Algorithmica (1986) 1: 395–407

Algorithmica

A Modification of Karmarkar's Linear Programming Algorithm

Robert J. Vanderbei,[1] Marc S. Meketon,[1] and Barry A. Freedman[1]

Abstract. We present a modification of Karmarkar's linear programming algorithm. Our algorithm uses a recentered projected gradient approach thereby obviating *a priori* knowledge of the optimal objective function value. Assuming primal and dual nondegeneracy, we prove that our algorithm converges. We present computational comparisons between our algorithm and the revised simplex method. For small, dense constraint matrices we saw little difference between the two methods.

Key Words. Linear programming, Karmarkar's algorithm, Projected gradient methods, Least squares.

1. Introduction. This paper proposes a modification to Karmarkar's original algorithm [6] for solving linear programs. Our algorithm is formulated in the positive orthant instead of the simplex. This makes it easier to conceptualize and leads to computational simplicity. Karmarkar's sliding-objective-function method is replaced by a projected gradient search for the optimum. Empirically, this leads to a decrease in the number of iterations the algorithm requires to solve a problem.

In describing our algorithm, we show how to start it, when to stop it, and how to identify infeasibility and unboundedness easily. Assuming primal and dual nondegeneracy, we prove convergence to the optimal solution. (In practice, the algorithm works equally well on problems not satisfying these assumptions.) We also show that duality plays an important role.

Finally, we present results comparing the performance of our algorithm with the revised simplex method for problems with fewer than 200 variables and randomly generated, dense constraint matrices. For this class of problems we saw little difference between the two methods. However, our results may not be indicative of the behavior for large sparse problems.

Some recent papers on Karmarkar's algorithm are [2], [3], [8], [9], and [10]. This algorithm has been independently proposed in [2] and [3]. No convergence proof was presented in [3]. There is a convergence proof in [2] that is substantially different from the one presented here.

2. Interior Methods. Any linear program can be transformed into the following *canonical* form:

[1] AT&T Bell Laboratories, Holmdel, NJ 07733, USA.

Received June 18, 1985; revised June 16, 1986. Communicated by Nimrod Megiddo.

$$\text{find} \quad x^* \in \Omega = \{x \geq \mathbf{0}: \mathbf{A}x = b\} \quad \text{such that} \tag{1}$$
$$c \cdot x^* = \min_{x \in \Omega} c \cdot x,$$

where $\mathbf{A}$ is a given $m \times n$ *constraint matrix*, c is a given *cost vector*, and $c \cdot x$ denotes the inner product between the vectors c and x. Any such point x^* will be called *optimal*. The linear function $c \cdot x$ is called the *objective function*. The polytope Ω is called the *feasible region*. If Ω is nonempty then the problem is *feasible*. If $\min_{x \in \Omega} c \cdot x > -\infty$, the problem is *bounded*.

In the absence of degeneracy, the set $\Omega^0 = \{x > \mathbf{0}: \mathbf{A}x = b\}$ is the interior of Ω relative to the affine space $\{x: Ax = b\}$. We assume that we are given an *interior feasible point* $x^0 \in \Omega^0$. In practice, this can be done by adding one artificial variable to the matrix $\mathbf{A}$. Indeed, if we put

$$\mathbf{A}' = [\mathbf{A} \vdots b - \mathbf{A}\mathbf{1}],$$

where $\mathbf{1}$ denotes the vector of all ones, then the $(n+1)$-dimensional vector of all ones satisfies $\mathbf{A}'\mathbf{1} = b$. By assigning infinite cost (i.e., big M) to the new variable x_{n+1}, we can assure that it will vanish at the optimum. In fact, if it does not, then the original problem is infeasible. If it does vanish, then a solution of the enlarged problem is also a solution of the original problem.

When considering interior methods for linear programming, one immediately thinks of going in the gradient direction. However, this will give a substantial reduction in the objective function only if the current feasible point is somewhat centered in the polytope so that all the walls are sufficiently distant. Therefore, we are led to consider a method that alternates between "centering" the feasible point and taking a step in the gradient direction. Many existing interior methods are of this type. The efficiency and the elegance of any particular method depends on the choice of the centering scheme. In the rest of the paper we consider one special centering scheme.

