

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

TECHNICAL REPORT NO. 626

March 1984

THE ELLIPSOID METHOD GENERATES
DUAL VARIABLES

by

Bruce P. Burrell
and
Michael J. Todd¹

¹Research supported in part by a fellowship from the Alfred P. Sloan Foundation and by NSF Grant ECS82-15361.

Abstract

We show that the ellipsoid algorithm applied to a system of linear inequalities can be implemented in such a way that at each iteration there is a short proof of the containment of the feasible region in the current ellipsoid. Moreover, the data describing each ellipsoid also generate dual variables that provide bounds on the linear functions appearing in the inequalities.

1. Introduction

The ellipsoid method was introduced by Yudin and Nemirovsky [12] for convex programming and became famous (and later notorious) when Khachian [7] used it to demonstrate that linear programming problems could be solved in polynomial time. For a survey on these results and the ensuing research, see Bland, Goldfarb, and Todd [2].

In this paper we are concerned with the application of the ellipsoid method to find a point in

$$X = \left\{ x \in \mathbb{R}^n : A^T x \leq u \right\} \quad (1)$$

if one exists. We assume that X is either empty or of full dimension. This problem is equivalent to the solution of linear programming problems, and we shall describe briefly in the concluding section how to modify the algorithm for feasibility to handle linear optimization efficiently.

The standard ellipsoid method generates a sequence E_k of ellipsoids, each containing X . The ellipsoid E_k is represented by its center y_k and a symmetric positive definite matrix B_k , so that

$$E_k = \left\{ x : (x - y_k)^T B_k^{-1} (x - y_k) \leq 1 \right\} \quad (2)$$

In its simplest form, we check at each iteration whether $y_k \in X$; if so we stop, otherwise we choose j with $a_j^T y_k > u_j$, where a_j is the j^{th} column of A and u_j the corresponding component of u . Then E_{k+1} is chosen as the smallest ellipsoid containing the half-ellipsoid

$$\left\{ x \in E_k : a_j^T x \leq a_j^T y_k \right\};$$

E_{k+1} is given as in (2) by its center y_{k+1} and the symmetric positive definite matrix B_{k+1} defined by

$$y_{k+1} = y_k - \frac{\tau B_k a_j}{(a_j^T B_k a_j)^{1/2}} \quad (3)$$

$$B_{k+1} = \delta \left\{ B_k - \frac{\sigma B_k a_j a_j^T B_k}{a_j^T B_k a_j} \right\} \quad (4)$$

where $\tau = \frac{1}{n+1}$, $\delta = \frac{n^2}{n^2-1}$ and $\sigma = \frac{2}{n+1}$.

Many more sophisticated strategies have been proposed, which generally lead to a similar update with modified parameters τ , δ , and σ . Even the simplest version above guarantees that $\text{vol}(E_{k+1})/\text{vol}(E_k) \leq \exp(-1/2(n+1))$, and this systematic reduction of volume gives bounds on the number of steps of the algorithm before it can be concluded that X is empty, under certain conditions.

If the algorithm above is carried out in exact arithmetic, it is known that $E_k \supset X$ for each k , but the only way to demonstrate this to a skeptic is to show that $E_0 \supset X$ and that E_k was generated by a sequence of valid steps as above. If finite precision arithmetic is used, then it is possible that E_k fails to contain X . Rigorous proofs of the convergence of the ellipsoid algorithm require one to specify the precision to which the computations must be carried out (in terms of the length of input of the initial data A and u , assumed to be integer), and to make a slight increase in δ above to ensure that, even after rounding errors, E_{k+1} will contain X if E_k did. The accuracy required is generally very high. See e.g., [3,6].

Most implementations of the ellipsoid algorithm, by contrast, use floating-point arithmetic, and try to ensure as much accuracy as possible by updating factorizations of B_k rather than B_k itself. For example, several authors (see, e.g., [2,5]) have recommended using $B_k = L_k \Delta_k L_k^T$, where L_k is unit lower triangular and Δ_k is diagonal, with positive diagonal entries. One cannot easily guarantee, however, that E_k will contain X .

Next, note that the minimum value of $a_j^T x$ over $x \in E_k$ is $a_j^T y_k - (a_j^T B_k a_j)^{1/2}$; if

this is greater than u_j , then X must be empty. Once again, however, it is very hard to convince a skeptic of this fact; it is necessary to go through all of the iterations again. A much more satisfactory way to convince an observer would be to exhibit a vector μ with

$$A\mu = 0, \mu \geq 0, \text{ and } u^T \mu < 0, \quad (5)$$

clearly demonstrating that X is empty. The Farkas lemma guarantees that such a μ exists if X is empty.

We will show in this paper that these two drawbacks of the ellipsoid method are illusory. Each ellipsoid can be represented in such a way that a skeptic can be easily convinced that it contains X , and infeasibility can be demonstrated by a theorem of the alternative. Indeed, we shall maintain a dual vector λ_j that certifies a lower bound on $a_j^T x$ for all feasible x , for all j . As we shall see, these two features are very closely related. For simplicity, we work with exact arithmetic; if there are inaccuracies due to rounding errors, we may use the short proofs mentioned above to restart the method as accurately as desired, without starting from scratch.

The two desirable features, guaranteed containment and dual variables, have previously been studied for variants of the ellipsoid method. Levin and Yamnitsky [8] described a "simplex method", in which X was contained in a sequence of shrinking simplices, rather than ellipsoids. Each simplex S_k is of the form

$$S_k = \left\{ x: B'_k A^T x \leq B'_k u \right\} \quad (6)$$

for some nonnegative B'_k with $n+1$ rows. Thus B'_k provides a short proof that $S_k \supset X$. While a great deal of flexibility in updating S_k is present, [8] described only a method that guarantees $\log \text{vol } S_{k+1} - \log \text{vol } S_k \leq \vartheta(n^{-2})$, compared to $\vartheta(n^{-1})$ for ellipsoid algorithms.

Wolsey [11] considered relaxation methods [1,9], which can be viewed as

precursors of the ellipsoid methods with B_k held fixed -- see [2]. For a certain sequence of nonnegative scalars $\{\lambda_k\}$, the iterates are given by

$$y_{k+1} = y_k - \frac{\lambda_k a_{j(k)}}{\|a_{j(k)}\|}, \quad (7)$$

where $a_{j(k)}^T y_k > u_{j(k)}$. Wolsey noted that, if y_{1+p} were close to y_1 ,

$$\text{then } \sum_{k=1+p}^{1+p} \frac{\lambda_k a_{j(k)}}{\|a_{j(k)}\|} \approx 0; \quad (8)$$

thus we have an approximate nonnegative solution to $A\mu = 0$. The accuracy depends on the distance $\|y_{1+p} - y_1\|$, the size of p , and lower bounds on the λ_k 's; one can only achieve good approximate dual solutions μ if the convergence of the $\{y_k\}$ is slow. Moreover, it appears to be impossible to modify the argument for the ellipsoid algorithm where $a_{j(k)}$ is replaced by $B_k a_{j(k)}$. We will show that exact dual solutions can be generated at each iteration of the ellipsoid method.

