Algorithmica (1986) 1: 409–424

Algorithmica

An Extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables

Michael J. Todd[1,2] and Bruce P. Burrell[1]

Abstract. We describe an extension of Karmarkar's algorithm for linear programming that handles problems with unknown optimal value and generates primal and dual solutions with objective values converging to the common optimal primal and dual value. We also describe an implementation for the dense case and show how extreme point solutions can be obtained naturally, with little extra computation.

Key Words. Linear programming, Karmarkar's algorithm, Duality.

1. Introduction. This paper is concerned with the new algorithm of Karmarkar [16] to solve linear programming problems. If the data of a linear programming problem are all integer, then the running time of the algorithm is polynomial in the length of a binary encoding of the data. The basic method is an iterative technique for solving a linear programming problem of a certain type. If the data are integer, the iterations can be terminated after a polynomial number of steps, and an exact solution obtained by rounding. Various subsidiary problems of obtaining a feasible solution and dealing with problems not in the required form can be reduced to solving problems in canonical form.

Let us ignore integrality of the data and consider the infinite sequence of iterates produced by the basic algorithm. The objective function values of the iterates converge at least linearly to the optimal value, but, in contrast to simplex methods, each iterate is a strictly positive vector lying in the relative interior of the feasible region. Convergence is proved by monitoring a cleverly constructed potential function, which is invariant under certain projective transformations. By making such a transformation so that the current iterate is mapped into a point far away from all inequality constraints Karmarkar is able to assure a fixed decrease in this potential function at every iteration. This translates into an aggregate decrease of the objective function as a geometric progression.

One drawback of Karmarkar's method is that it does not generate dual solutions, which are of great economic significance as well as of potential use during the algorithm. Here we show how dual solutions can be generated naturally during the course of the algorithm and how they can be used to extend the applicability

[1] School of Operations Research and Industrial Engineering, College of Engineering, Cornell University, Ithaca, New York, 14850, USA.
[2] Research supported in part by a fellowship from the Alfred P. Sloan Foundation and by NSF Grant ECS-15361.

Received March 1985; revised January 2, 1986. Communicated by Nimrod Megiddo.

of the basic algorithm. We also discuss some aspects of an implementation of the algorithm for the dense case. We show in particular that, at a very modest extra cost, extreme point solutions can be generated at each iteration.

In Section 2 we describe Karmarkar's basic algorithm for problems in a certain canonical form, with optimal value known to be zero. Section 3 shows that the potential function can be decreased by a constant at each iteration. This proof uses a result (Lemma 3.1) which shows that, in the transformed space, the objective function moves a substantial fraction of the way toward a lower bound given by duality, rather than toward zero. This replaces a geometrical argument of Karmarkar, and also applies when the optimal value is not zero. Section 4 discusses a sequence of dual solutions that arises naturally from the proofs of Section 3. We prove convergence under a certain nondegeneracy assumption. However, this assumption may fail for many practical problems. Section 5 introduces an extended algorithm that applies even in degenerate cases and when the optimal value is not known. Dual variables are naturally generated, and both the primal and dual objective values converge to their optimal value.

In Section 6 we discuss an implementation of the method using the *QR* factorization. In particular we show how to generate extreme point solutions.

Karmarkar uses a clever modification of his method to avoid computing a fresh factorization at each iteration. Instead, a lower rank update of the appropriate matrix is carried out at each step; on average, updates of rank $\sqrt{n}$ are sufficient. We avoid this modification for several reasons. First, it is hard to see how dual variables could be extracted and used effectively if low-rank updates are employed. Second, Karmarkar's proof that $\sqrt{n}$-rank updates suffice on average depends on his taking a step of fixed length in the transformed space. It seems to be much more efficient to perform some form of line search on the potential function along the direction generated by the algorithm. Such a line search may lead to much longer steps that preclude the application of Karmarkar's average-rank analysis. A line search also eliminates the difficulty of simple small examples leading to slow linear convergence—see Charnes *et al.* [3].

To conclude this introduction, we note that several previous algorithms have attempted to cut through the interior of the feasible region, rather than generate a sequence of adjacent extreme points on its boundary. See, for example, the reduced gradient methods in [15]. Karmarkar's treatment is distinguished by its ingenious transformations allowing substantial steps to be taken at each iteration and its use of the sophisticated potential function to assure reasonable progress. We also mention that projective transformations have also been used, first to handle linear fractional programming problems (Charnes and Cooper [2]), but also in linear complementary [18] and probabilistic analyses [11].

The first version of this paper contained additional material on solving problems in general form and on implementation for the large sparse case; many of the ideas were discussed independently by Tomlin [19]. There has been considerable further work in these areas. We would like to mention in particular the work of Anstreicher [1], Gay [6], Gonzaga [10], and Jensen and Steger [14] on handling general problems, and of Gill *et al* [8] on a Newton barrier method (similar to Karmarkar's projective method) and its efficient implementation. This last paper also contains the most encouraging computational results for an interior method obtained outside AT&T Bell Laboratories.

2. Karmarkar's Algorithm. In this section, we consider the linear programming problem

$$(P)\qquad \begin{aligned} \min\ & c^T x \\ & Ax = 0, \\ & e^T x = n, \\ & x \geq 0, \end{aligned}$$

where A is $m \times n$, and c, x, and $e = (1, 1, \ldots, 1)^T$ are n-vectors. We assume that:

(a) $Ae = 0$, so that $x^0 = e$ is an initial feasible point;
(b) the rank of A is m; and
(c) the optimal value of (P), denoted $v(P)$, is zero.

