SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 933–960, November 1996

003

SOLVING LP PROBLEMS VIA WEIGHTED CENTERS*

AIPING LIAO† AND MICHAEL J. TODD‡

Abstract. The feasibility problem for a system of linear inequalities can be converted into an unconstrained optimization problem by using ideas from the ellipsoid method, which can be viewed as a very simple minimization technique for the resulting nonlinear function. This function is related to the volume of an ellipsoid containing all feasible solutions, which is parametrized by certain weights which we choose to minimize the function. The center of the resulting ellipsoid turns out to be a feasible solution to the inequalities. Using more sophisticated nonlinear minimization algorithms, we develop and investigate more efficient methods, which lead to two kinds of weighted centers for the feasible set. Using these centers, we develop new algorithms for solving linear programming problems.

Key words. weighted center, the ellipsoid method, Newton's method, linear programming

AMS subject classifications. 65K, 90C

1. Introduction and a history of centers. In this paper we will consider a linear programming problem of the form

$$\text{(LP)} \qquad \begin{aligned} &\min\ c^T x \\ &l \le A^T x \le u \end{aligned}$$

and the associated feasibility problem

$$\text{(FP)} \qquad l \le A^T x \le u,$$

where A is an $n \times m$ matrix with full rank. We assume that $m \ge n \ge 2$ and $l < u$. As is well known, any linear programming problem can be reduced to so-called inequality form:

$$\text{(ILP)} \qquad \begin{aligned} &\min\ c^T x \\ &\tilde{A}^T x \le b, \\ &x \ge 0. \end{aligned}$$

If the input data are all integers, (ILP) is equivalent to

$$\text{(ILPB)} \qquad \begin{aligned} &\min\ c^T x \\ &\tilde{A}^T x \le b, \\ &(l_x =)0 \le x \le u_x, \end{aligned}$$

where $u_x = 2^L e$ with L the input length of (ILP) and e the vector of ones, which can be easily adapted to the form (LP) (see, e.g., Burrell and Todd [3]). So, theoretically speaking, our form (LP) is not restrictive. On the other hand, since "many linear

* Received by the editors August 23, 1993; accepted for publication (in revised form) August 10, 1995.

† Advanced Computing Research Institute and Center for Applied Mathematics, Cornell University, Ithaca, NY 14853. The research of this author was partially supported by the Cornell Theory Center, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from the State of New York and its Corporate Research Institutes; and by the NSF, the AFOSR, and the ONR through grant DMS-8920550.

‡ School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. The research of this author was partially supported by the NSF, the AFOSR, and the ONR through grant DMS-8920550.

programming problems involve explicit upper bounds on individual variables [in the standard form]" (Chvátal [4])—in particular, the upper bounds may correspond to limits on resources—we thus often, in practice also, are confronted with linear programming problems in the form (ILPB), which as above can be converted to the form (LP). Thus (LP) is not very restrictive in practice either.

In §§2 and 3 we will discuss methods for solving the linear inequality system (FP) and introduce two weighted centers. Then §4 develops algorithms for solving (LP) via these two kinds of center. We also report preliminary encouraging numerical results. Section 5 presents a summary and conclusion.

One recent development in linear programming can be regarded as a study of various concepts of center for a polytope $P \subset R^n$. Such a center must be in the (relative) interior of P. Since 1984 when Karmarkar proposed his famous projective-scaling algorithm [9], interior-point methods have become the mainstream of research in linear programming. Basically speaking, in this field, there are two aspects under study. One is the appropriate definition of center, which provides the interior point; the other is the method of moving the center and the merit function for measuring the progress made by this movement.

The first kind of center of a polytope that was used in optimization is the center of gravity or centroid used by Levin [12] in his algorithm, the method of central sections, for minimizing a convex function f over a convex polytope P (see also Newman [18]). In that paper the centroid is used as the test point. If the current centroid, say x^k, satisfies the convergence criterion, stop with a satisfactory approximate solution; otherwise, cut the current polytope P^k into two pieces by a hyperplane through x^k whose normal is a subgradient of f at x^k, and replace P^k by that part containing all optimal solutions. The volume of the polytope under consideration is thus reduced, and this procedure can be repeated until a satisfactory centroid is obtained. This method is very concise and $1 - \exp(-1)$ is a guaranteed reduction of the volume of successive polytopes. The disadvantage is the difficulty of calculating the centroid.

