



## Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### The Ellipsoid Method Generates Dual Variables

Bruce P. Burrell, Michael J. Todd,

To cite this article:

Bruce P. Burrell, Michael J. Todd, (1985) The Ellipsoid Method Generates Dual Variables. *Mathematics of Operations Research* 10(4):688-700. <http://dx.doi.org/10.1287/moor.10.4.688>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

© 1985 INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

## THE ELLIPSOID METHOD GENERATES DUAL VARIABLES\*

BRUCE P. BURRELL AND MICHAEL J. TODD†

*Cornell University*

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

$$X = \{x \in R^n : A^T x < u\} \quad (1)$$

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

$$E_k = \{x : (x - y_k)^T B_k^{-1} (x - y_k) < 1\}. \quad (2)$$

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  $\{x \in E_k : a_j^T x < a_j^T y_k\}$ ;  $E_{k+1}$  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  $\tau = 1/(n+1)$ ,  $\delta = n^2/(n^2-1)$  and  $\sigma = 2/(n+1)$ .

\*Received April 12, 1984.

AMS 1980 subject classification. Primary: 65K05. Secondary: 90C05.

IAOR 1973 subject classification. Main: Optimization. Cross references: Numerical analysis.

OR/MS Index 1978 subject classification. Primary: 644 Programming/linear/algorithms/ellipsoidal.

Key words. Ellipsoid method, Khachian method, linear programming, linear inequalities.

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

Many more sophisticated strategies have been proposed, which generally lead to a similar update with modified parameters  $\tau$ ,  $\delta$ , and  $\sigma$ . Even the simplest version above, however, guarantees that  $\text{vol}(E_{k+1})/\text{vol}(E_k) < \exp(-1/2(n+1))$  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  $\delta$  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  $\Delta_k$  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  $\mu$  with

$$A\mu = 0, \quad \mu > 0, \quad \text{and} \quad u^T \mu < 0, \tag{5}$$

clearly demonstrating that  $X$  is empty. The Farkas lemma guarantees that such a  $\mu$  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  $\lambda_j$  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

$$S_k = \{x : B'_k A^T x < B'_k u\} \tag{6}$$

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  $\log \text{vol} S_{k+1} - \log \text{vol} S_k < O(n^{-2})$ , compared to  $O(n^{-1})$  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

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

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

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

thus we have an approximate nonnegative solution to  $A\mu = 0$ . The accuracy depends on the distance  $\|y_{l+p} - y_l\|$ , the size of  $p$ , and lower bounds on the  $\lambda_k$ 's; one can only achieve good approximate dual solutions  $\mu$  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  $X$ . §3 discusses the generation of lower bounds via dual solutions, while updating the ellipsoid is addressed in §4. §5 gives the resulting algorithm, and in §6 we provide some concluding discussion. The Appendix provides justification for our choice of  $\lambda$ .

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  $\tilde{A}^T x < \tilde{b}$ ,  $x < u_x$ ,  $-x < -l_x$ . 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  $X = \{x \in R^n : A^T x < u\}$ .

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

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

We can alternately define  $X$  by quadratic inequalities:

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

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

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

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

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

$$r = \frac{u+l}{2}, \quad B = (ADA^T)^{-1}, \quad \text{and} \quad y = BAdr. \quad (11)$$

Then the inequality defining  $E$  can be rewritten as  $x^T ADA^T x - 2r^T DA^T x + l^T Du < 0$ , or

$$(x - y)^T B^{-1}(x - y) < y^T B^{-1}y - l^T Du. \tag{12}$$

Thus  $y$  is the center of the ellipsoid  $E$ .

There is another way of looking at the problem that is useful. Define  $v = (u - l)/2$  so that  $u = r + v$ ,  $l = r - v$ . Then, defining  $s = s(x) = A^T x - r$ , we see that  $X = \{x \in R^n : -v < s < v\}$ .

Again, with  $D$  a nonnegative diagonal matrix we find  $X \subset E$ , where  $E = \{x \in R^n : s^T D s < v^T D v\}$ . The left side is minimized by solving the least squares problem of minimizing  $\|D^{1/2}(A^T x - r)\|_2$ ; 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  $t^T D t + (x - y)^T B^{-1}(x - y) < v^T D v$  (since  $t^T DA^T(x - y) = 0$ ), or  $(x - y)^T B^{-1}(x - y) < v^T D v - t^T D t$ . This inequality and (12) are identical.