We also assume that $c^T e > 0$, since otherwise we could stop immediately with e optimal. This implies that $c^T x$ is not constant on the feasible region, and hence that it is (strictly) positive at any strictly positive feasible point.

We will outline Karmarkar's algorithm for this problem, and show that in k iterations it generates a strictly positive feasible vector x^k with

$$c^T x^k \leq \exp(-k/5n) c^T x^0, \tag{1}$$

which can be written as

$$n \log c^T x^k \leq n \log c^T x^0 - k/5. \tag{2}$$

This inequality would be trivial if at each iteration $n \log c^T x$ could be reduced by $\frac{1}{5}$. This appears difficult, due to the presence of the inequalities $x \geq 0$. Thus Karmarkar considers instead the "potential function"

$$\begin{aligned} f(x) \equiv f(x; c) &\equiv n \log c^T x - \sum_j \log x_j \\ &= \sum \log(c^T x / x_j) \end{aligned} \tag{3}$$

defined for all points in

$$F = \{x \in R^n_{++}: Ax = 0,\ e^T x = n\},$$

where

$$R^n_{++} = \{x \in R^n: x_j > 0 \text{ for all } j\}.$$

Karmarkar shows that f can be reduced by a constant at every iteration. Intuitively, if $n \log c^T x$ is not decreased, then $-\sum_j \log x_j$ is; this can be thought of as "gaining altitude" to allow a greater decrease later, or moving away from the close boundaries of R^n_{++} to allow a greater decrease later. Note that $-\sum_j \log x_j$ is minimized for $x \in F$, indeed also for x in the simplex

$$S = \{x \in R^n_{++}: e^T x = n\},$$

by $x = x^0 = e$. Hence it is easy to see that guaranteeing a decrease in f of $\frac{1}{5}$ at each iteration assures (1) (Lemma 2.2).

It is worthwhile to note the resemblance of f to the objective function of an interior penalty or barrier method. Using Frisch's logarithmic barrier function [5] and ignoring the easily maintained equality constraints, one might choose x^k to minimize

$$(1/r^k)c^T x - \sum_j \log x_j$$

over $x \in F$, where $\{r^k\}$ is a sequence of parameters converging to zero. Thus the parameters are chosen to magnify the effect of the objective function as k increases. There are also parameter-free methods, from which we single out Huard's method of centers [13]. At iteration k, Huard would minimize

$$-\log(c^T x^{k-1} - c^T x) - \sum_j \log x_j$$

over $x \in F$ with $c^T x < c^T x^{k-1}$. Note that, as $c^T x^{k-1}$ approaches $v(P)$, the effect of the objective function is automatically magnified.

There is a close resemblance between Karmarkar's potential function f and Huard's penalty function. Again, the effect of the objective function in f is magnified as $c^T x$ approaches 0, the assumed optimal value of (P). Huard's function is convex, while f is not; on the other hand, f does not depend on the iteration number and has some particularly nice properties.

We shall see that Karmarkar's algorithm is basically a projected gradient method to minimize (or reduce) f, but at each iteration it works in a transformed space. The transformation moves the current iterate $x^k \in S$ to the center e of the simplex; at the same time it maintains the form of the potential function f.

The transformation is defined by

$$x \to \hat{x} = \frac{nX_k^{-1}x}{e^T X_k^{-1} x}, \tag{4}$$

where $X_k = \operatorname{diag}(x_1^k, \ldots, x_n^k)$. It is a projective transformation taking S into itself, x^k into e, and the feasible solutions F into

$$\hat{F} = \{x \in S: \hat{A}x = 0\},$$

where $\hat{A} = AX_k$. A crucial property of the potential function f is that it is transformed into a function of the same form:

LEMMA 2.1. *Let x and $\hat{x}$ correspond as above, and let $\hat{c} = X_k c$. Then*

$$f(\hat{x};\, \hat{c}) = f(x;\, c) + \log \det X_k.$$

The proof is direct from the definition. Thus if we can assure a certain drop in $\hat{f} \equiv f(\cdot\,;\, \hat{c})$ from $\hat{f}(e)$, we will have obtained a similar drop in f in the original space. The significance of this is shown by:

LEMMA 2.2. *Let $x \in F$ satisfy $f(x) \le f(e) - \gamma$. Then*

$$c^T x \le \exp(-\gamma/n) c^T e.$$

PROOF. Note that $f(e) = n \log c^T x - \sum_j \log 1 = n \log c^T e$. Thus

$$n \log c^T x \le n \log c^T e - \gamma + \sum_j \log x_j.$$

Now the x_j's are positive numbers with arithmetic mean 1 (since $e^T x = n$). Thus their geometric mean is at most 1, which implies $\sum_j \log x_j \le 0$. Hence $\log c^T x \le \log c^T e - \gamma/n$, from which the result follows. □

The key inequality (1) will follow from Lemma 2.2 if we can show that each iteration decreases f by at least $\frac{1}{5}$. By Lemma 2.1 we only need to decrease $\hat{f}$ by at least $\frac{1}{5}$ from its value at e.

We now show how this is done (the proof is deferred to the next section). For any $p \times n$ matrix M of rank p, P_M denotes the projection onto the nullspace of M, $\{d \in R^n\colon Md = 0\}$, so that $P_M = I - M^T(MM^T)^{-1}M$. We denote by P the projection $I - ee^T/n$ onto $\{d \in R^n\colon e^T d = 0\}$.