Yudin and Nemirovskii [30] discuss the computational difficulties of Levin's method and propose a modified method of centered cross-sections, using ellipsoids instead of polyhedra. This modified method is computationally implementable. They also point out that this ellipsoid method is a special case of Shor's algorithm [21] with space dilation in the direction of the subgradient. Shor [22] independently developed the ellipsoid method. Later, in 1979, Khachiyan [10] showed that the ellipsoid method is a polynomial-time algorithm for solving linear programming problems.

We note that John's result [8] shows that for every convex polytope $P \subset R^n$, the minimum volume ellipsoid containing P exists and is unique. It is easy to show that the center of this minimum volume ellipsoid is in P; otherwise, by one step of the ellipsoid algorithm, a smaller ellipsoid containing P can be obtained. Thus, the center of this ellipsoid can be used as a center of P. However, the smallest ellipsoid containing P is hard to find in general, and so is its center.

Tarasov, Khachiyan, and Erlich [24] study the method of inscribed ellipsoids, using the center of the maximal inscribed ellipsoid for a polytope as a center of the polytope. In [11], Khachiyan and Todd discuss the problem of approximating the maximal inscribed ellipsoid and related problems. They also propose algorithms for finding these ellipsoids.

Karmarkar's projective-scaling algorithm [9] uses centers in a different way. At each iteration, the current interior point is mapped by a projective transformation which is chosen so that its image becomes a certain center (the *analytic* center, see

Sonnevend [23]) of the transformed feasible region. Then it is easy to move this transformed point in order to make a sufficient decrease in a certain potential function.

Renegar [19] also uses the analytical center to develop his algorithm for linear programming problems, which is polynomial-time bounded. The analytical center is easier to approximate compared with the centroid. Actually, in his algorithm, Renegar showed that an ε-analytical center is enough. Unlike the centroid, the analytical center is not analytically independent; it depends on the way in which the polytope P is represented. Renegar [19] makes use of this property to improve the convergence of his algorithm by adding some extra constraints.

In the paper [28] Vaidya introduces the *volumetric center* which is the center of the ellipsoid with largest volume among a certain set of ellipsoids that are contained in P and proposes an algorithm with a better global convergence rate and time complexity than the ellipsoid method.

The centers of P we propose here are the centers of the ellipsoids with the smallest volumes among certain sets of ellipsoids that contain P. These ellipsoids are of simpler structure and are relatively easier to construct.

2. Model I and the associated center. We consider the feasibility problem:

$$\text{find } \bar{x} \text{ such that } \bar{x} \in P := \{x : l \leq A^T x \leq u\}. \tag{1}$$

For convenience we denote $r := \frac{l+u}{2}$ and $s := \frac{u-l}{2}$. We also denote by e the all-one vector and by e_j the jth column of the identity matrix. In [3] Burrell and Todd proposed a parallel-cut ellipsoid algorithm based on the results of Todd [26].

Note that P can be alternately written as

$$P = \{x \in R^n : (a_i^T x - l_i)(a_i^T x - u_i) \leq 0, i = 1, \ldots, m\}, \tag{2}$$

where a_i is the ith column of A and l_i, u_i are the corresponding components of l, u, respectively. Now choose a nonnegative diagonal matrix $D := \text{diag}(d) := \text{diag}(d_1, \ldots, d_m)$, and combine the inequalities above with weights d_i. We thus obtain a set

$$E := E(d) := \{x \in R^n : (A^T x - l)^T D (A^T x - u) \leq 0\}. \tag{3}$$

It is obvious that $P \subset E$. We suppose that ADA^T is nonsingular. Then E is actually an ellipsoid. Further calculation shows that

$$E = \{x \in R^n : (x - x_c)^T ADA^T (x - x_c) \leq x_c^T (ADA^T) x_c - l^T D u\}, \tag{4}$$

where $x_c := x_c(d) := (ADA^T)^{-1} ADr$ is the center of E.

If the current center violates some constraint, say, $l_i \leq a_i^T x \leq u_i$, by the results of Todd [26] we can construct a new ellipsoid that contains that part of the previous one between the parallel hyperplanes $a_i^T x = l_i$ and $a_i^T x = u_i$, and the volume of the ellipsoid decreases by a factor which is, at worst, $\exp(-\frac{1}{2(n+1)})$. Only one component of d changes in this process.

