show that certain combinatorial optimization problems are in P (i.e., have polynomial algorithms) and others are NP-hard (i.e., are unlikely to have polynomial algorithms). A key to its use is the fact that the ellipsoid method does

not need to have all the constraints listed in advance--they can be generated as needed. Thus it is possible to handle problems with an exponential number of constraints. On the other hand, it appears that Karmarkar's algorithm requires all the constraints and variables to be explicitly present, and thus it cannot be used to solve such problems with our present knowledge.

2. The ellipsoid method

The preferred problem for the ellipsoid method is that of determining feasibility of a system of linear inequalities. To facilitate comparison with Karmarkar's algorithm, we assume that we seek a point in

$$Y = \{y \in \mathbb{R}^m: A^T y \leq c\} \quad (2.1)$$

where A is an $m \times n$ matrix and c an n -vector. We assume that a bound is known for all solutions to Y :

$$(A1) \text{ There is a known } y^0 \in \mathbb{R}^m \text{ and } R > 0 \text{ such that} \\ Y \subseteq S(y^0, R),$$

where $S(y^0, R)$ denotes the ball $\{y \in \mathbb{R}^m; \|y - y^0\| \leq R\}$ and $\|\cdot\|$ denotes the Euclidean norm.

The ellipsoid method generates a sequence of ellipsoids $\{E_k\}$ with centers $\{y^k\}$ such that

$$E_0 = S(y^0, R); \quad (2.2)$$

$$Y \subseteq E_k, \quad k \geq 0; \quad (2.3)$$

if $y^k \notin Y$, then

$$\text{vol } E_{k+1} / \text{vol } E_k < \exp(-1/2(m+1)). \quad (2.4)$$

Of course, if $y^k \in Y$ for some k , the algorithm stops; otherwise by (2.4) the ellipsoids generated have volumes shrinking geometrically to zero. If we further assume

- (A2) If $Y \neq \emptyset$, there is some $\hat{y} \in \mathbb{R}^m$, $\rho > 0$, such that
- $$S(\hat{y}, \rho) \subseteq Y,$$

then if the algorithm does not generate a feasible iterate y^k in $[2m(m+1)\ln(R/\rho)] + 1$ steps, we can stop and conclude that $Y = \emptyset$. This follows from (2.2)-(2.4); in this number of steps, the volume of E_k has shrunk from that of a ball of radius R to less than that of one of radius ρ , and by (A2) Y must be empty. Assumptions (A1) and (A2) imply that Y is bounded and has a nonempty interior; if (A1) is not satisfied, the algorithm cannot be started while if (A2) fails it will usually not terminate. If the data in A and c are integer with an input size L , then bounds can be added to the variables and the right-hand sides c increased a little so that the resulting polyhedron \hat{Y} satisfies (A1) and (A2). Moreover, Y will be nonempty iff \hat{Y} is for suitable bounds (or order 2^L) and perturbations of c (of order 2^{-L}). In this way, one can show that solving the feasibility problem for a polytope Y satisfying (A1) and (A2) in polynomial time allows a polynomial algorithm for general linear programming. For more details, see for example Bland, Goldfarb and Todd [1981].

The ellipsoid E_k is usually represented by its center y^k and a symmetric positive definite matrix B_k^{-1} :

$$E_k = \{y \in \mathbb{R}^m: (y-y^k)^T B_k^{-1} (y-y^k) \leq 1\}. \quad (2.5)$$

Ignoring finite precision, we can give the update formulae for E_{k+1} as follows. If $y^k \notin Y$, find some violated constraint $a^T y \leq \gamma$ in (2.1), so that $a^T y^k > \gamma$. Then

$$y^{k+1} = y^k - \tau_k B_k a / (a^T B_k a)^{1/2}, \quad (2.6)$$

$$B_{k+1} = \delta_k \left[B_k - \sigma_k \frac{B_k a a^T B_k}{a^T B_k a} \right]. \quad (2.7)$$

The simplest method to choose the scalar parameters is

$$\tau_k = 1/(m+1), \quad \sigma_k = 2/(m+1), \quad \delta_k = m^2/(m^2-1),$$

but different values depending on $a^T y^k - \gamma$ can give a greater decrease in the volume of E_{k+1} .

