J. Austral. Math. Soc. Ser. B 31(1990), 367–378

AN INTERIOR POINT METHOD FOR LINEAR PROGRAMMING

M. R. OSBORNE[1]

(Received 5 September, 1988; revised 28 February, 1989)

Abstract

Design of an interior point method for linear programming is discussed, and results of a simulation study reported. Emphasis is put on guessing the optimal vertex at as early a stage as possible.

1. Introduction

In this paper certain aspects both of the design and the implementation of an interior point method for linear programming are discussed.The motivation for this work has been the satisfaction of the author's curiosity about certain aspects of the developments that have stemmed from the impetus that the work of Narendra Karmarkar (see for example the remarkable paper [4]) gave to consideration of these methods; and, in particular to try and answer the question 'should I be using these methods instead of active set, reduced gradient methods for the typically rectangular (many more constraints that variables) and dense problems encountered in approximation theory and data analysis'? For this purpose attention is given to two distinct components of the interior point method:

(i) a basis solution step which generates at each point a new feasible point with a lower objective function value, and which would converge eventually to the desired solution in an iterative fashion; and

[1] Statistics Research Section, School of Mathematical Sciences, Australian National University, Box 4, GPO, Canberra, A. C. T. 2601, Australia.

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

(ii) an existential step in which, at selected points produced by the above procedure, an attempt is made to guess the optimal vertex; and the algorithm terminates if the step is successful.

In a previous paper [5] modified barrier functions were used in constructing the iterative step, and these were designed to give accelerated convergence to the associated Kuhn-Tucker multiplier estimates. It was hoped that this multiplier information would provide an effective basis for making the existential step; but this proved to be the case only if the minimization of the barrier function in each iterative step was computed with sufficient precision with an accompanying cost in expensive 'inner' iterations. This contrasts unfavourably with the single step 'inner' iterations in the methods associated with the method of centres (for example [4], [7]). However, these methods have the disadvantage that dual information of adequate integrity now has to be fought for (for example [3], [8]). The method considered here is described in the next section. It belongs to the class of immediate descendents of the Karmarkar algorithm, and has the advantage that it gives accelerated convergence of the multiplier estimates in a form which is adequate for the existential step. In fact it is possible to modify the algorithm to give the same kind of superfast convergence of the multiplier estimates as that which formed the basis for the investigation in [5]. Implementation is described briefly in Section 3 (the method has proved agreeably stable in the experiments so far conducted), and numerical results based on quite extensive simulations using randomly generated problems are discussed in Section 4. In summary, while interesting trends are evident which may suggest advantages in large and very sparse problems there is no doubt that active set methods are superior for the problems reported here by a wide margin on conventional computing machinery. However, the algorithm does appear to lend itself to implementation on vector processors.

2. Development of an interior point method

The linear programming problem is considered in the form:

$$\min_{\mathbf{x}\in X} \mathbf{c}^T\mathbf{x}; \qquad X = \{\mathbf{x} : \mathbf{A}\mathbf{x} \geq \mathbf{b}\} \tag{2.1}$$

where $\mathbf{A}: R^p \to R^n$ is assumed to have its full rank p, and where the problem is assumed to have a bounded minimum at $\mathbf{x} = \mathbf{x}^*$. Let the currently (strictly feasible) point be $\bar{\mathbf{x}}$, and set

$$\mathbf{y} = \mathbf{A}\bar{\mathbf{x}} - \mathbf{b} > \mathbf{0}, \qquad \mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}.$$

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

Following Barnes [1] who considers problems in standard form (the dual of (2.1)) the problem

$$\min_{\mathbf{x}} \mathbf{c}^T\mathbf{x} \quad \text{subject to} \sum_{i=1}^{n} \left(\frac{r_i - y_i}{y_i}\right)^2 \le R^2 \tag{2.2}$$

is used as the basis of the iterative method (that Barnes' method gives the basic step for his problem for suitably chosen R is actually noted by Karmarkar in [4]). The first result needed follows by a simple contradiction argument.

Lemma 2.1. *Let* $\mathbf{x}$ *solve* (2.2), *then* $\mathbf{x}$ *is strictly feasible provided* $|R| < 1$.