3. Centering Scheme. The centering scheme that we use can be described as follows. Since the walls of the polytope arise from the inequality constraints $x \geq \mathbf{0}$, it makes sense to change units in each variable so that the current feasible point x^0 becomes the vector of all ones. In these new coordinates, the constraint matrix $\mathbf{A}$ and the cost vector c are changed but the inequality constraints $x \geq \mathbf{0}$ are not. Hence, in these coordinates, our feasible point is at least one unit away from every wall of the polytope.

From a physicist's point of view, the above centering scheme must sound particularly nice. After all, taking the gradient of a function where the various coordinates have different units is unreasonable. But the above centering scheme first makes all coordinate directions unitless and consequently they play an equal role.

We now describe the details. Given an interior feasible point x^0, we make the following change of units

$$y = \mathbf{D}_{x^0}^{-1} x, \tag{2}$$

where $\mathbf{D}_x$ denotes the diagonal matrix containing the components of x. Clearly x^0 is mapped to $y^0 = \mathbf{1}$. In terms of y coordinates, we get a new linear program with $\mathbf{A}$ and c replaced by

$$\tilde{\mathbf{A}} = \mathbf{A}\mathbf{D}_{x^0}, \tag{3}$$

$$\tilde{c} = \mathbf{D}_{x^0} c. \tag{4}$$

To obtain the maximum rate of decrease of the transformed objective function, we would move in the negative of the gradient direction, $\tilde{c}$. However, to preserve the equality constraints we project $\tilde{c}$ onto the null space of $\tilde{\mathbf{A}}$:

$$\tilde{c}_p = \mathbf{P}_{x^0} \tilde{c}, \tag{5}$$

where $\mathbf{P}_x$ denotes the projection onto the null space of $\mathbf{A}\mathbf{D}_x$.

Next, we move from $y^0 = \mathbf{1}$ to y^1 in the direction of $-\tilde{c}_p$. The step size is chosen so that $y^1 > \mathbf{0}$:

$$y^1 = \mathbf{1} - \alpha \tilde{c}_p / \max_i (e_i \cdot \tilde{c}_p), \tag{6}$$

where $\alpha \in (0, 1)$, and e_i is the ith unit vector. If $\tilde{c}_p \leq \mathbf{0}$, the problem is unbounded (see Proposition 1). It is easy to see that y^1 is an interior feasible point in the unitless coordinate system. Finally, we map back to get a new interior feasible point for (1):

$$x^1 = \mathbf{D}_{x^0} y^1. \tag{7}$$

Substituting (4), (5), and (6) into (7), we get

$$x^1 = T(x^0),$$

where

$$T(x) = x - \frac{\alpha}{\gamma} \mathbf{D}_x \mathbf{P}_x \mathbf{D}_x c, \tag{8}$$

and

$$\gamma = \max_i (e_i \cdot \mathbf{P}_x \mathbf{D}_x c). \tag{9}$$

The algorithm consists of generating the sequence of points

$$x^{k+1} = T(x^k). \tag{10}$$

PROPOSITION 1. *Letting* $\tilde{c}_p(x) = \mathbf{P}_x\mathbf{D}_x c$, *the following properties hold*:

(a) *If the objective function is not constant on* Ω, *then the sequence* $c \cdot x^k$ *is strictly decreasing.*
(b) *If, for some* $x \in \Omega^0$, *we have* $\tilde{c}_p(x) \leq \mathbf{0}$ *and* $\tilde{c}_p(x) \neq \mathbf{0}$, *then the problem is unbounded.*
(c) *If, for some* $x \in \Omega^0$, *we have* $\tilde{c}_p(x) = \mathbf{0}$, *then every feasible point is optimal.*

PROOF. Property (a) follows from the fact that the algorithm is a projected gradient method. Since $\tilde{c}_p$ is the projected gradient in the rescaled coordinate system, the hypotheses in (b) clearly imply that the ray emanating from the vector $\mathbf{1}$ in the direction of $-\tilde{c}_p$ never leaves the positive orthant. Since this a direction of strict decrease, the problem is unbounded. Finally, the hypothesis in (c) implies that every point in the rescaled polytype is optimal which, of course, implies the same thing for Ω. □

4. Convergence. In this section we prove, under certain assumptions, that the algorithm converges to the optimal solution. To show this, we introduce dual variables and implicitly reconstruct duality theory.