One of the disadvantages of this implementation of the ellipsoid method is that it provides no easy way to prove infeasibility. Usually, if $Y = \emptyset$, at a certain k we will have

$$\{y \in E_k: a^T y \leq \gamma\} = \emptyset, \quad (2.8)$$

where $a^T y \leq \gamma$ is one of the inequalities in (2.1). However, if round-off errors may have occurred during the algorithm, we cannot be certain that

(2.3) holds. A much more satisfactory way to demonstrate infeasibility would be to generate a vector $x \in \mathbb{R}^n$ with

$$Ax = 0, \quad c^T x < 0, \quad x \geq 0, \quad (2.9)$$

i.e., a solution to the alternative system in Farkas' lemma. Similarly, if we are using the ellipsoid method to solve

$$\min\{(a^0)^T y : y \in Y\}, \quad (2.10)$$

then we would like to use the value

$$\min\{(a^0)^T y : y \in E_k\} \quad (2.11)$$

to provide a lower bound guaranteeing the quality of a feasible solution. However, in the presence of round-off error, such a bound may not be valid. Linear programming duality provides a more satisfactory way to derive a bound: $-c^T x$ is a lower bound if $x \in \mathbb{R}^n$ satisfies

$$Ax = -a^0, \quad x \geq 0. \quad (2.12)$$

A different representation of each ellipsoid E_k due to Burrell and Todd [1985] avoids these drawbacks. At iteration k , we have a vector $\ell = \ell^k \in \mathbb{R}^n$ such that

$$Y = \{y \in \mathbb{R}^m : \ell \leq A^T y \leq c\}; \quad (2.13)$$

using (A1), we can easily obtain a suitable lower bound ℓ^0 initially. We also have a nonnegative diagonal $n \times n$ matrix $D = D_k$ such that ADA^T is positive definite. Then let

$$E = \{y \in \mathbb{R}^m: (A^T y - c)^T D (A^T y - \ell) \leq 0\}. \quad (2.14)$$

Since each component of $A^T y - c$ ($A^T y - \ell$) is nonpositive (nonnegative), we have a short proof that $Y \subseteq E$, in contrast to our earlier discussion. The inequality in (2.14) can be written as $y^T (ADA^T) y - y^T AD(c+\ell) + c^T D \ell \leq 0$. By "completing the square," we can express the set E as

$$E = \{y \in \mathbb{R}^m: (y-\bar{y})^T (ADA^T) (y-\bar{y}) \leq \bar{y}^T ADA^T \bar{y} - c^T D \ell\}$$

where \bar{y} is the solution to the system

$$(ADA^T)y = ADr \quad (2.15)$$

with $r = (c+\ell)/2$. Since ADA^T is positive definite, E is an ellipsoid with center \bar{y} . Note that (2.15) forms the normal equations for the weighted least squares problem

$$\min_y \|D^{1/2}(A^T y - r)\| \quad (2.16)$$

with $D^{1/2}$ the diagonal matrix whose diagonal entries are the square roots of those of D . Note that $D^{1/2}$ weights the components of the residual $A^T y - r$.

At each iteration, we check as before whether \bar{y} is feasible. If not, we choose some constraint $a^T y \leq \gamma$, say the j th constraint of (2.1), that is violated by \bar{y} . If (2.8) holds, then using this representation we can generate a vector x satisfying (2.9) and thereby proving infeasibility. Otherwise, we update our representation. This requires possibly updating the lower bound ℓ_j on $(A^T y)_j$, and then increasing the j th diagonal entry of D . Since D changes in such a simple way, the matrix ADA^T in (2.15) is modified by a rank-one matrix, and thus \bar{y} can be updated cheaply by a formula similar to (2.6). However, the precise formulae are complicated, and we refer to Burrell and Todd [1985] for details.