In the next section we describe how an ellipsoid can be represented in such a way as to certify its containment of the polyhedron X . Section 3 discusses the generation of lower bounds via dual solutions, while updating the ellipsoid is addressed in section 4. Section 5 gives the resulting algorithm, and in section 6 we provide some concluding discussion. The Appendix provides justification for our choice of λ .

Our notation is as follows. As above, we use subscripts both for indexing (as in y_k) and to indicate components (as in u_j); no confusion should result. In particular, a_i denotes the i^{th} column of the matrix A . We use e_i for the i^{th} unit vector of appropriate dimension. For a matrix $C = [c_{ij}]$, C_+ denotes $(\max \{0, c_{ij}\})$, and $C_- = (\max \{0, -c_{ij}\})$, so that $C_+, C_- \geq 0$ and $C = C_+ - C_-$.

2. Ellipsoids Containing The Feasible Region

We will assume that the original system of linear inequalities includes bounds on all the variables, i.e., that it is of the form $\tilde{A}^T x \leq \tilde{b}$, $x \leq u_x$, $-x \leq -l_x$. If such bounds are not included, they may be added without affecting the feasibility of the system as long as \tilde{A} and \tilde{b} are integer -- we may take each component of u_x to be 2^L and each component of l_x to be -2^L , where L is the length of the encoding of the problem (see [3]).

Let us write $A = [I, -I, \tilde{A}]$ and $u^T = [u_x^T, -l_x^T, \tilde{b}^T]$, and let m denote the number of columns of A . We assume without loss of generality that each column a_i of A is nonzero. Then $X = \left\{ x \in \mathbb{R}^n : A^T x \leq u \right\}$.

Moreover, because we have bounds on all the variables, we may deduce a lower bound on $A^T x$ for $x \in X$. Set $l = A_+^T l_x - A_-^T u_x$ and note that $l^T = [l_x^T, -u_x^T, l_x^T \tilde{A}_+ - u_x^T \tilde{A}_-]$. Then

$$X = \left\{ x \in \mathbb{R}^n : l \leq A^T x \leq u \right\}. \quad (9)$$

We can alternately define X by quadratic inequalities:

$$X = \left\{ x \in \mathbb{R}^n : (a_i^T x - l_i)(a_i^T x - u_i) \leq 0 \text{ for all } i \right\}.$$

Now choose any nonnegative diagonal matrix $D = \text{diag}(d_1, \dots, d_m)$. By combining the quadratic inequalities above with weights d_i we see that X is contained in the set $E = E(D, l)$, where

$$E = \left\{ x \in \mathbb{R}^n : (A^T x - l)^T D (A^T x - u) \leq 0 \right\}. \quad (10)$$

The fact that D is nonnegative and diagonal provides a short proof that $X \subset E$.

Note that $ADA^T = (AD^{\frac{1}{2}})(D^{\frac{1}{2}}A^T)$ is positive semi-definite, where $D^{\frac{1}{2}}$ is $\text{diag}(d_1^{\frac{1}{2}}, \dots, d_m^{\frac{1}{2}})$. Suppose that ADA^T is nonsingular, and hence positive definite. Then E is an ellipsoid. Define

$$\begin{aligned} r &= \frac{u + l}{2}, \\ B &= (ADA^T)^{-1}, \text{ and} \\ y &= BADr. \end{aligned} \tag{11}$$

Then the inequality defining E can be rewritten as

$$\begin{aligned} x^T ADA^T x - 2r^T DA^T x + l^T Du &\leq 0, \text{ or} \\ (x - y)^T B^{-1} (x - y) &\leq y^T B^{-1} y - l^T Du. \end{aligned} \tag{12}$$

Thus y is the center of the ellipsoid E.

There is another way of looking at the problem that is useful. Define

$$v = \frac{u - l}{2}$$

so that $u = r + v$, $l = r - v$. Then, defining $s = s(x) = A^T x - r$, we see that

$$X = \left\{ x \in \mathbb{R}^n : -v \leq s \leq v \right\}.$$

Again, with D a nonnegative diagonal matrix we find $X \subset E$, where

$$E = \left\{ x \in \mathbb{R}^n : s^T D s \leq v^T D v \right\}.$$

The left side is minimized by solving the least squares problem of minimizing

$\|D^{\frac{1}{2}}(A^T x - r)\|_2$; if ADA^T is positive definite, the solution is $x = y = BADr$, where $B = (ADA^T)^{-1}$. Let t denote $s(y) = A^T y - r$, and note that $ADt = 0$. Thus writing $s = t + A^T(x - y)$, the quadratic inequality defining E is $t^T Dt + (x - y)^T B^{-1} (x - y) \leq v^T Dv$, (since $t^T DA^T(x - y) = 0$), or

$$(x - y)^T B^{-1} (x - y) \leq v^T Dv - t^T Dt$$

This inequality and (12) are identical.

Consider the right hand side of these inequalities,

$$\Psi = y^T B^{-1} y - l^T Du = v^T Dv - t^T Dt.$$

If Ψ is negative then E is empty, and hence so is X. If Ψ is zero, E degenerates to a single point $\{y\}$. Thus either $X = \{y\}$ or X is empty. Hence,

in general, $\Psi > 0$. In this case, the matrix D (and therefore B) can be scaled so that $\Psi = 1$, and we assume this scaling henceforth. We now have a representation of the ellipsoid E ,

$$E = \{x \in \mathbb{R}^n: (x - y)^T B^{-1} (x - y) \leq 1\}, \quad (13)$$

as in [2,3,6], with the advantage of a succinct certificate ($D \geq 0$) that it contains X .

To obtain an initial ellipsoid E of this form is trivial.