Consider the right-hand side of these inequalities,  $\Psi = y^T B^{-1}y - l^T Du = v^T D v - t^T D t$ . If  $\Psi$  is negative then  $E$  is empty, and hence so is  $X$ . If  $\Psi$  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  $E$ ,

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

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

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

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

$$B = \text{diag}(nv_1^2, \dots, nv_n^2).$$

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  $\Lambda$  of "dual variables" establishing that  $l$  is indeed a valid lower bound. We will have

$$A\Lambda = -A, \quad \Lambda > 0, \quad \text{and} \quad u^T \Lambda = -l^T, \tag{15}$$

so that  $A^T x < u$  implies  $A^T x = -\Lambda^T A^T x > -\Lambda^T u = l$  as desired. Note that  $\lambda_i$ , the  $i$ th column of  $\Lambda$ , satisfies  $A^T \lambda_i = -a_i$ , the  $i$ th column of  $A$ , while  $u^T \lambda_i = -l_i$ ; thus  $\lambda_i$  is a vector of dual variables certifying the lower bound  $l_i$ . Initially, (15) holds with

$$\Lambda^T = (A^T, A_+^T, 0). \tag{16}$$

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

$$y^T B^{-1}y - l^T Du = v^T D v - t^T D t = 1. \tag{17}$$

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  $\lambda_j$  of

Downloaded from informs.org by [132.236.181.20] on 26 March 2014, at 14:28 . For personal use only, all rights reserved.

dual variables as above. In §3 we show how to obtain a vector  $\bar{\lambda}_j > 0$  from the current ellipsoid with  $A\bar{\lambda}_j = -a_j$  and  $\bar{l}_j = -u^T\bar{\lambda}_j > a_j^Tz$ ; 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  $\Lambda$  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  $\mu > 0$ ,  $A\mu = 0$ , and  $u^T\mu < 0$ ; thus  $\mu$  provides a short proof of infeasibility.

After a possible update of  $l_j$ , and assuming that we have not demonstrated infeasibility, we have  $u_j > l_j$ ,  $a_j^Ty > u_j$ , and the hyperplanes  $a_j^Tx = l_j$  and  $a_j^Tx = u_j$  both intersect the current ellipsoid  $E$ . In such a case, Todd [10] has shown how to obtain the minimum volume ellipsoid  $E'$  with  $E' \supset \{x \in E : l_j < a_j^Tx < u_j\}$ . Moreover, the proof in [10] shows that the quadratic inequality defining  $E'$  is a convex combination of that defining  $E$  and  $(a_j^Tx - l_j)(a_j^Tx - u_j) < 0$ . 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  $(a_j^Tx - \hat{l}_j)(a_j^Tx - u_j)$ , 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 §4, obtaining a new ellipsoid  $\bar{E} = E(\bar{D}, l)$  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  $E(D, l)$  for some nonnegative diagonal  $D$ , and decreasing in volume by a factor of at worst  $\exp(-\frac{1}{2}(n+1))$  at each step. The algorithm is summarized in §5. Some concluding remarks are made in §6.

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

Consider the linear programming problem

$$\min a_j^Tx, \quad A^Tx > l, \quad -A^Tx > -u. \quad (18)$$

Its dual is

$$\max l^T\lambda' - u^T\lambda'', \quad A\lambda' - A\lambda'' = a_j, \quad \lambda', \lambda'' > 0.$$

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

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

$$\lambda = \gamma_j^{1/2}D(A^Tz - r) = \gamma_j^{1/2}Ds(z) \quad (19)$$

suffices. (This choice of  $\lambda$  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  $\lambda$  as in (19),  $A\lambda = -a_j$ .*

**PROOF.** We have

$$\begin{aligned} A\lambda &= \gamma_j^{1/2}AD(A^Ty - r - A^T(y - z)) \\ &= -\gamma_j^{1/2}ADA^T(y - z) \\ &= -a_j, \end{aligned}$$

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

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

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

for all  $i$ , and note that if  $\lambda_i \neq 0$  (so  $d_i \neq 0$  and  $a_i z \neq r_i$ ), 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  $\lambda_i$  and  $a_i^T z - r_i$  have the same sign and

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

Similarly if  $\lambda_i < 0$ , then

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

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

$$(a_i^T z - l_i)(a_i^T z - u_i) < 0 \quad \text{and} \quad \mu_i = -\frac{\gamma_j^{1/2}}{2} d_i (a_i^T z - l_i)(a_i^T z - u_i) > 0.$$