To conclude this section, we note that during the algorithm each component of ℓ , a lower bound on a component of $A^T y$ for all $y \in Y$, can be shown valid using duality; that is, $\ell_j = -c^T x^j$ for some $x^j \in \mathbb{R}^n$ satisfying

$$Ax^j = -a^j, \quad x \geq 0$$

if a^j is the j th column of A . Thus $-\ell_j$ is the objective value of a solution feasible in the dual of $\max\{(-a^j)^T y: y \in Y\}$. In particular, if we are solving the optimization problem (2.10), the vector x^0 is dual feasible to (2.10) and provides a certified lower bound on the optimal value. Details may be found in the cited paper.

3. Karmarkar's algorithm

The preferred form for the projective algorithm is a linear programming problem

$$\begin{aligned}
\min \quad & c^T x \\
& Ax = 0 \\
& e^T x = 1 \\
& x \geq 0,
\end{aligned} \tag{3.1}$$

where the following assumptions hold:

- (Ai) A is an $m \times n$ matrix of rank m ;
- (Aii) $e \in \mathbb{R}^n$ is a vector of ones and $Ae = 0$, so that $x^0 = e/n$ is feasible; and
- (Aiii) the optimal value v of (3.1) is zero.

Once again, these assumptions are without loss of generality in devising a polynomial algorithm for general linear programming problems. Karmarkar [1984a,b] describes two different ways to handle the latter; see also Todd and Burrell [1985] for an extension that permits the relaxation of (Aiii). This extension uses the dual problem of finding $y \in \mathbb{R}^m$, $z \in \mathbb{R}$ to

$$\begin{aligned}
\text{maximize} \quad & z \\
& A^T y + ez \leq c
\end{aligned} \tag{3.2}$$

to derive lower bounds on v .

Note that assumption (Aiii) implies that the optimal value of (3.2) is also zero; hence solving (3.2) amounts to finding y with $A^T y \leq c$, which is the feasibility problem considered in the last section.

Karmarkar's algorithm generates a sequence $\{x^k\}$ of strictly feasible (i.e., $x^k > 0$, componentwise) solutions to (3.1), with objective function values satisfying

$$c^T x^k \leq \exp(-k/5n) c^T x^0. \quad (3.3)$$

At each step, given a strictly feasible solution $\bar{x} = x^k$, so that

$$A\bar{x} = 0, \quad e^T \bar{x} = 1, \quad \bar{x} > 0, \quad (3.4)$$

we first make a projective transformation that takes \bar{x} into the center e/n of the simplex

$$\Delta = \{x \in \mathbb{R}^n: e^T x = 1, x \geq 0\}. \quad (3.5)$$

This transformation is defined using the diagonal matrix

$$\bar{X} = \text{diag}(\bar{x}) \quad (3.6)$$

by

$$T(x) = \hat{x} = \frac{\bar{X}^{-1} x}{e^T \bar{X}^{-1} x}, \quad (3.7)$$

$$T^{-1}(\hat{x}) = x = \frac{\bar{X} \hat{x}}{e^T \bar{X} \hat{x}}. \quad (3.8)$$

It is easy to see that T carries Δ into itself, the feasible region $\{x \in \Delta: Ax = 0\}$ into $\{\hat{x} \in \Delta: \hat{A} \hat{x} = 0\}$ where $\hat{A} = A\bar{X}$, and the linear objective function $c^T x$ into the fractional objective function $\hat{c}^T \hat{x} / \bar{c}^T \hat{x}$, where $\hat{c} = \bar{X}c$. Thus (3.1) is equivalent to

$$\begin{aligned} \min \quad & \hat{c}^T \hat{x} / \bar{c}^T \hat{x} \\ & \hat{A} \hat{x} = 0 \\ & e^T \hat{x} = 1 \\ & \hat{x} \geq 0. \end{aligned}$$