The Kuhn-Tucker conditions for (2.2) are

$$-\mathbf{c} = \gamma \sum_{i=1}^{n} \frac{r_i - y_i}{y_i^2} a_i, \qquad \gamma \ge 0 \tag{2.3}$$

where a_i^T denotes the i'th row of A.

Remark 2.1. The multiplier satisfies $\gamma > 0$ provided x is strictly feasible.

Lemma 2.2.

$$\mathbf{c}^T\bar{\mathbf{x}} - \mathbf{c}^T\mathbf{x} = \gamma R^2. \tag{2.4}$$

Proof. This follows on multiplying (2.3) by $\mathbf{x}$ and $\bar{\mathbf{x}}$, subtracting, and using that the constraint (2.3) must be active.

The following result is an immediate consequence of the boundedness of the solution of (2.1), but the conditions are worth stressing.

Theorem 2.1. *Let* (2.2) *be applied iteratively to generate a sequence* $\{x^{(k)}\}$ *with the properties that*

(i) $r^{(k)}$ *is strictly positive* (*so that* $c^T\mathbf{x}^{(k)}$ *is bounded below*), *and*
(ii) $R^{(k)}$ *is bounded,*

for all k. *Then the iteration is weakly convergent in the sense that* $\{\mathbf{c}^T\mathbf{x}^{(k)}\}$ *converges and the series of positive terms* $\sum \gamma^{(k)}$ *is convergent.*

Another contradiction argument gives the result:

Lemma 2.2. *Let*

$$\xi = \min(y_i,\ i = 1, 2, \dots, n) \tag{2.5}$$

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

then $\lim_{k\to\infty}\xi^{(k)} = 0$ *so that the limit points of the iteration must lie in the boundary of* X.

The next step is to solve (2.3) for $\mathbf{x}$. To do this note that it can be written

$$-\mathbf{c} = \gamma \sum_{i=1}^{n} \frac{1}{y_i^2} a_i a_i^T (\mathbf{x} - \overline{\mathbf{x}})$$

so that

$$\mathbf{x} = \overline{\mathbf{x}} - \frac{\xi^2}{\gamma}\left\{\mathbf{A}^T\mathbf{D}\left[\frac{\xi}{y}\right]^2\mathbf{A}\right\}^{-1}\mathbf{c}, \tag{2.6}$$

where

$$\mathbf{D}(\mathbf{z}) = \operatorname{diag}\{z_i,\ i = 1, 2, \ldots, n\}.$$

It follows from (2.4) that

$$\gamma R^2 = \frac{\xi^2}{\gamma}\mathbf{c}^T\left\{\mathbf{A}^T\mathbf{D}\left[\frac{\xi}{y}\right]^2\mathbf{A}\right\}^{-1}\mathbf{c}, \tag{2.7}$$

In particular,

$$\frac{\gamma R}{\|\mathbf{c}\|}\sigma_{\min} \le \xi \le \frac{\gamma R}{\|\mathbf{c}\|}\sigma_{\max} \tag{2.8}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the extreme singular values of $D[\frac{\xi}{y}]A$. As this matrix has uniformly bounded elements it follows that the smallest residual converges to zero with γ.

REMARK 2.2. Note that the constraint (2.2) is independent of the scaling of the constraints (2.1) as is the multiplier γ determined by (2.3). Also this invariance property extends to the sequence $\{x^{(k)}\}$,

To discuss the nature of the limit points of the iteration it is convenient to assume that the sequence of matrices $\{\mathbf{A}^T\mathbf{D}(\xi^{(k)}/y^{(k)})^2\mathbf{A}\}$ have uniformly bounded inverses.

THEOREM 2.2. *With this assumption it follows that*

(a) *the sequence* $\{\mathbf{x}^{(k)}\}$ *converges, and*
(b) *the limit point of the iteration is a vertex of the feasible region.*

PROOF. Part (a) is an immediate consequence of (2.6), (2.7), and the convergence of $\sum \gamma^{(k)}$. Part (b) follows because at least p of the elements $D[\frac{\xi}{y}]$ must remain bounded away from zero to ensure boundedness of the inverse matrices.

To discuss the optimality of this vertex let σ be an index set pointing to the active constraints. Subscript σ will be used to denote corresponding subvectors and matrices.