Let

$$\begin{aligned} D &= \text{diag} \left[\frac{1}{nv_1^2}, \dots, \frac{1}{nv_n^2}, 0, \dots, 0 \right], \\ y &= \frac{(u_x + l_x)}{2}, \text{ and} \\ B &= \text{diag} (nv_1^2, \dots, nv_n^2). \end{aligned} \quad (14)$$

We show in the ensuing sections that the ellipsoid algorithm can be implemented in such a way that each ellipsoid generated is of the form $E(D, l)$, possibly with an updated and improved l . We also maintain throughout a matrix A of "dual variables" establishing that l is indeed a valid lower bound. We will have

$$AA^T = -A, \quad A \geq 0, \quad \text{and} \quad u^T A = -l^T, \quad (15)$$

so that $A^T x \leq u$ implies $A^T x = -A^T A^T x \geq -A^T u = l$ as desired. Note that λ_i , the i^{th} column of A , satisfies $A^T \lambda_i = -a_i$, the i^{th} column of A , while $u^T \lambda_i = -l_i$; thus λ_i is a vector of dual variables certifying the lower bound l_i .

Initially, (15) holds with

$$A^T = (A_-^T, A_+^T, 0). \quad (16)$$

Suppose that we have some ellipsoid $E = E(D, l)$, where D is scaled so that

$$y^T B^{-1} y - l^T D u = v^T D v - t^T D t = 1.$$

Thus E is given by (10) and (13). If $y \in X$ then we have obtained a feasible solution; hence assume that y is not in X , so that $a_j^T y > u_j$ for some j .

Equivalently, $t_j > v_j$.

Let $\gamma_i = a_i^T B a_i$ for each i . Then $a_j^T x$ is minimized over E by $z = y - \gamma_j^{-1/2} B a_j$, which gives a value of $a_j^T z = a_j^T y - \gamma_j^{1/2}$. (A short proof of this is given in the Appendix.) If $a_j^T z > l_j$, we have an improved lower bound on $a_j^T x$ for all $x \in X$. This bound is not derived in a very natural way -- in particular, it does not correspond to a vector λ_j of dual variables as above. In section 3 we show how to obtain a vector $\bar{\lambda}_j \geq 0$ from the current ellipsoid with $A \bar{\lambda}_j = -a_j$ and $\bar{l}_j = -u^T \bar{\lambda}_j \geq a_j^T z$; thus $\bar{\lambda}_j$ certifies a new lower bound at least as good as that provided by the "ellipsoid minimizer" z . We therefore update l_j to \bar{l}_j and replace the j^{th} column of A by $\bar{\lambda}_j$.

We may find that the updated l_j is greater than u_j . In this case it is clear that X is empty. Indeed, if $\mu = e_j + \bar{\lambda}_j$ (where e_j is the j^{th} unit vector), we have $\mu \geq 0$, $A\mu = 0$, and $u^T \mu < 0$; thus μ provides a short proof of infeasibility.

After a possible update of l_j , and assuming that we have not demonstrated infeasibility, we have $u_j \geq l_j$, $a_j^T y > u_j$, and the hyperplanes $a_j^T x = l_j$ and $a_j^T x = u_j$ both intersect the current ellipsoid E . In such a case, Todd [10] has shown how to obtain the minimum volume ellipsoid E' with

$$E' \supset \left\{ x \in E: l_j \leq a_j^T x \leq u_j \right\}.$$

Moreover, the proof in [10] shows that the quadratic inequality defining E' is a convex combination of that defining E and $(a_j^T x - l_j)(a_j^T x - u_j) \leq 0$. It may appear, then, that this new ellipsoid is also of the form $E(D', l)$ for some D' obtained by increasing d_j (and scaling). This is indeed true, unless $d_j > 0$ and l_j has been updated. In this case, E already contains as part of its

defining inequality some multiple of $(a_j^T x - \hat{l}_j)(a_j^T x - u_j)$, where \hat{l}_j is the lower bound before updating.

The simplest way to avoid this difficulty is to remove this part of the inequality before updating. We describe this process in section 4, obtaining a new ellipsoid $\bar{E} = E(\bar{D}, 1)$ that contains X and has volume no bigger than that of E' .

The combination of these modifications yields an ellipsoid algorithm that generates a sequence of ellipsoids, all of form $E(D, 1)$ for some nonnegative diagonal D , and decreasing in volume by a factor of at worst $\exp(-1/2(n+1))$ at each step. The algorithm is summarized in section 5. Some concluding remarks are made in section 6.

3. Generating Lower Bounds By Duality

Suppose we have a current ellipsoid $E = E(D,1)$ as in (10) and (13), with B , r , and y as in (11). We wish to obtain a lower bound on $a_j^T x$ for all $x \in X$ that is at least as good as $a_j^T z$, where $z = y - \gamma_j^{-1/2} B a_j$, $\gamma_j = a_j^T B a_j$, minimizes $a_j^T x$ over $x \in E$.

Consider the linear programming problem

$$\begin{aligned} \min \quad & a_j^T x \\ & A^T x \geq 1 \\ & -A^T x \geq -u \end{aligned} \tag{18}$$

Its dual is

$$\begin{aligned} \max \quad & 1^T \lambda' - u^T \lambda'' \\ & A \lambda' - A \lambda'' = a_j \\ & \lambda', \lambda'' \geq 0 \end{aligned}$$

Since $1 \leq u$, an equivalent statement is to maximize $f(\lambda) \equiv 1^T \lambda_- - u^T \lambda_+$, subject to $A \lambda = -a_j$.

By linear programming duality, any λ with $A \lambda = -a_j$ affords a lower bound $f(\lambda)$ on $a_j^T x$ for all $x \in X$. We seek such a λ with $f(\lambda) \geq a_j^T z$, and we show below that

$$\lambda = \gamma_j^{-1/2} D (A^T z - r) = \gamma_j^{-1/2} D s(z) \tag{19}$$

suffices. (This choice of λ is motivated by consideration of the convex relaxation of (18) where the feasible region X is replaced by E , and is discussed in the Appendix.)

Proposition 3.1 With λ as in (19), $A \lambda = -a_j$.

Proof We have $A \lambda = \gamma_j^{-1/2} A D (A^T y - r - A^T (y - z))$

$$\begin{aligned}
 &= -\gamma_j^{\frac{1}{2}} ADA^T(y - z) \\
 &= -a_j
 \end{aligned}$$

since $ADt = 0$ and $ADA^T(y - z) = B^{-1}(y - z) = \gamma_j^{-\frac{1}{2}}a_j$. ■

Thus λ is dual feasible. To show that it yields a better bound than $a_j^T z$, define

$$\mu_i = \left(\frac{\gamma_j^{\frac{1}{2}}}{2}\right) d_i \{2a_i^T z(a_i^T z - r_i) - (a_i^T z - l_i)(a_i^T z - u_i)\},$$

for all i , and note that if $\lambda_i \neq 0$ (so $d_i \neq 0$ and $a_i^T z \neq r_i$), then

$$\begin{aligned}
 \frac{\mu_i}{\lambda_i} &= a_i^T z - \frac{(a_i^T z - l_i)(a_i^T z - u_i)}{2(a_i^T z - r_i)} = l_i + \frac{(a_i^T z - l_i)^2}{2(a_i^T z - r_i)} \\
 &= u_i + \frac{(a_i^T z - u_i)^2}{2(a_i^T z - r_i)}.
 \end{aligned}$$

