SIAM J. NUMER. ANAL.
Vol. 24, No. 4, August 1987

© 1987 Society for Industrial and Applied Mathematics
013

CONVERGENCE IN KARMARKAR'S ALGORITHM FOR LINEAR PROGRAMMING*

JOEL FRANKLIN†

Abstract. Karmarkar's algorithm is formulated so as to avoid the possibility of failure because of unbounded solutions. A general inequality gives an easy proof of the convergence of the iterations. It is shown that the parameter value $\alpha = 0.5$ more than doubles the originally predicted rate of convergence. To go from the last iterate to an exact optimal solution, an $O(n^3)$ termination algorithm is prescribed. If the data have maximum bit length independent of n, the composite algorithm is shown to have complexity $O(n^{4.5} \log n)$.

Key words. linear programming, Karmarkar's algorithm, projective-iterative method

AMS(MOS) subject classifications. 90C05, 90C06

1. Introduction. This is an expository article. It is not intended to describe an efficient implementation of Karmarkar's algorithm, but rather to give a self-contained mathematical description of its structure.

We will give a simple proof of the convergence of Karmarkar's algorithm [10] for linear programming. We will prove that for large problems the best choice for his parameter alpha is not $\alpha = 0.25$ but $\alpha = 0.5$. The choice $\alpha = 0.5$ more than doubles the rate of convergence. This point has been made by Blair [3] and Padberg [13]; see also Anstreicher [1], Goldfarb and Mehrotra [6] and Kalantari [9].

In § 2 we state an elementary inequality that provides the results on convergence. The proof of the inequality is given in the Appendix.

In § 3 we respond to the criticism by Michael Todd [15] that Karmarkar's algorithm may fail in certain cases. We show how to introduce an additional constraint that removes the possibility of failure.

In § 4 we give a self-contained description of Karmarkar's algorithm.

In § 5 we prove our theorem on the rate of convergence of Karmarkar's algorithm, and we derive an optimal value for Karmarkar's parameter α.

In § 6 we present an $O(n^3)$ *termination algorithm.* This algorithm starts with the last Karmarkar iterate and it ends with an exact basic optimal solution. The algorithm depends on the numerically stable QR orthogonalization and it is faster than the well-known $O(n^4)$ termination algorithms.

In § 7 we estimate the complexity $T(n)$ of the composite modified algorithm, which produces an exact optimal solution in a finite number of steps. In his paper [10], Karmarkar estimates $T(n) = O(n^{3.5}L)$ if L is the total number of bits in the input. If all the data are integers with absolute value $<2^s$, where s is independent of n, then the number L may tend to infinity as fast as n^2 and Karmarkar's estimate becomes $T(n) = O(n^{5.5})$. We prove that the complexity is smaller: $T(n) = O(n^{4.5} \log n)$.

At any rate, Karmarkar's algorithm surely works in polynomial time. By contrast, as Klee and Minty showed in [12], the simplex algorithm may require exponential time in ingeniously contrived examples. Khachian discovered the first algorithm [11] that solves linear programming in polynomial time. Khachian's algorithm seems to

* Received by the editors April 14, 1986; accepted for publication (in revised form) September 25, 1986.
† Applied Mathematics 217-50, Firestone Laboratory, California Institute of Technology, Pasadena, California 91125.

work badly in practice, while the simplex algorithm [5] continues to work well. Only years of computing experience can show us the value of Karmarkar's algorithm. Eventually, it may become a preferred method for very large-scale linear programming.

2. An inequality. We consider a plane and a sphere:

$$x_1+\cdots+x_n=1, \qquad x_1^2+\cdots+x_n^2=\rho^2. \tag{1}$$

Call the intersection S. Assume $n\geqq 3$.

If $\rho^2<1/n$, the set S is empty. If $\rho^2=1/n$, the set S is the single point with all $x_i=1/n$. If $\rho^2>1/n$, the set S is a sphere of dimension $n-2$.

If $\rho^2=1/(n-1)$, the sphere S contains n points with zero coordinates. If $\rho^2>1/(n-1)$, the sphere S contains points with negative coordinates.

If $1/n<\rho^2<1/(n-1)$, the sphere S contains only points x with positive coordinates, $x_i>0$. In this case we will write

$$\rho^2=\frac{1}{n}+\frac{\alpha^2}{n(n-1)} \qquad (0<\alpha<1). \tag{2}$$

Then S is the $(n-2)$-dimensional sphere described by the equations

$$x_1+\cdots+x_n=1, \qquad \left(x_1-\frac{1}{n}\right)^2+\cdots+\left(x_n-\frac{1}{n}\right)^2=\alpha^2 r^2 \quad (0<\alpha<1), \tag{3}$$

where $r^2=1/(n(n-1))$. The parameter α governs the convergence of Karmarkar's algorithm. From now on we will suppose $0<\alpha<1$, or $1/n<\rho^2<1/(n-1)$.

We will use those points on S for which one coordinate x_i takes one value, u, and all the other coordinates x_j take another value, v. The two values u and v must satisfy the equations

$$u+(n-1)v=1, \qquad u^2+(n-1)v^2=\rho^2. \tag{4}$$

If we eliminate u and solve the quadratic equation for v, we get the solution pairs

$$u_1=\frac{1}{n}(1-\alpha), \qquad v_1=\frac{1}{n}\left(1+\frac{\alpha}{n-1}\right), \tag{5}$$

$$u_2=\frac{1}{n}(1+\alpha), \qquad v_2=\frac{1}{n}\left(1-\frac{\alpha}{n-1}\right) \tag{6}$$

where $0<\alpha<1$, according to the definition (2).

THEOREM 1. *For $n\geqq 3$ we define the sphere S by* (1), *where $n^{-1}<\rho^2<(n-1)^{-1}$. Assume that a given function $g(t)$ has a convex derivative for $0<t<2/n$. Then, if the point x has coordinates x_i,*

$$\min_{x\in S}\sum_{i=1}^{n} g(x_i)=g(u_1)+(n-1)g(v_1), \tag{7}$$

and

$$\max_{x\in S}\sum_{i=1}^{n} g(x_i)=g(u_2)+(n-1)g(v_2), \tag{8}$$

where the pairs u_i, v_i are defined by (5) *and* (6).

The proof of the theorem appears in the Appendix.

Example: *an inequality.* In our analysis of Karmarkar's algorithm, we will set $g(t) = \ln t$. Then formula (7) yields this inequality on S:

$$\sum_{i=1}^{n} \ln x_i \geqq \ln\left(\frac{1}{n} - \frac{\alpha}{n}\right) + (n-1)\ln\left(\frac{1}{n} + \frac{\alpha}{n(n-1)}\right). \tag{9}$$

Here $0 < \alpha < 1$, and the coordinates $x_1, \cdots, x_n$, satisfy

$$\sum x_i = 1, \qquad \sum\left(x_i - \frac{1}{n}\right)^2 = \frac{\alpha^2}{n(n-1)}.$$

3. Linear programming in projective coordinates. We begin with a linear program in inequality form:

$$Ax \geqq b, \qquad x \geqq 0, \qquad c^T x = \min. \tag{1}$$

Here A is an $m \times n$ matrix. *We will assume that b and c are not both zero.* We will make no assumption about the rank of A. For instance, as the extreme case of degeneracy, we could let A be the zero matrix.