Because of assumption (Aiii), the optimal solution of this problem has $\hat{c}^T \hat{x} = 0$. We therefore consider the related linear programming problem

$$\begin{aligned} \min \quad & \hat{c}^T \hat{x} \\ & \hat{A} \hat{x} = 0 \\ & \mathbf{e}^T \hat{x} = 1 \\ & \hat{x} \geq 0. \end{aligned} \tag{3.9}$$

Thus we take a step based on this "equivalent" linear programming problem. Note that $T(\bar{x}) = \mathbf{e}/n$, which is as far from all the inequality constraints $\hat{x} \geq 0$ as possible. We therefore ignore these constraints, and take a step in the negative projected gradient direction,

$$\hat{d} = -(I - B^T(BB^T)^{-1}B)\hat{c}, \tag{3.10}$$

where

$$B = \begin{bmatrix} \hat{A} \\ \mathbf{e}^T \end{bmatrix} = \begin{bmatrix} A\bar{X} \\ \mathbf{e}^T \end{bmatrix}, \quad \hat{c} = \bar{X}c. \tag{3.11}$$

To compute \hat{d} , note that (3.4) implies

$$BB^T = \begin{bmatrix} A\bar{X}^2 A^T & 0 \\ 0 & n \end{bmatrix},$$

so that it is sufficient to find \bar{y} solving

$$(A\bar{X}^2 A^T)y = A\bar{X}^2 c, \tag{3.12}$$

which are the normal equations for the weighted least-squares problem

$$\min_y \|\bar{X}(A^T y - c)\|. \quad (3.13)$$

Having \bar{y} , we set $\bar{z} = c^T \bar{x}/n$ and then $\hat{d} = -\bar{X}(c - A^T \bar{y}) - e\bar{z}$.

We now replace $\hat{x} = e/n$ by $\hat{x} = e/n + \alpha \hat{d}/n\|\hat{d}\|$ for some suitable α ($\alpha \leq 1$ ensures $\hat{x} > 0$); the next iterate is then given by $x^{k+1} = T^{-1}(\hat{x}) = (x^k + \alpha \bar{X} \hat{d}/\|\hat{d}\|)/(1 + \alpha \bar{X}^T \hat{d}/\|\hat{d}\|)$, and the algorithm continues.

The convergence analysis uses the fact that, since the optimal value of (3.9) is zero, that of its relaxation

$$\begin{aligned} \min \quad & c^T \hat{x} \\ \text{s.t.} \quad & A \hat{x} = 0 \\ & e^T \hat{x} = 1 \\ & \hat{x} \in S(e/n, ((n-1)/n)^{1/2}) \end{aligned} \quad (3.14)$$

is nonpositive. But the optimal solution to (3.14) is $\hat{x} = e/n + ((n-1)/n)^{1/2} \hat{d}/\|\hat{d}\|$, and we deduce that

$$c^T (e/n + \alpha \hat{d}/n\|\hat{d}\|) \leq (1 - \alpha/n) c^T (e/n). \quad (3.15)$$

Unfortunately, this inequality does not yield the desired performance guarantee (3.3), for some fixed α , say $\alpha = 1$; the reason is that the linear objective functions $c^T x$ and $\hat{c}^T \hat{x}$ do not correspond, although they are both zero at optimal solutions. Karmarkar thus introduces the fundamental potential function

$$f(x; c) = \sum_j \ell n(c^T x / x_j) \quad (3.16)$$

which balances improvements in the objective function $c^T x$ with distances x_j from the inequality constraints. The crucial property of f is that it is invariant under projective transformations, or more precisely,

$$f(T(x); \hat{c}) = f(x; c) + \ell n \det \bar{X}. \quad (3.17)$$

Thus a decrease in the transformed potential function $\hat{f} = f(\cdot; \hat{c})$ assures an equal decrease in f in the original space. Karmarkar shows that a fixed decrease of $1/7$ at each iteration is possible by choosing $\alpha \approx 1/4$. Various authors have given improvements--for example Todd and Burrell [1985] show that $\alpha = 1/3$ guarantees a decrease of $1/5$ in f ; but in practice a line search to approximately minimize f seems advisable. In any case, since $\sum_j \ell n(1/x_j)$ is minimized for $x \in \Delta$ by $x^0 = e/n$, a decrease of $1/5$ in f at each iteration easily implies the key inequality (3.3), and hence provides a polynomial algorithm.