Now if $\lambda_i > 0$, then λ_i and $a_i^T z - r_i$ have the same sign and

$$\mu_i = \lambda_i \left\{ \frac{\mu_i}{\lambda_i} \right\} = \lambda_i \left\{ u_i + \frac{(a_i^T z - u_i)^2}{2(a_i^T z - r_i)} \right\} \geq \lambda_i u_i.$$

Similarly if $\lambda_i < 0$, then

$$\mu_i = \lambda_i \left\{ \frac{\mu_i}{\lambda_i} \right\} = \lambda_i \left\{ l_i + \frac{(a_i^T z - l_i)^2}{2(a_i^T z - r_i)} \right\} \geq \lambda_i l_i.$$

Finally, if $\lambda_i = 0$, then either $d_i = 0$ so $\mu_i = 0$, or $a_i^T z = r_i$ so

$$(a_i^T z - l_i)(a_i^T z - u_i) \leq 0 \text{ and}$$

$$\mu_i = -\frac{\gamma_j^{\frac{1}{2}}}{2} d_i (a_i^T z - l_i)(a_i^T z - u_i) \geq 0.$$

Hence $\sum \mu_i \geq -f(\lambda)$. Finally,

$$\sum \mu_i = \lambda^T A^T z - \frac{\gamma_j^{\frac{1}{2}}}{2} \sum d_i (a_i^T z - l_i)(a_i^T z - u_i)$$

$$\begin{aligned}
 &= -a_j^T z - \frac{\gamma_j^{1/2}}{2} \sum d_i (a_i^T z - 1_i)(a_i^T z - u_i) \\
 &= -a_j^T z,
 \end{aligned}$$

since z satisfies the quadratic inequality defining E with equality. We have therefore proved

Theorem 3.2 With λ as in (19), $f(\lambda) \geq a_j^T z$. ■

The vector λ above proves a lower bound $\bar{l}_j = f(\lambda)$ on $a_j^T x$ that is at least as good as $a_j^T z$; we may not, however, have $\lambda \geq 0$, i.e., \bar{l}_j may use some previously established lower bounds. Let us therefore define

$$\bar{\lambda}_j = A\lambda_- + \lambda_+ \geq 0 \quad (20)$$

Then $A\bar{\lambda}_j = A\lambda_- + A\lambda_+ = A(\lambda_+ - \lambda_-) = A\lambda = -a_j$, and

$-u^T \bar{\lambda}_j = -u^T A\lambda_- - u^T \lambda_+ = 1^T \lambda_- - u^T \lambda_+ = f(\lambda)$. Thus $\bar{\lambda}_j$ has precisely the

properties we desire. (Note that we have used λ_i above for the i^{th} component of λ , whereas $\bar{\lambda}_j$ is the vector certifying a new lower bound for $a_j^T x$; no confusion should result.)

4. Updating The Ellipsoid

Again we have the current ellipsoid $E(D,1)$ and B, r, v, y , and t as in section 2. We have determined that y is not in X , and have j with $a_j^T y > u_j$, i.e. $t_j > v_j$. Also, let $\gamma_j = a_j^T B a_j$ and set $z = y - \gamma_j^{-1/2} B a_j$.

If $l_j < a_j^T z$, we could use the method of section 3 to derive a new lower bound at least as great as $a_j^T z$. As pointed out in section 2, however, if $d_j > 0$ the current ellipsoid is defined in part by the old lower bound l_j ; moreover, if we updated the ellipsoid by replacing l_j by its improved value \bar{l}_j , we might find that $\bar{l}_j < a_j^T \bar{z}$, where \bar{z} is the minimizer of $a_j^T x$ over the new ellipsoid. We shall first remove the effect of the j^{th} constraint on the current ellipsoid, then obtain a new lower bound if necessary, and finally update this intermediate ellipsoid.

First we prove that the removal is possible. In fact, we show under what condition the i^{th} constraint can be removed.

Proposition 4.1 Let $\bar{D} = D - d_i e_i e_i^T$ and $\gamma_i = a_i^T B a_i$. We have

$d_i \gamma_i \leq 1$, with equality only if $t_i = 0$. If $t_i \neq 0$, then $d_i \gamma_i < 1$ and $\bar{A} \bar{D} \bar{A}^T$ is positive definite.

Proof Note that in any case $\bar{A} \bar{D} \bar{A}^T$ is positive semi-definite.

Since $\bar{A} \bar{D} \bar{A}^T = \bar{A} D \bar{A}^T + d_i a_i a_i^T$, multiplying on the left and right by $B a_i$ yields

$$a_i^T B a_i = a_i^T B (\bar{A} \bar{D} \bar{A}^T) B a_i + d_i (a_i^T B a_i)^2, \text{ so}$$

$$a_i^T B (\bar{A} \bar{D} \bar{A}^T) B a_i = \gamma_i (1 - d_i \gamma_i).$$

The left hand side is nonnegative. Since $\gamma_i > 0$, we have $1 - d_i \gamma_i \geq 0$. If equality holds, the left hand side is zero, so

$$\bar{D} \bar{A}^T B a_i = 0; \tag{21}$$

thus $a_i^T y = a_i^T B A D r = a_i^T B (A D r + d_i r_i a_i) = d_i \gamma_i r_i = r_i$; hence $d_i \gamma_i = 1$ yields $t_i = 0$. Finally, if $t_i \neq 0$ then we have shown that $d_i \gamma_i < 1$, and since $\det(A D A^T) = (1 - d_i \gamma_i) \det(A D A^T)$ is positive, $A D A^T$ is nonsingular and hence positive definite. ■

For $i = j$, we have $t_j > v_j \geq 0$, so with $\tilde{D} = D - d_j e_j e_j^T$, $A \tilde{D} A^T$ is certainly positive definite and $d_j \gamma_j < 1$.

