

**DUAL BARRIER FUNCTIONS WITH SUPERFAST RATES
OF CONVERGENCE
FOR THE LINEAR PROGRAMMING PROBLEM**

M. R. OSBORNE¹

(Received 30 January 1986; revised 16 May 1986).

Abstract

It is shown that barrier functions applied to the dual linear program can be modified to give multiplier estimates that converge to the solution of the primal problem. Newton's method is considered for implementing this approach and numerical results presented. It has been shown that there is a connection between these methods and Karmarkar's algorithm, but for the class of problems considered further improvements are still required before those methods become competitive with active set methods.

1. Introduction

Recent developments in linear programming have put an emphasis on methods which (in contrast to the ubiquitous simplex algorithm) do not make direct and explicit use of the geometry of the set of feasible points, and spectacular results have been claimed (and disputed) for one such procedure due to Karmarkar [10]. Investigation of this procedure has thrown up links with barrier functions [6] and with the closely related entropy methods [2]. The present paper considers the application of these methods to the linear programming problem. The main aim is to point out their advantages and difficulties, and to report some (limited) numerical experience.

Consider the linear programming problem

$$\min_{\mathbf{x} \in X} \mathbf{c}^T \mathbf{x}; X = \{\mathbf{x}: A\mathbf{x} \geq \mathbf{b}\} \quad (LP1)$$

¹Department of Statistics, Research School of Social Sciences, Australian National University, G. P. O. Box 4, Canberra, A. C. T. 2601, Australia

© Copyright Australian Mathematical Society 1987, Serial-fee code 0334-2700/87

where $A: R^p \rightarrow R^n$ has full column rank $p(n > p)$. The standard barrier function algorithm for solving this problem and making use of the log barrier function has the form:

- 0) Set $\alpha < 1$, μ_1 , *tol* small, and \mathbf{x}^0 feasible
- i) Starting from \mathbf{x}^k compute

$$\mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} B(\mathbf{x}, \mu_k)$$

where

$$B(\mathbf{x}, \mu) = \mathbf{c}^T \mathbf{x} - \mu \sum_{i=1}^n \log(\mathbf{a}_i^T \mathbf{x} - b_i), \tag{1.1}$$

and \mathbf{a}_i is the i 'th row of A ;

- ii)

$$\mu_{k+1} = \alpha \mu_k,$$

- iii) Stop if

$$\begin{aligned} &\mu_{k+1} < \textit{tol} \\ &\text{else } k := k + 1; \text{ repeat (i)}. \end{aligned}$$

Methods of this kind have been analysed extensively for nonlinear programming problems [4], but it is usual for the successive minimizations in the step equivalent to (i) above to be ill-conditioned and too expensive for this approach to be competitive. However, the situation is somewhat different here.

Barrier functions do have some nice properties [4]. For example

- i) let $\mathbf{x}^k \rightarrow \mathbf{x}^*$ as $\mu_k \rightarrow 0$, then \mathbf{x}^* minimises LP1,
- ii) let

$$u_i^k = \mu_k / r_i(\mathbf{x}^k), \quad i = 1, 2, \dots, n, \tag{1.2}$$

where $r_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - b_i$. Then

$$u_i^k \rightarrow u_i^*, \quad i = 1, 2, \dots, n$$

where u_i^* is a Kuhn–Tucker multiplier for the i 'th constraint.

REMARK 1.1. Let σ be an index set defined by

$$\sigma = \{i: r_i(\mathbf{x}^*) = 0\}. \tag{1.3}$$

If $|\sigma| \geq p$ and $\dim\{\mathbf{a}_i, i \in \sigma\} = p$ then \mathbf{x}^* is a vertex of the feasible region X . In this case

$$\lim_{\mu_k \rightarrow 0} \mu_k \nabla^2 B(\mathbf{x}^k, \mu_k) = \sum_{i \in \sigma} (u_i^*)^2 \mathbf{a}_i \mathbf{a}_i^T \tag{1.4}$$

is nonsingular. In particular the limiting Hessian has a finite condition number. This result is in marked contrast to the usual result in the nonlinear programming problem [5].

Rate of convergence results for the nonlinear programming problem are also known [8]. These are usually derived under the restrictive conditions of

- (a) strict complementarity ($u_i^* > 0$ if $r_i(x^*) = 0$), and
- (b) second order sufficiency.

They ensure that $x^k \rightarrow x^*$ like $O(\mu_k)$, and are necessary in the sense that if strict complementarity is relaxed then the convergence rate drops to exactly $O(\mu_k^{1/2})$. However, for LP1 second order sufficiency cannot hold, and strict complementarity is equivalent to uniqueness in the usual case that $|\sigma| = p$ [13]. The nonunique case has some interest, and the main results are summarised below.

THEOREM 1.1. *Let LP1 have a connected set of optima (not a singleton) bounded by vertices of the feasible region X . Let $x^k \rightarrow x^*$. Then*

- (i) x^* is an interior point of the optimal set,
- (ii) the rate of convergence is $O(\mu_k)$, and
- (iii) $\mu_k \nabla^2 B(x^k, \mu_k) \rightarrow$ singular matrix as $\mu_k \rightarrow 0$.

For the standard linear programming algorithms, degeneracy is more of a problem than nonuniqueness. This occurs when $|\sigma| > p$ (only the optimal vertex is relevant here), and typically rate of convergence behaviour of barrier function methods is unaffected.