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

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

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

**THEOREM 3.2.** *With  $\lambda$  as in (19),  $f(\lambda) > a_j^T z$ . ■*

The vector  $\lambda$  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  $\lambda > 0$ , i.e.,  $\bar{l}_j$  may use some previously established lower bounds. Let us therefore define

$$\bar{\lambda}_j = \Lambda \lambda_- + \lambda_+ > 0. \tag{20}$$

Then  $A\bar{\lambda}_j = A\Lambda\lambda_- + A\lambda_+ = A(\lambda_+ - \lambda_-) = A\lambda = -a_j$ , and  $-u^T \bar{\lambda}_j = -u^T \Lambda \lambda_- - u^T \lambda_+ = l^T \lambda_- - u^T \lambda_+ = f(\lambda)$ . Thus  $\bar{\lambda}_j$  has precisely the properties we desire. (Note that we have used  $\lambda_i$  above for the  $i$ th component of  $\lambda$ , 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  $E(D, l)$  and  $B, r, v, y$ , and  $t$  as in §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 §3 to derive a new lower bound at least as great as  $a_j^T z$ . As pointed out in §2, however, if  $d_j > 0$  the current ellipsoid is defined in part by the older 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  $\check{D} = D - d_i e_i e_i^T$  and  $\gamma_i = a_i^T B a_i$ . We have  $d_i \gamma_i < 1$ , with equality only if  $t_i = 0$ . If  $t_i \neq 0$ , then  $d_i \gamma_i < 1$  and  $A \check{D} A^T$  is positive definite.*

**PROOF.** Note that in any case  $A \check{D} A^T$  is positive semidefinite. Since  $ADA^T = A \check{D} A^T + d_i a_i a_i^T$ , multiplying on the left and right by  $B a_i$  yields

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

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

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

$$\check{D}^{1/2} A^T B a_i = 0; \quad (21)$$

thus  $a_i^T y = a_i^T B A D r = a_i^T B (A \check{D} r + d_i r_i a_i) = d_i \gamma_i r_i = r_i$ ; 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

$$\det(A \check{D} A^T) = (1 - d_i \gamma_i) \det(ADA^T)$$

is positive,  $A \check{D} A^T$  is nonsingular and hence positive definite. ■

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

**4.1. Removing  $a_j$ .** Here we obtain the updated quantities  $\check{B}$ ,  $\check{D}$ ,  $\check{y}$ , and  $\check{t}$  corresponding to changing  $D$  to  $\check{D} = D - d_j e_j e_j^T$ . Note that  $\check{D} \neq \check{D}$ , since we wish to preserve our scaling, i.e.  $v^T \check{D} v - \check{t}^T \check{D} \check{t} = 1$ . The resulting ellipsoid is denoted  $\check{E}$ . 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  $A \check{D} A^T = ADA^T - d_j a_j a_j^T$ , we find  $\check{B} = (A \check{D} A^T)^{-1} = B + \theta_j B a_j a_j^T B$ . Set  $\check{y} = \check{B} A \check{D} r$ . Then

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

Hence we obtain  $\check{t} = s(\check{y}) = t + A^T(\check{y} - y)$ , or  $\check{t} = t + \theta_j t_j A^T B a_j$ , and, in particular,

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



We are assuming that  $D$  is scaled so that  $v^T D v - t^T D t = 1$  as in (17); we wish to scale  $\tilde{D}$  (and  $\tilde{B}$ ) similarly. Note that  $v^T \tilde{D} v = v^T D v - d_j v_j^2$ . Next

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

We therefore define

$$\tilde{\delta} = v^T \tilde{D} v - \tilde{t}^T \tilde{D} \tilde{t} = 1 - d_j v_j^2 + \theta_j t_j^2, \quad \tilde{\sigma} = -\theta_j \gamma_j. \tag{24}$$

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

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

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.**

$$\{x \in \tilde{E} : l_j < a_j^T x < u_j\} \subset \{x \in E : l_j < a_j^T x < u_j\}.$$

**PROOF.** Suppose  $x$  lies in the left-hand set. Then  $(A^T x - l)^T \tilde{D} (A^T x - u) < 0$ . Also we clearly have  $(a_j^T x - l_j) d_j (a_j^T x - u_j) < 0$ . Adding gives  $(A^T x - l)^T D (A^T x - u) < 0$ , 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

$$\tilde{\gamma}_j = a_j^T \tilde{B} a_j = \tilde{\delta} (1 - \tilde{\sigma}) \gamma_j. \tag{26}$$

Then we know that  $\tilde{l}_j > a_j^T \tilde{z}$ , where  $\tilde{z} = \tilde{y} - \tilde{\gamma}_j^{-1/2} \tilde{B} a_j$  is the minimizer of  $a_j^T x$  over  $\tilde{E}$ . Moreover, by Proposition 4.2, if  $a_j^T z > l_j$  then  $\tilde{z} \in E$  and  $\tilde{l}_j > a_j^T \tilde{z} > a_j^T z > l_j$ . 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 §2, we have a short proof of infeasibility. So assume  $u_j > l_j$ , so that  $v_j > 0$ .