The dual program is

$$y^T A \leqq c^T, \qquad y \geqq 0, \qquad b^T y = \max. \tag{2}$$

The duality theorem of linear programming states that the problem (1) has an optimal solution x if and only if the dual problem (2) has an optimal solution y, with

$$c^T x = b^T y. \tag{3}$$

We restate the optimality condition (3) and the feasibility conditions in the primal and dual problems as follows:

$$c^T x - b^T y = 0, \qquad -Ax + u = -b, \qquad A^T y + v = c, \qquad x, y, u, v \geqq 0. \tag{4}$$

Here we have introduced slack vectors u and v. The original problem (1) has an optimal solution if and only if the problem (4) has a solution (a *feasible* solution).

In his article [15] Michael Todd points out that Karmarkar's algorithm might fail if the program has an unbounded set of solutions. But, as Papadimitriou and Steiglitz point out in their book [14], it is easy to give a bound for the *basic* feasible solutions; and we know that a program has a feasible solution only if it has a *basic* feasible solution.

In the program (4) suppose that all basic feasible solutions must have components with sum less than or equal to the bound β; see formula (21). Then we append to (4) the equation

$$\sum_{1}^{n} x_j + \sum_{1}^{m} y_i + \sum_{1}^{m} u_i + \sum_{1}^{n} v_j + \omega = \beta, \tag{5}$$

where $\omega \geqq 0$ is a new slack variable. The augmented program (4), (5) cannot have an unbounded feasible set and so Karmarkar's algorithm must work for the augmented program.

Let z be the vector with the components $x_j, y_i, u_i, v_j, \omega$, so z has $2(m+n)+1$ components. Let g be the vector whose $1+m+n+1$ components appear on the right-hand sides of the equations in (4) and (5). Then the problem (4), (5) has the form

$$Mz = g, \qquad z \geqq 0, \tag{6}$$

where M is the composite matrix

$$M = \begin{bmatrix} c^T & -b^T & 0 & 0 & 0 \\ -A & 0 & I_m & 0 & 0 \\ 0 & A^T & 0 & I_n & 0 \\ 1 \cdots 1 & 1 \cdots 1 & 1 \cdots 1 & 1 \cdots 1 & 1 \end{bmatrix}. \tag{7}$$

This matrix has $m+n+2$ rows and $2(m+n)+1$ columns. Because we have assumed that c and b are not both zero, *the rows of M are linearly independent.* We shall need this independence to implement Karmarkar's algorithm.

We have proved that the original linear program (1) *has an optimal solution if and only if the problem* $Mz = g$, $z \geqq 0$, *has a feasible solution. The matrix M has independent rows, and the set of feasible solutions z is either empty or bounded.*

We will convert the feasibility problem (6) into an optimality problem with a known feasible solution. Define $z_j^0 = 1$ for all j. Introduce a new unknown λ and consider the problem

$$\lambda(g - Mz^0) + Mz = g, \quad \lambda \geqq 0, \quad z \geqq 0, \quad \lambda = \min. \tag{8}$$

This problem has an optimal solution with $\min \lambda = 0$ if and only if the problem (6) has a feasible solution z. Also, the problem (8) has the feasible solution

$$\lambda = 1, \qquad z = z^0. \tag{9}$$

The new problem (8) involves a matrix M_1 whose first column is $g - Mz^0$ and whose remaining columns are the columns of M. Since the rows of M are independent, we have rank $M_1 =$ rank $M = m+n+2$.

Finally, we introduce homogeneous coordinates and map the unbounded orthant $\lambda \geqq 0$, $z \geqq 0$ into a simplex, which is bounded. If we replace the coordinates λ and z_j by λ/μ and z_j/μ, we get the homogeneous system

$$\lambda(g - Mz^0) + Mz - \mu g = 0, \quad \lambda \geqq 0, \quad z \geqq 0, \quad \mu \geqq 0. \tag{10}$$

Now we normalize the coordinates by requiring

$$\lambda + \sum z_j + \mu = 1 \tag{11}$$

and once again we state the optimality condition

$$\lambda = \min. \tag{12}$$

We will use the letter P to designate the linear programming problem defined by formulas (10), (11), (12).

The problem P has $2m+2n+3$ unknowns: $\lambda, z_1, \cdots, z_{2m+2n+1}, \mu$. *We get a first feasible solution by setting all these unknowns equal to* $1/N$, *where* $N = 2m+2n+3$.

We now assert that the problem P has the optimal value $\lambda = 0$ if and only if the problem (8) has optimal value $\lambda = 0$. This we prove as follows.

Suppose $\lambda = 0$ in (8). Then we get an optimal solution to problem P by assigning the values

$$z_j := z_j/(1+\sum z_j), \qquad \mu := 1/(1+\sum z_j), \tag{13}$$

with $\min \lambda = 0$ in (12).

Conversely, suppose $\lambda = 0$ in problem P. Then formulas (10) and (11) become

$$Mz - \mu g = 0, \quad z \geqq 0, \quad \mu \geqq 0, \tag{14}$$

$$\sum z_j + \mu = 1. \tag{15}$$

Can $\mu = 0$? No, for then we should have

$$Mz = 0, \qquad z \geqq 0, \qquad \sum z_j = 1. \tag{16}$$

This we have made impossible by introducing the last row of 1's in the definition (7) of the matrix M. (We must introduce that last row; otherwise, Todd's criticism [15] is valid.)

Let us summarize the work of this section. We begin with a linear program in inequality form

$$Ax \geqq b, \qquad x \geqq 0, \qquad c^T x = \min. \tag{17}$$

We assume only that the vectors b and c are not both zero. We relate this program to the problem P in formulas (10), (11), (12). This problem has the form

$$\bar{A}\bar{x} = 0, \quad \sum \bar{x}_j = 1, \quad \bar{x} \geqq 0, \quad \bar{c}^T \bar{x} = \min. \tag{18}$$

In this problem the rows of the matrix $\bar{A}$ are linearly independent, and $\bar{c}^T = (1, 0, \cdots, 0)$. We know that a vector $\bar{x}^0$ with identical coordinates is feasible. We know that the set of feasible solutions is bounded and we know that all feasible $\bar{x}$ satisfy $\bar{c}^T \bar{x} \geqq 0$. Also, we know that $\min \bar{c}^T \bar{x} = 0$ if and only if the original problem (17) has an optimal solution x. In this case the components of x and of the optimal dual solution, y, are found by dividing the corresponding components of $\bar{x}$ by its last component, μ, which must be positive. Then, of course, the minimum cost is found by computing $c^T x$ or $b^T y$.