Let

$$\hat{B} = \begin{pmatrix} \hat{A} \\ e^T \end{pmatrix}.$$

Note that since $\hat{A}$ has full row rank and $\hat{A}e = 0$, $\hat{B}$ has full row rank and

$$P_{\hat{B}} = P_{\hat{A}} P = P P_{\hat{A}}.$$

The gradient of $\hat{f}$ evaluated at e is

$$\nabla \hat{f}(e) = n\hat{c}/\hat{c}^T e - e.$$

Thus a reasonable direction in which to search for a suitable decrease in $\hat{f}$ is the projection of the negative gradient,

$$\hat{d}' = -P_{\hat{B}} \nabla \hat{f}(e).$$

Since $P_{\hat{B}}e = 0$, this is proportional to

$$\hat{d} = -P_{\hat{B}}\hat{c}.$$

Karmarkar shows that at least some fixed decrease in f is obtained by moving from e to $\hat{x} = e + \alpha\hat{d}/\|\hat{d}\|$, where α is a sufficiently small constant. In the next section we show that $\alpha = \frac{1}{3}$ guarantees a decrease in $\hat{f}$ of $\frac{1}{5}$.

In terms of the original variables, we have:

ALGORITHM 1. Set

$$B_k = \begin{pmatrix} AX_k \\ e^T \end{pmatrix}.$$

Compute

$$\hat{d}^k = -P_{B_k}X_k c, \qquad d^k = X_k\hat{d}^k.$$

Set

$$\tilde{x}^{k+1} = x^k + \alpha d^k/\|\hat{d}^k\|$$

and

$$x^{k+1} = n\tilde{x}^{k+1}/e^T\tilde{x}^{k+1}.$$

In fact, it is worth while to perform a line search on f in the direction d^k from x^k (or on $\hat{f}$ in the direction $\hat{d}^k$ from e) rather than choosing a fixed α. We show below that along such a line, f has at most one stationary point, which is a minimizer.

LEMMA 2.3. *Let $\varphi(\alpha) = f(x+\alpha d) = n \log c^T(x+\alpha d) - \sum_j \log(x_j + \alpha d_j)$, where x and d are not proportional. Then φ has at most one stationary point, and if it has one, it is a minimizer.*

PROOF. We find directly

$$\varphi'(\alpha) = \frac{nc^Td}{c^T(x+\alpha d)} - \sum_j \frac{d_j}{x_j + \alpha d_j},$$

$$\varphi''(\alpha) = -\frac{n(c^Td)^2}{(c^T(x+\alpha d))^2} + \sum_j \frac{d_j^2}{(x_j+\alpha d_j)^2}.$$

If $\varphi'(\alpha) = 0$, then the arithmetic mean of the quantities $\delta_j = d_j/(x_j + \alpha d_j)$ equals $c^Td/c^T(x+\alpha d)$. Because $\delta \to \delta^2$ is a strictly convex function, it follows that the arithmetic mean of the δ_j^2 is at least $[c^Td/c^T(x+\alpha d)]^2$, with equality only if all δ_j's are equal. Since we assume that x and d are not proportional, the δ_j's are not all equal, so that $\varphi'(\alpha) = 0$ implies $\varphi''(\alpha) > 0$. This gives the result. □

3. Guaranteed Decrease in f. Here we show that f can be decreased by a constant at each iteration. To allow its use in more general contexts in Section 5, the proof is different from that presented by Karmarkar.

The analysis of the previous section demonstrates that all iterations reduce to the same situation as the first, with c and A replaced by $\hat{c}$ and $\hat{A}$. Thus we consider here just the first iteration. However, to allow for general use of the results in Section 5, we replace c and A by $\bar{c}$ and $\bar{A}$. Thus we consider the linear programming problem

$$(\bar{P}) \qquad \begin{aligned} \min\ & \bar{c}^T x \\ & \bar{A}x = 0, \\ & e^T x = n, \\ & x \geq 0, \end{aligned}$$

satisfying the assumptions (a) and (b) of Section 2. Its dual is

$$(\bar{D}) \qquad \begin{aligned} \max\ & n\bar{z} \\ & \bar{A}^T\bar{y} + e\bar{z} \leq \bar{c}, \end{aligned}$$

where $\bar{y} \in R^m$ and $\bar{z} \in R$.

Note that, given any $\bar{y}$, $(\bar{y}, \bar{z})$ with $\bar{z} = \min_j(\bar{c} - \bar{A}^T\bar{y})_j$ is feasible in the dual with objective function $n\bar{z}$. If $v(\bar{P}) = 0$, we have $\bar{z} \leq 0$.

The idea is to show first that Karmarkar's step assures a reasonable decrease in $\bar{f}(x) = n \log \bar{c}^T x$ and then to show that the extra term $-\sum_j \log x_j$ does not increase f too much.

Our first result shows that, in fact, $\bar{c}^T x$ moves a substantial fraction of the way toward a lower bound given by a particular feasible dual solution by searching in the negative projected gradient direction.