We conclude this section by describing briefly two extensions of Karmarkar's algorithm. First, as noted above, Todd and Burrell [1985] have shown how the assumption (Aiii) on the optimal value of (3.1) can be relaxed. The basic idea is to replace c by $c - ez^k$ at the k th iteration, where z^k is a lower bound on the optimal value of (3.1) given by duality. Then \hat{c} is replaced by $\bar{X}(c - ez^k)$ in (3.9) and (3.14). Note that (3.14) provides a computable lower bound for the optimal value of (3.9), and that the optimal values of (3.9) and (3.14) are unchanged if we replace \hat{c} by

$$\hat{c}_p = (I - \hat{A}^T (\hat{A} \hat{A}^T)^{-1} \hat{A}) \hat{c}. \quad (3.18)$$

A better bound on the optimal value of (3.9) is provided by

$$\begin{aligned} \min \quad & \hat{c}^T \hat{x} \\ & \hat{e}^T \hat{x} = 1 \\ & \hat{x} \geq 0. \end{aligned} \tag{3.14159}$$

(Since it provides a better lower bound on an interesting quantity than does (3.14), we cannot resist labelling this problem (3.14159).) The value of z^k is then adjusted to z^{k+1} if necessary so that the value of (3.14159) is nonpositive. The search direction is then computed using $c - ez^{k+1}$. This extension also provides dual solutions: if y^{k+1} solves

$$(\overline{A}^2)^T y = \overline{A}^2 (c - ez^{k+1}), \tag{3.19}$$

then (y^{k+1}, z^{k+1}) is feasible in (3.2) and its objective function value z^{k+1} converges to the optimal value. In fact, the duality gap $c^T x^k - z^k$ converges linearly to zero as in (3.3).

Finally, we note that several researchers (Anstreicher [1985], Gay [1985], Gonzaga [1985] and Jensen and Steger, see Steger [1985]) have used these ideas to develop an efficient algorithm for the standard form problem

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & x \geq 0. \end{aligned} \tag{3.20}$$

4. Comparison

The key similarity between the two algorithms that emerges from the analysis above is that in both each iteration requires solving the normal equations (2.15), (3.12) or equivalently the weighted least-squares subproblems (2.16), (3.13). The solutions to these subproblems give respectively the center $\bar{y} = y^k$ of the current ellipsoid and the vector \bar{y} that determines the search direction \hat{d} . Moreover, in the extension of Karmarkar's algorithm by Todd and Burrell [1985], the solution to a related subproblem (see (3.19)) gives a vector y^{k+1} that is part of a feasible dual solution which converges to the optimal dual solution if it is unique.

The weighted least-squares subproblems differ in their right-hand sides-- $r = (c+\ell)/2$ in the ellipsoid method, and c or $c-ez$ for some z in Karmarkar's--and especially in their diagonal weighting matrices-- $D^{1/2}$ in the ellipsoid method and \bar{X} in Karmarkar's projective algorithm. It seems impossible to compare the two subproblems meaningfully at any particular iteration, but it is very instructive to see how the subproblems change from one iteration to the next in each algorithm.