4.1 Removing a_j

Here we obtain the updated quantities \tilde{B} , \tilde{D} , \tilde{y} , and \tilde{t} corresponding to changing D to $\tilde{D} = D - d_j e_j e_j^T$. Note that $\tilde{D} \neq \tilde{D}$, since we wish to preserve our scaling, i.e.

$$v^T \tilde{D} v = \tilde{t}^T \tilde{D} \tilde{t} = 1.$$

The resulting ellipsoid is denoted \tilde{E} . It is convenient to let

$$\theta_j = \frac{d_j}{(1 - d_j \gamma_j)}; \quad (22)$$

as noted above, we have $1 - d_j \gamma_j > 0$. Since $A \tilde{D} A^T = A D A^T - d_j a_j a_j^T$,

we find

$$\tilde{B} = (A \tilde{D} A^T)^{-1} = B + \theta_j B a_j a_j^T B.$$

Set $\tilde{y} = \tilde{B} A D r$. Then

$$\begin{aligned} \tilde{y} &= (B + \theta_j B a_j a_j^T B) (A D r - d_j r_j a_j) \\ &= B A D r + (\theta_j a_j^T B A D r - d_j r_j - \theta_j \gamma_j d_j r_j) B a_j \\ &= y + \theta_j (a_j^T y - r_j) B a_j, \text{ so} \\ \tilde{y} &= y + \theta_j t_j B a_j. \end{aligned}$$

Hence we obtain $\tilde{t} = s(\tilde{y}) = t + A^T(\tilde{y} - y)$, or

$$\tilde{t} = t + \theta_j t_j A^T B a_j,$$

and, in particular,

$$\tilde{t}_j = (1 + \theta_j \gamma_j) t_j. \quad (23)$$

We are assuming that D is scaled so that $v^T D v - t^T D t = 1$; we wish to scale \tilde{D} (and \tilde{B}) similarly. Note that

$$v^T D v = v^T D v - d_j v_j^2.$$

Next

$$\begin{aligned} \tilde{t}^T D \tilde{t} &= (t + \theta_j t_j A^T B a_j)^T (D - d_j e_j e_j^T) (t + \theta_j t_j A^T B a_j) \\ &= t^T D t - d_j t_j^2 + \theta_j^2 t_j^2 \{a_j^T B A D A^T B a_j - d_j \gamma_j^2\} - 2 d_j \theta_j \gamma_j t_j^2 \\ &= t^T D t - t_j^2 \{d_j - \theta_j^2 \gamma_j + \theta_j^2 d_j \gamma_j^2 + 2 \theta_j d_j \gamma_j\} \\ &= t^T D t - \theta_j t_j^2. \end{aligned}$$

We therefore define

$$\begin{aligned} \tilde{\delta} &= v^T D v - \tilde{t}^T D \tilde{t} = 1 - d_j v_j^2 + \theta_j t_j^2, \\ \tilde{\sigma} &= -\theta_j \gamma_j. \end{aligned} \quad (24)$$

Note that $t_j > v_j$ and $\theta_j \geq d_j$, so that $\tilde{\delta} \geq 1$. Then

$$\begin{aligned} \tilde{D} &= \tilde{\delta}^{-1} (D - d_j e_j e_j^T); \\ \tilde{B} &= \tilde{\delta} \left\{ B - \tilde{\sigma} \frac{B a_j a_j^T B}{a_j^T B a_j} \right\}; \\ \tilde{y} &= y - \frac{\tilde{\sigma} t_j}{\gamma_j} B a_j; \end{aligned} \quad (25)$$

these quantities define the new ellipsoid \tilde{E} . The next result shows that obtaining a minimum volume ellipsoid containing part of \tilde{E} , instead of part of E , entails no loss in volume reduction.

Proposition 4.2

$$\left\{ x \in \tilde{E}: l_j \leq a_j^T x \leq u_j \right\} \subset \left\{ x \in E: l_j \leq a_j^T x \leq u_j \right\}$$

Proof Suppose x lies in the left hand set. Then

$$(A^T x - l)^T D (A^T x - u) \leq 0.$$

Also we clearly have

$$(a_j^T x - l_j) d_j (a_j^T x - u_j) \leq 0.$$

Adding gives $(A^T x - l)^T D (A^T x - u) \leq 0$, so that x lies in E . ■

We remark that if $d_j = 0$, the analysis of this section remains valid and yields the following results: $\tilde{\delta} = 1$, $\tilde{\sigma} = 0$, $\tilde{D} = D$, $\tilde{B} = B$, $\tilde{y} = y$, and $\tilde{t} = t$.

4.2 Adding a_j

We now use \tilde{E} to generate a lower bound \tilde{l}_j on $a_j^T x$ for $x \in X$. Set

$$\tilde{r}_j = a_j^T \tilde{B} a_j = \tilde{\delta}(1 - \tilde{\sigma}) r_j. \quad (26)$$

Then we know that $\tilde{l}_j \geq a_j^T \tilde{z}$, where $\tilde{z} = \tilde{y} - \frac{\tilde{r}_j}{\tilde{r}_j^T \tilde{B} a_j} \tilde{B} a_j$ is the minimizer of $a_j^T x$ over \tilde{E} . Moreover, by proposition 4.2, if $a_j^T z \geq l_j$ then $\tilde{z} \in E$ and

$$\tilde{l}_j \geq a_j^T \tilde{z} \geq a_j^T z \geq l_j.$$

Thus we have an improved lower bound. We therefore update l_j and correspondingly r_j and v_j . It is important to notice that \tilde{B} , \tilde{D} , and \tilde{y} are unchanged, since $\tilde{d}_j = 0$. We next update \tilde{t}_j . If $l_j > u_j$ then X is empty, and as discussed in section 2, we have a short proof of infeasibility. So assume $u_j \geq l_j$, so that $v_j \geq 0$.

Since $a_j^T y > u_j$, $t_j \geq 0$, and then $\tilde{\sigma} \leq 0$ implies $a_j^T \tilde{y} \geq a_j^T y > u_j$. Thus the

center of the new ellipsoid still violates the j^{th} constraint. Also, $l_j \geq a_j^T \tilde{z}$, so $\tilde{\gamma}_j^{\frac{1}{2}} \geq \tilde{t}_j + v_j \geq \tilde{t}_j - v_j > 0$. Hence $\tilde{\gamma}_j \geq (\tilde{t}_j + v_j)^2$, $\tilde{\gamma}_j \geq (\tilde{t}_j - v_j)^2$, and so $\tilde{\gamma}_j \geq \tilde{t}_j^2 + v_j^2$.