LEMMA 3.1. *Let* $\bar{d} = -P_{\bar{B}}\bar{c}$, *where* $\bar{B} = \begin{pmatrix} \bar{A} \\ e^T \end{pmatrix}$. *Let* $\bar{y} = (\bar{A}\bar{A}^T)^{-1}\bar{A}\bar{c}$ *and* $\bar{z} = \min_j(\bar{c} - \bar{A}^T\bar{y})_j$. *Then either* $\bar{d} = 0$, *in which case e is an optimal solution to* $(\bar{P})$ *and* $(\bar{y}, \bar{z})$ *to* $(\bar{D})$, *or*

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

PROOF. If $\bar{d} = -P_{\bar{B}}\bar{c} = -PP_{\bar{A}}\bar{c} = -P(\bar{c} - \bar{A}^T\bar{y})$ is zero, then $\bar{c} - \bar{A}^T\bar{y}$ is a multiple of e and hence $n\bar{z} = (\bar{c} - \bar{A}^T\bar{y})^T e = \bar{c}^T e$. Hence e and $(\bar{y}, \bar{z})$ are primal and dual optimal. Suppose not. Then we have $\|\bar{d}\|^2 = \bar{d}^T\bar{d} = \bar{c}^T P_{\bar{B}}\bar{c} = -\bar{c}^T\bar{d}$. Thus $\bar{c}^T(e + \alpha\bar{d}/\|\bar{d}\|) = \bar{c}^T e - \alpha\|\bar{d}\|$. It suffices to show that $\|\bar{d}\| \geq \bar{c}^T e/n - \bar{z}$. Now

$$\begin{aligned} \bar{d} &= -P(\bar{c} - \bar{A}^T\bar{y}) \\ &= -(\bar{c} - \bar{A}^T\bar{y} - ee^T(\bar{c} - \bar{A}^T\bar{y})/n). \end{aligned}$$

Since $\bar{A}e=0$, we get $\bar{d}=-(\bar{c}-\bar{A}^T\bar{y})+(\bar{c}^Te/n)e$. Also, $\bar{c}^Te\geq v(\bar{P})\geq n\bar{z}$, so $(\bar{c}^Te/n)\geq \bar{z}$. For some i, we have $\bar{z}=(\bar{c}-\bar{A}^T\bar{y})_i$, so $\bar{d}_i=\bar{c}^Te/n-\bar{z}\geq 0$. Now $\|\bar{d}\|\geq |\bar{d}_i|=\bar{c}^Te/n-\bar{z}$, which completes the proof. □

COROLLARY 3.2. *With the notation of Lemma* 3.1, *suppose further that* $\bar{z}\leq 0$ *and* $v(\bar{P})\geq 0$ (*which holds if* $(\bar{P})$ *satisfies assumption* (c) *of Section* 2) *and* $\bar{d}\neq 0$. *Then if* $0<\alpha<1$, $n\log \bar{c}^T(e+\alpha\bar{d}/\|\bar{d}\|)\leq n\log \bar{c}^Te-\alpha$.

PROOF. Since $\bar{d}\neq 0$, Lemma 3.1 gives $\bar{c}^Te>\bar{c}^T(e+\alpha\bar{d}/\|\bar{d}\|)\geq v(\bar{P})$ so that $\bar{c}^Tx>v(\bar{P})\geq 0$ for all strictly positive x feasible in $(\bar{P})$. From $\bar{z}\leq 0$, Lemma 3.1 yields $\bar{c}^T(e+\alpha\bar{d}/\|\bar{d}\|)\leq(1-\alpha/n)\bar{c}^Te$. Since all these quantities are positive, the result follows from $\log(1-\alpha/n)\leq -\alpha/n$. □

The following lemmas are essentially proved in Karmarkar [16].

LEMMA 3.3. *If* $|\varepsilon|\leq\alpha<1$ *then*

$$\varepsilon-\varepsilon^2/2(1-\alpha)^2\leq\log(1+\varepsilon)\leq\varepsilon.$$

LEMMA 3.4. *If* $\|x-e\|\leq\alpha<1$, $e^Tx=n$, *then*

$$0\leq-\sum_j\log x_j\leq\alpha^2/2(1-\alpha^2).$$

Now from Corollary 3.2 and Lemma 3.4 we deduce:

THEOREM 3.5. *With notation as in Lemma* 3.1,

$$f(e+\alpha d/\|d\|)\leq f(e)-\alpha+\alpha^2/2(1-\alpha)^2.$$

In particular, if $\alpha=\frac{1}{3}$,

$$f(e+(\tfrac{1}{3})d/\|d\|)<f(e)-\tfrac{1}{5}. \tag{5}$$

Inequality (1) follows from (5) and the discussion following Lemma 2.2.

4. Dual Variables, I. The proof of Lemma 3.1 suggests that we might obtain good dual solutions by setting

$$y^k=(AX_k^2A^T)^{-1}AX_k^2c \quad\text{and}\quad z^k=\min_j(c-A^Ty^k)_j. \tag{6}$$

This is indeed the case as long as suitable nondegeneracy assumptions hold:

THEOREM 4.1. *Suppose the iterates* x^k *defined by Karmarkar's Algorithm* 1 *converge to a nondegenerate basic solution* x^* *to* (P). *Then* $\{(y^k,z^k)\}$ *given by* (6) *converges to an optimal solution to* (D).