ASSUMPTIONS

(1) *The problem is bounded and feasible.*
(2) *The problem is primal nondegenerate.*
(3) *The problem is dual nondegenerate.*

One can show that assumptions (2) and (3) taken together are equivalent to the following:

(2′) *The matrix* $\mathbf{A}\mathbf{D}_x$ *has full row rank for every* $x \in \Omega$.
(3′) *The vector* $c - \mathbf{A}^T w$ *has at most* m *zeros for every* $w \in R^m$.

It follows from (2′) that $(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}$ exists for all $x \in \Omega$, and from this it can be shown that it is continuous in x. Finally, it can be shown that assumptions (1) and (2′) imply the existence of an interior feasible solution.

Before proceeding, we will give a formula for projection matrices. If $\mathbf{Q}$ is any full row rank matrix, the projection onto the null space of $\mathbf{Q}$ is $\mathbf{I} - \mathbf{Q}^T(\mathbf{Q}\mathbf{Q}^T)^{-1}\mathbf{Q}$. Since $\mathbf{P}_x$ is the projection onto the null space of $\mathbf{A}\mathbf{D}_x$, we have

$$\mathbf{P}_x = \mathbf{I} - \mathbf{D}_x\mathbf{A}^T(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}_x.$$

From this we see by direct substitution that

$$\tilde{c}_p = \mathbf{P}_x\mathbf{D}_x c = \mathbf{D}_x r, \tag{11}$$

where

$$r = c - \mathbf{A}^T w,$$

and

$$w = (\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}_x^2 c. \tag{12}$$

In terms of $\tilde{c}_p$, r, and w, we may rewrite $T(x)$ in the following ways:

$$T(x) = x - \frac{\alpha}{\gamma}\mathbf{D}_x\tilde{c}_p \tag{13}$$

$$= x - \frac{\alpha}{\gamma}\mathbf{D}_x^2 r$$

$$= x - \frac{\alpha}{\gamma}\mathbf{D}_x^2(c - \mathbf{A}^T w).$$

We call r the vector of *reduced costs* corresponding to x, and w is the vector of *dual variables* corresponding to x. To emphasize the dependence of r and w on x we will occasionally write $r(x)$ and $w(x)$ instead of r and w. We will also denote the ith element of $r(x)$ by $r_i(x)$.

Note that if x is a vertex, then (12) reduces to the usual formula for dual variables. Namely, $w^T = c_B^T\mathbf{B}^{-1}$ where $\mathbf{B}$ is the submatrix of $\mathbf{A}$ consisting of the basic columns (i.e., columns for which the corresponding components of x are nonzero) and c_B is the vector of the corresponding components of c.

We start by showing that the limit points of the sequence x^k, $k = 0, 1, 2, \ldots$, belong to the set

$$S = \{x \in \Omega: \mathbf{D}_x r(x) = \mathbf{0}\}. \tag{14}$$

Assumptions (2′) and (3′) imply that S is the set of vertices of Ω. Indeed, assumption (3′) implies that $r(x)$ has at most m zeros. But if x is not a vertex, then it follows from Assumption (2′) that x has at most $n-m-1$ zeros and so $D_x r \neq 0$. The fact that every vertex x belongs to S can easily be seen by partitioning according to the vanishing and nonvanishing components of x and doing simple block matrix manipulations.

PROPOSITION 2. $\lim_{k\to\infty} \mathbf{D}_{x^k} r(x^k) = \mathbf{0}$.

PROOF. Since $c \cdot x^k$ is monotonically decreasing and bounded below, $c \cdot x^k$ converges. Hence its difference sequence tends to zero:

$$0 = \lim_{k\to\infty}(c \cdot x^k - c \cdot x^{k+1}) = \lim_{k\to\infty} \frac{\alpha \|\tilde{c}_p(x^k)\|_2^2}{\gamma(x^k)}, \tag{15}$$

where we have written $\tilde{c}_p(x)$ and $\gamma(x)$ to emphasize the dependence of $\tilde{c}_p$ and γ on x. Since for any vector y, $\max_i y_i \leq \|y\|_2$, it follows by taking $y = \tilde{c}_p(x^k)$ that

$$0 \leq \|\tilde{c}_p(x^k)\|_2 \leq \frac{\|\tilde{c}_p(x^k)\|_2^2}{\gamma(x^k)}.$$

Since the right-hand side tends to zero, we see that

$$\lim_{k\to\infty} \mathbf{D}_{x^k} r(x^k) = \lim_{k\to\infty} \tilde{c}_p(x^k) = \mathbf{0}. \qquad \square$$

We will say that S consists of the points in Ω for which *complementary slackness* holds. We also need to introduce the set

$$F = \{x \in \Omega\colon r(x) \geq 0\}.$$

We say that F consists of those points for which *dual feasibility* holds.