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

Theorem 2.3. *Under the same assumptions* (a) *the limiting vertex is optimal,* (b) *there exists a subsequence of the* $\{\gamma^{(k)}[y_i^{(k)} - r_i^{(k)}]/y_i^{(k)^2},\ i = 1, 2, \cdots\}$ *converging to a non-negative Kuhn-Tucker vector, and* (c) *if* $|\sigma| = p$ *then the sequence converges to the unique Kuhn-Tucker vector and the rate of convergence is* $O(\gamma^2)$.

Proof. It follows from (2.6) and the above assumptions that

$$y_i - r_i = \frac{\xi^2}{\gamma} a_i^T \left\{ \mathbf{A}_\sigma^T D \left[\frac{\xi}{y}\right]_\sigma^2 \mathbf{A}_\sigma \right\}^{-1} \mathbf{c} + O(\xi^3). \tag{2.9}$$

Let

$$u_i = \gamma \frac{y_i - r_i}{y_i^2}, \tag{2.10}$$

then $u_i = O(\gamma^2)$ if $i \in \sigma^c$ by (2.9) as $y_i \nrightarrow 0$. If $i \in \sigma$ then

$$u_\sigma = D \left[\frac{\xi}{y}\right]_\sigma^2 \mathbf{A}_\sigma \left\{ \mathbf{A}_\sigma^T D \left[\frac{\xi}{y}\right]_\sigma^2 \mathbf{A}_\sigma \right\}^{-1} c + O(\xi^2). \tag{2.11}$$

Because $r_i^{(k)} \to 0$, $k \to \infty$ an infinite number of $u_i^{(k)} \geq 0$. It follows from (2.10), (2.11) that there exists a bounded, non-negative vector $\overline{u}_\sigma$ such that (by selecting subsequences if necessary)

$$0 \leq u_\sigma^{(k)} \to \overline{u}_\sigma, \qquad k \to \infty, \tag{2.12}$$

and that

$$\mathbf{A}_\sigma^T \overline{u}_\sigma = c. \tag{2.13}$$

This establishes parts (a) and (b). Part (c) follows from (2.11) on noting that if $|\sigma| = p$ then $\mathbf{A}_\sigma$ must be invertible so that u_σ converges to a unique limit which must be non-negative.

Remark 2.3. This convergence result suggest that it may be worth sorting the multiplier estimates $\gamma(r_i - y_i)/y_i^2$ and testing the index set determined by the p largest to see if it points to the optimal vertex. An alternative is to sort the r_i and test the index set determined by the p smallest. But it follows from (2.8) that the rate of convergence of the quantities tested to their final value is slower in this case. Thus it would be satisfying if the first heuristic proved superior (these are just heuristics as we hope to stop before final convergence sets in). Note that r_i transforms with the scale of the ith constraint, while the multiplier estimate transforms with its reciprocal. Thus it is presumably sensible to ensure that the constraint scales are comparable before making these comparisons.

REMARK 2.4. In Brophy and Smith [2] an alternative derivation of Barnes' algorithm is given motivated by trajectory arguments. If in fact the $r_i^{(k)}$ did correspond to values $r(t_k)$, $t_k = \sum_{j=k}^{\infty} \gamma^{(j)}$, lying on a smooth trajectory then we would have

$$\frac{y_i^{(k)} - r_i^{(k)}}{y_i^{(k)}} = -R^{(k)} \frac{y_i^{(k)} u_i}{\left\{\sum_o \left[y_j^{(k)} \overline{u}_j\right]^2\right\}^{1/2}} + O(\xi^2) = \frac{\gamma^{(k)}}{t_k} + \text{smaller terms} \quad (2.14)$$

so that $y_i u_i$ would be independent of i to first order, and (2.14) would give $(r_i - y_i)/y_i = -R/p^{1/2}$. This might even suggest that an optimum choice of R giving a faster rate of convergence is possible. But this would contradict (2.8), while (2.14) is not born out by numerical experimentation. Thus the trajectory following analogy cannot be pushed too far.

REMARK 2.5. The assumption that $\mathbf{A}^T\mathbf{D}(\xi/y)^2\mathbf{A}$ has a bounded inverse does not exclude at least some degenerate behaviour, but does exclude the important case of nonuniqueness. In this latter case it is easy to construct examples which show that convergence is to a boundary point which is not a vertex in general, and that the particular limit point will depend on the initial feasible $\mathbf{x}$. However, the above argument can be extended to show that this limit point is optimal and that the rate of convergence to zero of the active constraint residuals is $O(\gamma)$.

