SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NEW YORK 14853

TECHNICAL REPORT NO. 721

October 1986

IMPROVED BOUNDS AND CONTAINING
ELLIPSOIDS IN KARMARKAR'S
LINEAR PROGRAMMING ALGORITHM

By

Michael J. Todd*

*Research supported in part by NSF Grant ECS-8602534 and the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University.

Improved Bounds and Containing Ellipsoids
in Karmarkar's Linear Programming Algorithm

Michael J. Todd*

Abstract. Karmarkar's projective algorithm for linear programming provides not only primal solutions but dual solutions giving bounds on the optimal value. Here we show how improved bounds can be obtained at the expense of solving a two-dimensional linear programming problem at every iteration, and also how an ellipsoid containing all dual optimal solutions can be generated from available information. We also give the results of limited computational experiments related to these topics.

*Research supported in part by NSF Grant ECS-8602534 and the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University.

1. Introduction

We describe two techniques that can be used in Karmarkar's projective algorithm [8] to provide further information about the optimal value and optimal dual solutions for a linear programming problem. This information can also be used to accelerate the algorithm.

Our discussion focusses on the variant of Karmarkar's algorithm developed by a number of authors (Anstreicher [1], Gay [4], Gonzaga [5], Jensen and Steger [6] and Ye and Kojima [12]) that uses an arbitrary normalization to handle standard form problems gracefully. Thus the original problem is

$$\begin{array}{ll} \min & c_0^T x \\ (P_0) & A_0 x = 0 \\ & g_0^T x = 1 \\ & x \geq 0 \end{array}$$

where A_0 is $m \times n$ with rank m , e , x and g_0 are in R^n , with g_0 nonnegative and nonzero. The standard form problem

$$\begin{array}{ll} \min & \tilde{c}^T \tilde{x} \\ (\tilde{P}) & \tilde{A} \tilde{x} = \tilde{b} \\ & \tilde{x} \geq 0 \end{array}$$

is equivalent to the instance of (P_0) with

$$\begin{aligned} x^T &= (\tilde{x}^T, \xi) \\ c^T &= (\tilde{c}^T, 0) \\ A_0 &= (\tilde{A}, -b), \text{ and} \\ g_0^T &= (0^T, 1). \end{aligned} \tag{1}$$

We will assume that we have a strictly positive feasible solution x_0 to (P_0) (without loss of generality, $x_0 = e$, the vector of ones in R^n) and a lower bound z_0 on its optimal value $v(P_0)$. We also assume that the feasible region of (P_0) is bounded. (These assumptions follow from similar hypotheses on $(\tilde{P})$ if (P_0) is obtained as above.)

At iteration k , we have a strictly positive feasible solution x_k to (P_0) and a lower bound z_k on its value. Let X_k be the diagonal matrix with diagonal entries equal to the components of x_k , so that $X_k e = x_k$. Then e is a feasible solution to the rescaled problem

$$\begin{aligned} \min \quad & c^T x \\ (P) \quad & A x = 0 \\ & g^T x = 1 \\ & x \geq 0 \end{aligned}$$

where $A = A_k \equiv A_0 X_k$, $c = c_k \equiv X_k c_0$, and $g = g_k \equiv X_k g_0$. The

iteration to obtain x_{k+1} is then based on taking a step from e in (P) and rescaling.

More precisely, let

$$B = \begin{bmatrix} A \\ e^T \end{bmatrix}, \quad (2)$$

and, for any $u \in R^n$, let u_p (respectively, u_q) denote the projection of u onto the null space of B (respectively, of A). It can be shown that

$$u_p = u_q - (e^T u_q / n) e. \quad (3)$$

The iteration then consists of three steps:

- (i) obtaining a lower bound $z = z_{k+1} \geq z_k$ on $v(P) = v(P_0)$;
- (ii) computing

$$d = -(c_p - \underline{z} g_p), \quad (4)$$

and setting

$$x = e + \alpha d / \|d\| \quad (5)$$

for suitable $\alpha > 0$ so that $x > 0$; and

(iii) letting

$$x_{k+1} = \frac{X_k x}{g^T x}. \quad (6)$$

See, for instance, [11].