PROPOSITION 3. *If $x \in S \cap F$, then x is optimal.*

This theorem is a standard result from duality theory (see, e.g., [7], p. 51 or [4], p. 62). We include a proof here as a warm-up for the discussion of stopping rules in Section 5.

PROOF. The fact that $x \in S$ allows us to establish the third equality in the following chain:

$$\begin{aligned} c \cdot x &= c \cdot \mathbf{D}_x \mathbf{1} \\ &= \mathbf{D}_x c \cdot \mathbf{1} \\ &= \mathbf{D}_x \mathbf{A}^T w \cdot \mathbf{1} \\ &= w \cdot \mathbf{A}\mathbf{D}_x \mathbf{1} \\ &= w \cdot \mathbf{A}x \\ &= w \cdot b. \end{aligned}$$

Now suppose that z is any point in Ω. Then, since $x \in F$, we have

$$\begin{aligned} c \cdot z &\geq \mathbf{A}^T w(x) \cdot z \\ &= w(x) \cdot \mathbf{A}z \\ &= w(x) \cdot b. \end{aligned}$$

Combining these two chains we see that

$$c \cdot z \geq w(x) \cdot b = c \cdot x$$

for any $z \in \Omega$. Hence x is optimal. $\square$

PROPOSITION 4. *If the sequence $\{x^k\}$ is convergent, then $x^\infty = \lim x^k$ is optimal.*

PROOF. Suppose, to the contrary, that

$$\lim_{k\to\infty} c \cdot x^k > \min_{\Omega} c \cdot x. \tag{16}$$

By the continuity of $r(x)$ and Proposition 2, we see that

$$\mathbf{D}_{x^\infty} r(x^\infty) = \lim_{k\to\infty} \mathbf{D}_{x^k} r(x^k) = \mathbf{0}$$

so that $x^\infty \in S$. By (16) and Proposition 3, it follows that $x^\infty \notin F$, hence there exists an index i such that $r_i(x^\infty) < 0$. Since $x^\infty \in S$, we see that $x_i^\infty = 0$. By the continuity of $r(x)$, it follows that there exists an integer K such that, for every $k \geq K$, $r_i(x^k) < 0$. From (13) we see that

$$x_i^{k+1} = x_i^k - \alpha\gamma(x^k)^{-1}(x_i^k)^2 r_i(x^k).$$

Since $(x_i^k)^2 r_i(x^k) < 0$, we find that $x_i^{k+1} > x_i^k > 0$ for every $k \geq K$, which contradicts the fact that $x_i^k \to x_i^\infty = 0$. □

PROPOSITION 5. *The sequence x^k is convergent.*

PROOF. Our boundedness and nondegeneracy assumptions imply that $\Omega \cap \{x: c \cdot x \leq const\}$ is compact for any value of *const*. Hence, the sequence $\{x^k\}$ has a limit point $\hat{x}$. Proposition 2 together with the continuity of $r(x)$ shows that $\hat{x}$ belongs to S and hence is a vertex. Let N denote the set of nonbasic indices of $\hat{x}$ (i.e., indices of the vanishing components). For $\delta > 0$, let

$$B_\delta = \{x \in \Omega: x_N < \delta \mathbf{1}\}.$$

It follows from Assumption (3′) that we can fix an ε satisfying

$$0 < \varepsilon < \min_{i \in N} |r_i(\hat{x})|.$$

Since S is a finite set and $r(x)$ is continuous, there is a $\delta > 0$ such that

$$B_{2\delta} \cap S = \hat{x}$$

and

$$\min_{i \in N} |r_i(x)| > \varepsilon \qquad \text{for all} \quad x \in B_\delta. \tag{17}$$