A note on the bound β. We made sure that μ was positive by introducing a bound β in (5). Let us give an explicit value for β. Suppose the data are scaled and rounded so that all the coefficients a_{ij}, b_i, c_j are integers. Let us consider any basic solution of (4). If Δ is the determinant of the basis, it must satisfy an inequality

$$1 \leqq |\Delta| \leqq l_1 l_2 \cdots l_{m+n+1}, \tag{19}$$

where $l_1, l_2, \cdots$, are the lengths of the columns in the basis. Every positive component of a basic solution of (4) is a quotient Δ'/Δ, where the determinant Δ' also satisfies a Hadamard inequality. Let $l_{\max}$ be the greatest of the column lengths in the system (4), including the right-hand column. Then every component of a basic solution has absolute value

$$|\Delta'/\Delta| \leqq l_{\max}^{m+n+1} \tag{20}$$

and so the sum of the positive basic components has the upper bound

$$\beta = (m+n+1) l_{\max}^{m+n+1}. \tag{21}$$

This bound may be a very large number, though it is smaller than the bound given by Papadimitriou and Steiglitz [14, p. 30]. If β is too large, its use in (5) may cause round-off errors in single-precision calculations. But usually (5) is superfluous, because most linear programs have unique optimal primal and dual solutions. We seldom have to worry that the sets of optimal solutions may be unbounded.

4. Karmarkar's algorithm. We consider a linear programming problem that has been reduced to projective form:

$$Ax = 0, \quad \sum x_j = 1, \quad x \geqq 0, \quad c^T x = \text{minimum}. \tag{1}$$

Here A is an $m \times n$ matrix with rank m and we know that $c^T x \geqq 0$ for all feasible x. We are given a first feasible solution x^0 with components $x_j^0 = 1/n$ $(j = 1, \cdots, n)$. We may suppose $c^T x^0 > 0$.

If $\min c^T x = 0$ in (1), the purpose of the algorithm is to construct a feasible solution $x^{(k)}$ satisfying

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

where q is a given positive integer. The algorithm prescribes an upper bound for the number of iterations, k, required to achieve the inequality (2). If this inequality fails after k iterations, the algorithm deduces that $\min c^T x$ is positive.

In § 5 we will discuss how to prescribe an upper bound for k. In the present section we will simply call the prescribed upper bound k max. In § 6 we will briefly discuss how to get an exact optimal solution x^* from an approximate optimal solution $x^{(k)}$.

The algorithm uses a parameter α, with $0 < \alpha < 1$. In § 5 we will discuss how to choose α.

The algorithm will compute and use certain matrices

$$B = \begin{bmatrix} AD \\ e^T \end{bmatrix}. \tag{3}$$

Here A is the given $m \times n$ matrix with rank m; D is a diagonal matrix with n positive components; and e^T is a row of n 1's. A is fixed, but the matrix D varies from one iteration to the next, namely,

$$D = D^{(k)} = \operatorname{diag}(x_1^{(k)}, \cdots, x_n^{(k)}). \tag{4}$$

Thus, the kth iterate $x^{(k)}$ equals $D^{(k)}e$.

The matrix AD has rank m because A has rank m and D has an inverse. The m rows of AD are independent of e^T, because $x = x^{(k)}$ satisfies

$$e^T x = 1, \qquad ADx = 0.$$

Therefore, the $m+1$ rows of B are linearly independent. Therefore, *the matrix BB^T has an inverse.* This is true for each $B = B^{(k)}$ $(k = 0, 1, 2, \cdots)$.

Now we will show how to go from $x := x^{(k)}$ to $x := x^{(k+1)}$. We start with $x^{(0)} = n^{-1}e \equiv a$.

(i) Form the diagonal matrix

$$D = \operatorname{diag}(x_1, x_2, \cdots, x_n).$$

(ii) Set

$$B = \begin{bmatrix} AD \\ e^T \end{bmatrix}.$$

(iii) Define y to be the solution of the problem

$$By = 0, \qquad y_1^2 + \cdots + y_n^2 = (\alpha r)^2, \qquad c^T D y = \text{maximum}. \tag{5}$$

Here r is the distance from the point a with all $a_i = 1/n$ to the boundary of the simplex $x \geqq 0$, $\sum x_j = 1$; namely, $r = (n(n-1))^{-1/2}$. The parameter α is fixed between 0 and 1; say $\alpha = 0.25$ or, as we shall recommend, $\alpha = 0.5$.

(iv) Set $z = a - y$ and now define the new iterate $x := x^{(k+1)}$ by the projective transformation

$$x_i^{(k+1)} = \frac{d_{ii} z_i}{\sum_{j=1}^n d_{jj} z_j} \qquad (i = 1, \cdots, n). \tag{6}$$

In step (iii) the solution is found by using Lagrange multipliers. For some scalar λ and vector u we require

$$Dc = \lambda y + B^T u. \tag{7}$$

Since $By = 0$ we can eliminate u to obtain

$$\lambda y = [I - B^T(BB^T)^{-1}B]Dc. \tag{8}$$

Now we choose the positive value λ that makes $\sum y_i^2 = \alpha^2 r^2$.

This completes the description of the iterative transformation $x^{(k)} \mapsto x^{(k+1)}$. For efficient implementation, much more is required than to write the formulas. We will not discuss efficient implementation in this paper.

The iterations stop when $x = x^{(k)}$ meets the required convergence check

$$c^T x \leqq 2^{-q}(c^T x^0) \tag{9}$$

or when $k = k\max$. If k reaches $k\max$ before the convergence check is passed, we may deduce that the linear program (1) has no solution with $\min c^T x = 0$. In § 5, formula (37), we will show that if $\alpha = 0.5$, we may define $k\max = \lceil 2.2589nq \rceil$.

This is a better result than Karmarkar's Theorem 1 [10, p. 379], which states: "In $O(n(q+\log n))$ steps the algorithm finds a feasible point x such that $c^T x \leqq 2^{-q} c^T a_0$." (His a_0 equals our x^0.)

Speed of computation. Most of the computing time is spent in formula (8) evaluating y. Each evaluation requires $O(n^3)$ arithmetic operations. We shall require at most $k\max$ evaluations.

We will show that we may choose α to make $k\max = O(nq)$, yielding a cumulative number $O(n^4 q)$. In Karmarkar's modified algorithm, we can reduce this number to $O(n^{3.5}q)$. Our improved value for α will apply both to the original form and to the modified form of Karmarkar's algorithm.