EXAMPLE 1.1. Consider the problem

$$\min e_3^T x$$

subject to

$$\begin{bmatrix} 1 & 0 & 1 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix} x \geq 0.$$

This problem has a degenerate optimum ($|\sigma| = 4$) at $x = 0$. To determine $x(\mu)$ set

$$0 = \nabla B = e_3^T - \mu \left\{ \frac{1}{x_1 + x_3} (e_1 + e_3)^T + \frac{1}{-x_1 + x_3} (-e_1 + e_3)^T + \frac{1}{x_2 + x_3} (e_2 + e_3)^T + \frac{1}{-x_2 + x_3} (-e_2 + e_3)^T \right\}.$$

This gives

$$\begin{aligned} x_1 = x_2 = 0, \quad x_3 = 4\mu, \\ u_1 = u_2 = u_3 = u_4 = 1/4, \end{aligned}$$

and

$$\mu \nabla^2 B = \frac{1}{16} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}.$$

The problem dual to LP1 is

$$\max_{\mathbf{u} \in U} \mathbf{b}^T \mathbf{u}; U = \{\mathbf{u}: A^T \mathbf{u} = \mathbf{c}, \mathbf{u} \geq 0\}. \tag{LP2}$$

The natural barrier function to consider for this problem is

$$D(\mathbf{u}, \mu) = \mathbf{b}^T \mathbf{u} + \mu \sum_{i=1}^n \log u_i \tag{1.5}$$

subject to

$$A^T \mathbf{u} = \mathbf{c}, \tag{1.6}$$

and the main result connecting these barrier function formulations is as follows.

THEOREM 1.2. *Let \mathbf{x}^k minimize $B(\mathbf{x}, \mu_k)$, and define \mathbf{u}^k by (1.2). Then \mathbf{u}^k maximizes $D(\mathbf{u}, \mu_k)$ subject to the constraints (1.6). Conversely, let \mathbf{u}^k maximize (1.5) subject to (1.6) and set $\pi_s^k = \mu_k / u_s^k, s = 1, 2, \dots, n$, then $\pi_s^k = r_s(\mathbf{x}^k)$ where \mathbf{x}^k minimizes $B(\mathbf{x}, \mu_k)$.*

PROOF. The necessary conditions for the constrained maximization of $D(\mathbf{u}, \mu_k)$ subject to (1.6) gives

$$\mathbf{b}^T + \sum_{s=1}^n \frac{\mu_k}{u_s^k} \mathbf{e}_s^T = \mathbf{y}^T A^T, \tag{1.7}$$

where \mathbf{y} is the vector of Lagrange multipliers. This equation is satisfied by $\mathbf{y} = \mathbf{x}^k, \pi_s^k = r_s(\mathbf{x}^k)$, and this establishes the desired result as $B(\mathbf{x}, \mu)$ is strictly convex and $D(\mathbf{u}, \mu)$ strictly concave for $\mu > 0$.

REMARK 1.2. This is an important result because it guarantees that

(a) by minimizing $B(\mathbf{x}, \mu)$ we estimate not only \mathbf{x}^* : we simultaneously estimate \mathbf{u}^* maximizing LP2 to the same order of accuracy, and

(b) the multiplier estimate \mathbf{u}^k obtained by minimizing $B(\mathbf{x}, \mu)$ is stable in the sense that it exactly maximizes $D(\mathbf{u}, \mu)$ and these maxima generate a sequence converging to \mathbf{u}^* .

REMARK 1.3. The log barrier function possesses another attractive property. The sequence of minima \mathbf{x}^k obtaining by minimizing $B(\mathbf{x}, \mu_k)$ is independent of the row scaling of the constraint matrix A . The corresponding result for the dual variable \mathbf{u} is

$$\mathbf{a}_i \rightarrow d_i \mathbf{a}_i \Rightarrow u_i \rightarrow u_i / d_i.$$

The challenge to make barrier function methods competitive is offered on at least two fronts:

(a) to use rate of convergence information to accelerate convergence of the method. This is usually attempted using extrapolation techniques [4], [8]. However, in the next section an alternative approach is suggested that appears attractive. Also, in section 3 an alternative barrier function closely related to entropy methods is considered. It offers the potential of even more rapid convergence.

(b) to reduce the amount of computation involved in each minimization. Here an implementation of Newton's method is studied in section 4. This has some nice features (in particular, a line search procedure suggested in [14] proves very satisfactory). However, the barrier function approach appears to have intrinsic difficulties.

To check out these ideas the LP problem generator described in [13] has been used. The numerical results are limited (actually by the capacity of the computer used), but they are not incompatible with the suggestions that this class of methods has potential advantages for very large problems for which the barrier function minimizations are relatively easy. They do confirm that numerical problems can occur in the minimizations, and it does appear that these methods are more subject to ill-conditioning problems than are simplex and its variants.

2. Modified barrier functions

Clearly, if the optimal reference pointed to by the index set σ defined by (1.3) is known then the solutions both to LP1 and LP2 can be obtained by solving

$$A_{\sigma}x^{*} = \mathbf{b}_{\sigma} \quad (2.1)$$

and

$$A_{\sigma}^T \mathbf{u}_{\sigma}^{*} = \mathbf{c} \quad (2.2)$$

where the subscript σ defines the appropriate submatrices and vectors. Thus it is necessary to continue the sequential minimizations only until the optimal reference can be identified with sufficient confidence. It turns out that a modification of the basic barrier function assists this. Let \mathbf{v}^{k+1} be an estimate of the multiplier vector and define

$$M^k(\mathbf{x}) = \mathbf{c}^T \mathbf{x} - \mu_k \sum_{i=1}^n w_i v_i^{k-1} \log r_i, \quad (2.3)$$

where $w_i > 0$ is chosen to transform under row scaling according to

$$\mathbf{a}_i \rightarrow d_i \mathbf{a}_i \Rightarrow w_i \rightarrow d_i w_i,$$

and where the initial multiplier estimate transforms according to

$$\mathbf{a}_i \rightarrow d_i \mathbf{a}_i \Rightarrow v_i^0 \rightarrow v_i^0/d_i.$$

Suitable choices include $w_i = \|\mathbf{a}_i\|$, $v_i^0 = 1/\|\mathbf{a}_i\|$, $i = 1, 2, \dots, n$. The main step of the barrier algorithm becomes

$$\mathbf{x}^k = \arg \min_{\mathbf{x}} M^k(\mathbf{x}), \quad v_i^k = \mu_k w_i v_i^{k-1} / r_i(\mathbf{x}^k). \tag{2.4}$$

This procedure was suggested in [12] and analysed for general nonlinear programming problems in [9]. The main result in the case that LP1 has a unique solution is an immediate adaptation of the results of [9].