PROOF. Let $X_* = \operatorname{diag}(x_1^*, \ldots, x_n^*)$ and let $\bar{X}_*$ denote the principal submatrix of X_* corresponding to the basic variables in x^*. Let $\begin{bmatrix} \bar{A} \\ e^T \end{bmatrix}$ be the basis matrix of (P) corresponding to x^*. Then $\bar{A}$ has rank m, and hence so does $\bar{A}\bar{X}_*$. Thus $\bar{A}\bar{X}_*^2\bar{A}^T = AX_*^2A^T$ is nonsingular. Now $(AX_k^2A^T)y^k = AX_k^2c$. Since $AX_k^2A^T$ converges to the nonsingular matrix $AX_*^2A^T$, and AX_k^2c converges to AX_*^2c, it follows that y^k converges to the unique solution y^* to

$$AX_*^2A^Ty^* = AX_*^2c. \tag{7}$$

Now the optimal solution $\bar{y}$ to (D) satisfies $A^T\bar{y} \le c$ (the optimal z is zero) and by complementary slackness $X_*A^T\bar{y} = X_*c$. Thus $\bar{y}$ also solves (7), whence $y^* = \bar{y}$ as desired. □

While Theorem 4.1 appears at first sight attractive, there are some difficulties. Many linear programming problems arising in practice have degenerate optimal solutions, and Theorem 4.1 fails to apply. In particular, one might hope to be able to find a feasible solution to a system $\tilde{A}^Ty \le \tilde{c}$ by considering $(\tilde{D})$ with $\bar{A} = [\tilde{A}, 0]$, $\bar{c}^T = [\tilde{c}^T, 0]$. If the linear inequality system has a solution, this has optimal value 0, and one might be tempted to apply Karmarkar's algorithm to $(\bar{P})$ and generate $\{y^k\}$ as in (6). Unfortunately, one optimal solution of $(\bar{P})$ is then $x = (0, \ldots, 0, n)^T$, which is highly degenerate. While the argument of Theorem 4.1 can be somewhat refined, we see no way to prove convergence in such a case.

Next, one might hope that, even if $\{y^k\}$ failed to converge, z^k would converge to 0 and hence every limit point of $\{y^k\}$ would be optimal. (It is possible to show that $\{y^k\}$ remains bounded, so that limit points exist.) The basis for such a hope comes from Lemma 3.1, where it appears that c^Tx^k makes progress toward the lower bound nz^k at each iteration. Thus, with $c^Tx^k \to 0$, we should also have $z^k \to 0$. Unfortunately, Lemma 3.1 applies only to the *first* iteration; at a later iteration z^k should be replaced with $\tilde{z}^k = \min_j(X_kc - X_kA^Ty^k)_j$. Since some components of x^k are approaching zero, the fact that $\tilde{z}^k$ tends to zero does not help.

We shall see in the next section how dual variables can be obtained in any case, and how they can be used to solve problems whose optimal value is not necessarily zero.

5. Dual Variables, II. We consider again problem (P), but now we *drop* assumption (c), that $v(P) = 0$. If we knew that $v(P)$ equalled nz^*, for some known z^* (part of an optimal solution to the dual (D) of (P)), then it would follow that the optimal value of $(\bar{P})$ with $\bar{c} = c - z^*e$ would be zero. Indeed, $\bar{c}^Tx = c^Tx - z^*e^Tx = c^Tx - nz^*$ for every feasible x. Then we could apply the algorithm with $\bar{c} = c - z^*e$ replacing c.

Unfortunately, z^* is unknown. Instead, we will use lower bounds z^k on z^* and consider $\bar{c} = c(z^k) \equiv c - z^ke$ at the kth iteration instead of c. We will show that the algorithm can be modified to generate sequences $\{x^k\}$ and $\{(y^k, z^k)\}$ satisfying

$$x^k \in F, \tag{8}$$

$$y^k \in R^m, \tag{9}$$

$$z^k = \min_j (c - A^T y^k)_j, \tag{10}$$

such that

$$\gamma^k = f(x^0; c(z^k)) - f(x^k; c(z^k)) \geq k/5 \tag{11}$$

for all k. For $k = 0$ we choose $x^0 = e$, $y^0 = (AA^T)^{-1}Ac$ and the corresponding z^0, and (8)-(11) are trivially satisfied.

If $c - A^T y^0$ is a multiple of e, then $c^T x$ is constant on the feasible region and we stop with e optimal. If not, then $c^T x$ is not constant and so $c^T x > nz$ for any strictly positive feasible x and any lower bound nz on $v(P)$.

Suppose (8)-(11) hold at the kth iteration. Since (y^k, z^k) is feasible in (D), $z^k \leq z^*$. The following lemma shows that (11) implies also that

$$f(x^k; c(z^*)) \leq f(x^0; c(z^*)) - k/5,$$

so that convergence of the algorithm follows as before.

LEMMA 5.1. *Let $z \leq z' \leq z^*$. Then for all $x \in F$ with $c^T x > nz'$,*

$$f(x^0; c(z)) - f(x; c(z)) \geq \gamma \geq 0$$

implies

$$f(x^0; c(z')) - f(x; c(z')) \geq \gamma.$$

PROOF. By Lemma 2.2, $c(z)^T x \leq c(z)^T x^0$, so $c^T x \leq c^T x^0$. The hypothesis is equivalent to

$$n \log\left(\frac{c^T x - nz}{c^T x^0 - nz}\right) \leq \sum_j \log x_j - \gamma.$$

Note that, since $nz \leq nz' < c^T x \leq c^T x^0$,

$$\frac{c^T x - nz'}{c^T x^0 - nz'} \leq \frac{c^T x - nz}{c^T x^0 - nz};$$

thus

$$n \log\left(\frac{c^T x - nz'}{c^T x^0 - nz'}\right) \leq \sum_j \log x_j - \gamma,$$

which implies the conclusion. □

We now show how to find x^{k+1}, y^{k+1}, and z^{k+1} satisfying (8)-(11).