Now we obtain the smallest ellipsoid containing

$\left\{ x \in \tilde{E}: l_j \leq a_j^T x \leq u_j \right\}$. We use the formulae in [10] and some algebraic manipulation. Define

$$\begin{aligned} \eta &= \tilde{\gamma}_j - \tilde{t}_j^2 - v_j^2 \geq 0, \\ \xi &= \left\{ \eta^2 + 4(n^2 - 1)\tilde{t}_j^2 v_j^2 \right\}^{\frac{1}{2}} \geq 0. \end{aligned} \quad (27)$$

(In terms of the quantities defined in [10], $\eta = \tilde{\gamma}_j \frac{(2 - \alpha^2 - \beta^2)}{2}$ and $\xi = \tilde{\gamma}_j \rho/2$.) Then set

$$\begin{aligned} \hat{\sigma} &= 1 - \frac{2(n-1)v_j^2}{\xi + \eta} \text{ and} \\ \hat{\delta} &= n \frac{(n\eta + \xi)}{(n^2 - 1)\tilde{\gamma}_j} \end{aligned} \quad (28)$$

Finally, we obtain

$$\begin{aligned} \bar{D} &= \hat{\delta}^{-1} \left\{ \tilde{D} + \frac{\hat{\sigma}}{\tilde{\gamma}_j(1 - \hat{\sigma})} e_j e_j^T \right\}; \\ \bar{B} &= \hat{\delta} \left\{ \tilde{B} - \frac{\hat{\sigma} \tilde{B} a_j a_j^T \tilde{B}}{a_j^T \tilde{B} a_j} \right\}; \\ \bar{y} &= \tilde{y} - \frac{\hat{\sigma} \tilde{t}_j}{\tilde{\gamma}_j} \tilde{B} a_j. \end{aligned} \quad (29)$$

These define the new ellipsoid \bar{E} . The volume of \bar{E} is at most $\exp(-1/2(n+1))$ times that of E . The scaling factor $\hat{\delta}$ ensures that the inequality defining \bar{E} has a right hand side of one, i.e., that

$$\bar{E} = \left\{ x: (x - \bar{y})^T \bar{B}^{-1} (x - \bar{y}) \leq 1 \right\}.$$

4.3 Combining The Updates

It is unnecessary to form the intermediate quantities \tilde{D} , \tilde{B} , \tilde{y} , and \tilde{t} ; we can obtain \bar{D} , \bar{B} , and \bar{y} directly from D , B , y , and t . Note first that

$$\tilde{B}a_j = \tilde{\delta}(1 - \tilde{\sigma})Ba_j \text{ and so } \tilde{y}_j = \tilde{\delta}(1 - \tilde{\sigma})y_j.$$

Also,

$$\tilde{t} = t - \frac{\tilde{\sigma}t_j}{y_j} A^T Ba_j.$$

Thus the vector λ required for the lower bound on $a^T x$ for $x \in \tilde{E}$ is

$$\begin{aligned} \lambda &= \tilde{y}_j^{\frac{1}{2}} \tilde{D} \left\{ A^T \tilde{z} - r \right\} \\ &= \tilde{y}_j^{\frac{1}{2}} \tilde{D} \left\{ A^T \tilde{y} - r - \tilde{y}_j^{-\frac{1}{2}} A^T \tilde{B}a_j \right\} \\ &= \tilde{y}_j^{\frac{1}{2}} \tilde{D} t - \tilde{D} A^T \tilde{B}a_j \\ &= \tilde{y}_j^{\frac{1}{2}} \tilde{D} t - \tilde{D} A^T Ba_j \left\{ \tilde{\delta}(1 - \tilde{\sigma}) + \frac{\tilde{y}_j^{\frac{1}{2}} \tilde{\sigma} t_j}{y_j} \right\}, \text{ so} \\ \lambda &= \left\{ D - d_j e_j e_j^T \right\} \left\{ \frac{t y_j^{\frac{1}{2}} (1 - \tilde{\sigma})^{\frac{1}{2}}}{\tilde{\delta}_j^{\frac{1}{2}}} - A^T Ba_j \left[1 - \tilde{\sigma} + \frac{t_j \tilde{\sigma} (1 - \tilde{\sigma})^{\frac{1}{2}}}{\tilde{\delta}_j^{\frac{1}{2}} y_j^{\frac{1}{2}}} \right] \right\}. \end{aligned} \quad (30)$$

Hence we calculate the scalars $\tilde{\sigma}$ and $\tilde{\delta}$ and then compute λ from the original B , D , and t .

From λ we obtain the new l_j and hence r_j , v_j , and \tilde{t}_j . Thus we can calculate η , ξ , $\hat{\sigma}$ and $\hat{\delta}$. Now note that

$$\bar{y} = y - \left\{ \frac{\tilde{\sigma} t_j + \tilde{\sigma} t_j}{y_j} \right\} Ba_j. \quad (31)$$

Now set

$$\bar{\delta} = \hat{\delta}\hat{\delta}$$

and

$$\bar{\sigma} = \tilde{\sigma} + \hat{\sigma} - \hat{\sigma}\tilde{\sigma},$$

(so that $(1 - \bar{\sigma}) = (1 - \tilde{\sigma})(1 - \hat{\sigma})$). Then

$$\begin{aligned}\bar{D} &= \hat{\delta}^{-1} \left\{ \tilde{D} + \frac{\hat{\sigma}}{\tilde{\gamma}_j(1 - \hat{\sigma})} e_j e_j^T \right\} \\ &= \bar{\delta}^{-1} \left\{ D - d_j e_j e_j^T + \frac{\hat{\sigma}\tilde{\sigma}}{\gamma_j \tilde{\delta}(1 - \tilde{\sigma})(1 - \hat{\sigma})} e_j e_j^T \right\} \\ &= \bar{\delta}^{-1} \left\{ D + \frac{\hat{\sigma} - d_j \gamma_j (1 - \tilde{\sigma})(1 - \hat{\sigma})}{\gamma_j (1 - \hat{\sigma})} e_j e_j^T \right\}.\end{aligned}$$

But $-d_j \gamma_j (1 - \tilde{\sigma}) = \tilde{\sigma}$, so the numerator above is $\hat{\sigma} + (1 - \hat{\sigma})\tilde{\sigma} = \bar{\sigma}$ and

$$\bar{D} = \bar{\delta}^{-1} \left\{ D + \frac{\bar{\sigma}}{\gamma_j (1 - \bar{\sigma})} e_j e_j^T \right\}. \quad (33)$$

Finally,

$$\begin{aligned}\bar{B} &= \hat{\delta} \left\{ \tilde{B} - \frac{\tilde{B} a_j a_j^T \tilde{B}}{a_j^T \tilde{B} a_j} \right\} \\ &= \hat{\delta} \left\{ \tilde{\delta} \left\{ B - \frac{\tilde{B} a_j a_j^T \tilde{B}}{a_j^T \tilde{B} a_j} \right\} - \hat{\sigma} \tilde{\delta} (1 - \tilde{\sigma}) \frac{B a_j a_j^T B}{a_j^T B a_j} \right\}.\end{aligned}$$

Therefore

$$\bar{B} = \bar{\delta} \left\{ B - \frac{B a_j a_j^T B}{a_j^T B a_j} \right\}. \quad (34)$$

5. The Algorithm

Below we summarize the method that can be assembled from the ingredients presented in earlier sections. We suppose X is given by (9).