We have written $O(n^{3.5}q)$ instead of $O(n^{3.5})$ for a reason: *the integer q may have to depend on n.* We will consider this point in § 7.

5. Convergence of the iterations. Following Karmarkar, we discuss the problem in the form

$$Ax = 0, \quad \sum_{j=1}^{n} x_j = 1, \quad x \geqq 0, \quad c^T x = \min. \tag{1}$$

We have an initial feasible solution x^0 at the center of the simplex $\sum x_j = 1, x \geqq 0$:

$$x_j^0 = a_j = \frac{1}{n} \quad (j = 1, \cdots, n). \tag{2}$$

We assume that $c^T x^0 > 0$ and that $c^T x^* = 0$ for a basic optimal solution x^*.

If $x^{(k)}$ is the kth iterate, we form $x^{(k+1)}$ as follows. We form the diagonal matrix D with positive diagonal elements

$$d_{ii} = x_i^{(k)} > 0 \quad (i = 1, \cdots, n). \tag{3}$$

Setting $e^T = (1, 1, \cdots, 1)$, we make the change of variable

$$x = \frac{Dz}{(e^T Dz)}, \tag{4}$$

so that $x = x^k$ if $z = a$. We now require

$$ADz = 0, \qquad e^T z = 1. \tag{5}$$

(The projective transformation (4) maps the simplex into itself.)

We now pick $\hat{z}$ so that

$$c^T D\hat{z} = \min \{c^T Dz : ADz = 0,\ e^T z = 1,\ |z - a| = \alpha r\}, \tag{6}$$

where r is the distance from the center of the simplex to its boundary, namely $r = (n(n-1))^{-1/2}$. The purpose of the following analysis will be to choose an optimal value for the parameter α between 0 and 1. (We will finally choose $\alpha = \frac{1}{2}$.)

Having picked $\hat{z}$ by the rule (6), we use the change of coordinates (4) to determine the new feasible solution:

$$x^{(k+1)} = \frac{D\hat{z}}{e^T D\hat{z}} \tag{7}$$

where $D = D^{(k)} = \operatorname{diag}(x_1^k, \cdots, x_n^k)$.

On p. 379 of his paper [10], Karmarkar defines the logarithmic metric

$$f(x) = \sum_{j=1}^{n} \ln\left(\frac{c^T x}{x_j}\right). \tag{8}$$

The purpose of the iterations is to drive $f(x^k)$ to $-\infty$. On his p. 384 he completes a proof the following result.

THEOREM (Karmarkar). *For the problem* (1), *if* $0 < \alpha < 1$, *define*

$$\beta_n = \alpha\left(\frac{n}{n-1}\right)^{1/2}, \qquad \delta_n = \ln(1+\alpha) - \frac{\beta_n^2}{1-\beta_n}. \tag{9}$$

Then the successive iterates satisfy

$$f(x^{k+1}) \leqq f(x^k) - \delta_n. \tag{10}$$

For large n the decrement δ_n *tends to the limit*

$$\delta^* = \ln(1+\alpha) - \frac{\alpha^2}{1-\alpha} \tag{11}$$

and this limit takes its maximum value near $\alpha = 0.25$:

$$\delta^* = 0.1391 \quad \text{if } \alpha = 0.25. \tag{12}$$

Thus, Karmarkar suggests 0.25 as a good choice for his parameter α. We shall, instead, suggest $\alpha = 0.5$ because of the following result.

THEOREM 2. *For the problem* (1), *if* $0 < \alpha < 1$, *define*

$$\varepsilon_n(\alpha) = -n \ln\left(1 - \frac{\alpha}{n-1}\right) + (n-1)\ln\left(1 + \frac{\alpha}{n-1}\right) + \ln(1-\alpha). \tag{13}$$

Then the successive iterates satisfy

$$f(x^{k+1}) \leqq f(x^k) - \varepsilon_n(\alpha). \tag{14}$$

For large n the decrement $\varepsilon_n(\alpha)$ *tends to the limit*

$$\varepsilon^*(\alpha) = 2\alpha + \ln(1-\alpha) \tag{15}$$

and this limit takes its maximum value at $\alpha = 0.5$:

$$\varepsilon^* = 0.30685 \quad \text{if } \alpha = 0.5. \tag{16}$$

Comparison for finite n. For fixed α, the decrement $\varepsilon_n(\alpha)$ in (13) is a *decreasing* function of n. By contrast, Karmarkar's decrement $\delta_n(\alpha)$ in (8) is an *increasing* function of n. Therefore, for all n,

$$\delta_n(0.25) < 0.1391 < 0.30685 < \varepsilon_n(0.5). \tag{17}$$

Proof of Theorem 2. If x^* is a basic optimal solution to the program (1), then

$$|x^*-a| \leqq \left(\frac{n-1}{n}\right)^{1/2} = (n-1)r \tag{18}$$

since the right-hand side is the greatest distance from the center a to the boundary of the simplex $x_1+\cdots+x_n=1$, $x \geqq 0$. If z^* is related to x^* by the projection (4), then z^* lies on the boundary of the same simplex, and so

$$|z^*-a| \leqq (n-1)r. \tag{19}$$

Assuming $c^T x^* = 0$, we have $c^T D z^* = 0$. Now we draw the line segment from the center a to the point z^*. This segment intersects the sphere $|z-a| = \alpha r$ at the point

$$z^1 = (1-\theta)a + \theta z^* \tag{20}$$

where

$$\theta = \frac{\alpha r}{|z^*-a|} \geqq \frac{\alpha r}{(n-1)r} = \frac{\alpha}{n-1}. \tag{21}$$

Then we deduce

$$c^T D z^1 = (1-\theta)c^T Da + \theta c^T D z^*, \qquad c^T D z^1 \leqq \left(1-\frac{\alpha}{n-1}\right) c^T Da. \tag{22}$$

The point z^1 satisfies the constraints

$$ADz = 0, \qquad e^T z = 1, \qquad |z-a| = \alpha r. \tag{23}$$

Since $\hat{z}$ was picked in (6) to minimize $c^T Dz$ under these constraints, (22) implies

$$c^T D\hat{z} \leqq c^T D z^1 \leqq \left(1-\frac{\alpha}{n-1}\right) c^T Da. \tag{24}$$

Using the change of coordinates (4) in the definition (8), we find

$$f(x) = \sum_{j=1}^{n} \ln \frac{c^T Dz}{d_{jj} z_j}$$

or

$$f(x) = n \ln(c^T Dz) - \sum_{j=1}^{n} \ln z_j - \sum_{j=1}^{n} \ln d_{jj}. \tag{25}$$

Applying this formula for $z = \hat{z}$ and $z = a$, we get

$$\begin{aligned} f(x^{k+1}) &= n \ln (c^T D\hat{z}) - \sum_{1}^{n} \ln \hat{z}_j - \sum_{1}^{n} \ln d_{jj}, \\ f(x^k) &= n \ln (c^T Da) - \sum_{1}^{n} \ln a_j - \sum_{1}^{n} \ln d_{jj}. \end{aligned} \tag{26}$$