The idea of this paper is to consider the volume of $E(d)$, or some surrogate of this volume, as a nonlinear function of d; then by applying an efficient algorithm (usually a variant of Newton's method) to minimize this function, we obtain a d^* for which we can show that $x_c(d^*)$ lies in P.

There is another way in which ellipsoids of a form very similar to $E(d)$ arise. Suppose that we model (FP) as the linear programming problem

$$\text{(FPP)}\qquad \begin{array}{lrcrcr} \max & & & \lambda & & \\ & A^T x & + & e\lambda & \leq & u, \\ & -A^T x & + & e\lambda & \leq & -l. \end{array}$$

The dual of this problem is

$$\text{(FPD)}\qquad \begin{array}{lrcrcl} \min & u^T y_u & - & l^T y_l & & \\ & A y_u & - & A y_l & = & 0, \\ & e^T y_u & + & e^T y_l & = & 1, \\ & y_u, & & y_l & \geq & 0, \end{array}$$

a problem in Karmarkar's canonical form. Todd [27] and Ye [29] show that at each iteration of Karmarkar's algorithm applied to (FPD), an ellipsoid is naturally generated that contains all optimal solutions of (FPP). This ellipsoid has the form

$$E'(d, w, \nu) := \{x : x^T ADA^T x - 2w^T A^T x + \nu \leq 0\}$$

for $D := \mathrm{diag}(d) := Y_u^2 + Y_l^2$, where (y_u, y_l) is the current feasible iterate for (FPD) and $Y_l := \mathrm{diag}(y_l)$, $Y_u := \mathrm{diag}(y_u)$, which is similar to $E(d)$ above. Moreover, Todd and Ye show that the logarithm of the volume of this ellipsoid is closely related to Karmarkar's potential function and hence argue that Karmarkar's algorithm is typically much more efficient than the ellipsoid method, although the iterations are more expensive. Our motivation is similar, but our methods are much closer in spirit to the ellipsoid method. We hope that by using sophisticated methods to minimize the nonlinear function of d, we will need far fewer iterations than the ellipsoid method, even though our iterations are again more expensive. There are also considerable differences between our methods and the interpretation of Karmarkar's method above: w and ν are not generally equal to Dr and $l^T Du$, respectively, so $E(d)$ and $E'(d, w, \nu)$ typically differ. Further, in Karmarkar's method, y_u and y_l constitute a feasible solution to (FPD), while our d only has to lie in a set whose closure is the nonnegative orthant. Lastly, our method for updating d is very different from Karmarkar's for updating (y_u, y_l).

Consider the problem

$$(\mathrm{P}_{fh})\qquad \begin{array}{ll} \min & v(d) := f(d) \cdot h(d) \\ \text{s.t.} & d \in \mathcal{D}, \end{array}$$

where

$$\begin{aligned} f(d) &:= r^T DA^T (ADA^T)^{-1} ADr - l^T Du = x_c^T ADA^T x_c - l^T Du, \\ h(d) &:= (\det(ADA^T))^{-\frac{1}{n}}, \\ \mathcal{D} &:= \{d : d \geq 0, ADA^T \text{ is nonsingular}\}. \end{aligned}$$

Note that the volume of $E(d)$ is $\kappa_n \cdot (v(d))^{\frac{n}{2}}$, where κ_n is the volume of the unit ball in R^n. Liao and Todd [14] show that the Burrell–Todd algorithm is basically the coordinate descent method applied to (P_{fh}), together with rules for updating the bounds l and u. They also propose a simpler way to perform the updating.

Newton's method is typically fast in practice as well as in theory, while the coordinate descent algorithm is usually considered not fast enough for practical use. But here the function $v(d)$ is a homogeneous function of degree 0, and hence it can easily be shown that

$$\nabla v(d) + \nabla^2 v(d)^T d = 0,$$
$$d^T \nabla v(d) = 0,$$

so d is the Newton direction, as well as a direction of constancy for the function, and therefore useless as a search direction. We could solve a constrained problem to alleviate this difficulty, but we still face a nonconvex minimization.

The first new model is as follows:

$$(\mathrm{P}_{f+h}) \qquad \begin{array}{l} \min\ F(d) := f(d) + h(d) \\ \text{s.t. } d \in \mathcal{D}. \end{array}$$

We note that if $f(d) > 0$, by the arithmetic–geometric mean inequality, $F(d) \geq 2\sqrt{v(d)}$. Moreover, it is easy to see that f is homogeneous of degree 1 and h of degree -1; thus, for any given $d \in \mathcal{D}$, $F(d)$ attains its minimum over the half-line $\{\lambda d : \lambda \in R, \lambda > 0\}$ at $d^* := \sqrt{h(d)/f(d)} \cdot d$ which gives the value $F(d^*) = 2\sqrt{v(d^*)}$.