The existence of convergent multiplier estimates suggests that it might be possible to proceed as in [5] and develop procedures which give estimates possessing superfast rates of convergence. The idea is to introduce positive weights into the constraint inequality (2.2) which now has the form

$$\sum_{i=1}^{n} w_i^{(k)} \left[\frac{y_i^{(k)} - r_i^{(k)}}{y_i^{(k)}}\right]^2 \leq R^2. \quad (2.15)$$

The previous argument still goes through with the only change being to accommodate the weights. The idea used in [5] is to choose $W_i^{(k)}$ as the best estimate of corresponding Kuhn-Tucker multiplier, and this suggest the update

$$w_i^{(k+1)} = \gamma^{(k)} \frac{y_i^{(k)} - r_i^{(k)}}{y_i^{(k)2}} w_i^{(k)}. \quad (2.16)$$

However, defined in this way, $W_i^{(k)}$ does not transform the right way under charge of constraint scale (if $a_i \to \rho_i a_i$ then it is required that $w_i \to \frac{1}{\rho_i} a_i$). This can be remedied by modifying the constraint inequality (2.15) to

$$\sum_{i=1}^{n} w_i^{(k)} \|a_i\| \left[\frac{r_i - y_i^{(k)}}{y_i^{(k)}}\right]^2 < R^2, \quad (2.17)$$

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

choosing $w_i^{(0)}$ to transform correctly (for example, by setting $w_i^{(0)} = 1/\|a_i\|$), and modifying (2.16) to

$$w_i^{(k+1)} = \gamma^{(k)} \frac{y_i^{(k)} - r_i^{(k)}}{y_i^{(k)2}} w_i^{(k)} \|a_i\|. \tag{2.18}$$

But there is a further problem with $w_i^{(k)}$ defined by (2.16) or (2.18). This is that $w_i^{(k)}$ is not guaranteed to be positive. This does not seem to be serious as (2.10) shows that the multiplier estimates of most interest will necessarily be positive eventually. As the remainder must become small the simplest solution would seem to be to define $W_i^{(k+1)}$ as the absolute value of the right hand side of (2.18). To demonstrate the superfast convergence note that if $u_i = 0$ corresponding to $i \notin \sigma$ then

$$w_i^{(k+1)} \to 0, \quad \text{with } \prod_{j=0}^{k} \gamma^{(j)2}.$$

Thus the argument used previously to improve the estimate given by (2.12) shows that

$$|w_i^{(k+1)} - u_i = O\left(\prod_{j=0}^{k} \gamma^{(j)2}\right), \qquad i \in \sigma. \tag{2.19}$$

3. Implementation notes

The numerical results presented in the next section have been obtained using a neat implementation in which the major component of the work done in each basic solution step corresponds to using modified Gram Schmidt orthogonalisation [6] to compute the tableau transformation

$$[\mathbf{A}^T \mathbf{D}(\xi/y) I] \to [\mathbf{Q}_1^T U^{-T}]$$

where $\mathbf{Q}_1$ is an orthogonal basis for the range of $\mathbf{D}(\xi/y)\mathbf{A}$. We then compute in sequence

$$\begin{aligned}
&\mathbf{v} = U^{-T}\mathbf{c}, \\
&[d\mathbf{r}^T \mathbf{D}(\xi/y)\, d\mathbf{x}^T] = v^T[\mathbf{Q}_1^T \ U^{-T}], \\
&\alpha = \min(y_i/dr_i, dr_i > 0), \quad \text{and} \\
&[\mathbf{r}^T \ \mathbf{x}^T] = [\mathbf{y}^T \ \overline{\mathbf{x}}^T] - \Delta\alpha[d\mathbf{r}^T \ d\mathbf{x}^T]
\end{aligned}$$

where Δ is an under relaxation factor ($0 < \Delta < 1$) which ensures feasibility is maintained, before performing the existential step by guessing a vertex (Remark 2.2) and checking it for feasibility and optimality using

$$(\mathbf{Q}_1)_\sigma U\mathbf{x} = \mathbf{D}(\xi/y)_\sigma b_\sigma, \qquad (\mathbf{Q}_1)_\sigma^T u = \mathbf{v}.$$