But we know

$$c^T D\hat{z} \leqq \left(1-\frac{\alpha}{n-1}\right) c^T Da \quad \text{and} \quad -\sum_{1}^{n} \ln a_j = n \ln n,$$

so if we subtract in (26), we get

$$f(x^{k+1}) - f(x^k) \leqq n \ln\left(1-\frac{\alpha}{n-1}\right) - \sum_{1}^{n} \ln \hat{z}_j - n \ln n. \tag{27}$$

Now all we need is a lower bound for $\sum \ln \hat{z}_j$. We can get a lower bound from our Theorem 1, since the function $\ln t$ has a convex derivative. Since $\hat{z}$ satisfies

$$\sum_1^n z_j = 1 \quad \text{and} \quad \sum_1^n \left(z_j - \frac{1}{n}\right)^2 = \frac{\alpha^2}{n(n-1)}, \tag{28}$$

formula (9) in § 2 gives the inequality

$$\sum_1^n \ln \hat{z}_j \geqq \ln(1-\alpha) + (n-1)\ln\left(1+\frac{\alpha}{n-1}\right) - n \ln n. \tag{29}$$

If we subtract this inequality from the inequality (27), we get

$$f(x^{k+1}) - f(x^k) \leqq n \ln\left(1-\frac{\alpha}{n-1}\right) - (n-1)\ln\left(1+\frac{\alpha}{n-1}\right) - \ln(1-\alpha). \tag{30}$$

Here the right-hand side is $-\varepsilon_n$ and we have proved the assertion (14).

As $n \to \infty$ we get

$$\lim_{n\to\infty} \varepsilon_n(\alpha) = 2\alpha + \ln(1-\alpha) \equiv \varepsilon^*(\alpha) \tag{31}$$

and now calculus gives $\min \varepsilon^*(\alpha) = \varepsilon^*(0.5) = 0.30685$.

It remains only to show that ε_n is a decreasing function of n. To do this, we write ε_n in the form

$$\varepsilon_n = -\ln\left(1-\frac{\alpha}{n-1}\right) + (n-1)\left[\ln\left(1+\frac{\alpha}{n-1}\right) - \ln\left(1-\frac{\alpha}{n-1}\right)\right] + \ln(1-\alpha)$$

and so

$$\varepsilon_n = -\ln\left(1-\frac{\alpha}{n-1}\right) + 2\sum_{k=0}^{\infty} \frac{1}{2k+1} \frac{\alpha^{2k+1}}{(n-1)^{2k}} + \ln(1-\alpha).$$

This expression decreases as n increases and so we have finished the proof of Theorem 2 (see Padberg [13]).

COROLLARY. *If $\alpha = \frac{1}{2}$, the costs of the iterates in Karmarkar's algorithm tend to zero as follows*:

$$c^T x^{(k)} \leqq \left(\frac{2}{e}\right)^{k/n} (c^T x^0) \qquad (k = 1, 2, \cdots). \tag{32}$$

This result holds for all problem sizes n. (The ratio $2/e = 0.73576$.)

Proof. We have set the problem up so that the initial iterate x^0 has identical coordinates $x_j^0 = 1/n$. Since the iterate $x^{(k)}$ also has positive coordinates with unit sum, we know

$$\sum_{j=1}^n \ln x_j^{(k)} \leqq \sum_{j=1}^n \ln x_j^0 = n \ln \frac{1}{n} \tag{33}$$

by the inequality of the geometric and arithmetic means.

In Theorem 2 we proved

$$f(x^{k+1}) \leqq f(x^k) - \varepsilon_n(\alpha)$$

where $\varepsilon_n(0.5) \geqq \varepsilon^* = 1 - \ln 2$, according to formula (15). So we deduce, for $\alpha = 0.5$,

$$f(x^k) \leqq f(x^0) - k\varepsilon^* = f(x^0) + k \ln\left(\frac{2}{e}\right). \tag{34}$$

By definition,

$$f(x^k) = n \ln (c^T x^k) - \sum_{j=1}^{n} \ln x_j^{(k)}, \qquad f(x^0) = n \ln (c^T x^0) - \sum_{j=1}^{n} \ln x_j^0.$$

Subtracting and using the inequality (33), we get

$$f(x^k) - f(x^0) \geqq n[\ln (c^T x^{(k)}) - \ln (c^T x^0)]. \tag{35}$$

Now (34) yields

$$k \ln \left(\frac{2}{e}\right) \geqq n[\ln (c^T x^{(k)}) - \ln (c^T x^0)], \tag{36}$$

which proves the asserted inequality (32). This proves the corollary.

Finally, we recall that we wish to achieve an inequality $c^T x^k \leqq 2^{-q}(c^T x^0)$ in at most $k = k$ max steps. We used the integer k max in our description of the algorithm in § 4. Now we can define k max.

We shall have $c^T x^{(k)} \leqq 2^{-q} c^T x^0$ when k satisfies

$$\ln c^T x^{(k)} - \ln c^T x^0 \leqq -q \ln 2.$$

Now (36) says this will be true when k satisfies

$$\left(\frac{k}{n}\right) \ln \left(\frac{2}{e}\right) \leqq -q \ln 2$$

or, equivalently,

$$k \geqq \frac{\ln 2}{1 - \ln 2} nq = 2.25889\, nq. \tag{37}$$

Therefore, we may define $k \max = \lceil 2.2589\, nq \rceil$.

6. An $O(n^3)$ termination algorithm. Karmarkar's iterative method applies to a linear program set up in the form

$$Ax = 0, \qquad e^T x = 1, \qquad x \geqq 0, \qquad c^T x = \min \tag{1}$$

where we know that the minimum cost $c^T x^*$ equals zero and we are only looking for an optimal vector x^*. We prescribe a large integer q and the iterations stop when we obtain an iterate $x^{(k)}$ that satisfies

$$x^{(k)} > 0, \qquad c^T x^{(k)} \leqq 2^{-q} c^T x^0. \tag{2}$$

For practical purposes the criterion (2) may be sufficient. If 2^{-q} is less than the unavoidable roundoff error in fixed-precision calculation, it makes no sense to keep on computing after we have achieved $c^T x < 2^{-q}$.

Let us assume, instead, that all arithmetic operations are exact. We want to compute an exact solution x^* in polynomial time.

We will suppose that all the coefficients a_{ij}, c_j in the problem (1) are integers. Let us rewrite the problem in the form

$$Bx = g, \quad x \geqq 0, \quad c^T x = \min \quad \text{(known to be zero)}. \tag{3}$$

We suppose B is an $m \times n$ matrix with rank m. We begin with $x^{(k)} > 0$ satisfying (2), and we look for an optimal *basic* solution $\hat{x}$, satisfying $c^T \hat{x} = 0$.