The usual way of updating z_k (see section 2) implies that, when $z_{k+1} > z_k$, we can construct $y_{k+1} \in R^m$ so that (y_{k+1}, z_{k+1}) is feasible in the dual of (P_0)

$$(D_0) \quad \begin{array}{ll} \max & z \\ & A^T y + g_0 z \leq c_0 \end{array}$$

Thus the algorithm generates primal solutions $\{x_k\}$ and dual solutions $\{(y_k, z_k)\}$ (except that we may not have y_0 corresponding to the initial bound z_0). With a suitable choice of bound z in (i) and step size α in (ii) at each iteration one can establish

$$c_0^T x_k - z_k \leq (e^T x_k) \exp(-k/5n) (c_0^T x_0 - z_0). \quad (7)$$

The first factor on the right-hand side is bounded by assumption, while the second converges to zero. Hence the duality gap tends to zero.

The inequality (7) is based on worst-case estimates and a choice of $\alpha = 1/3$ at each iteration. Far better performance results from choosing $\alpha > 0$ at each iteration by performing an approximate line search on Karmarkar's potential function,

$$f(x; c - \underline{z}g) = n \ln(c - \underline{z}g)^T x - \sum_j \ln(e_j^T x),$$

where e_j denotes the j th unit vector in R^n . It appears that the direction d is better, and hence the line search provides a greater decrease in the potential function, if the bound $\underline{z}$ in (i) is improved. Section 2 discusses how to obtain cheaply a better bound than those proposed thus far. In one important case, our improved bound coincides with one proposed by de Ghellinck and Vial [3] and Ye and Kojima [12], but our derivation was performed independently.

Section 3 is concerned with optimal solutions to the dual problem (D_0) . We show that it is easy to obtain at each iteration an ellipsoid E_k that contains the y -part of every optimal dual solution. The relationships between the ellipsoid method of Khachian [9] and Karmarkar's algorithm were discussed in [10], but the result that the latter generates containing ellipsoids is apparently new. Similar ellipsoids can be generated in the affine variant of the algorithm if the optimal value is known.

Finally, section 4 contains some preliminary computational experience related to the material in sections 2 and 3.

2. An Improved Bound.

The dual of (P) is

$$(D) \quad \begin{array}{ll} \max & z \\ & A^T y + gz \leq c, \end{array}$$

and any (y, z) feasible in (D), and hence also in the unscaled version (D_0) , provides a lower bound z on $v(P) = v(P_0)$. The usual method at each iteration uses

$$y(z) = (AA^T)^{-1} A(c - zg) \quad (8)$$

in the following way:

if $(y(z_k), z_k)$ is not feasible in (D), set

$$y_{k+1} = y_k, \quad z_{k+1} = z_k;$$

otherwise, find (9)

$$\underline{z} = \max \{z: (y(z), z) \text{ feasible in (D)}\}$$

$$\text{and set } y_{k+1} = y(\underline{z}), \quad z_{k+1} = \underline{z}.$$

From (8), $(y(z), z)$ is feasible in (D) iff

$$c_q - zg_q \geq 0;$$

thus obtaining z_{k+1} requires (given c_q and g_q) at most a minimum-ratio test; moreover, each bound z_k obtained (except possibly z_0) is certified valid by a feasible solution to (D_0) .

This updating scheme examines a line segment--consisting of points of the form $(y(z), z)$ --in the feasible region of (D) . At very little extra work we can examine a two-dimensional subset. Note that we already have to compute c_q and g_q . We therefore consider points of the form $(y(w), z)$.

Lemma 2.1. The point $(y(w), z)$ is feasible in (D) iff

$$c_q - wg_q - (z-w)g \geq 0. \quad (10)$$

The proof is trivial.

Hence we suggest the following procedure to update the lower bounds:

if $(y(z_k), z_k)$ is not feasible in (D) , set

$$y_{k+1} = y_k, \quad z_{k+1} = z_k;$$

otherwise, find the optimal solution $(\underline{z}', \underline{w}')$ to (11)

$\max \{z: (z, w) \text{ satisfies (10)}\}$ and set

$$y_{k+1} = y(\underline{w}'), \quad z_{k+1} = \underline{z}'.$$

Note that $z = w = \underline{z}$, where $\underline{z}$ is as in (9), is feasible in (11). Hence we have:

Theorem 2.2. The bound z_{k+1} produced by (11) is valid; indeed, (y_{k+1}, z_{k+1}) is feasible in (D). Moreover, the z_{k+1} produced by (11) is at least as great as that produced by (9).