In the ellipsoid method, each iteration possibly changes one component of the lower bound vector ℓ and hence of the right-hand side r and increases one component of the diagonal of the weighting matrix $D^{1/2}$. The advantage of such a simple modification is that each subproblem can be trivially solved--we can update $B = (ADA^T)^{-1}$ or a Cholesky factorization of B or of ADA^T in $O(m^2)$ real operations. The disadvantage is that convergence is very slow--theoretically $O(m^2L)$ iterations are necessary, and in practice $O(m)$ are required for a constant factor decrease in the volume of E_k , and about $O(m^2)$ for a constant factor decrease in its diameter. One reason for this slow convergence can be seen from a

different viewpoint. Suppose ℓ remains fixed. Then we can forget the centers y^k and the ellipsoids E_k , and consider just the volume of E_k as a nonlinear function of the vector d (the diagonal entries of D) which we want to minimize. The ellipsoid method changes one component at a time; in other words, it is a coordinate descent method, and such methods are known to be very slow for unconstrained optimization problems. This insight suggests extensions of the ellipsoid method that permit several components of d to change simultaneously.

In Karmarkar's projective algorithm, the right-hand side is always c (in the extensions, $c - ze$ for varying z) but the weighting matrix \bar{X} changes in all its diagonal entries from one iteration to the next. In this case, each least squares subproblem must be solved again from scratch, requiring (for a dense problem) $O(n^3)$ real operations. (Karmarkar [1984a,b] has proposed a version in which, apart from a common scale factor, only on average $n^{1/2}$ diagonal entries of \bar{X} change; this allows the corresponding improvement in the worst-case bound. We will not discuss this variant further.) To compensate for this additional work per iteration, the number of iterations seems to be very modest, fewer than 50 being typical.

To conclude the paper, we discuss briefly two rationales for this very attractive behavior, which is not explained by the theoretical linear convergence rate or $O(nL)$ theoretical bound. First, Gill, Murray, Saunders, Tomlin and Wright [1985] have shown that Karmarkar's algorithm is very closely related to a projected Newton barrier method. One might therefore hope that it would converge like other Newton methods at a typically quadratic rate in the limit.

Second, we can view Karmarkar's algorithm as designed to "minimize" (more accurately, reduce, since it is unbounded below) the potential function $f = f(\cdot; c)$ subject to the equality constraints $Ax = b$ and implicitly $x > 0$. We are therefore interested in the gradient

$$\nabla f(\bar{x}) = nc/c^T \bar{x} - \bar{X}^{-1} e \quad (4.1)$$

where $\bar{X} = \text{diag}(\bar{x})$, and the Hessian matrix

$$\nabla^2 f(\bar{x}) = \bar{X}^{-1} (I - \hat{c} \hat{c}^T / (\hat{c}^T e)^2) \bar{X}^{-1} \quad (4.2)$$

where $\hat{c} = \bar{X}c$. Let us ignore temporarily the equality constraints. It is easy to see that $-\nabla f(x^0) = -\nabla f(e/n)$ is precisely the direction \hat{d} chosen at the first direction in Karmarkar's algorithm. On the other hand, for any \bar{x} , $\nabla^2 f(\bar{x}) \bar{x} = -\nabla f(\bar{x})$, so if the Hessian matrix is nonsingular, \bar{x} is the Newton direction. This is a singularly inappropriate direction in which to move, however; since f is positively homogeneous of degree zero, it is constant in the direction \bar{x} . Note that usually $I - \hat{c} \hat{c}^T / (\hat{c}^T e)^2$ has $n-1$ positive eigenvalues equal to 1, and one negative one equal to $1 - \hat{c}^T \hat{c} / (\hat{c}^T e)^2$, corresponding to the eigenvector \hat{c} . (Since $e^T e = n$, the last eigenvalue is nonpositive and zero only if \hat{c} is proportional to e , in which case (Aiii) implies all feasible solutions are optimal.) Hence $\nabla^2 f(\bar{x})$ also has $n-1$ positive and one negative eigenvalue. Thus a good second-order direction would be the eigenvector corresponding to the negative eigenvalue.