First we will show how to prescribe q so that any *basic* feasible solution x satisfying $c^T x \leqq 2^{-q}(c^T x^0)$ must be optimal. This has been done before; see Aspvall and Stone [2] and Grötschel, Lovász and Schrijver [8].

As we observed before, every $m \times m$ determinant formed from m columns of the matrix B must satisfy a Hadamard inequality

$$|\Delta| \leqq l_1 \cdots l_m \leqq (l_{\max})^m, \tag{4}$$

where $l_1 \cdots l_m$ are the lengths of the columns of Δ and where $l_{\max}$ is the maximum of the lengths of the columns of B. *Pick q so large that*

$$2^{-q}(c^T x^0) < (l_{\max})^{-m}. \tag{5}$$

Since all b_{ij}, c_j are integers, Cramer's rule implies that the cost $c^T x$ of a basic feasible solution must be a rational number with some denominator Δ satisfying (4). *So if* (5) *holds and if $c^T x \leqq 2^{-q}(c^T x^0)$, we must have $c^T x = 0$. Then x is the required basic optimal solution x^*.*

So the problem reduces to finding *any* basic feasible solution x^* with cost

$$c^T x^* \leqq c^T x^{(k)} \leqq 2^{-q}(c^T x^0). \tag{6}$$

As Aspvall and Stone observed in [2], it is surely possible to go from a feasible solution x to some better *basic* feasible solution x^* in $O(mn^3)$ arithmetic operations. If m/n does not tend to zero, we have $O(n^4)$ operations. But n^4 may be too big for us if we wish to use Karmarkar's modified algorithm, which gives us the feasible solution $x^{(k)}$ in $O(n^{3.5}q)$ operations. For this reason we will now discuss a *termination algorithm* that takes us from $x^{(k)}$ to $\hat{x}$ in $O(n^3)$ operations.

We begin by assigning $x := x^{(k)}$, a feasible solution of the linear program (3), for which we know that the set of feasible solutions is bounded (see § 3).

By the use of successive Householder reflections, we perform a QR transformation on the matrix B^T:

$$Q^T B^T = \begin{bmatrix} R \\ 0 \end{bmatrix}. \tag{7}$$

This process is described in the book by Golub and van Loan, [7, p. 148]. *This process requires $O(n^3)$ arithmetic operations* and we will do it only once. The result is an $n \times n$ orthogonal matrix Q and an $m \times m$ right-triangular matrix R, whose diagonal elements are positive because the $m \times n$ matrix B is known to have rank m. In (7) the zero matrix O has $n - m$ rows and m columns. Since (7) implies $BQ = [R^T, O^T]$, we deduce that *the last $n - m$ columns of Q provide a basis for the null space of B.*

In general, if x is any feasible solution of our program (3), we will say that x *depends* on the set J if J is a subset of $\{1, \cdots, n\}$ such that $x_j \geqq 0$ for $j \in J$ and $x_j = 0$ for j in the complement J'. (Thus x_j *may* be zero if $j \in J$, but *must* be zero if $j \in J'$.)

We begin with $x := x^k > 0$. So we begin with $J = \{1, \cdots, n\}$ and $J' = \varnothing$, the null set.

At any stage, if we have a feasible solution x that depends on J, suppose we have a basis $z^1, \cdots, z^s$ for the linear space

$$\{z\colon Bz = 0, \text{ and } z \text{ depends on } J\}. \tag{8}$$

Initially, $s = n - m$ and $z^1, \cdots, z^{n-m}$ are the last $n - m$ columns of Q.

Starting with x, if $s > 0$ we will construct a new feasible solution that depends on a smaller index set J, with equal or smaller cost. Assign $z := z^s$. If $c^T z < 0$, assign $z := -z$. Now we have $c^T z \geqq 0$. Set

$$\lambda = \min \{x_j / z_j\colon z_j > 0\}. \tag{9}$$

(Some z_j must be positive; otherwise $x-\sigma z$, $\sigma\to+\infty$, would provide an unbounded set of feasible solutions.) If the minimum λ is achieved for $j=r$, assign

$$\begin{aligned} &x := x-\lambda z, \qquad J := J-\{r\},\\ &z^j := z^j-(z_r^j/z_r)z \quad \text{for } j := 1 \text{ to } s-1,\\ &s := s-1. \end{aligned} \tag{10}$$

Since $\lambda \geqq 0$ and $c^T z \geqq 0$, the new feasible x has cost less than or equal to the previous cost. The size of J has been reduced by 1. The new set $\{z^j\}$ provides a basis for the new space $\{z\colon Bz=0$, and z depends on $J\}$.

We repeat this procedure until $s=0$. Then x is the required *basic* feasible solution x^* with cost $c^T x^* \leqq$ the initial cost $c^T x^{(k)}$. Then if (5) holds, we must have $c^T x^* = 0$.

Our termination algorithm begins with a single $O(n^3)$ QR transformation. Then for $s := n-m$ down to 1 the algorithm performs the assignments (9) and (10). Each of these assignments requires $O(ns)=O(n^2)$ arithmetic operations. *Therefore, the entire termination algorithm requires $O(n^3)$ arithmetic operations.*

7. The complexity of Karmarkar's algorithm. We have shown how to reduce the standard minimum problem of linear programming to the form

$$Ax=0, \qquad \sum_{j=1}^{n} x_j = 1, \qquad x \geqq 0, \qquad c^T x = \min. \tag{1}$$

Here A is an $m\times n$ matrix with rank m and we know the initial feasible solution x^0 with identical coordinates $x_j^0 = 1/n$. We assume $c^T x^0 > 0$.

If the original standard minimum problem has an optimal solution, then the problem (1) has the optimal value $c^T x^* = 0$. We will now estimate the total number of arithmetic operations required to compute an *exact* optimal solution x^*.

If we prescribe $q>0$, Karmarkar's iterative algorithm computes a sequence of feasible solutions $x^1, \cdots, x^k$. Each of the k iterations requires $O(n^3)$ arithmetic operations. The final iterate satisfies

$$c^T x^k \leqq 2^{-q} c^T x^0. \tag{2}$$

In § 5 we showed that if $\alpha = 0.5$, it suffices to take

$$k = k \max = \lceil 2.2589nq \rceil. \tag{3}$$

We must now ask how large q should be.

To go from the approximate optimal solution x^k to an exact solution x^*, we must know that any *basic* feasible solution x must be optimal if it has cost $c^T x \leqq c^T x^k$. Knowing this, we can produce the basic optimal solution $x = x^*$ from the feasible solution x^k by using the $O(n^3)$ *termination algorithm* described in § 6.