This result implies that the performance guarantee in (7) remains valid if the lower bounds are updated as in (11). Our hope is that, in practice, improved bounds will lead to better directions and hence fewer iterations. Another possible advantage arises from an observation of Gay [4]. Instead of computing c_q and g_q , Gay suggests first computing $(c - z_k g)_q$; if it has a negative component then $(y(z_k), z_k)$ is not feasible in (D) and z_k and y_k are left unchanged. If $(c - z_k g)_q \geq 0$, then compute g_q , and, since c_q and g_q are thus known, proceed as in (9) or (11). Hence, when the bound z_k is not updated, one projection is saved. This is likely to occur more frequently with improved bounds. Section 4 contains some computational results relevant to these questions.

In one case of special interest, $(\underline{z}', \underline{w}')$ is very simple to calculate. Suppose that $g_0 = e_n$, the n th unit vector as in the problem (P_0) resulting from $(\tilde{P})$ using (1). Then $g = e_n$ also, and (10) requires

$$e_j^T c_q - w e_j^T g_q \geq 0, \quad j = 1, 2, \dots, n-1$$

$$z \leq e_n^T c_q + w(1 - e_n^T g_q).$$

Since $1 = \|g\|^2 \geq \|g_q\|^2 = g^T g_q = e_n^T g_q$, the coefficient of w in the last inequality is nonnegative. Hence w should be taken as large as possible subject to satisfying the first $n-1$ inequalities above, and then z chosen to make the last inequality tight. The resulting value of z' is precisely that proposed by de Ghellinck and Vial [3] (procedure B' of section 5) and by Ye and Kojima [12].

Another special case is when $g_0 = e$. Then (10) can be written as

$$z \leq w + \min_j \{e_j^T X_k^{-1} (c_q - w g_q)\}.$$

Then instead of setting $w = z$ so that the second term vanishes, we choose w to maximize the piecewise-linear concave right-hand side, i.e. find w with -1 a generalized gradient of the second term.

In general, (10) will have some inequalities independent of z (corresponding to zero components of g_0) which define a feasible interval for w . Then an easy search among the breakpoints of

$$w + \min_{j: e_j^T g > 0} \{e_j^T (c_q - w g_q) / e_g^T g\}$$

gives the optimal w' and hence z' .

3. Dual Ellipsoids.

At every iteration we have a lower bound z_k and a strictly positive feasible solution x_k so that, with $A = A_0 X_k$, $c = X_k c_0$ and $g = X_k g_0$, we have

$$Ae = 0 \quad (12)$$

and, for all optimal solutions (y, z_*) to (D) or (D_0) ,

$$A^T y \leq c - z_* g \leq c - z_k g. \quad (13)$$

Let the current duality gap be denoted γ_k , so that

$$\gamma_k = c_0^T x_k - z_k = (c_0 - z_k g_0)^T x_k = (c - z_k g)^T e. \quad (14)$$

For notational simplicity, we will suppress the subscript k . Let a_j denote the j th column of A . Then, using (12) we find

$$a_j = Ae_j = A(e_j - e), \text{ and so}$$

$$a_j^T y = (e_j - e)^T A^T y \geq (e_j - e)^T (c - zg) = (c - zg)^T e_j - \gamma \quad (15)$$

for all dual optimal y . Hence such y 's satisfy

$$(c - zg) - \gamma e \leq A^T y \leq c - zg. \quad (16)$$

We now use the method of Burrell and Todd [2] to derive an ellipsoid E containing all solutions to (16). Let

$$u = c - zg, \quad \ell = c - zg - \gamma e,$$

$$r = (u + \ell)/2 = c - zg - \frac{1}{2}\gamma e, \quad v = (u - \ell)/2 = \frac{1}{2}\gamma e, \quad (17)$$

$$M = (AA^T)^{-1}, \quad \bar{y} = MAr \quad \text{and} \quad t = A^T \bar{y} - r.$$