Substituting (11) into (15), we see that

$$\lim_{k \to \infty} \gamma(x^k)^{-1} \|\mathbf{D}_{x^k} r(x^k)\|_2^2 = 0$$

and hence

$$\lim_{k \to \infty} \gamma(x^k)^{-1} [x_i^k r_i(x^k)]^2 = 0.$$

Consider the set

$$C_{\varepsilon,\delta} = \{x \in B_\delta: \alpha\gamma(x)^{-1} x_i^2 r_i(x)^2 < \varepsilon\delta \text{ for all } i \in N\}.$$

It follows from the continuity of $r(x)$ that $x \in C_{\varepsilon,\delta}$ for every $x \in B_\delta$ sufficiently close to $\hat{x}$. Fix $x \in C_{\varepsilon,\delta}$. Since $C_{\varepsilon,\delta} \subset B_\delta$, we see from (17) that

$$\alpha\gamma(x)^{-1}x_i^2|r_i(x)| < \delta \qquad \text{for all} \quad i \in N.$$

Now from (13) it follows that

$$T_i(x) = x_i - \alpha\gamma(x)^{-1}x_i^2 r_i(x) < 2\delta \qquad \text{for all} \quad i \in N$$

so that $T(C_{\varepsilon,\delta}) \subset B_{2\delta}$.

Now, since $\hat{x}$ is a limit point, x^k belongs to $C_{\varepsilon,\delta}$ infinitely often. Suppose at this point that the sequence x^k has two or more limit points. Then the sequence x^k must leave $C_{\varepsilon,\delta}$ infinitely often. Each time it leaves, it belongs to $B_{2\delta} \setminus C_{\varepsilon,\delta}$. Since this set has compact closure, we see that the subsequence of x^k belonging to $B_{2\delta} \setminus C_{\varepsilon,\delta}$ must have a limit point. But every limit point must belong to S which is disjoint from the closure of $B_{2\delta} \setminus C_{\varepsilon,\delta}$. This is a contradication and consequently there can be only one limit point. □

5. Stopping Rules. In the previous section we described the behavior of the algorithm as it approaches a limit point. In this section we will analyze in more detail this behavior in a small neighborhood around a limit point. This information is used to establish stopping rules for our algorithm.

For $\gamma \geq 0$, let

$$S_\gamma = \{x \in \Omega: \mathbf{D}_x r \leq \gamma \mathbf{1}\}.$$

We say that points in S_γ satisfy *γ-complementary slackness.* If $x \in S_\gamma$ then

$$\begin{aligned} c \cdot x &= c \cdot \mathbf{D}_x \mathbf{1} \\ &= \mathbf{D}_x c \cdot \mathbf{1} \\ &\leq \mathbf{D}_x \mathbf{A}^T w \cdot \mathbf{1} + \gamma \mathbf{1} \cdot \mathbf{1} \\ &= w \cdot \mathbf{A}x + \gamma n \\ &= w(x) \cdot b + \gamma n. \end{aligned} \tag{18}$$

Let

$$F_\delta = \{x \in \Omega: c \geq \mathbf{A}^T w - \delta \mathbf{1}\}.$$

We say that points in F_δ satisfy *δ-dual feasibility.* If $x \in F_\delta$, then for every $z \in \Omega$ we have

$$\begin{aligned} c \cdot z &\geq \mathbf{A}^T w(x) \cdot z - \delta \mathbf{1} \cdot z \\ &= w(x) \cdot \mathbf{A}z - \delta \mathbf{1} \cdot z \\ &= w(x) \cdot b - \delta n \bar{z}, \end{aligned}$$

where $\bar{z} = (1/n)\mathbf{1} \cdot z$. Therefore,

$$\min_{\Omega} c \cdot z \geq w(x) \cdot b - \delta n \max_{\Omega} \bar{z}. \tag{19}$$

Combining (18) and (19) we see that if $x \in S_\gamma \cap F_\delta$ then

$$c \cdot x \leq \min_{\Omega} c \cdot z + \gamma n + \delta n M, \tag{20}$$

where

$$M \geq \max_{\Omega} \bar{z}.$$

If Ω is bounded, then M is finite. Note that this assumption is stronger than the boundedness assumption in the previous section.