In the following analysis, *we assume that all the coefficients a_{ij}, c_j in* (1) *are integers.* Then the number of bits in the input is approximately equal to the exponent L defined by the equation

$$2^L = \left(\prod_{i=1}^{m} \prod_{j=1}^{n} (1+|a_{ij}|) \right) \prod_{j=1}^{n} (1+|c_j|). \tag{4}$$

A basic solution x solves a nonsingular system of $m+1$ linear equations whose coefficients are integers. If Δ is the determinant of this system, we have the crude upper bound

$$|\Delta| \leqq \prod_{i=1}^{m} \prod_{j=1}^{n} (1+|a_{ij}|). \tag{5}$$

Furthermore,

$$0 < c^T x^0 = n^{-1} \sum_{j=1}^{n} c_j < \prod_{j=1}^{n} (1+|c_j|). \tag{6}$$

From the last three formulas, we deduce

$$0 < |\Delta| c^T x^0 < 2^L. \tag{7}$$

Our basic feasible solution x satisfies

$$0 \leqq c^T x \leqq c^T x^k \leqq 2^{-q} c^T x^0. \tag{8}$$

If we prescribe $q = L$, we deduce

$$0 \leqq c^T x < |\Delta|^{-1}. \tag{9}$$

But $c^T x$ is a rational number with denominator Δ. Therefore $c^T x = 0$ and x equals the required optimal solution x^*.

So it suffices to prescribe $q = L$. Then we can compute an exact optimal solution x^* in $O(n^4 L)$ arithmetic operations. By using Karmarkar's modified algorithm [6, p. 388], we can reduce this number to $O(n^{3.5} L)$.

A different analysis. In the preceding analysis we looked at the total bit-length of the input. In the definition (4) of L we implicitly allowed the coefficients a_{ij}, c_j to have very different sizes. *Let us now suppose that all these integers are at most s-bit numbers*:

$$|a_{ij}| < 2^s, \qquad |c_j| < 2^s \tag{10}$$

where s is independent of n, say $s = 64$. We will now estimate the computational complexity as a function of n alone.

As before, we must obtain upper bounds for $c^T x^0$ and $|\Delta|$. The first is easy:

$$c^T x^0 = n^{-1} \sum_{j=1}^{n} c_j < 2^s. \tag{11}$$

It is also easy to get a bound for $|\Delta|$. Hadamard's inequality gives

$$|\Delta| < 2^{ms} (m+1)^{(m+1)/2}. \tag{12}$$

(Remember, Δ is a determinant of order $m+1$; the last row has all components equal to 1.)

Now choose $q = (m+1)s + \frac{1}{2}(m+1) \lg (m+1)$. Then the last two formulas imply

$$2^{-q} c^T x^0 < |\Delta|^{-1}. \tag{13}$$

Now formula (8) implies

$$0 \leqq c^T x \leqq 2^{-q} c^T x^0 < |\Delta|^{-1}. \tag{14}$$

As before, we deduce $c^T x = 0$; and the basic solution x is the required optimal solution x^*.

In this analysis, according to (3), the number of iterations is

$$k \max = O(nq) = O(n^2 \log n). \tag{15}$$

Multiplying k max by n^3, we obtain this estimate for the complexity:

$$T(n) = O(n^5 \log n). \tag{16}$$

As before, by using Karmarkar's modified algorithm we can reduce this number to $O(n^{4.5} \log n)$.

Comparison. Is the estimate $n^5 \log n$ smaller or bigger than the original estimate, $n^4 L$? It should be smaller, because in formula (10) we assumed more.

Indeed, formula (10) gives at most the bit-length

$$L = mns + ns = (m+1)ns \tag{17}$$

as we see from the definition (4). With s fixed, L may go to infinity as fast as n^2 and so the original estimate $O(n^4 L)$ becomes $O(n^6)$. For the modified algorithm the estimate $O(n^{3.5} L)$ becomes $O(n^{5.5})$.

Appendix. Here we complete § 2 with further examples and with a proof of Theorem 1.

Further examples. The function $g(t) = t^p$ has a convex derivative if $p \leqq 0$ or if $p \geqq 1$. Then we deduce

$$u_1^p + (n-1)v_1^p \leqq \sum_{i=1}^{n} x_i^p \leqq u_2^p + (n-1)v_2^p \tag{10}$$

for all points x on S.

If $0 < p < 1$ the function $-t^p$ has a convex derivative and so we deduce the opposite inequalities:

$$u_2^p + (n-1)v_2^p \leqq \sum_{i=1}^{n} x_i^p \leqq u_1^p + (n-1)v_1^p. \tag{11}$$

In these inequalities $u_1 < v_1$ while $u_2 > v_2$.

Proof of the theorem. We have assumed that $g'(t)$ is convex. Without loss of generality, we may assume that $g'(t)$ is *strictly* convex:

$$g'((1-\theta)t_1 + \theta t_2) < (1-\theta)g'(t_1) + \theta g'(t_2) \tag{12}$$

for $0 < \theta < 1$, $0 < t_1 < t_2 < 2/n$. If $g'(t)$ is convex but not strictly convex, we define the function $g_\varepsilon(t) = g(t) + \varepsilon t^3$. Now $g'_\varepsilon(t)$ is strictly convex if $\varepsilon > 0$. If the inequalities (7) and (8) are proved for $g_\varepsilon(t)$, they follow by continuity for $g(t)$ if we let $\varepsilon \to 0$. So from now on we shall assume *strict* convexity (12).

For all x on S the coordinates x_i satisfy

$$0 < \frac{1}{n} - \frac{\alpha}{n} \leqq x_i \leqq \frac{1}{n} + \frac{\alpha}{n} < \frac{2}{n} \qquad (i = 1, \cdots, n), \tag{13}$$

so the coordinates x_i range in a closed subinterval of the open interval $0 < t < 2/n$, where $g(t)$ is differentiable and hence continuous.

For x on S, let $\sum g(x_i)$ take its minimum value at $x = a$. *We assert that the coordinates a_i take just two distinct values.* (They cannot all take the same value because we have assumed $\rho^2 > n^{-1}$.)

To prove the assertion, assume that some three of the coordinates were different, say $a_1 < a_2 < a_3$. Define the partial sums

$$a_1 + a_2 + a_3 = s_1, \qquad a_1^2 + a_2^2 + a_3^2 = s_2, \tag{14}$$

and look at the three equations

$$\begin{gathered} x_1 + x_2 + x_3 = s_1, \qquad x_1^2 + x_2^2 + x_3^2 = s_2, \\ g(x_1) + g(x_2) + g(x_3) = \lambda, \end{gathered} \tag{15}$$

for λ in the neighborhood of $\lambda_0 = g(a_1) + g(a_2) + g(a_3)$.