2.1. Properties of ($\mathbf{P}_{f+h}$). In the following we give some properties of (P_{f+h}).

PROPOSITION 2.1. *$\mathcal{D}$ defined above is a convex set.*

Proof. The straightforward derivation is omitted. □

Now we show that both $f(d)$ and $h(d)$ are convex functions over $\mathcal{D}$. We first prove some lemmas.

LEMMA 2.2.

$$\frac{\partial (ADA^T)^{-1}}{\partial d_i} = -(ADA^T)^{-1} a_i a_i^T (ADA^T)^{-1},$$
$$\frac{\partial (\det(ADA^T))}{\partial d_i} = \det(ADA^T) a_i^T (ADA^T)^{-1} a_i. \tag{5}$$

Proof. The proof is easy using the rank-1 update formulae. □

PROPOSITION 2.3. *For $i = 1, \ldots, m$, the ith components of $\nabla f(d)$ and $\nabla h(d)$ are*

$$\frac{\partial f(d)}{\partial d_i} = -(a_i^T x_c - l_i)(a_i^T x_c - u_i),$$
$$\frac{\partial h(d)}{\partial d_i} = -\frac{1}{n} \det(ADA^T)^{-\frac{1}{n}} a_i^T (ADA^T)^{-1} a_i.$$

Proof. This result can be obtained by direct calculation using Lemma 2.2. □

If $B = (b_{ij})_{m\times n}$, $C = (c_{ij})_{m\times n}$ are two m-by-n matrices, the Schur product of B and C is defined by $B \circ C := (b_{ij}c_{ij})_{m\times n}$. The following result can be found, for example, in [7].

LEMMA 2.4. *If B and C are m-by-m positive semidefinite matrices, so is $B \circ C$.*

PROPOSITION 2.5. *f and h are convex functions over $\mathcal{D}$, and hence so is $F = f + h$.*

Proof. By using Lemma 2.2, direct calculations give the Hessian of f as follows:

$$H_f = 2 \cdot \mathrm{diag}(A^T x_c - r) A^T (ADA^T)^{-1} A \cdot \mathrm{diag}(A^T x_c - r), \tag{6}$$

which is obviously positive semidefinite for all $d \in \mathcal{D}$. As for h, calculations using Lemma 2.2 give us the following expression for the (i,j)-element of the Hessian of h:

$$H_h(i,j) = \frac{1}{n^2}\det(ADA^T)^{-\frac{1}{n}}[a_i^T(ADA^T)^{-1}a_i \cdot a_j^T(ADA^T)^{-1}a_j + n(a_i^T(ADA^T)^{-1}a_j)^2]. \tag{7}$$

Let $\nu_i := (ADA^T)^{-\frac{1}{2}}a_i$. Then

$$H_h(i,j) = \frac{1}{n^2}\det(ADA^T)^{-\frac{1}{n}}[(\nu_i^T\nu_i)\cdot(\nu_j^T\nu_j) + n(\nu_i^T\nu_j)^2], \tag{8}$$

which shows that H_h is positive semidefinite, since the first term in the square brackets is a matrix of a product of a vector by its transpose, which is thus positive semidefinite, while the second term is positive semidefinite from Lemma 2.4. ∎

We now make the following assumption:

$$\text{int}(P) \neq \emptyset \text{ and } \|a_i\| \neq 0 \quad \text{for all } i = 1,\dots,m. \tag{A1}$$

Here and throughout we denote by $\|\cdot\|$ the l_2-norm.

Theoretically speaking, if P is not empty then by a perturbation we can get an equivalent system which satisfies $\text{int}(P) \neq \emptyset$. Thus (A1) is not that restrictive.

In the following we show that, under (A1), the solution set of (P_{f+h}) is nonempty and $x_c(d^*)$ is an interior point of P and unique for any solution d^* of (P_{f+h}). The feasibility problem for the system (FP) is thus converted into an unconstrained optimization problem.

First we prove a simple lemma.

LEMMA 2.6. *If* $\text{int}(P) \neq \emptyset$, *then* $f(d) > 0$ *for all* $d \in \mathcal{D}$.