For any $x \in \Omega$, we can measure the degree of complementary slackness and dual feasibility by putting

$$\gamma(x) = \max_i x_i r_i(x)$$

$$\delta(x) = -\min_i r_i(x).$$

Note that this definition of γ agrees with the definition given earlier in formula (9). We can now formulate an effective stopping rule:

PROPOSITION 6. *Fix $\varepsilon > 0$. If M is an upper bound on $\bar{z}$, $z \in \Omega$, and if we stop at the first iteration at which*

$$\gamma(x^k) + \delta(x^k) M \leq \varepsilon / n$$

then

$$c \cdot x^k \leq \min_{\Omega} c \cdot z + \varepsilon.$$

PROOF. It is not hard to show that $\delta(x^k) \geq 0$ and that $x^k \in S_{\gamma(x^k)} \cap F_{\delta(x^k)}$. Hence the proposition follows from formula (20). □

To implement this stopping rule, we need a value for M. A conservative approach would be to choose a very large number. However, it may be better to choose M dynamically by simply setting $M(x) = \bar{x}$. This makes the precise inequality in Proposition 6 into an approximate inequality.

From (18) we see that $w(x) \cdot b$ is an estimate of the optimal value of the objective function. For the rest of this section we show that this tends to be a better estimate than $c \cdot x$. For practical purposes, it may be useful to monitor the duality gap, $c \cdot x - w \cdot b$, together with the degree of dual feasibility, $\delta(x)$, and stop when both are sufficiently small. For the remainder of this section we impose Assumptions (1)-(3) of the previous section.

Let x^* be the optimal vertex. Let N denote the $n-m$ indices of x^* that correspond to vanishing components and let B denote the remaining m indices. Using this partition of the indices, we can partition our matrices and vectors as follows:

$$\mathbf{A}=[\mathbf{B}\vdots\mathbf{N}], \qquad x=\begin{bmatrix} x_B \\ x_N \end{bmatrix}, \qquad c=\begin{bmatrix} c_B \\ c_N \end{bmatrix}.$$

For any $x\in\Omega$, it follows from $\mathbf{A}x=b$ that $x_B=\mathbf{B}^{-1}(b-\mathbf{N}x_N)$. Hence, we can write

$$c\cdot x=c_B\cdot\mathbf{B}^{-1}b+r_N\cdot x_N, \tag{21}$$

where

$$r_N=c_N-(\mathbf{B}^{-1}\mathbf{N})^Tc_B.$$

The optimal vertex x^* corresponds to $x_N^*=\mathbf{0}$ and $x_B^*=\mathbf{B}^{-1}b$. Hence we see that $c_B\cdot\mathbf{B}^{-1}b$ is the optimal value of the objective function.

We will now derive an expression for $w(x)\cdot b$ that is valid for x in a neighborhood of x^*. Simple block matrix algebra shows that

$$w(x)\cdot b=(c_B^T\mathbf{D}_{x_B}^2\mathbf{B}^T+c_N^T\mathbf{D}_{x_N}^2\mathbf{N}^T)(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T+\mathbf{N}\mathbf{D}_{x_N}^2\mathbf{N}^T)^{-1}b. \tag{22}$$

For x_N sufficiently close to $\mathbf{0}$, we can expand the inverted expression in a geometric series to get

$$(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T+\mathbf{N}\mathbf{D}_{x_N}^2\mathbf{N}^T)^{-1}=(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T)^{-1}-(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T)^{-1}\mathbf{N}\mathbf{D}_{x_N}^2\mathbf{N}^T(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T)^{-1}+o(\|x_N\|^2).$$

Substituting this into (22) and simplifying, we get

$$w(x)\cdot b=c_B\cdot\mathbf{B}^{-1}b-r_N\cdot\mathbf{D}_{x_N}^2\mathbf{N}^T(\mathbf{B}\mathbf{D}_{x_B}^2\mathbf{B}^T)^{-1}b+o(\|x_N\|^2). \tag{23}$$

Comparing formulas (21) and (23), we see that there is a neighborhood of x^* in which $w(x)\cdot b$ is closer to $c\cdot x^*$ than $c\cdot x$ is. Note that this neighborhood can be small if the problem is nearly degenerate.