THEOREM 2.1. *Let LP1 have a unique solution with $|\sigma| = p$, and let $\{\mu_k\} \rightarrow 0$. Then $\mathbf{x}^k \rightarrow \mathbf{x}^*$, $\mathbf{v}^k \rightarrow \mathbf{v}^*$, and the rate of convergence is given by*

$$\|\mathbf{x}^k - \mathbf{x}^*\| = O(\mu_k), \quad \|\mathbf{v}^k - \mathbf{v}^*\| = O\left(\prod_1^k \mu_s\right).$$

REMARK 2.1. The feature of greatest interest is the superfast convergence of the multiplier estimates (like $\mu_0^k \alpha^{k^2/2}$ against $\mu_0 \alpha^k$ for the unmodified barrier function). The proof of the rate of convergence result follows on noting that the estimate is appropriate for the components of \mathbf{x}^k associated with the inactive constraints ($r_s \rightarrow 0$). But because \mathbf{v}^k is dual feasible by definition (from (2.4))

$$A_\sigma^T(\mathbf{v}_\sigma^k - \mathbf{u}_\sigma^*) = -A_\sigma^T \mathbf{v}_\sigma^k \tag{2.5}$$

where σ^c is the complementary index set. The result now follows because the estimate is correct for the inactive constraints.

The key point is that $\mathbf{v}^k \rightarrow \mathbf{u}^*$ maximizing LP2. Thus the minimizing sequence generated by M^k provides an auxiliary sequence converging superfast to the maximizer of LP2. This result is obtained without any artificial inspection and selection process or the need for an explicit extrapolation procedure and recovers the best result possible for extrapolation procedures.

The barrier function minimized by \mathbf{v}^k follows from (2.4) by inspection. We have

$$\mathbf{v}^k = \arg \max_{\mathbf{v}} N^k(\mathbf{v})$$

subject to (1.6), where

$$N^k(\mathbf{v}) = \mathbf{b}^T \mathbf{v} + \mu_k \sum_{i=1}^n w_i v_i^{k-1} \log(v_i). \tag{2.6}$$

REMARK 2.2. This dual barrier function has the interesting feature that the effective barrier parameter penalising the active constraints is relatively very small because of the weighting by v_i^{k-1} . Experience suggests that the use of small weighting parameters for the constraints tending to zero makes minimization

difficult. Thus it is a further bonus that the minimization of $M^k(\mathbf{x})$ does not involve such small parameters associated with the active constraints and can be expected to be relatively easier.

3. The exponential barrier

In [3] a procedure very similar to that considered in the previous section is developed from entropy methods. However, it can be derived directly by our arguments. Here, instead of a log barrier, exponentials are used to define the penalising objective function associated with LP1. The basic form of this exponential barrier function is

$$E(\mathbf{x}, \mu) = \mathbf{c}^T \mathbf{x} + \mu \sum_{i=1}^n e^{-r_i(\mathbf{x})/\mu w_i} \tag{3.1}$$

where $w_i > 0$ is chosen to make the minimizer of $E(\mathbf{x}, \mu)$ independent of row scaling of the constraint matrix (as before $w_i = \|\mathbf{a}_i\|$ will do). In this case the main step of the algorithm computes

$$\mathbf{x}^k = \arg \min_{\mathbf{x}} E(\mathbf{x}, \mu_k)$$

and

$$u_i^k = \frac{1}{w_i} e^{-r_i(\mathbf{x}^k)/\mu_k w_i}, \quad i = 1, 2, \dots, n. \tag{3.2}$$

Convergence results are essentially the same as for the log barrier function.

THEOREM 3.1. (i) *Let $\{\mu_k\} \rightarrow 0$ and $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$. Then \mathbf{x}^* minimizes LP1 and $\{\mathbf{u}^k\} \rightarrow \mathbf{u}^*$ maximizing LP2.*

(ii) *Let LP1 have a unique solution. Then*

$$\|\mathbf{x}^k - \mathbf{x}^*\| = O(\mu_k), \quad \|\mathbf{u}^k - \mathbf{u}^*\| = O(e^{-\rho/\mu_k}),$$

where

$$\rho = \min\{r_i(\mathbf{x}^*)/w_i, \quad i \in \sigma^c\} \tag{3.3}$$

is independent of the row scaling of A .

PROOF. (i) The standard argument [4] shows that $\{\mathbf{c}^T \mathbf{x}^k\}$ is decreasing and that $\lim\{\mathbf{x}^k\}$ feasible. It is now easy to show that \mathbf{x}^* must minimize LP1.

(ii) For $i \in \sigma$, $u_i^k \rightarrow u_i^* > 0$ only if $r_i(\mathbf{x}^k) = O(\mu_k)$. Thus

$$A_\sigma(\mathbf{x}^k - \mathbf{x}^*) = O(\mu_k)$$

and the first result follows because A_σ has full rank (uniqueness implies $|\sigma| \geq p$ so that \mathbf{x}^* is a vertex of X). The second part follows on noting that

$$A_\sigma^T(\mathbf{u}_\sigma^k - \mathbf{u}_\sigma^*) = -A_\sigma^T \mathbf{u}_\sigma^{k,c} = O(e^{-\rho/\mu_k}),$$

as \mathbf{u}^k satisfies (1.6).