Proof. For $d \in \mathcal{D}$, $E(d)$ is an ellipsoid that contains P. If $\text{int}(P)$ is not empty, $E(d)$ is a full-dimensional ellipsoid; thus the volume of $E(d)$ is positive, i.e., $v(d) > 0$. Therefore $f(d)h(d) > 0$, and so $f(d) > 0$, since $h(d) > 0$. ∎

THEOREM 2.7. *Suppose* (A1) *holds. Then* (P_{f+h}) *is a convex programming problem, and the solution set, denoted* $\mathcal{S}_{f+h}$, *is not empty*; *moreover,* $x_c(d^*)$ *is an interior point of* P *for any optimal solution* d^* *of* (P_{f+h}).

Proof. From the previous propositions, (P_{f+h}) is a convex program. We now prove its solution set is not empty. Define

$$\tilde{F}(d) := \begin{cases} f(d) + h(d) & \text{if } d \in \mathcal{D}, \\ +\infty & \text{otherwise.} \end{cases}$$

Since $\tilde{F}$ is not identically $+\infty$ and $\tilde{F}(d) > -\infty$ for every d, $\tilde{F}$ is a proper convex function.

On the other hand, since $\tilde{F}(d)$ is continuous on $\text{dom}(\tilde{F}) = \mathcal{D}$, if d_k lies in the relative interior of $\mathcal{D}$ and $d_k \to d$, then, by Lemma 2.6,

$$\tilde{F}(d_k) \longrightarrow \left\{ \begin{array}{ll} F(d) & \text{if } d \in \mathcal{D}, \\ +\infty & \text{if } d \notin \mathcal{D}, \end{array} \right\} = \tilde{F}(d).$$

So $\tilde{F}$ is closed (since $\text{cl}\tilde{F} = \tilde{F}$).

In the following we prove that $\tilde{F}$ has no (nonzero) direction of recession in R^m. Suppose we are given a direction, say y. We show that y cannot be a recession direction of $\tilde{F}$. Since $\tilde{F}(d) = \infty$ for $d \notin R^m_+$ where $R^m_+ := \{d \in R^m : d \geq 0\}$, no $y \notin R^m_+$ can be a recession direction for $\tilde{F}$. Therefore we need only consider $y \in R^m_+$.

Suppose we are given $y \in R^m_+$ and $y \neq 0$. Without loss of generality we assume $y_1 \neq 0$. Since A is of full rank and $a_1 \neq 0$, there is a set of n columns of A including a_1 which is linearly independent; further, by Cauchy's formula,

$$h(e + \lambda y) \leq \alpha^{-\frac{2}{n}}(1 + \lambda y_1)^{-\frac{1}{n}},$$

where α is the determinant of the matrix formed by the n independent columns. Thus $h(e + \lambda y) \longrightarrow 0$. On the other hand, since $\text{int}(P)$ is not empty and $P \subset E$, there is a $\delta > 0$ such that $f(e + \lambda y)h(e + \lambda y) > \delta > 0$. Thus

$$f(e + \lambda y) \longrightarrow \infty \text{ as } \lambda \longrightarrow \infty,$$

which shows f, thus $\tilde{F}$, has no recession direction in R^m_+.

We have thus proved that $\tilde{F}$ has no recession direction in R^m. Therefore, by Theorem 27.3 of Rockafellar [20], $\tilde{F}$ attains its minimum in R^m. Equivalently, F attains its minimum over $\mathcal{D}$, i.e., $\mathcal{S}_{f+h}$ is not empty.

On the other hand, by the Karush–Kuhn–Tucker (KKT) conditions, d^* is optimal if and only if there are $\lambda^T = (\lambda_1, \ldots, \lambda_m) \geq 0$ such that $\nabla F(d^*) = \lambda$, i.e.,

$$-(a_i^T x_c(d^*) - l_i)(a_i^T x_c(d^*) - u_i) = \frac{1}{n}\det(ADA^T)^{-\frac{1}{n}} a_i^T (ADA^T)^{-1} a_i + \lambda_i \tag{9}$$

for all $i = 1, \ldots, m$. The right-hand side of (9) is positive; hence $x_c(d^*)$ is an interior point. □

The following corollaries show that, under assumption (A1), (P_{f+h}) and (P_{fh}) are essentially the same.