Initialization Let $r = (u + 1)/2$, $v = (u - 1)/2$.

The initial matrices are given by (14).

Define Λ as in (16).

Calculate $\gamma_i = a_i^T B a_i$ for each i .

Iteration Calculate $t = A^T y - r$.

If $t \leq v$, stop with feasible solution y .

Otherwise, choose j with $t_j > v_j$.

(A plausible choice is to maximize $(t_i - v_i)^2/\gamma_i$
over i for which $t_i > v_i$.)

Compute $w = B a_j$ and recalculate $\gamma_j = a_j^T B a_j$.

Set $q = A^T w$.

Compute θ_j from (22), and hence \tilde{t}_j from (23) and
 $\tilde{\delta}$ and $\tilde{\sigma}$ from (24).

Calculate λ from (30), using $q = A^T w$.

If $l^T \lambda_- - u^T \lambda_+ > l_j$, compute $\bar{\lambda}_j$ from (20), replace
the j^{th} column of Λ by $\bar{\lambda}_j$, update l_j to $-u^T \bar{\lambda}_j$
and correspondingly update r_j , v_j , and \tilde{t}_j .

If $l_j > u_j$, stop with $\mu = \bar{\lambda}_j + e_j$ indicating
infeasibility.

Calculate $\tilde{\gamma}_j$ from (26) and hence η , ξ , $\hat{\sigma}$, and $\hat{\delta}$
from (27)-(28).

Obtain $\bar{\delta}$ and $\bar{\sigma}$ from (32).

Compute the updated quantities \bar{y} , \bar{D} , and \bar{B} from

(31), (33) and (34).

Update γ_i to $\bar{\gamma}_i = \bar{\delta}(\gamma_i - \bar{\sigma}q_i^2/\gamma_j)$ for all i .

Remove overbars from update quantities and repeat.

It is advisable for numerical reasons to update a factorization of B rather than B itself. Suppose

$$B = L\Delta L^T \quad (35)$$

where L is unit lower triangular and Δ is diagonal with positive diagonal entries. Then, even if L and Δ are contaminated by round-off errors, the resulting B is positive definite. We compute first $w' = L^T a_j$, thence $\gamma_j = \sum g_{ij} (w'_i)^2 > 0$, and then $w = L\Delta w'$. The update formula (34) for B is exactly as in earlier versions of the ellipsoid method, and the factorization (35) can be updated efficiently and in a numerically stable way as in, e.g., [5], based on the method of Gill, Murray, and Saunders [4]. Note that, since $B^{-1} = \Delta A^T$, we may recompute the factorization (35) from the current D and the original data. Thus we use the method of [4] to compute the factorization

$$AD^{1/2} = \bar{L}\bar{\Delta}\bar{V}$$

with \bar{L} unit lower triangular, $\bar{\Delta}$ diagonal with positive diagonal entries, and $\bar{\Delta}^{1/2}\bar{V}$ orthogonal. Then $L = \bar{L}^{-1}$ and $\Delta = \bar{\Delta}^{-1}$ give (35).

It is also worth remarking that the improved lower bound can lead to values of $\bar{\sigma}$ that are close to 1. It is therefore important to use the formulae

$$1 - \tilde{\sigma} = \frac{1}{1 - d_j \gamma_j},$$

$$1 - \hat{\sigma} = \frac{2(n-1)v_j^2}{\xi + \eta}, \quad \text{and}$$

$$1 - \bar{\sigma} = (1 - \tilde{\sigma})(1 - \hat{\sigma})$$

to avoid excessive cancellation when calculating $1 - \bar{\sigma}$, which is then employed in the updates of L and Δ and of D .

6. Discussion

Suppose that we wish to solve the linear programming problem

$$\min \{ a_0^T x, x \in X \} \quad (36)$$

with X given by (9). We are still assuming that X either is empty or has full dimension. We start by applying the algorithm of section 5 to find a point in X . If one exists, we find that the center y of the current ellipsoid is feasible after a finite number of steps.

Now we compute $w = Ba_0$ and obtain a lower bound l_0 on $a_0^T x$ for all $x \in X$ as in section 3. Moreover, we produce a dual solution $\bar{\lambda}_0$ establishing this bound. To obtain an upper bound u_0 on $a_0^T x$, we find $\bar{v} = \max\{v : (y - vw) \in X\}$ and set $\bar{x} = y - \bar{v}w$ and $u_0 = a_0^T \bar{x}$. Next we continue the algorithm of section 5 to find a point in

$$\bar{X} = \{x : \bar{l} \leq \bar{A}^T x \leq \bar{u}\},$$

where $\bar{A} = [a_0, A]$, $\bar{l}^T = (l_0, l^T)$, and $\bar{u}^T = (u_0, u^T)$. We start with the current ellipsoid; its center y violates the 0th inequality. We continue similarly; each time a feasible solution is generated, we improve u_0 and possibly l_0 . There are, however, two new possibilities that we must discuss briefly.

First, the lower bounds that are obtained are functions of the upper bounds \bar{u} . Now that u_0 is varying during the algorithm, we keep the best feasible vector \bar{x} found so far, and $u_0 = a_0^T \bar{x}$. It could be, however, that the lower bound l_i was derived from a previous, weaker upper bound on $a_0^T x$. Thus we have $-u^T \lambda_i \geq l_i$, where λ_i is the dual vector certifying l_i , and strict inequality is possible if the 0th component of λ_i is positive and u_0 has been improved since λ_i was generated. (We could update all such l_i 's when u_0 is decreased, but this is more work, since the ellipsoid changes.)

Second, when we obtain the subsequent feasible solutions \bar{x} , our current ellipsoid already involves the 0th constraint. Proposition 4.1, which assures

that $d_i \gamma_i < 1$ when $t_i > v_i$, no longer implies that we can drop the 0th constraint and proceed with the update in section 4. Suppose therefore that $d_0 \gamma_0 = 1$. The proof of proposition 4.1 implies that $t_0 = 0$ and $d_i a_i^T B a_0 = 0$ for $i > 0$. Since y is feasible, $-v_i \leq t_i \leq v_i$ for all $i > 0$, so that $1 = \sum (d_i v_i^2 - d_i t_i^2)$ implies that $d_0 v_0^2 \leq 1$ and so $v_0 \leq \gamma_0^{1/2}$. Thus the hyperplanes $a_0^T x = l_0$ and $a_0^T x = u_0$ both intersect the current ellipsoid E .