If $\lambda = \lambda_0$, the system (15) has the solution $x_1 = a_1$, $x_2 = a_2$, $x_3 = a_3$. To show that the system remains solvable if we vary λ in the neighborhood of λ_0, and hence that λ_0 is not the minimum value, it suffices to prove that the Jacobian determinant is nonzero. For $x_i = a_i$ the Jacobian determinant is

$$J = \begin{vmatrix} 1 & 1 & 1 \\ 2a_1 & 2a_2 & 2a_3 \\ g'(a_1) & g'(a_2) & g'(a_3) \end{vmatrix}. \tag{16}$$

Since we assume $a_1 < a_2 < a_3$, we may write

$$(1-\theta)a_1 + \theta a_3 = a_2 \qquad (0 < \theta < 1). \tag{17}$$

If we subtract $1-\theta$ times the first column of J from the second column and then subtract θ times the last column from the second column, we get

$$J = \begin{vmatrix} 1 & 0 & 1 \\ 2a_1 & 0 & 2a_3 \\ g'(a_1) & \delta & g'(a_3) \end{vmatrix}, \tag{18}$$

where

$$\delta = g'((1-\theta)a_1 + \theta a_3) - (1-\theta)g'(a_1) - \theta g'(a_3). \tag{19}$$

Now the strict convexity of $g'(t)$ implies $\delta < 0$ and we compute

$$J = -2(a_3 - a_1)\delta > 0. \tag{20}$$

This completes the proof that the coordinates of the minimizing point a cannot take three or more distinct values.

So the coordinates a_i take exactly two distinct values, $u \neq v$. *We now assert that one coordinate takes the smaller value, and the other $n-1$ coordinates take the larger value.*

To prove this, consider any three coordinates a_i that are not identical, say

$$a_1 = u, \quad a_2 = a_3 = v \quad \text{with } u \neq v. \tag{21}$$

Then we must show $u < v$.

Define the partial sums

$$a_1 + a_2 + a_3 = u + 2v = s_1, \qquad a_1^2 + a_2^2 + a_3^2 = u^2 + 2v^2 = s_2. \tag{22}$$

We assume $u > 0$, $v > 0$, $u \neq v$. If s_1 and s_2 are fixed, these equations have two solution pairs:

$$u_1 = \tfrac{1}{3}(s_1 - 2\Delta), \qquad v_1 = \tfrac{1}{3}(s_1 + \Delta) \tag{23}$$

and

$$u_2 = \tfrac{1}{3}(s_1 + 2\Delta), \qquad v_2 = \tfrac{1}{3}(s_1 - \Delta), \tag{24}$$

where

$$\Delta = \left(\frac{3s_2 - s_1^2}{2}\right)^{1/2} > 0. \tag{25}$$

(Here $3s_2 - s_1^2 > 0$ because $u \neq v$ in (22).)

Now we must compare the partial sums

$$g(u_1) + 2g(v_1) \quad \text{and} \quad g(u_2) + 2g(v_2) \tag{26}$$

and pick the smaller value to minimize $g(a_1)+g(a_2)+g(a_3)$, and hence to minimize $g(a_1)+\cdots+g(a_n)$.

Looking at the formulas (23), (24), we see expressions of the form

$$u_1 = t-2h, \quad v_1 = t+h, \qquad u_2 = t+2h, \quad v_2 = t-h \tag{27}$$

where $0<2h<t$. We will now prove that

$$g(t-2h)+2g(t+h) < g(t+2h)+2g(t-h), \tag{28}$$

which implies that we must pick the first pair ($u_1 < v_1$), and not the second pair ($u_2 > v_2$), to minimize $\sum g(a_i)$.

To prove (28) we use the strict convexity of the derivative g'. We have

$$g'(t+\phi) < \tfrac{1}{2}[g'(t+h+\phi)+g'(t-h+\phi)]. \tag{29}$$

If we integrate this inequality for $-h<\phi<h$, we get

$$\int_{-h}^{h} g'(t+\phi)\,d\phi < \frac{1}{2}\int_{-2h}^{2h} g'(t+\phi)\,d\phi$$

or

$$g(t+h)-g(t-h) < \tfrac{1}{2}[g(t+2h)-g(t-2h)]. \tag{30}$$

This proves the assertion (28).

We have now completed a proof of the assertion (7), which states that the coordinates of a minimizing point a must have the form $a_i = u_1$ for one i, $a_j = v_1$ for all $j \neq i$, where $u_1 < v_1$.

It remains to prove the assertion (8) for a maximizing point $x = b$. The proof is exactly like the proof of the assertion (7) for a minimizing point $x = a$. First we show that the coordinates b_i take exactly two values, $u \neq v$. We then show that if any one coordinate equals u and any two other coordinates equal v, then we must choose $u = u_2 > v = v_2$ in order to maximize the partial sum $g(u)+2g(v)$ under the constraints (22). This result follows, as before, from the inequality (28), and so the theorem is proved.

REFERENCES

[1] K. ANSTREICHER, *A monotonic projective algorithm for fractional linear programming*, Preprint, Yale School of Management, New Haven, CT; Algorithmica, 1985, to appear.

[2] B. ASPVALL AND R. E. STONE, *Khachian's linear programming algorithm*, J. Algorithms, 1 (1980), pp. 1-13.

[3] C. BLAIR, *The iterative step in the linear programming algorithm of N. Karmarkar*, Manuscript, College of Business Administration, University of Illinois, Urbana-Champaign (February 1985).

[4] R. G. BLAND, *New finite pivoting rules for the simplex method*, Math. Oper. Res., 2 (1977), pp. 103-107.

[5] G. B. DANTZIG, *Linear Programming and Extensions*, Princeton Univ. Press, Princeton, NJ, 1963.

[6] D. GOLDFARB AND S. MEHROTRA, *A relaxed version of Karmarkar's method*, Technical Report, Department of Industrial Engineering and Operations Research, Columbia Univ., New York, NY, 1986.

[7] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[8] M. GRÖTSCHEL, L. LOVÁSZ AND A. SCHRIJVER, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169-197.

[9] B. KALANTARI, *Karmarkar's algorithm with improved steps*, Technical Report LCSR-TR-80, Department of Computer Science, Rutgers University, New Brunswick, NJ, 1986.

[10] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373-395.

[11] L. V. KHACHIAN, *A polynomial algorithm in linear programming*, Soviet Math. Dokl., 20 (1979), pp. 191–194 (Amer. Math. Soc. Transl.).

[12] V. KLEE AND G. MINTY, *How good is the simplex algorithm?*, Inequalities, 3 (1972), pp. 159–175.

[13] M. PADBERG, *A different proof for the projective method for linear programming*, Manuscript, New York University, New York (February 1985).

[14] C. H. PAPADIMITRIOU AND K. STEIGLITZ, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[15] M. TODD, *A review of N. Karmarkar, "A new polynomial-time algorithm for linear programming,"* Computing Reviews, 27 (February 1986), pp. 95–96.