Note that \mathbf{u}^k is dual feasible, but feasibility (except in the limit) is not forced on \mathbf{x}^k . In fact, if $\mathbf{u}_i^k > 1/w_i$ then it follows from (3.2) that $r_i(\mathbf{x}^k) < 0$. In this sense the exponential defines a soft barrier, in contrast to the log barrier which defines a hard barrier. However, \mathbf{u}^k is necessarily dual feasible, and we can show that it maximizes a hard barrier function. Assume that it has the form

$$F(\mathbf{u}, \mu_k) = \mathbf{b}^T \mathbf{u} + \mu_k \sum_{i=1}^n w_i f_i(u_i).$$

Then minimizing F subject to (1.6) gives

$$\mathbf{b}^T + \mu_k \sum_{i=1}^n w_i f_i'(u_i) \mathbf{e}_i^T = \mathbf{x}^T A^T.$$

Equating terms gives

$$\mu_k w_i f_i'(u_i) = r_i,$$

or, using (3.2),

$$f_i'(u_i) = -\log w_i u_i.$$

Thus

$$f_i(u) = -(u \log w_i u - u).$$

Note that f_i is strictly concave on $u \geq 0$ so that uniqueness of the maximum, which is assumed implicitly in this development, is justified.

The trick used in the previous section of weighting the barrier terms to obtain a more rapidly convergent sequence of dual variables can be used here also. Let \mathbf{v}^{k-1} be an estimate of the dual vector and consider the barrier function

$$G^k(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \mu_k \sum_{i=1}^n w_i v_i^{k-1} e^{-r_i(\mathbf{x})/\mu_k w_i}. \tag{3.4}$$

The main step of the algorithm computes

$$\mathbf{x}_k = \arg \min_{\mathbf{x}} G^k(\mathbf{x}),$$

and

$$v_i^k = v_i^{k-1} e^{-r_i(\mathbf{x}^k)/\mu_k w_i}, \quad i = 1, 2, \dots, n. \tag{3.5}$$

As usual \mathbf{x}^k is independent of the row scaling of A provided

$$\mathbf{a}_i \rightarrow d_i \mathbf{a}_i \Rightarrow w_i \rightarrow d_i w_i, \quad v_i^0 \rightarrow v_i^0/d_i.$$

These conditions are satisfied by setting $w_i = \|\mathbf{a}_i\|$, $v_i^0 = 1/\|\mathbf{a}_i\|$.

The rate of convergence result now becomes the following.

THEOREM 3.2. *Let LP1 have a unique solution, $u_i^* > 0$ for $i \in \sigma$, and ρ be given by (3.3). Then*

$$\|x^k - x^*\| = O(\mu_k),$$

and

$$\|v^k - v^*\| = O\left(\prod_{s=1}^k e^{-\rho/\mu_s}\right). \tag{3.6}$$

REMARK 3.1. The use of G^k is suggested in [3] for solving linear programming problems. The spectacular rate of convergence result (3.6) is not noted, but the comment is made that the dual maximization problem for v^k which considers the objective function

$$b^T v - \mu_k \sum_{i=1}^n w_i \left(v_i \log \frac{v_i}{v_i^{k-1}} - v_i \right)$$

is slowly convergent. This should be compared with Remark 2.2. Also convergence will be through infeasible points when using G^k if some multipliers increase to their limiting values.

The order of convergence behaviour of the different barrier functions is compared in Table 3.1 for the barrier parameter sequence $\mu_k = \mu_0 \alpha^k$. The entries in this table follow immediately from the previous discussion. For example consider the M^k barrier function. If

$$\min_{i \in \sigma^c} r_i(x^*)/w_i = \pi,$$

TABLE 3.1. Barrier functions convergence rates ($\mu_k = \mu_0 \alpha^k$).

Barrier Function	Convergence Order
B	α^k
M^k	$\alpha^{k^2}/2$
E	$(e^{-1/\alpha})^k$
G^k	$(e^{-1/\alpha})^{k^2}/2$

then (using 2.4) an estimate of the order of convergence as $k \rightarrow \infty$ is obtained from

$$\begin{aligned} v^k &= \frac{\mu_k}{\pi} v^{k-1} = \left(\frac{\mu_0}{\pi} \right)^k \sum_{i=1}^k \alpha^i \\ &= O\left(\left\{ \alpha (\mu_0/\pi)^{2/k} \right\}^{k^2/2} \right) \\ &= O\left(\alpha^{k^2/2} \right) \end{aligned}$$

in the sense that $v^k/(\alpha + \varepsilon)^{k^2/2} \rightarrow 0$ for any fixed $\varepsilon > 0$.

4. A modified Newton's method

Newton's method applied to minimize $F(\mathbf{x})$, where F can be any one of our barrier functions, and stabilized by requiring that the successive values of $F(\mathbf{x}^k)$ decrease is considered here in the form:

(i) Compute

$$\mathbf{H}^t = -\nabla^2 F(\mathbf{x}^t)^{-1} \nabla F(\mathbf{x}^t)^T, \quad (4.1)$$

(ii) compute

$$\lambda^t = \arg \min_{\lambda} F(\mathbf{x}^t + \lambda \mathbf{H}^t),$$

(iii) set

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \lambda^t \mathbf{H}^t,$$

(iv) check convergence.

The quantities which must be computed for the different barrier functions are displayed in Table 4.1.

Several points can be made immediately.

(i) The limiting condition number of $\nabla^2 F$ depends only on the optimal vertex and is independent of the barrier parameter μ . This should be contrasted with cases usually considered in nonlinear programming where the limiting condition number is infinite [5].

(ii) The standard log barrier function has a different limiting form to the others, which are all identical. Potentially the appearance of the squared multipliers could exacerbate differences in scale between the individual terms in the case of some relatively small multipliers.

TABLE 4.1. Quantities required for Newton's method.