This eigenvector is in general hard to compute; but if $\bar{x} = e/n$, it is proportional to $\hat{c} = nc$. Because f is homogeneous, the direction $-nc$ is equivalent to the direction $-\nabla f(e/n) = -c/c^T e + e$; hence:

$$\text{At } \bar{x} = e/n, \text{ good first- and second-order directions coincide.} \quad (4.3)$$

If we include the equality constraints $Ax = b$, the statement (4.3) remains valid; now the directions are $-P\nabla f(e/n)$ and an eigenvector of $P\nabla^2 f(e/n)P$ in the range of P with a most negative eigenvalue, where P is the orthogonal projection onto the null space of A . Now f is invariant under projective transformations, while neither the gradient nor the eigenvector are. Hence it is natural to make such a transformation at each iteration to cause these directions to coincide, and by (4.3) Karmarkar's projective transformation does the trick. Thus Karmarkar's algorithm, which we motivated by first-order arguments, chooses a direction that is very natural also using second-order reasoning. Again, this viewpoint suggests that the projective method is likely to perform far better than its linear convergence rate promises, and such behavior seems to occur in practice.

References

- I. Adler, N.K. Karmarkar, M.G.C. Resende and G. Veiga (1986), "An implementation of Karmarkar's algorithm for linear programming," manuscript, Department of Industrial Engineering and Operations Research, University of California, Berkeley, California.
- K.M. Anstreicher (1985), "Analysis of Karmarkar's algorithm for fractional linear programming," manuscript, School of Organization and Management, Yale University, New Haven, Connecticut.
- R.G. Bland, D. Goldfarb and M.J. Todd (1981), "The ellipsoid method: a survey," Operations Research 29, 1039-1091.
- B.P. Burrell and M.J. Todd (1985), "The ellipsoid method generates dual variables," Mathematics of Operations Research 10, 688-700.
- D. Gay (1985), "A variant of Karmarkar's linear programming algorithm for problems in standard form," Numerical Analysis Manuscript 85-10, AT&T Bell Laboratories, Murray Hill, New Jersey.
- P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright (1985), "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method," Technical Report SOL85-11, Department of Operations Research, Stanford University, California, to appear in Mathematical Programming.
- C. Gonzaga (1985), "A conical projection algorithm for linear programming," manuscript, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California.
- "
M. Grötschel, L. Lovász and A. Schrijver (1981), "The ellipsoid method and its consequences in combinatorial optimization," Combinatorica 1, 169-197.
- "
M. Grötschel, L. Lovász and A. Schrijver (1986), The Ellipsoid Method and Combinatorial Optimization, Springer-Verlag, Berlin, to appear.
- N.K. Karmarkar (1984a), "A new polynomial-time algorithm for linear programming," Proceedings of the 16th Annual ACM Symposium on the Theory of Computing, 302-311.
- N.K. Karmarkar (1984b), "A new polynomial-time algorithm for linear programming," Combinatorica 4, 373-395.
- N.K. Karmarkar (1985), "Further developments in the new polynomial-time algorithm for linear programming," presented at the 12th International Symposium on Mathematical Programming, Boston, Massachusetts.
- N.K. Karmarkar and L.P. Sinha (1985), "Application of Karmarkar's algorithm to overseas telecommunications facilities planning," presented at the 12th International Symposium on Mathematical Programming, Boston, Massachusetts.

- L.G. Khachian (1979), "A polynomial algorithm in linear programming," Soviet Mathematics Doklady 20, 191-194.
- L.G. Khachian (1980), "Polynomial algorithms in linear programming," USSR Computational Mathematics and Mathematical Physics 20, 53-72.
- A. Steger (1985), "An extension of Karmarkar's algorithm for bounded linear programming problems," M.S. Thesis, SUNY at Stonybrook, New York.
- M.J. Todd and B.P. Burrell (1985), "An extension of Karmarkar's algorithm for linear programming using dual variables," Technical Report 648, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.
- D.B. Yudin and A.S. Nemirovsky (1977), "Informational complexity and effective methods of solution for convex extremal problems," Matekon 13, 24-45.