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

If this test is not satisfied then we set

$$[\mathbf{y}^T \quad \bar{\mathbf{x}}^T] := [\mathbf{r}^T \quad \mathbf{x}^T],$$

and the iteration sequence is repeated. The parameter which determines the step in $\mathbf{x}$ in the current situation is the under relaxation factor Δ rather than R. The implementation ensures that the relative change in residuals satisfies

$$\frac{y_i - r_i}{y_i} \le \Delta, \qquad i = 1, 2, \ldots, n,$$

and this seems appropriate in view of the particular form of the constraint on (2.2) as it ensures both uniform boundedness of R^2 (actually

$$R^2 \le \Delta^2(\|\mathbf{v}^T\mathbf{Q}_1^T\| / \max(\mathbf{v}^T q_i; v^T q_i \ge 0))^2),$$

and strict feasibility as required by Theorem 2.1 (this method of ensuring feasibility is suggested in Brophy and Smith [2]). However, both R and γ are readily available. For example, from (2.3),

$$-\xi\mathbf{v} = \gamma\mathbf{Q}_1^T\mathbf{z}, \qquad z_i = (r_i - y_i)/y_i, \quad i = 1, 2, \ldots, n$$

so that

$$\gamma = \xi\|\mathbf{v}\|/\|\mathbf{Q}_1^T\mathbf{z}\| = \xi\|\mathbf{v}\|/R$$

as $\mathbf{z}$ is the range of $D(\xi/y)\mathbf{A}$.

4. Numerical experiments

The basis for the testing of the interior point method is the routine for generating random linear programs with known solution and multiplier vectors described in [6]. A MATLAB program is given in the appendix. The basic idea is that given σ, $|\sigma| = p$, $\mathbf{r}$ satisfying $r_i = 0$, $i \in \sigma$, $r_i > 0$, $i \in \sigma^c$, $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{u} > 0$ then $\mathbf{c}$, $\mathbf{B}$ can be found from

$$\mathbf{c} = \mathbf{A}_\sigma^T\mathbf{u}, \qquad \mathbf{b} = \mathbf{A}\mathbf{x} - \mathbf{r}.$$

The implementation of the interior point method then standardizes the problem data so that $\|a_i\| = 1$, $i = 1, \ldots, n$, and considers the penalised problem

$$\min[\mathbf{c}^T \omega] \begin{bmatrix} \mathbf{x} \\ x_{p+1} \end{bmatrix}$$

subject to

$$\begin{bmatrix} \mathbf{A} & \mathbf{e} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_{p+1} \end{bmatrix} \le \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix}$$

which has the same optimum if ω is large enough, and has the strictly feasible point $\mathbf{x} = 0$, $x_{p+1} = 1 + \max(\max(b_i, i = 1, 2, \ldots, n), 0)$ which can be used to start the computation. From the manner of construction of u it is known that $\sum 1/w_i$ where w_i are the row multipliers that standardize the data matrix is a suitable upper bound for ω.

Two types of preliminary calculation were carried out:

(i) Clearly the existential step is easier if the optimal vertex is well separated from its competitors in an appropriate sense. One factor affecting this is the scale of the residual vector **r**, and it was observed that by choosing this smaller it was possible to make small problems ($n = 20, p = 5$ say) more difficult. However, this effect proved to be confounded with problem size and rapidly became less significant as the dimensions were increased. Subsequent calculations held the scale at 1.

(ii) Problems were solved for a range of values of Δ to obtain information on the effect of different choices. While these could not be considered exhaustive, being restricted to comparatively small problems, it is clear that Δ needs to be chosen relatively large in $(0, 1)$. Most consideration was given to the choices .9, .95, .99; and the final choice for the more extensive calculations was .95, but the evidence for this choice over .99 was not clear cut.