Then any y satisfying (16) lies in

$$\begin{aligned} E = E_k &= \{y \in R^m: (A^T y - u)^T (A^T y - \ell) \leq 0\} \\ &= \{y \in R^m: (A^T (y - \bar{y}) - t - v)^T (A^T (y - \bar{y}) - t + v) \leq 0\} \\ &= \{y \in R^m: (y - \bar{y})^T M^{-1} (y - \bar{y}) \leq v^T v - t^T t\} \end{aligned} \quad (18)$$

This is an ellipsoid with center $\bar{y} = (AA^T)^{-1} A(c - zg - \frac{1}{2}\gamma e)$; by (12),

$$\bar{y} = (AA^T)^{-1} A(c - zg) = y(z). \quad (19)$$

Moreover, substituting for r and $\bar{y}$ gives

$$t = \frac{1}{2}\gamma e - (c - zg)_q, \quad (20)$$

so that the right-hand side of the ellipsoid inequality in (18) is

$$\begin{aligned}
 v^T v - t^T t &= \|\frac{1}{2}\gamma e\|^2 - \|\frac{1}{2}\gamma e - (c-zg)_q\|^2 \\
 &= \gamma e^T (c-zg)_q - \|(c-zg)_q\|^2, \\
 &= \gamma^2 - \|(c-zg)_q\|^2,
 \end{aligned} \tag{21}$$

since $(c-zg)_q^T e = (c-zg)^T e_q = (c-zg)^T e = \gamma$. Note that all the entities defining the ellipsoid E , namely M , $\bar{y}$ and $v^T v - t^T t$, arise naturally or can be computed simply from entities that appear in the algorithm described in the introduction.

While the ellipsoid method is concerned with volumes of ellipsoids, it is more illuminating in our context to examine the radius ρ of an n -dimensional ball of equal volume to E . From (18), (17) and (21),

$$\begin{aligned}
 \rho &= (v^T v - t^T t)^{\frac{1}{2}} (\det B)^{1/2n} \\
 &= (\gamma^2 - \|(c-zg)_q\|^2)^{\frac{1}{2}} / (\det A A^T)^{1/2n} \\
 &\leq \gamma / (\det A A^T)^{1/2n}.
 \end{aligned} \tag{22}$$

Note that, if A^T is factored as QR , where Q has orthonormal columns and R is nonsingular and upper triangular, then

$(\det AA^T)^{1/2n} = (\det R)^{1/n}$ is easily computed from the diagonal entries of R .

To analyze the behavior of ρ , we establish a result on $\det AA^T$; a similar result was proved by Kallio [7].

Lemma 3.1. If the feasible region of (P_0) is nonempty, bounded and nondegenerate, i.e., each feasible x has at least $m + 1$ positive components, then $\det AA^T = \det A_0 X_k^2 A_0^T$ is bounded and bounded away from zero as x_k ranges over all feasible points.

Proof. Since $\det A_0 X_k^2 A_0^T$ is a continuous function of x_k , it attains its maxima and minima over the compact feasible region. It is also nonnegative since $A_0 X_k^2 A_0^T$ is positive semi-definite. Hence we only need to show that $A_0 X_k^2 A_0^T$ is never singular.

Suppose for some x_k , $A_0 X_k$ has row rank less than m . Then

$\begin{bmatrix} A_0 X_k \\ g_0^T X_k \end{bmatrix}$ has row rank less than $m + 1$, and we can find a

feasible solution x to $A_0 X_k x = 0$, $g_0^T X_k x = 1$, $x \geq 0$, with fewer than $m + 1$ positive components. This violates our nondegeneracy assumption.

Thus the behavior of ρ is related to that of γ as k tends to infinity. Using (7), and reinstating the subscripts k , we know that $\{\gamma_k\}$ is bounded by a sequence converging linearly to zero. To summarize the results of this section, we have:

Theorem 3.2. At each iteration of the projective algorithm, the ellipsoid E_k given by (18) contains the y-parts of all dual optimal solutions. Moreover, if ρ_k is the radius of a ball of equal volume to E_k , then $\{\rho_k\}$ is bounded by a sequence converging linearly to zero, and decreasing by a constant factor every n iterations.