Set $\tilde{y} = (AX_k^2A^T)^{-1}AX_k^2c(z^k)$, and $\hat{z} = \min_j(c(z^k) - A^T\tilde{y})_j$. There are two cases, depending on whether $\hat{z} \le 0$. Note that $\hat{z} = \tilde{z} - z^k$, where $\tilde{z} = \min_j(c - A^T\tilde{y})_j$; $n\tilde{z}$ is the bound on $v(P)$ given by $\tilde{y}$. Hence $\hat{z} > 0$ iff $\tilde{y}$ generates an improved bound ($n\tilde{z} > nz^k$) on $v(P)$.

LEMMA 5.2. *If $\hat{z} \le 0$, set $\hat{d}^k = -P_{B_k}X_kc(z^k)$ and $d^k = X_k\hat{d}^k$. Then with $y^{k+1} = y^k$, $z^{k+1} = z^k$, $\tilde{x}^{k+1} = x^k + \alpha d^k/\|\hat{d}^k\|$ and $x^{k+1} = n\tilde{x}^{k+1}/e^T\tilde{x}^{k+1}$ (where $\alpha = \frac{1}{3}$ or α is obtained by a line search on $f(\cdot\,; c(z^{k+1}))$), (8)-(11) hold for $k+1$.*

PROOF. Since $\hat{z} \le 0$, we also have

$$\bar{z} = \min_j(X_kc(z^k) - X_kA^T\tilde{y})_j \le 0.$$

Thus Lemma 3.1 and Corollary 3.2 can be applied with $\bar{c} = X_kc(z^k)$ and $\bar{A} = AX_k$. Hence the potential function $f(\cdot\,; c(z^k))$ can be reduced by at least $\frac{1}{5}$ as in Section 3, by moving in the direction d^k in the statement of the lemma. Hence (8)-(11) follow for $k+1$. □

Now we suppose $\hat{z} > 0$, so that

$$\min_j(X_kc(z^k) - X_kA^Ty)_j > 0. \tag{12}$$

Note that

$$\begin{aligned} X_kc(z^k) - X_kA^T\tilde{y} &= P_{AX_k}X_kc(z^k) \\ &= P_{AX_k}(X_kc - z^kx^k). \end{aligned}$$

Let $u = P_{AX_k}X_kc$, $v = P_{AX_k}x^k$.

LEMMA 5.3. *If $\hat{z} > 0$, there is a unique z^{k+1} with $z^k < z^{k+1}$ such that*

$$\min_j(u - z^{k+1}v)_j = 0.$$

Define

$$y^{k+1} = (AX_k^2A^T)^{-1}AX_k^2c(z^{k+1}).$$

Then

$$\min_j(c(z^{k+1}) - A^Ty^{k+1})_j = \min_j(u - z^{k+1}v)_j = 0. \tag{13}$$

Moreover, if

$$\hat{d}^k = -P_{B_k}X_kc(z^{k+1}), \qquad d^k = X_k\hat{d}^k,$$

$$\tilde{x}^{k+1} = x^k + \alpha d^k/\|\hat{d}^k\| \quad \textit{and} \quad x^{k+1} = n\tilde{x}^{k+1}/e^T\tilde{x}^{k+1}$$

(*where* $\alpha=\frac{1}{3}$ *or* α *is obtained by a line search on* $f(\cdot\,;c(z^{k+1}))$), *then* (8)–(11) *hold for* $k+1$.

PROOF. The function $\Psi(z)\equiv\min_j(u-zv)_j$ is piecewise linear and concave. Moreover, by (12), $\Psi(z^k)>0$. Now let $z=c^Tx^k/n\geq z^k$. Then

$$\begin{aligned} e^T(u-zv) &= (P_{AX_k}e)^T(X_kc-zx^k) \\ &= e^T(X_kc-zx^k) \\ &= c^Tx_k-nz=0. \end{aligned}$$

Since the sum of the components of $u-zv$ vanishes, we must have $\Psi(z)\leq 0$. Hence there is a unique z^{k+1}, $z^k<z^{k+1}\leq z$, with $\Psi(z^{k+1})=0$. (Note that z^{k+1} can be found by a simple minimum ratio test.)

Observe that $u-z^{k+1}v=X_k(c(z^{k+1})-A^Ty^{k+1})$, so that (13) holds. Thus nz^{k+1} is a lower bound on $v(P)$, since (y^{k+1},z^{k+1}) is feasible in (D). From $z^k<z^{k+1}\leq z^*$, Lemma 5.1 gives

$$\tilde{\gamma}^k=f(x^0;c(z^{k+1}))-f(x^k;c(z^{k+1}))\geq k/5.$$

Moreover, from (13),

$$\min_j(X_kc(z^{k+1})-X_kA^Ty^{k+1})_j=0,$$

so that Lemma 3.1 and Corollary 3.2 can be applied with $\bar{c}=X_kc(z^{k+1})$ and $\bar{A}=AX_k$. The corresponding $\bar{z}$ equals 0, so that the potential function

$$f(\cdot\,;c(z^{k+1}))$$

can be decreased by at least $\frac{1}{5}$ as in Section 3 by searching in the direction d^k. Hence (8)–(11) also hold for $k+1$. □