Since  $a_j^T y > u_j$ ,  $t_j > 0$ , and then  $\tilde{\sigma} < 0$  implies  $a_j^T \tilde{y} > a_j^T y > u_j$ . Thus the center of the new ellipsoid still violates the  $j$ th constraint. Also,  $l_j > a_j^T \tilde{z}$ , so  $\tilde{\gamma}_j^{1/2} > t_j + v_j > \tilde{t}_j - v_j > 0$ . Hence  $\tilde{\gamma}_j > (t_j + v_j)^2$ ,  $\tilde{\gamma}_j > (t_j - v_j)^2$ , and so  $\tilde{\gamma}_j > t_j^2 + v_j^2$ .

Now we obtain the smallest ellipsoid containing  $\{x \in \tilde{E} : l_j < a_j^T x < u_j\}$ . We use the formulae in [10] and some algebraic manipulation. Define

$$\eta = \tilde{\gamma}_j - \tilde{t}_j^2 - v_j^2 > 0, \quad \xi = \{ \eta^2 + 4(n^2 - 1) \tilde{t}_j^2 v_j^2 \}^{1/2} > 0. \tag{27}$$

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

$$\hat{\delta} = 1 - \frac{2(n-1)v_j^2}{\xi + \eta} \quad \text{and} \quad \hat{\delta} = n \frac{(n\eta + \xi)}{(n^2 - 1)\tilde{\gamma}_j}. \tag{28}$$

Finally, we obtain

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

These define the new ellipsoid  $\bar{E}$ . The volume of  $\bar{E}$  is at most  $\exp(-\frac{1}{2}(n+1))$  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  $\bar{E} = \{x : (x - \bar{y})^T \bar{B}^{-1} (x - \bar{y}) < 1\}$ .

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  $\tilde{B} a_j = \hat{\delta}(1-\hat{\sigma}) B a_j$  and so  $\tilde{\gamma}_j = \hat{\delta}(1-\hat{\sigma}) \gamma_j$ . Also,

$$\tilde{t} = t - \frac{\hat{\sigma} \tilde{t}_j}{\gamma_j} A^T B a_j.$$

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

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

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

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

$$\bar{y} = y - \left\{ \frac{\hat{\sigma} \tilde{t}_j + \hat{\sigma} \tilde{t}_j}{\gamma_j} \right\} B a_j. \quad (31)$$

Now set

$$\bar{\delta} = \hat{\delta} \hat{\delta} \quad \text{and} \quad \bar{\sigma} = \hat{\sigma} + \hat{\sigma} - \hat{\sigma} \hat{\sigma}, \quad (32)$$

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

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

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

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

Finally,

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

Therefore

$$B = \bar{\delta} \left\{ B - \bar{\sigma} \frac{Ba_j a_j^T B}{a_j^T Ba_j} \right\}. \tag{34}$$

**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  $r = (u + l)/2$ ,  $v = (u - l)/2$ . The initial matrices are given by (14). Define  $\Lambda$  as in (16). Calculate  $\gamma_i = a_i^T B a_i$  for each  $i$ .

**Iteration.** Calculate  $t = A^T y - r$ . If  $t < v$ , 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  $\theta_j$  from (22), and hence  $\tilde{t}_j$  from (23) and  $\tilde{\delta}$  and  $\tilde{\sigma}$  from (24).

Calculate  $\lambda$  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  $\Lambda$  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  $\eta$ ,  $\xi$ ,  $\hat{\sigma}$ , 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  $\gamma_i$  to  $\bar{\gamma}_i = \bar{\delta}(\gamma_i - \bar{\sigma} q_i^2 / \gamma_j)$  for all  $i$ .

Remove overbars from updated quantities and repeat.