COROLLARY 2.8. *Under* (A1)*, the solution set of*

$$(\text{P}_{fh}) \qquad \begin{array}{ll} \min & v(d) := f(d) \cdot h(d) \\ \text{s.t.} & d \in \mathcal{D} \end{array}$$

is not empty.

Proof. Suppose d^* is an optimal solution to (P_{f+h}). We show that d^* is an optimal solution to (P_{fh}). Otherwise, there exists $\bar{d}$ such that $v(d^*) > v(\bar{d})$. Since $f(d) > 0$ for $d \in \mathcal{D}$, we can let $\lambda = \sqrt{h(\bar{d})/f(\bar{d})}$ and have

$$2\sqrt{v(\bar{d})} = 2\sqrt{v(\lambda\bar{d})} = f(\lambda\bar{d}) + h(\lambda\bar{d}),$$

which leads to

$$f(d^*) + h(d^*) \geq 2\sqrt{v(d^*)} > 2\sqrt{v(\bar{d})} = 2\sqrt{v(\lambda\bar{d})} = f(\lambda\bar{d}) + h(\lambda\bar{d}),$$

a contradiction. □

As a matter of fact, if we let $\mathcal{S}_{fh}$ and $\mathcal{S}_{f+h}$ be the solution sets of (P_{fh}) and (P_{f+h}), respectively, we have the following result.

COROLLARY 2.9. *Under assumption* (A1)*,* $\mathcal{S}_{f+h} \subset \mathcal{S}_{fh}$ *and* $\mathcal{S}_{fh} = \text{cone}(\mathcal{S}_{f+h}) \setminus \{0\}$.

The following example shows that the Hessian of F might be only positive semidefinite, and the optimal solution set $\mathcal{S}_{f+h}$ can be a segment.

Example. Let

$$A = \begin{pmatrix} 1 & 0 & 1 & -2 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

and

$$l = \left(-\sqrt{\frac{1}{2}}, -\sqrt{\frac{1}{2}}, -1, -\sqrt{\frac{5}{2}}\right)^T, \quad u = \left(\sqrt{\frac{1}{2}}, \sqrt{\frac{1}{2}}, 1, \sqrt{\frac{5}{2}}\right)^T.$$

Then, if we take $d^0 = (\frac{1}{4}, \frac{5}{8}, \frac{1}{4}, \frac{1}{8})^T$, $\nabla^2 F(d^0)$ is singular. Indeed, $r = 0$ so $x_c(d) = 0$ for all d, whence $H_f(d) = 0$. Also, it is easy to see that $(6, 3, -2, -1)^T$ lies in the null space of H_h (see (7)). Moreover, let

$$d = \left(0, \frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right)^T, \quad \bar{d} = (1, 1, 0, 0)^T, \quad \text{and} \quad d_\lambda = \lambda d + (1-\lambda)\bar{d}, \quad \lambda \in [0, 1].$$

Then

$$\begin{aligned} F(d_\lambda) &= f(d_\lambda) + h(d_\lambda) \\ &= r^T D_\lambda A^T (A D_\lambda A^T)^{-1} A D_\lambda r - l^T D_\lambda u + \det(A D_\lambda A^T)^{-\frac{1}{n}} \\ &\equiv 2 \text{ (note that } r = 0,\ -l^T D_\lambda u = 1,\ A D_\lambda A^T = I), \end{aligned}$$

and $\nabla F(d_\lambda) = 0$ (see Proposition 2.3). Therefore, the solution set of (P_{f+h}) is (at least) $\{d_\lambda = \lambda d + (1-\lambda)\bar{d} : \lambda \in [0, 1]\}$.

We now show that $x_c(d^*)$ is unique even though the optimal solution set $\mathcal{S}_{f+h}$ may not be a singleton. We can thus define $x_c(d^*)$ as a (weighted) center. We first require several lemmas.

LEMMA 2.10. *For all* $d, \bar{d} \in \mathcal{S}_{f+h}$,

$$f(d) = f(\bar{d}) = h(d) = h(\bar{d}). \tag{10}$$

Proof. Since $\mathcal{S}_{f+h}$ is a convex set, $F(d) = F(d')\ \forall d, d' \in \mathcal{S}_{f+h}$; on the other hand, because f and h are nonnegative functions which are homogeneous with degrees 1 and -1, respectively, for $d \in \mathcal{S}_{f+h}$, $f(d) = h(d) = \frac{1}{2}F(d)$. The lemma follows immediately. ∎

LEMMA 2.11. *If* $\det(I + \varepsilon\Lambda) \equiv 1$ *for all* $\varepsilon \in (0, \eta)$, *where* η *is some positive number, and* $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, *then* $\Lambda = 0$.