TABLE 4.1. Cumulative iteration counts for 25 randomly generated linear programs for $\Delta = .95$.

p \ n	5		10		20		40	
σ	min r	max u	min r	max u	min r	max u	min r	max u
10	161	137						
20	214	209	219	229				
40	252	251	306	274	316	308		
80	308	296	380	351	420	389	406	384
160	411	350	475	438	479	470	512	475
320	513	34	561	516	588	603	602	573
640	587	486	796	688	841	890	745	785
1280	683	525	924	853	959	871		

The more extensive calculations are summarized in Table 4.1. Here cumulative iterations for each of 25 randomly generated problems are reported

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

for each of the two possibilities for the existential step in Remark 2.3, for the range of values of n and p indicated. It will be seen that there is clear evidence of growth with n, and some evidence of slow growth with p. There is some evidence also that 25 is not really large enough for the statistic to have stabilized (a more robust statistic may have been preferable as the occasional large outlier was noted). However, by and large, the choice of σ based on the large multiplier estimates seems more serviceable that that based on small residuals.

TABLE 4.2. Cumulative iteration counts for reduced gradient solution of 25 randomly generated linear programs.

n \ p	5	10	20	40
10	211			
20	259	483		
40	309	673	1120	
80	362	806	163	2534
160	411	953	2013	4067
320	494	1070	1388	5192
640	529	1191	2752	6009
1280	554	1820	2961	

A basis for comparison with an active set method is given in Table 4.2. Here similar tests have been run using an implementation of the reduced gradient algorithm described in [6]. The general characteristics show a slower growth with n and a much faster growth with p (at least linear in p). But these are dense problems so the relative work in performing the modified Gram-Schmidt orthogonalisation (costing $O(np^2)$ flops) destroys any possible advantage of the interior point method as each reduced gradient step costs only $O(np)$ flops. The result is that the reduced gradient wins very easily. the comparison may be more interesting for very sparse problems when the significantly slower growth with p for the interior point method may prove an important advantage.

There remains the question of the utility of the accelerated multiplier estimates $w_i^{(k)}$ given by (2.16) and (2.18). Using (2.16) to guess the optimal vertex in similar tests to those described above gave the following hopeful

TABLE 4.3. Results for the accelerated multiplier estimate.

n p	20	40	80
5	65	78	82

results for the cumulative totals for 10 replications for each of the indicated parameters values (Table 4.3).

Unfortunately, the results for (2.18) are not nearly so satisfactory, and do not prove competitive with either of the other two methods for selecting the vertex to test for optimality.

Appendix

The code that follows is the MATLAB text for a procedure that generates random linear programs in the form (2.1) with known solution and multiplier vectors. It is based on the discussion in [6], section 7.4.

```
   a = −.7 + rand (n,p).^(−1/1.2); % set matrix
   x = rand(p, 1); % set solution
   z = rscl*rand(n, 1); % rscl scales residual
   ones = ones(n, 1)
   % determine optimal vertex by sampling p distinct integers from 1 − n
for i = 1 : p
   j = ceil(n*rand);
while one (j) == 0, j = ceil(n*rand); end
   one(j) = 0; end
   in =find (one==0); % in points to optimal vertex
   z(in)=zeros(in); %zero residuals at optimal vertex
   u = rand(p, 1); % set multiplier vector
   c = (u' ⋆ A(in, : ))'; % set objective
   b = A ⋆ x − z; % set right hand side
```

References

[1] E. R. Barnes, "A variation on Karmarkar's algorithm for solving linear programming problems", *Math. Prog.* **36** (1986) 174–182.

[2] J. F. Brophy and P. W. Smith, "Prototyping Karmarkar's algorithm using MATH/PROTRAN", *IMSL Directions* **5** (1988) 2-3.

[3] D. M. Gay, "A variant of Karmarkar's linear programming algorithm for problems in standard form", *Math. Prog.* **37** (1987) 81-90.

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press

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

[5] M. R. Osborne, "Dual barrier functions with superfast rates of convergence for the linear programming problem", *J. Austral. Math. Soc. Ser. B* **29** (1987) 39–58.

[6] ——, *Finite algorithms in Optimization and Data Analysis* (John Wiley, Chichester,/ 1986).

[7] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming", *Math. Prog.* **40** (1988) 59–93.

[8] Y. Ye and M. Kojima, "Researching optimal dual solutions in Karmarkar's polynomial algorithm for linear programming," *Math. Prog.* **39** (1987) 305–317.

https://doi.org/10.1017/S033427000000672X Published online by Cambridge University Press