6. Upper Bound Constraints. The centering scheme may easily be applied to problems with upper bound constraints $x_i\le u_i$. We wish to scale the problem in the transformed space to reflect how close x_i is to either 0 or u_i. Hence, the matrix $\mathbf{D}_x$ is now defined such that the ith diagonal element of $\mathbf{D}_x$ is $\min(x_i, u_i-x_i)$. The direction to move is still $\mathbf{D}_x\tilde{c}_p$ and the step size is chosen to preserve both the nonnegativity and upper bound constraints. In summary, the modifications to the algorithm are

$$\mathbf{D}_x=\operatorname{diag}(\min(x_i, u_i-x_i)),$$

$$\gamma(x)=\max_i\left(\max\left(\frac{e_i\cdot\mathbf{D}_x\tilde{c}_p}{x_i}, -\frac{e_i\cdot\mathbf{D}_x\tilde{c}_p}{u_i-x_i}\right)\right).$$

Results analogous to Propositions 1-5 are proved in a similar fashion for upper bound constraints. Here, the sets $S = \{x \in \Omega: \mathbf{D}_x r = \mathbf{0}\}$ and $F = \{x \in \Omega: \mathbf{S}_{u/2-x} r \geq 0\}$, where $\mathbf{S}_x$ is the diagonal matrix containing the signs of the components of x, represent complementary slackness and dual feasiblity, respectively. If $x \in S \cap F$, then it is easy to show that x is optimal (cf. Proposition 3) under Assumptions (1)-(3) of Section 4.

7. Algebraic Comparison with the Simplex Method. There are significant parallels, and differences, between our algorithm and the simplex method. We have found the relationship between the two algorithms noteworthy. The parallels are summarized in Table 1.

Both algorithms calculate *reduced costs* using the formula $r = c - \mathbf{A}^T(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}_x^2 c$. This is the vector of residuals from the weighted least-squares solution to the problem of minimizing the function $\|\mathbf{D}_x(c - \mathbf{A}^T w)\|_2$ with respect to w.

Using the reduced costs, both algorithms generate a direction p in which to move, so that $\mathbf{A}p = \mathbf{0}$. Both algorithms move from x to $x + (\alpha/\gamma)p$, where γ is related to the min-ratio test. In the simplex method, $\alpha = 1$, whereas in our algorithm $0 < \alpha < 1$.

Table 1. Comparison between simplex and our algorithm.

Simplex method	Our algorithm
(1) Initial x is a feasible vertex. $x = \begin{bmatrix} x_B \\ \cdots \\ \mathbf{0} \end{bmatrix}$, $c = \begin{bmatrix} c_B \\ \cdots \\ c_N \end{bmatrix}$, $\mathbf{A} = [\mathbf{B} \vdots \mathbf{N}]$.	(1) Initial x is a feasible, strictly interior solution ($x > \mathbf{0}$).
(2) Calculate reduced costs: $r = c - \mathbf{A}^T w = c - \mathbf{A}^T(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}_x^2 c$ $= \begin{bmatrix} 0 \\ \cdots\cdots\cdots \\ c_N - (\mathbf{B}^{-1}\mathbf{N})^T c_B \end{bmatrix}$.	(2) Calculate reduced costs: $r = c - \mathbf{A}^T w$ $= c - \mathbf{A}^T(\mathbf{A}\mathbf{D}_x^2\mathbf{A}^T)^{-1}\mathbf{A}\mathbf{D}_x^2 c$.
(2a) If $r \geq 0$, then x is optimal.	(2a) No corresponding *exact* stopping rule. See Proposition 6 for an ε-optimal stopping rule.
(3) Use r to find a direction, p, in null space of $\mathbf{A}$ that will decrease the objective function: choose i such that $r_i < 0$, and put $p = e_i - \begin{bmatrix} \mathbf{B}^{-1}\mathbf{A}e_i \\ \mathbf{0} \end{bmatrix}$.	(3) Use r to find a direction, p, in null space of $\mathbf{A}$ that will decrease the objective function: $p = -\mathbf{D}_x^2 r$.
(4) New feasible vertex is $x + \frac{p}{\max_{\{i: x_i > 0\}}(p_i/x_i)}$.	(4) New feasible, interior, point is $x + \frac{\alpha p}{\max_i(p_i/x_i)}$, where $0 < \alpha < 1$.

The main difference lies in how the reduced costs are used to generate a direction p. The simplex method dismisses some information contained in the reduced costs, while our algorithm does not. The price paid is that each iteration of the simplex method takes order mn computations, while each iteration of our algorithm takes order m^2n computations.

8. Computational Experience. We have coded our algorithm and the revised simplex method in Pascal. Both algorithms solve the canonical linear program (1).