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

$$B = L \Delta L^T \tag{35}$$

where  $L$  is unit lower triangular and  $\Delta$  is diagonal with positive diagonal entries. Then, even if  $L$  and  $\Delta$  are contaminated by round-off errors, the resulting  $B$  is positive definite. We compute first  $w' = L^T a_j$ , thence  $\gamma_j = \sum \Delta_{ii} (w'_i)^2 > 0$ , 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  $B^{-1} = A D A^T$ , 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  $A D^{1/2} = \bar{L} \bar{\Delta} \bar{V}$  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

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

to avoid excessive cancellation when calculating  $1 - \bar{\sigma}$ , which is then employed in the updates of  $L$  and  $\Delta$  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  $X$  is either empty or has full dimension. We start by applying the algorithm of §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 §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 §5 to find a point in  $\bar{X} = \{x : \bar{l} < \bar{A}^T x < \bar{u}\}$ , 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  $-u^T \lambda_i > l_i$ , where  $\lambda_i$  is the dual vector certifying  $l_i$ , and strict inequality is possible if the 0th component of  $\lambda_i$  is positive and  $u_0$  has been improved since  $\lambda_i$  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 §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,  $-v_i < t_i < v_i$  for all  $i > 0$ , so that  $1 = \sum (d_i v_i^2 - d_i t_i^2)$  implies that  $d_0 v_0^2 < 1$  and so  $v_0 < \gamma_0^{1/2}$ . 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  $Ba_0$ . 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 §5, which is initialized from the bounds  $l_x < x < u_x$ . Indeed, the positive semidefinite matrix  $A \bar{D} A^T$ , 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:

$$\min u^T \lambda, \quad A \lambda = -a_0, \quad \lambda > 0, \quad (37)$$

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 §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

$$\min a_j^T x, \quad A^T x > l, \quad -A^T x > -u, \quad (A.1)$$

is the convex programming problem

$$\min a_j^T x \quad (x - y)^T B^{-1}(x - y) < 1 \quad (A.2)$$

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):

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

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

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

where  $\lambda = \gamma_j^{1/2}D(A^T z - r)$ . Since the vector  $\lambda$  is obtained from the multiplier  $\pi$  for (A.2), it can be viewed as a disaggregated multiplier. Indeed,  $\lambda$  demonstrates that  $z$  also solves

$$\begin{aligned} \min a_j^T x \\ a_i^T x < a_i^T z \quad \text{if} \quad d_i > 0 \quad \text{and} \quad a_i^T z > r_i, \\ a_i^T x > a_i^T z \quad \text{if} \quad d_i > 0 \quad \text{and} \quad a_i^T z < r_i. \end{aligned}$$

The similarity of this problem to (A.1) suggests that  $\lambda$  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. *Canad. J. Math.* 6 382–392.  
 [2] Bland, R., Goldfarb, D. and Todd, M. J. (1981). The Ellipsoid Method: A Survey. *Oper. Res.* 29 1039–1091.  
 [3] Gacs, P., and Lovász, L. (1981). Khachian’s Algorithm for Linear Programming. *Math. Programming Stud.* 14 61–68.  
 [4] Gill, P. E., Murray, W. and Saunders, M. A. (1975). Methods for Computing and Modifying the *LDV* Factors of a Matrix. *Math. Comp.* 29 1051–1077.

Downloaded from informs.org by [132.236.181.20] on 26 March 2014, at 14:28 . For personal use only, all rights reserved.

- [5] Goldfarb, D., and Todd, M. J. (1982). Modifications and Implementation of the Ellipsoid Algorithm for Linear Programming. *Math. Programming* **23** 1–19.
- [6] Grötschel, M., Lovász, L. and Schrijver, A. (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. *Dokl. Akad. Nauk SSSR* **244** 1093–1096 (translated in *Soviet Math. Dokl.* **20** 191–194, 1979).
- [8] Levin, L. and Yamnitsky, B. (1982). An Old Linear Programming Algorithm Runs in Polynomial Time. *23rd Sympos. on Foundations of Computer Sci.*, IEEE, 327–328.
- [9] Motzkin, T. and Schoenberg, I. J. (1954). The Relaxation Method for Linear Inequalities. *Canad. 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 Nemirovsky, A. S. (1976). Informational Complexity and Effective Methods of Solution for Convex Extremal Problems. *Ekonom. i Mat. Metody* **12** 357–369 (translated in *Maketon: Transl. Russian and East European Math. Economics* **13** 25–45, Spring '77).

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

Copyright 1985, by INFORMS, all rights reserved. Copyright of Mathematics of Operations Research is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.