Let us state the resulting method precisely:

ALGORITHM 2. At iteration k, we have x^k, y^k, z^k. Set

$$A_k=AX_k.$$

Compute

$$u=P_{A_k}X_kc,\qquad v=P_{A_k}x^k.$$

If

$$\min_j(u-z^kv)_j\leq 0,$$

then set

$$y^{k+1}=y^k,\qquad z^{k+1}=z^k.$$

Otherwise, find

$$z^{k+1} > z^k \qquad \text{with} \quad \min_j(u - z^{k+1}v)_j = 0,$$

and set

$$y^{k+1} = (A_k A_k^T)^{-1} A_k X_k c(z^{k+1}).$$

Compute

$$\hat{d}^k = -P(u - z^{k+1}v), \qquad d^k = X_k \hat{d}^k,$$

and

$$\tilde{x}^{k+1} = x^k + (\tfrac{1}{3}) d^k / \|\hat{d}^k\|.$$

Set

$$x^{k+1} = n\tilde{x}^{k+1} / e^T \tilde{x}^{k+1}.$$

(Alternatively, $\tilde{x}^{k+1}$ is found by searching along $x^k + \alpha d^k$ to approximately minimize $f(\cdot\,; c(z^{k+1}))$.) Note that if the computation of u and v also yields $\bar{y} = (A_k A_k^T)^{-1} A_k X_k c$ and $\check{y} = (A_k A_k^T)^{-1} A_k x_k$ as by-products, then whenever y^k needs to be updated, we may use $y^{k+1} = \bar{y} - z^{k+1}\check{y}$.

From Lemmas 5.2 and 5.3, we obtain:

THEOREM 5.4. *The algorithm above generates a sequence $\{x^k\}$ of primal feasible and a sequence $\{(y^k, z^k)\}$ of dual feasible solutions, with $c^T x^k$ and nz^k approaching $nz^* = v(P)$. Indeed,*

$$(c^T x^k - nz^*) \le \exp(-k/5n)[c^T x^0 - nz^*], \tag{14}$$

and

$$(nz^* - nz^k) \le \frac{1}{1 - \exp(-k/5n)} \exp(-k/5n)[c^T x^0 - nz^*]. \tag{15}$$

PROOF. From (11) and Lemma 2.2 we have

$$(c^T x^k - nz^*) + (nz^* - nz^k) \le \exp(-k/5n)[(c^T x^0 - nz^*) + (nz^* - nz^k)].$$

Subtracting $\exp(-k/5n)(nz^* - nz^k)$ from both sides, we deduce inequalities (14) and (15) and hence the desired convergence. □

Note that obtaining the strongest possible inequality of form (11) assures both fast primal and fast dual convergence by the proof above. Thus the line search proposed above helps both primal and dual solutions to converge rapidly. Also, we see no way to prove that $c^T x^k$ approaches $v(P)$ monotonically or that nz^k (which is monotonic by construction) increases strictly at each iteration.

6. Implementation. Here we discuss how the algorithm in Section 5 can be implemented in the dense case, and how extreme point solutions can be generated at a modest extra cost for each iteration.

At each iteration, we perform a QR factorization of the matrix $A_k^T = X_k A^T$:

$$A_k^T = QR = [Q_1, Q_2]\begin{pmatrix} R_1 \\ 0 \end{pmatrix}, \tag{16}$$

where $Q = [Q_1, Q_2]$ is orthogonal, R_1 is $m \times m$ upper triangular, and Q_1 has m columns. We do not need Q explicitly—it is sufficient to be able to calculate Qv and $Q^T v$ efficiently. Thus Q can be stored as the product of Householder reflections or Givens rotations. The factorization (16) can be computed in about $m^2(n - m/3)$ floating-point operations using Householder reflections if A_k is dense; see [9]. For the sparse case, see, e.g., [7] and [12].

Next we compute $Q^T X_k c$ and $Q^T x^k$ (and thus $Q_i^T X_k c$ and $Q_i^T x^k$ for $i = 1, 2$) and hence $u = Q_2 Q_2^T X_k c$ and $v = Q_2 Q_2^T x^k$ in $O(mn)$ work. If y^k and z^k are to be updated, z^{k+1} requires $O(n)$ work and $y^{k+1} = R_1^{-1} Q_1^T (X_k c - z^{k+1} x^k)$ an additional $O(m^2)$. Finally, $\hat{d}^k$, d^k, and $\tilde{x}^{k+1}$, x^{k+1} need only $O(n)$ work together with any additional calculations performed in a linear search.