F	$\mu_k \nabla^2 F$	$\lim_{\mu_k \rightarrow 0} \mu_k \nabla^2 F$
B	$\sum_1^n \left(\frac{\mu_k}{r_i}\right)^2 \mathbf{a}_i \mathbf{a}_i^T$	$\sum_\sigma (u_i^*)^2 \mathbf{a}_i \mathbf{a}_i^T$
M^k	$\sum_1^n \left(\frac{\mu_k}{r_i}\right)^2 w_i v_i^{k-1} \mathbf{a}_i \mathbf{a}_i^T$	$\sum_\sigma \frac{u_i^*}{w_i} \mathbf{a}_i \mathbf{a}_i^T$
E^k	$\sum_1^n \frac{e^{-\frac{r_i}{\mu_k w_i}}}{w_i^2} \mathbf{a}_i \mathbf{a}_i^T$	$\sum_\sigma \frac{u_i^*}{w_i} \mathbf{a}_i \mathbf{a}_i^T$
G^k	$\sum_1^n \frac{e^{-\frac{r_i}{\mu_k w_i}}}{w_i} v_i^{k-1} \mathbf{a}_i \mathbf{a}_i^T$	$\sum_\sigma \frac{u_i^*}{w_i} \mathbf{a}_i \mathbf{a}_i^T$
F	$-\nabla F^T + \mathbf{c}$	$\lim_{\mu_k \rightarrow 0} -\nabla F^T + \mathbf{c}$
B	$\sum_1^n \frac{\mu_k}{r_i} \mathbf{a}_i$	$\sum_\sigma u_i^* \mathbf{a}_i$
M^k	$\sum_1^n \frac{\mu_k}{r_i} w_i v_i^{k-1} \mathbf{a}_i$	$\sum_\sigma u_i^* \mathbf{a}_i$
E^k	$\sum_1^n \frac{e^{-\frac{r_i}{\mu_k w_i}}}{w_i} \mathbf{a}_i$	$\sum_\sigma u_i^* \mathbf{a}_i$
G^k	$\sum_1^n e^{-\frac{r_i}{\mu_k w_i}} v_i^{k-1} \mathbf{a}_i$	$\sum_\sigma u_i^* \mathbf{a}_i$

(iii) The Newton correction is independent of the row scaling of the constraints. This is true also of the iteration modified to include a line search, since the minimizing condition determining λ is $\nabla F(\mathbf{x} + \lambda \mathbf{h})\mathbf{h} = 0$ and ∇F is independent of the scaling of the constraints.

REMARK 4.1. Unfortunately the nice representations of the limiting quantities in Table 4.1 may give somewhat misleading information about the behaviour of the Newton iteration. The problem is that these are based on converged values, while the Newton iteration must deal necessarily with perturbed quantities. Thus it is important to be able to say something about the growth resulting from these perturbations. Here an indication can be obtained by considering the Kantarovich lower bound for the radius of the sphere of attraction about the solution [11]. This is

$$R > \frac{1}{\xi} \{1 - \sqrt{1 - 2\xi}\} \eta$$

where

$$K \geq \| \nabla^3 F \|, \quad \delta = \| \nabla^2 F^{-1} \|, \quad \eta = \| \nabla^2 F^{-1} \nabla F^T \|$$

and

$$\xi = \eta \delta K < 1/2.$$

As R cannot exceed the distance from the constraint set, the rate of convergence result gives $R = O(\mu_k)$ at most, implying that

$$\delta K = O(\mu_k^{-1})$$

at least. This expression is unbounded as $\mu_k \rightarrow 0$. This suggests at least the possibility of difficulty, and this has tended to be confirmed by our numerical experiments.

The equation to be solved to find \mathbf{h}^k has the general form (compare Table 4.1)

$$A^T D_1^2 A \mathbf{h}^k = -\mu_k \{ \mathbf{c} - A^T D_2 \mathbf{e} \}, \tag{4.2}$$

where D_1 and D_2 are diagonal matrices. Also, $A^T D_1^2 A$ is positive definite provided LPI has a unique solution, so that an obvious strategy is to proceed by making a Choleski factorization. However, this has proved to be reasonably ill-conditioned, in the sense that occasional failures have occurred in solving members of the class of problems considered in the next section (this tends to confirm the suggestion of Remark 4.1). Another possibility is to note that if $\hat{\mathbf{u}}$ is known satisfying

$$\mathbf{c} = A^T \hat{\mathbf{u}}, \tag{1.6}$$

then (4.2) becomes

$$A^T D_1^2 A \mathbf{h}^k = A^T D_1 \{ -\mu_k (D_1^{-1} \hat{\mathbf{u}} - D_1^{-1} D_2 \mathbf{e}) \},$$

which is equivalent to the least squares problem

$$\min_{\mathbf{h}} \mathbf{s}^T \mathbf{s}; \mathbf{s} = D_1 A \mathbf{h} + \mu_k D_1^{-1} (\hat{\mathbf{u}} - D_2 \mathbf{e}). \tag{4.3}$$

Note that $\hat{\mathbf{u}}$ is not uniquely specified by the constraint equation (1.6). One way to proceed which avoids forming $D_1^{-1} \mathbf{u}$ by direct calculation (potentially awkward as elements of D_1 tend to zero with the inactive constraint multipliers) considers the problem

$$\min \| D_1^{-1} \mathbf{u} \|^2; \quad A^T D_1 (D_1^{-1} \mathbf{u}) = \mathbf{c}. \tag{4.4}$$

Both this problem and (4.3) can be solved by using a single factorization of $D_1 A$.

REMARK 4.2. One possible choice for $\hat{\mathbf{u}}$ which seems as if it should be attractive is $\hat{\mathbf{u}}^k = \mathbf{v}^{k-1}$. However, this has proved unsatisfactory in the numerical experiments. The likely reason is that the current minimization is not carried out accurately enough. In fact the test to terminate the current minimization has been

that $.8 < \lambda < 1.2$. Replacing this by a test on ∇F to try and force greater accuracy in $\hat{\mathbf{u}}$ (for example, $\|\nabla F(\mathbf{x})\|/\|\nabla F(\mathbf{x}^{k-1})\| < \text{tol}$) appeared to require rather tight tolerances and a consequent greater number of iterations in each minimization step. Remark 4.1 is probably again relevant.