This result contrasts quite strikingly with what can be established for the ellipsoid method. There, the volume decreases by a constant ratio every n iterations, while here the radius does, so that the volume decreases by a constant ratio every iteration (on average). Furthermore, in practice the ellipsoid method behaves similarly to its worst-case bound, while Karmarkar's method appears to decrease r_k (and hence ρ_k , on average) by a constant at every iteration. Section 4 presents some computational results corroborating this finding: the ellipsoids decrease in volume by a factor of about 2^{-n} each iteration, while the factor would be closer to $(1-2/n)$ for the ellipsoid method. Of course, each iteration of the latter is much simpler than an iteration of Karmarkar's algorithm.

The most obvious way to use the information provided by E_k is in eliminating primal variables. By complementary slackness, $e_j^T x = 0$ for all primal optimal x if

$$a_j^T y < e_j^T (c - z_* g)$$

for any dual optimal (y, z_*) ; since e is feasible in (P) , this holds if

$$a_j^T y < e_j^T (c - (c^T e)g) \quad (23)$$

for any dual optimal (y, z_*) . We can now use the technique in Burrell and Todd [2] to provide an upper bound to $a_j^T y$ for $y \in E$; if this bound is less than the right-hand side of (23), we know that $e_j^T x = 0$ in all optimal x .

Computing such a bound requires the solution of the linear system

$$(AA^T)y = a_j;$$

it therefore seems worthwhile to select candidate indices by some heuristic before considering them for removal.

To conclude this section we consider the affine variant of Karmarkar's algorithm, applied directly to a problem in standard form. To illustrate, assume that $x = e$ is feasible in the (possibly scaled) problem

$$\begin{aligned} \min \quad & c^T x \\ (Q) \quad & Ax = b \\ & x \geq 0. \end{aligned}$$

Any dual optimal solution y must satisfy $A^T y \leq c$. Since $Ae = b$, $-a_j = -b + A(e - e_j)$, and so

$$-a_j^T y = -b^T y + (A^T y)^T (e - e_j) \leq -b^T y + c^T (e - e_j).$$

Now, if z is a lower bound on the optimal value of (Q) , then $b^T y \geq z$ for all dual optimal y and hence

$$c - \gamma e \leq A^T y \leq c \quad (24)$$

for $\gamma = c^T e - z$ the current duality gap.

We may now use (24) as we used (16) to derive a containing ellipsoid for all dual optimal y . Unfortunately, the algorithm does not naturally provide any lower bound z to be used in this way; thus this approach may only be useful if the optimal value z_* of (Q) is known.

4. Computational results.

In this section we report on some computational experiments made to elucidate the theoretical analysis in sections 2 and 3. Random standard form problems of various dimensions $m \times n$ were generated as follows: each entry of $\tilde{A}$, $\tilde{c}$, and an n -vector $\tilde{x}$ was chosen as a uniform $[0,1]$ random variable independently, and then $\tilde{b}$ was set to $\tilde{A}\tilde{x}$. Each such problem was converted to a problem of the form (P) by setting

$$\begin{aligned} x^T &= (\tilde{x}^T, \xi, \lambda) \\ A_0 &= (\tilde{A}, -\tilde{b}, \tilde{b} - \tilde{A}\tilde{e}) \\ c_0^T &= (\tilde{c}^T, 0, \nu) \\ g_0^T &= (0^T, 1, 0); \end{aligned}$$

thus ξ is identically equal to one while λ is an artificial variable with cost $\nu \gg 1$. We applied the algorithm of section 1 with the modification that λ was set to zero and eliminated from the problem whenever the line search permitted; this typically occurred on the first or second iteration. We set $\nu = 10^{10}$ and $z_0 = -10^5$ in our code; thus while λ is reasonably large the lower bound is negligible compared to the primal objective function value, while when λ is eliminated, z_0 is probably a valid lower bound for well-scaled problems.

The code uses a QR factorization of A^T at each iteration to compute the projections (and the radius ρ). We use a crude safeguarded line search to approximately minimize the potential function, while guaranteeing a decrease at least that achieved by $\alpha = .33$.

The termination criterion used is

$$c_0^T x_k - z_k \leq 10^{-4} \max \{|c_0^T x_k|, 1\};$$

we do not attempt to obtain an exact optimal solution. The program is written in FORTRAN using single precision, and compiled using IBM's FORTRAN 2.0 compiler on an IBM PC.