Let us examine the effect on E of replacing u_0 by $\bar{u}_0 = u_0 - 2\rho$, an improved upper bound obtained by searching in the direction $B a_0$. Then r_0 becomes $\bar{r}_0 = r_0 - \rho$. Moreover, since $d_i a_i^T B a_0 = 0$ for all $i > 0$, $y = B A D r$ becomes $\bar{y} = y - d_0 \rho B a_0$ and $\bar{t} = s(\bar{y}) = t$. Thus we need only scale B and D to obtain the new representation. Since $\bar{v}_0 = v_0 - \rho$, the right hand side of the quadratic inequality has become $\bar{\Psi} = 1 - 2d_0 \rho v_0 + d_0 \rho^2 < 1$. We therefore replace D by $\bar{D} = D/\bar{\Psi}$ and B by $\bar{B} = \bar{\Psi} B$ to obtain the new representation. Note that the new center \bar{y} violates the constraint that defined the new feasible point \bar{x} with $a_0^T \bar{x} = \bar{u}_0$. We thus continue the iterations.

The occurrence of this case with $d_0 \gamma_0 = 1$ is in fact impossible with the version of the algorithm in section 5, which is initialized from the bounds $l_x \leq x \leq u_x$. Indeed, the positive semi-definite matrix $A D A^T$, in the notation of proposition 4.1, then contains $\sum d_i e_i e_i^T$, with $d_i > 0$ for $i = 1, \dots, n$, and thus is positive definite. We have included the remedy above to deal with the general case of some $E(D, l)$, derived in an arbitrary way. As a final remark, it can be shown that the modification above decreases the volume of E since $\bar{\Psi} < 1$.

If one wishes to solve the linear programming problem in canonical form:

$$\begin{aligned} \min \quad & u^T \lambda \\ & A \lambda = -a_0 \\ & \lambda \geq 0, \end{aligned} \tag{37}$$

with A $m \times n$ and $n > 2m$, then it is preferable to attack its dual (36) by the ellipsoid method. At any iteration one may obtain a feasible solution to (37) by calculating a corresponding dual vector $\bar{\lambda}_0$ for the linear function $a_0^T x$ as in section 3. Previous versions of the ellipsoid algorithm have not allowed this possibility. If the dual linear programming problem was solved, it was necessary to assume all data integer, work in extended precision arithmetic, and take a huge number of iterations in order to get, by rounding, an exact dual optimal solution; then the primal optimal solution could be obtained by complementary slackness.

Appendix

A relaxation of the linear programming problem

$$\begin{aligned} \min a_j^T x \\ A^T x &\geq 1 \\ -A^T x &\geq -u \end{aligned} \quad (A.1)$$

is the convex programming problem

$$\begin{aligned} \min a_j^T x \\ (x - y)^T B^{-1} (x - y) \leq 1 \end{aligned} \quad (A.2)$$

whose feasible region is the current ellipsoid E . Since the Slater condition holds for (A.2), the Karush-Kuhn-Tucker conditions below are necessary and sufficient for z to solve (A.2):

$$\begin{aligned} a_j + 2\pi B^{-1}(z - y) &= 0 \\ (z - y)^T B^{-1}(z - y) &\leq 1, \pi \geq 0, \\ \pi[(z - y)^T B^{-1}(z - y) - 1] &= 0. \end{aligned} \quad (A.3)$$

Since $a_j \neq 0$, we must have $\pi \neq 0$ and $z = y - Ba_j/2\pi$. Hence z satisfies the constraint of (A.2) with equality, and we find $\pi = \gamma_j^{1/2}/2$ and $z = y - \gamma_j^{-1/2} Ba_j$, where $\gamma_j = a_j^T Ba_j$.

Now $B^{-1} = ADA^T$ and $y = BADr$. Thus the first equation of (A.3) yields

$$\begin{aligned} a_j + A[\gamma_j^{1/2} D(A^T z - r)] &= 0, \\ \text{or} \quad A\lambda &= -a_j, \text{ where } \lambda = \gamma_j^{1/2} D(A^T z - r). \end{aligned} \quad (A.4)$$

Since the vector λ is obtained from the multiplier π for (A.2), it can be viewed as a disaggregated multiplier. Indeed, λ demonstrates that z also solves

$$\min a_j^T x$$

$$a_i^T x \leq a_i^T z \text{ if } d_i > 0 \text{ and } a_i^T z > r_i$$

$$a_i^T x \geq a_i^T z \text{ if } d_i > 0 \text{ and } a_i^T z < r_i$$

The similarity of this problem to (A.1) suggests that λ given in (A.4) is an excellent candidate to generate a better lower bound than $a_j^T z$.

References

- [1] Agmon, S. 1954. The Relaxation Method for Linear Inequalities. Can. J. Math. 6, 382-392.
- [2] Bland, R., D. Goldfarb and M. J. Todd. 1981. The Ellipsoid Method: A Survey. Opns. Res. 29, 1039-1091.
- [3] Gacs, P., and L. Lovasz. 1981. Khachian's Algorithm for Linear Programming. Math Program. Stud. 14, 61-68.
- [4] Gill, P. E., W. Murray and M. A. Saunders. 1975. Methods for Computing and Modifying the LDV Factors of a Matrix. Math. Comput. 29, 1051-1077.
- [5] Goldfarb, D., and M. J. Todd. 1982. Modifications and Implementation of the Ellipsoid Algorithm for Linear Programming. Math. Program. 23, 1-19
- [6] Grotschel, M., L. Lovasz and A. Schrijver. 1981. The Ellipsoid Method and Its Consequences in Combinatorial Optimization. Combinatorica 1, 169-197.
- [7] Khachian, L. G. 1980. A Polynomial Algorithm in Linear Programming. Doklady Akademii Nauk SSSR 244,1093-1096 (translated in Soviet Mathematics Doklady 20, 191-194, 1979).
- [8] Levin, L., and B. Yamitsky. 1982. An Old Linear Programming Algorithm Runs in Polynomial Time. 23rd Symp. on Found. of Comp. Sci., I. E. E. E. 327-328.
- [9] Motzkin, T., and I. J. Schoenberg. 1954. The Relaxation Method for Linear Inequalities. Can. J. Math. 6, 393-404.
- [10] Todd, M. J. 1980. Minimum Volume Ellipsoid Containing Part of a Given Ellipsoid. Math. Oper. Res. 7, 253-261.
- [11] Wolsey, L. Private communication.
- [12] Yudin, D. B., and A. S. Nemirovsky. 1976. Informational Complexity and Effective Methods of Solution for Convex Extremal Problems. Ekonomika i Matematicheskie Metody 12, 357-369 (translated in Math. Economics 13, 25-45, Spring '77).