The factorization (16) is normally obtained by dealing with the columns of A_k^T sequentially. A slightly modified procedure may be valuable in our context. Assume that $\{x^k\}$ is converging to a nondegenerate basic feasible solution. Suppose the columns of A_k are permuted to reflect decreasing size of x_j^k. For large k we might expect the first $m+1$ columns of the permuted A_k to correspond to the basic columns of A in an optimal solution. Suppose that these are the first $m+1$ columns and that

$$A_k = [\tilde{A}_k, \bar{A}_k] \tag{17}$$

with $\tilde{A}_k m \times (m+1)$. In about $(\frac{2}{3})m^3$ floating-point operations we compute the QR factorization of $\tilde{A}_k^T$:

$$\tilde{A}_k^T = \tilde{Q}\tilde{R} = [\tilde{Q}_1, \tilde{q}]\begin{bmatrix} \tilde{R}_1 \\ 0 \end{bmatrix}. \tag{18}$$

Now note that $\tilde{A}_k \tilde{q} = 0$. Thus if $A = [\tilde{A}, \bar{A}]$, $c^T = (\tilde{c}^T, \bar{c}^T)$ and $X_k = \operatorname{diag}(\tilde{X}_k, \bar{X}_k)$ are partitioned as in (17), we find $\tilde{A}\tilde{X}_k \tilde{q} = 0$. Hence a suitable normalization x of

$$\begin{bmatrix} \tilde{X}_k \tilde{q} \\ \hline 0 \end{bmatrix}$$

gives an extreme point solution of (P). Note that if $\tilde{Q}$ is stored in product form, $\tilde{q}$ can be obtained in $O(m^2)$ work. If x is nonnegative, we can compute the corresponding dual solution (y, z) to check its optimality. Let $[\tilde{r}^T, \tilde{\rho}] = \tilde{e}^T \tilde{Q}$, where $\tilde{e}$ is a vector of $m+1$ ones, and $w = \tilde{Q}^T \tilde{X}_k \tilde{c}$; then

$$\begin{bmatrix} \tilde{R}_1 & \tilde{r} \\ 0 & \tilde{\rho} \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix} = w$$

can be solved for y and z.

If x is not an optimal extreme point, we must continue with the algorithm. Note that

$$A_k^T = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{R} \\ \bar{A}_k^T \end{bmatrix}$$

from (18); thus we may just continue with the reduction of A_k^T to upper triangular form, an additional $\sim m^2(n-m)$ operations being required to obtain

$$A_k^T = \begin{bmatrix} \tilde{Q} & 0 \\ 0 & I \end{bmatrix} \bar{Q} \begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

with $\bar{Q}$ and $\tilde{Q}$ each stored in product form. Notice that the extra work required to calculate x, y, and z is only quadratic in m and thus negligible compared with that required for the factorization itself. Of course, it is only necessary to compute x, y, and z when the set of $m+1$ largest components of x^k differs from that obtained on the previous iteration.

The analysis above assumes that x^k is converging to a nondegerate basic optimal solution. In the case of degeneracy, we may still obtain the optimal x (if $\tilde{A}_k$ has rank m), but unless we have the right basis it is likely that (y, z) will not verify optimality. In addition, there may well be numerical problems since A_k^T is nearly rank deficient. We defer the treatment of these difficulties to future work.

References

[1] K. M. Anstreicher, Analysis of Karmarkar's algorithm for fractional linear programming, Manuscript, School of Organization and Management, Yale University, New Haven, CT, 1985.
[2] A. Charnes and W. W. Cooper, Programming with linear fraction functionals, *Naval Res. Logist. Quart.*, **9** (1962), 181-186.
[3] A. Charnes, T. Song, and M. Wolfe, An explicit solution sequence and convergence of Karmarkar's algorithm, Manuscript, University of Texas at Austin, Austin, TX, 1984.
[4] V. Chvatal, *Linear Programming*, Freeman: New York and San Francisco, 1983.
[5] K. R. Frisch, The logarithmic potential method of convex programming, unpublished, University Institute of Economics, Oslo, 1955.
[6] D. Gay, A variant of Karmarkar's linear programming algorithm for problems in standard form, Manuscript, AT&T Bell Laboratories, Murray Hill, NJ, 1985.
[7] A. George and M. T. Heath, Solution of sparse linear least squares problems using Givens rotations, *Linear Algebra Appl.*, **34** (1980), 69-83.
[8] P. E. Gill, W. Murray, M. E. Saunders, J. A. Tomlin, and M. H. Wright, On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, Manuscript, Department of Operations Research, Stanford University, Stanford, CA, 1985.
[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins Press, Baltimore, 1983.
[10] C. Gonzaga, A conical projection algorithm for linear programming, Manuscript, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1985.
[11] M. Haimovich, The simplex method is very good! On the expected number of pivot stops and related properties of random linear programs, Manuscript, Graduate School of Business, Columbia University, New York, 1983.

[12] M. Heath, Some extensions of an algorithm for sparse linear least squares problems, *SIAM J. Sci. Statist. Comput.*, **3** (1982), 223-237.

[13] P. Huard, Resolution of mathematical programming with nonlinear constraints by the method of centers, in *Nonlinear Programming* (J. Abadie, ed.), North-Holland, Amsterdam, 1967, pp. 207-219.

[14] D. Jensen and A. Steger, Private communication, Department of Applied Mathematics and Statistics, State University of New York at Stonybrook, Stonybrook, New York, 1985.

[15] M. Kallio and E. L. Porteus, A class of methods for linear programming, *Math. Programming*, **14** (1978), 161-16.

[16] N. Karmarkar; A new polynomial time algorithm for linear programming, *Combinatorica*, **4** (1984), 373-395.

[17] N. Megiddo, A variation on Karmarkar's Algorithm, Manuscript, IBM Research Laboratory, San Jose, CA, 1985.

[18] M. J. Todd, Extensions of Lemke's algorithm for the linear complementarity problem, *J. Optim. Theory Appl.*, **20** (1976), 397-416.

[19] J. A. Tomlin, An experimental approach to Karmarkar's projective method for linear programming, Manuscript, Ketron Inc., Mountain View, CA, 1985.