The second step of the computation involves carrying out the line search to determine λ^k . It has proved convenient to follow a suggestion made in [14], and to use local approximations of special form to $L(\lambda) = (dF/d\lambda)(\mathbf{x} + \lambda\mathbf{h})$. There are two cases.

(a) Log barrier. Let $w_i > 0$, $i = 1, 2, \dots, n$,

$$L(\lambda) = d + \sum_{i=1}^n \frac{w_i}{\alpha_i - \lambda}, \quad (4.5)$$

and define the interpolation $\phi(t, \lambda)$ by

$$\phi(t, \lambda) = \zeta_1 + \frac{\zeta_2}{\alpha - \lambda - t}, \quad (4.6)$$

where

$$\phi(0, \lambda) = L(\lambda), \quad \phi'(0, \lambda) = L'(\lambda),$$

and

$$\alpha = \min_{i, \alpha_i > 0} \alpha_i.$$

(b) Exponential barrier. Let $w_i \alpha_i > 0$, $i = 1, 2, \dots, n$, $\max_i \alpha_i > 0$, and

$$L(\lambda) = w_0 + \sum_{i=1}^n w_i e^{\alpha_i \lambda}, \quad (4.7)$$

and define the interpolation $\phi(t, \lambda)$ by

$$\phi(t, \lambda) = \zeta_1 + \zeta_2 e^{\alpha t}, \quad (4.8)$$

where

$$\phi(0, \lambda) = L(\lambda), \quad \phi'(0, \lambda) = L'(\lambda),$$

and

$$\alpha = \max_i \alpha_i.$$

The line search algorithm generates a sequence of points λ^s starting with $\lambda^0 = 0$ according to the rule

$$\lambda^{s+1} = \arg\{\phi(t, \lambda^s) = 0\}. \quad (4.9)$$

This condition can be solved explicitly. Specifically:

(a) log barrier.

$$\begin{aligned} \zeta_1 &= L(\lambda^s) - (\alpha - \lambda^s)L'(\lambda^s), \quad \zeta_2 = (\alpha - \lambda^s)^2 L'(\lambda^s), \\ \lambda^{s+1} &= (\alpha - \lambda^s)L(\lambda^s)/(L(\lambda^s) - (\alpha - \lambda^s)L'(\lambda^s)). \end{aligned}$$

(b) exponential barrier.

$$\begin{aligned} \zeta_1 &= L(\lambda^s) - L'(\lambda^s)/\alpha, \quad \zeta_2 = L'(\lambda^s)/\alpha, \\ \lambda^{s+1} &= \log(1 - \alpha L(\lambda^s)/L'(\lambda^s))/\alpha. \end{aligned}$$

The utility of this line search procedure is guaranteed by the following result.

THEOREM 4.1. *Let $L(0) < 0$, and let the conditions stated in the definitions of $L(\lambda)$ be satisfied. Then*

(i) (a) *log barrier:*

$$\phi(t, \lambda^s) \geq L(t + \lambda^s), \quad 0 \leq t < \alpha - \lambda^s,$$

(b) *exponential barrier:*

$$\phi(t, \lambda^s) \geq L(t + \lambda^s), \quad 0 \leq t;$$

(ii) *the line search procedure is convergent. Specifically, let*

$$L(\lambda^*) = 0.$$

Then

$$\lambda^1 < \lambda^2 < \dots < \lambda^s < \lambda^*,$$

and

$$\lim_{s \rightarrow \infty} \lambda^s = \lambda^*;$$

(iii) *the rate of convergence is second order.*

PROOF. Part (i) follows by elementary computations. Part (ii) follows from $L(0) < 0$, (i), and the property that $L(\lambda)$ increases monotonically to ∞ as $\lambda \rightarrow \alpha$ (log barrier), or $\lambda \rightarrow \infty$ (exponential barrier). Part (iii) also follows by direct computation which shows that

$$\lambda^{s+1} = -\frac{\phi(0, \lambda^s)}{\phi'(0, \lambda^s)} + O(\phi(0, \lambda^s)^2) = -\frac{L(\lambda^s)}{L'(\lambda^s)} + O(L(\lambda^s)^2)$$

as $L(\lambda^s) \rightarrow 0$.

REMARK 4.3. The identifications required in order to compute ϕ are easily made.

(a) Let

$$B = \mathbf{c}^T \mathbf{x} - \mu_k \sum_{i=1}^n \log r_i(\mathbf{x}).$$

Then

$$L(\lambda) = \mathbf{c}^T \mathbf{h} + \sum_{i=1}^n \frac{\mu_k}{-\frac{r_i(\mathbf{x})}{\mathbf{a}_i^T \mathbf{h}} - \lambda},$$

and

$$\alpha_i = -\frac{r_i(\mathbf{x})}{\mathbf{a}_i^T \mathbf{h}} > 0$$

holds if \mathbf{h} points in a direction in which the i 'th constraint is violated.

(b) Let

$$E = \mathbf{c}^T \mathbf{x} + \mu_k \sum_{i=1}^n e^{-r_i(\mathbf{x})/\mu_k}.$$

Then

$$L(\lambda) = \mathbf{c}^T \mathbf{h} + \sum_{i=1}^n -\mathbf{a}_i^T \mathbf{h} e^{-r_i(\mathbf{x})/\mu_k} e^{-(\mathbf{a}_i^T \mathbf{h}/\mu_k)\lambda},$$

so that

$$w_i \alpha_i = e^{-r_i(\mathbf{x})/\mu_k} (\mathbf{a}_i^T \mathbf{h})^2 / \mu_k > 0,$$

and

$$\alpha = -\frac{1}{\mu_k} \min_i \mathbf{a}_i^T \mathbf{h}.$$

Thus $\alpha > 0$ provided any constraint is violated in the direction determined by \mathbf{h} . This is not necessary as the approach to the minimum can be through infeasible points unless $u_i^* < 1$, $i = 1, 2, \dots, n$. It can always be achieved by scaling \mathbf{c} suitably.