To get an interior feasible point for our algorithm, the procedure adds a column to the **A** matrix as described in Section 2. At each iteration we check to see if the variable that would become zero by taking a step all the way to the wall (i.e., $\alpha = 1$) is the artificial variable. If it is, we set $\alpha = 1$ and in one step get an interior feasible point for the original problem. If it is not, then we set $\alpha = 0.97$.

We have only considered dense problems in our comparisons. For such problems, perhaps the best algorithm for computing the vector w of dual variables is the *QR* algorithm for solving least-squares problems. This is the algorithm we have implemented. We have borrowed this code from [1] to which we refer the reader for details.

To get a feasible vertex for the revised simplex method, we add an $m \times m$ identity matrix to the **A** matrix and assign "infinite" cost to each of the m added variables. In both algorithms we have used 10^6 for infinity.

We generated random problems as follows. The elements of the constraint matrix **A** are independent real values that are uniform on the interval $[0, 1)$. All random variables were generated using the *random* procedure in Pascal. Choosing all elements of **A** to be nonnegative, guarantees that the problem is bounded. The elements of the cost vector c were also independent and uniform on $[0, 1)$. The vector b was generated by putting $b = \mathbf{A}x$ where x is a randomly generated positive vector. This guarantees that the problem is feasible. For the stopping rule in our algorithm, we chose $\varepsilon = 10^{-3}$.

To get an idea of how our algorithm will do on larger problems, we generated 137 random problems. The first 100 had m uniform on $\{1, 2, \ldots, 30\}$ and n uniform on $\{m+1, m+2, \ldots, 190\}$. The last 37 problems had m uniform on $\{1, 2, \ldots, 60\}$ and n uniform on $\{m+1, m+2, \ldots, 90\}$. We collected the time and the number of iterations for each problem and for each algorithm. We did a regression on the logarithm of the time (number of iterations) as a linear function of the logarithms of m and n. The results are shown in Table 2. The regression is based on data collected from the computer programs, running on a Tandy 2000 computer. The most important observation is that, for the regression on run time,

Table 2. Performance comparison for dense constraint matrices.

Method	Time (minutes)	Iterations
Simplex method	$0.000119m^{2.1122}n^{0.9681}$	$0.7159m^{0.9522}n^{0.3109}$
Our algorithm	$0.001287m^{1.3602}n^{1.2391}$	$7.3885m^{-0.0187}n^{0.1694}$

the sum of the exponents in our algorithm is 0.47 less than the analogous sum for the simplex method. Roughly speaking, this says that to get a relative improvement of a factor of 10 of our algorithm over the simplex method one must increase m and n by a factor of 100. Two final observations are that the extrapolated crossover point is sensitive to the shape of the constraint matrix, and typically lies beyond the range of our data. As examples, if $m = \frac{1}{10}n$, then the extrapolated crossover point occurs when $n = 6250$, whereas if $m = \frac{1}{2}n$, it is 480. Lastly, these results may not apply to large sparse problems or problems with special structure. Further algorithmic work and computational testing is needed before conclusions regarding the practical impact of Karmarkar's algorithm can be drawn.

Acknowledgments. We thank N. Karmarkar and M. J. Todd (Cornell University) for helpful discussions. In particular, Proposition 5 is due to M. J. Todd. We are also grateful to N. Megiddo for carefully reading the paper and for suggesting several significant improvements.

References

[1] L. V. Atkinson and P. J. Harley, *An Introduction to Numerical Methods with Pascal*, Addison-Wesley, Reading, MA, 1983.

[2] E. R. Barnes, A variation on Karmarkar's algorithm for solving linear programming problems, Manuscript, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1985.

[3] T. M. Cavalier and A. L. Soyster, Some computational experience and a modification of the Karmarkar algorithm, ISME Working Paper 85-105, The Pennsylvania State University, 1985.

[4] V. Chvátal, *Linear Programming*, Freeman, New York and San Francisco, 1983.

[5] S. C. Eisenstat, Efficient implementation of a class of preconditioned conjugate gradient methods, *SIAM J. Sci. Statist. Comput.*, **2** (1985), 4-7.

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

[7] L. S. Lasdon, *Optimization Theory for Large Systems*, Macmillan, New York, 1970.

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

[9] M. J. Todd and B. P. Burrell, An extension of Karmarkar's algorithm for linear programming using dual variables, Technical Report No. 648, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1985.

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