For each size of problem $m \times n$ we generated 10 random problems and ran each three times: once with z_k updated as in (9) in section 2 (denoted $\underline{z}$), once with z_k updated as in (11) in section 2 (denoted $\underline{z}'$), and once with $z_k \equiv z_*$, the optimal value of the problem (denoted z_*).

For each size of problem and each such strategy we recorded: the average number of iterations, AVIT; the average decrease in the potential function achieved in the line search, AVPOT; the average decrease in the logarithm of the radius ρ of a ball of equal volume to E , AVLRAD; and the number of times the bound z_k was unchanged, NBDUNC. The results are given in table I.

Overall, we notice that the effects of the improved bound z' in reducing the number of iterations or increasing the reduction in potential are slight or nonexistent, but that use of this bound does lead to fewer updates in z_k . The use of the optimal value consistently reduces the number of iterations required, by .5-2.5 on average. The radius of the equivalent ball does seem to decrease by an average factor of at least 2 in every case, so that the volume is decreasing by 2^n on average. The reduction achieved lessens as the dimension increases, while the potential decrease increases slightly and the average number of iterations grows even more slowly. Definitive conclusions must await more extensive and realistic testing, but the rapidly shrinking ellipsoids do appear of potential use in eliminating variables.

Dimensions	Strategy	AVIT	AVPOT	AVLRAD	NBDUNC
5 × 10	<u>z</u>	9.6	4.7	1.3	13
	<u>z'</u>	10.0	5.3	1.3	17
	<u>z</u> *	9.1	8.3	1.4	-
10 × 20	<u>z</u>	11.8	5.7	1.1	11
	<u>z'</u>	10.7	7.5	1.1	15
	<u>z</u> *	9.5	13.7	1.2	-
20 × 40	<u>z</u>	13.1	9.7	.92	17
	<u>z'</u>	12.8	12.6	.94	18
	<u>z</u> *	12.4	20.1	.99	-
30 × 60	<u>z</u>	13.8	12.8	.86	14
	<u>z'</u>	13.6	10.3	.86	18
	<u>z</u> *	12.4	29.8	.89	-
40 × 80	<u>z</u>	15.4	21.4	.76	16
	<u>z'</u>	16.0	14.2	.74	19
	<u>z</u> *	14.0	37.0	.80	-
50 × 100	<u>z</u>	15.4	22.6	.78	18
	<u>z'</u>	15.4	21.8	.77	25
	<u>z</u> *	13.0	50.0	.84	-

Table I

References

- [1] K.M. Anstreicher, "Analysis of Karmarkar's algorithm for fractional linear programming", manuscript, School of Organization and Management, Yale University, New Haven, Connecticut (1985).
- [2] B.P. Burrell and M.J. Todd, "The ellipsoid method generates dual variables", Mathematics of Operations Research 10 (1985) 688-700.
- [3] G. de Ghellinck and J.-P. Vial, "A polynomial Newton method for linear programming", manuscript, CORE, Catholic University of Louvain, Belgium (1986).
- [4] D. Gay, "A variant of Karmarkar's linear programming algorithm for problems in standard form", to appear in Mathematical Programming (1985).
- [5] C. Gonzaga, "A conical projection algorithm for linear programming", manuscript, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA (1985).
- [6] D. Jensen and A. Steger, private communication, Department of Applied Mathematics and Statistics, State University of New York at Stonybrook, Stonybrook, New York (1985).
- [7] M. Kallio, "On gradient projection for linear programming", manuscript, School of Organization and Management, Yale University, New Haven, CT (1986).
- [8] N. Karmarkar, "A new polynomial time algorithm for linear programming", Combinatorica 4 (1984) 373-395.
- [9] L.G. Khachian, "Polynomial algorithms in linear programming", USSR Computational Mathematics and Mathematical Physics 20 (1980) 53-72.
- [10] M.J. Todd, "Polynomial algorithms for linear programming", manuscript, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY (1986).
- [11] M.J. Todd, "Exploiting special structure in Karmarkar's linear programming algorithm", manuscript, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY (1986).
- [12] Y. Ye and M. Kojima, "Recovering optimal dual solutions in Karmarkar's polynomial algorithm for linear programming", manuscript, Department of Engineering Economic Systems, Stanford University (1986).