REMARK 4.4. In the case that Newton's method is used to solve the minimization problem then the first step of the line search can be carried out immediately. This follows because (the case of the log barrier is typical)

$$\begin{aligned} L(0) &= \nabla B(\mathbf{x})\mathbf{h}, \\ L'(0) &= \mathbf{h}^T \nabla^2 B(\mathbf{x})\mathbf{h} \\ &= -\nabla B(\mathbf{x})\nabla^2 B(\mathbf{x})^{-1}\nabla^2 B(\mathbf{x})\mathbf{h} \\ &= -L(0), \end{aligned}$$

so that

- (a) log barrier: $\lambda^1 = \alpha/(1 + \alpha)$, and
- (b) exponential barrier: $\lambda^1 = \log(1 + \alpha)/\alpha$.

5. Numerical experiments

Numerical results reported by Karmarkar and by those working both on the implementation of his algorithm and on related methods appear to relate to a small set of comparatively large, sparse problems concerning which there seems to be a consensus regarding their suitability and relevance. Here an alternative approach is adopted. Barrier methods are applied to solve sequences of randomly generated test problems constructed by the procedure described in [13], and this permits rather different kinds of information to be obtained. For example, one may determine:

- (a) how the work required to solve members of this class of problems varies as functions of n and p .
- (b) how this work compares with the estimates for active set methods presented in [13], and
- (c) how the selection of tuning constants affects the performance of the algorithm. The test problems are obtained by sampling from a shifted Pareto distribution

$$F(\rho) = (1 - 2\rho)/(\rho - 1) + (1 - RND)**(-1/\rho),$$

where RND is a uniform random number generator on $[0, 1]$, and $\rho = 1.2$. This distribution has finite first moment only and is skewed. It appears to produce more difficult problems than did other distributions tested. Also, it does not meet the requirements of the recent average case results reported in [1] and [7]. To obtain a first feasible point x^0 it is convenient to add a slack variable and penalise the objective function (this is described in [13]). A suitable penalty can be calculated during problem set up, so that no additional computing is needed to adjust this. The numerical results obtained in [13] for the projected gradient algorithm are very well modelled in the range $p = 5 * 2^{s-1}$, $s = 1, 2, \dots, 5$, $n = 5 * 2^t$, $t = s, \dots, 6$ by an expression for the number of active set updates having the form

$$N = p + Bp^A \log n/p \quad (5.1)$$

with $B = .358$, and $A = 1.37$. In fact, the standard deviation of the residuals for this estimate is less than one update. The data is obtained by taking the median of 100 replications for each value of n , p .

The number of iterations (number of search directions \mathbf{h}^k generated) for the barrier function calculations are reported in Tables 5.1 and 5.2. These give the median of ten replications for the values of n and p indicated. The number of replications and the ranges of n and p were limited by both the speed and memory of the microprocessor used. A certain amount of preliminary tuning went into obtaining these results, and while the final strategy may not be optimal it was certainly easy to do much worse. First the data matrix $[A|\mathbf{b}]$ was scaled to have

TABLE 5.1. Performance of M^k barrier function on simulated data (median of 10 replications).

p	n	10	20	40	80
5		15	15.5	19.5	19
10		..	18	20	27.5
20		26	

TABLE 5.2. Performance of G^k barrier function on simulated data (median of 10 replications).

p	n	10	20	40	80
5		16	15	22.5	21.5
10		..	20	21	30
20		32.5	

column norms 1, and then the rows of A are scaled to have norm 1 (this simplifies the choice of w_i, u_i^0) as is c. For the log barrier μ_1 is chosen to minimise $\|\nabla M^1(x^0)\|$ (this is recommended in [4]) provided this is positive. Otherwise it is chosen to be $.5E - 3$ which proves to be of the same order of magnitude as the values obtained by the other procedure. For the exponential barrier function, this device leads to a nonlinear problem determining μ_1 , and it proved simpler to set μ using the size of the elements of r as a guide. The rule adopted is $\mu_1 = .03 \max_i |r_i(x^0)|$. The value $\alpha = .5$ was chosen to reduce μ after each minimization after some experimentation. The computation is stopped when the optimal reference is located to sufficient accuracy. Because the rate of convergence of the multiplier estimates is of direct interest, the test $u_{\nu(p+1)} > 1E3u_{\nu(p+2)}$ is used where ν is an index set pointing to the elements of u sorted into decreasing order. (Note that the use of the slack variable implies $|\sigma| = p + 1$). An alternative is to test if $\sigma = \{\nu(1), \nu(2), \dots, \nu(p + 1)\}$ gives $u_\sigma \geq 0$ in (2.2). A strategy of this kind is an essential component of the Karmarkar algorithm. It asks only weak convergence of the multiplier estimates.

The results tend to favour the log barrier function but have certain features in common. For example, the first two minimizations tend to take up the bulk of the computation. Presumably this implies

(a) that the initial feasible point generated by adding a slack variable is not particularly appropriate, and

(b) because u^0 does not come from a minimization step the choice of u^1 after the first minimization corresponds to a substantial change in the problem to be solved. The most surprising difference proved to be in the significantly fewer

TABLE 5.3. Typical results for M^k barrier function
(case $p = 5$, $n = 10$, components of u^k ordered).

u^k	1	2	3	4
1	3.3194 $E - 01$	3.5510 $E - 1$	3.5188 $E - 1$	3.5240 $E - 1$
2	3.2013 $E - 01$	3.2725 $E - 1$	3.1852 $E - 1$	3.1866 $E - 1$
3	2.1790 $E - 01$	2.0066 $E - 1$	1.9493 $E - 1$	1.9460 $E - 1$
4	1.7948 $E - 01$	1.8001 $E - 1$	1.7873 $E - 1$	1.7867 $E - 1$
5	1.1438 $E - 1$	1.7936 $E - 1$	1.7650 $E - 1$	1.7657 $E - 1$
6	4.0753 $E - 2$	5.5661 $E - 2$	5.4401 $E - 2$	5.4371 $E - 2$
7	3.6003 $E - 2$	3.2011 $E - 3$	1.2342 $E - 4$	2.4257 $E - 6$
8	3.3256 $E - 2$	1.4129 $E - 3$	2.9869 $E - 5$	4.9292 $E - 7$
9	1.6042 $E - 2$	8.8311 $E - 4$	2.7504 $E - 5$	2.7090 $E - 7$
10	1.4390 $E - 2$	8.6083 $E - 4$	2.1154 $E - 5$	2.6192 $E - 7$
11	1.4355 $E - 2$	7.3971 $E - 4$	8.3293 $E - 6$	4.7157 $E - 8$
Its	5	7	3	2

minimization steps required by the log barrier function. The exponential barrier function takes more steps (that generally prove easier to minimize), apparently having more trouble isolating the active constraints, but the final rate of convergence proves enormously fast in accordance with the predications. However, the Newton iteration proved harder to control with occasional cases of overflow in the exponential evaluations. Details of progress are given for the two barrier functions applied to one example in Tables 5.3 and 5.4.

REMARK 5.1. The above observations have a direct analogue in the results of [9]. There it is noted that a smooth, stable trajectory can be associated with the modified log barrier function, and that this trajectory is approached by the computed minima sufficiently rapidly to justify the use of extrapolation techniques to refine these solution estimates provided the first few points are ignored.

The justification for the recent interest in barrier methods for linear programming stems from the hope that they will prove superior to active set methods for important ranges of values of p and n . Such a conclusion would be clear for the results presented here if, for example, the number of iterations showed little or no tendency to grow with n and p . For if this number of steps is N then a break-even point would be reached when

$$N(np^2) = (p + Bp^4 \log n/p)\gamma np,$$

where γnp measures the work per step in the active set method, and np^2 is taken as the cost of solving the least squares problem (representing a Householder factorization of $D_1 A$ for example). But our figures give no such assurance. Thus it

TABLE 5.4. Typical results for G^k barrier function
(case $p = 5$, $n = 10$, components of u^k ordered).

u^k	1	2	3	4	5
1	3.1284 E - 1	3.4969 E - 1	3.3252 E - 1	3.2279 E - 1	3.5631 E - 1
2	2.6916 E - 1	2.6157 E - 1	2.8504 E - 1	3.1670 E - 1	3.1926 E - 1
3	2.3304 E - 1	2.6092 E - 1	2.6833 E - 1	2.0554 E - 1	1.9452 E - 1
4	1.9929 E - 1	2.5382 E - 1	2.2856 E - 1	2.0105 E - 1	1.7912 E - 1
5	1.7544 E - 1	1.8104 E - 1	2.0296 E - 1	1.8324 E - 1	1.7876 E - 1
6	5.1710 E - 2	5.5678 E - 2	8.5156 E - 2	5.1391 E - 2	5.5294 E - 2
7	2.1400 E - 2	2.0933 E - 2	4.0046 E - 3	1.7213 E - 3	1.5172 E - 8
8	1.5441 E - 2	9.3825 E - 3	6.9780 E - 4	3.8903 E - 7	3.8387 E - 13
9	1.4072 E - 2	6.4766 E - 3	1.6381 E - 5	1.4601 E - 11	1.1260 E - 21
10	1.3876 E - 2	1.3469 E - 3	1.2782 E - 6	5.2607 E - 13	4.8380 E - 25
11	9.4166 E - 3	2.8836 E - 5	1.6081 E - 10	4.5206 E - 21	8.0040 E - 42
Its	6	4	2	2	2

is necessary to consider the possibility of reducing the effective value of N . There appears some scope for doing this, and possibilities include keeping D_1A constant for several iterations, or updating to take account only of the largest change in D_1 at each step. The problem is that the Newton iteration is fairly sensitive (compare Remark 4.1) so that worthwhile strategies may prove elusive. But it is perhaps worth noting that Karmarkar improved his complexity estimate by this kind of device.

References

- [1] I. Adler, N. Megiddo and M. J. Todd, "New results on the average behaviour of simplex algorithms", *Bull. Amer. Math. Soc.* 11 (1984), 378–382.
- [2] J. Eriksson, *Algorithms for Entropy and Mathematical Programming*, Linköping Studies in Science and Technology Dissertations No. 63, Linköping University, 1981.
- [3] J. Eriksson, "An iterative primal-dual algorithm for linear programming", Report LiTH-MAT-R-1985-10, Linköping University, 1985.
- [4] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming* (Wiley, New York, 1968).
- [5] R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1981).
- [6] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and Margaret Wright, "On projective Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method", Report SOL 85-11, Department of Operations Research, Stanford University, 1985.
- [7] M. Haimovich, "The simplex algorithm is very good!—on the expected number of pivot steps and related properties of random linear programs", preprint, Columbia University, 1983.

- [8] K. Jittorntrum and M. R. Osborne, "Trajectory analysis and extrapolation in barrier function methods", *J. Aust. Math. Soc. Ser. B* **20** (1979), 352–369.
- [9] K. Jittorntrum and M. R. Osborne, "A modified barrier function method with improved rate of convergence for degenerate problems", *J. Aust. Math. Soc. Ser. B* **21** (1980), 305–329.
- [10] N. Karmarkar, "A new polynomial-time algorithm for linear programming", *Combinatorica*, **4** (1984), 373–395.
- [11] D. G. Luenberger, *Optimization by Vector Space Methods* (Wiley, New York, 1969).
- [12] M. R. Osborne, "Topics in optimization", Report CS-72-279, Computer Science Department, Stanford University, 1972.
- [13] M. R. Osborne, *Finite Algorithms in Optimization and Data Analysis* (Wiley, Chichester, 1985).
- [14] Margaret Wright, "Numerical methods for nonlinearly constrained optimization", Report CS-76-566, Computer Science Department, Stanford University, 1976.