Proof. We have $\det(I + \varepsilon\Lambda) = \prod_{i=1}^n (1 + \varepsilon\lambda_i) \equiv 1\ \forall \varepsilon \in (0, \eta)$. Thus the coefficient of ε^n should be zero, i.e., $\prod_{i=1}^n \lambda_i = 0$; so, at least one of the λ_i's is zero, say $\lambda_n = 0$. Thus we have

$$\det(I + \varepsilon\Lambda) = \prod_{i=1}^{n-1} (1 + \varepsilon\lambda_i).$$

The same argument implies that at least one of $\{\lambda_1, \ldots, \lambda_{n-1}\}$ is zero; we keep using this argument until eventually we have $\Lambda = 0$. ∎

PROPOSITION 2.12. *For all* $d, \bar{d} \in \mathcal{S}_{f+h}$, $ADA^T = A\bar{D}A^T$.

Proof. By Lemma 2.10,

$$\det(A(D + \varepsilon(\bar{D} - D))A^T) = \det(ADA^T) \quad \forall \varepsilon \in [0, 1]. \tag{11}$$

On the other hand,

$$\begin{aligned} &\det(A(D + \varepsilon(\bar{D} - D))A^T) = \det(ADA^T + \varepsilon A(\bar{D} - D)A^T) \\ &\quad = \det(ADA^T)^{\frac{1}{2}} \det(I + \varepsilon(ADA^T)^{-\frac{1}{2}} A(\bar{D} - D)A^T (ADA^T)^{-\frac{1}{2}}) \det(ADA^T)^{\frac{1}{2}}. \end{aligned}$$

Thus, (11) implies $\det(I+\varepsilon(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}}) = 1$. By Lemma 2.11, the eigenvalues of $(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}}$ are all zeros, so

$$(ADA^T)^{-\frac{1}{2}}A(\bar{D}-D)A^T(ADA^T)^{-\frac{1}{2}} = 0,$$

i.e., $A(\bar{D}-D)A^T = 0$. Hence $A\bar{D}A^T = ADA^T$. □

Thus the minimum volume ellipsoids containing P all have the same shape but possibly different locations. We now show that the location is also unique.

PROPOSITION 2.13. *For all $d, \bar{d} \in \mathcal{S}_{f+h}$, $ADr = A\bar{D}r$.*

Proof. From Lemma 2.10 and Proposition 2.12,

$$\begin{aligned} &r^T(D+\varepsilon(\bar{D}-D))A^T(ADA^T)^{-1}A(D+\varepsilon(\bar{D}-D))r - l^T(D+\varepsilon(\bar{D}-D))u \\ &\quad = r^TDA^T(ADA^T)^{-1}ADr - l^TDU \ \forall \varepsilon \in [0,1]. \end{aligned}$$

Thus by letting the coefficient of ε^2 be zero,

$$r^T(\bar{D}-D)A^T(ADA^T)^{-1}A(\bar{D}-D)r = 0.$$

Since $(ADA^T)^{-1}$ is positive definite, $A(\bar{D}-D)r = 0$, i.e., $A\bar{D}r = ADr$. □

THEOREM 2.14 (uniqueness). *If d^* solves (P_{f+h}), $x_c(d^*)$ and $E(d^*)$ are unique.*

Proof. By Propositions 2.12 and 2.13, for any $d \in \mathcal{S}_{f+h}$, $x_c^* = (ADA^T)^{-1} \cdot ADr$ is a constant vector. □

Now we show that f can be used to determine whether $P = \emptyset$. This result provides us a tool that is useful later in the analysis of the output of our model. Recall from Lemma 2.6 that $f(d) \leq 0$ implies that the interior of P is empty. To complement this, we have the following result; the hard part of the proof can be skimmed or skipped at a first reading.

THEOREM 2.15. *If $l < u$ and A is of full rank, then*

$$P = \emptyset \Longleftrightarrow f(d) < 0 \textit{ for some } d \in \mathcal{D}.$$

Proof. "$\Longleftarrow$". If there is a d such that $f(d) < 0$ and $P \neq \emptyset$, then, by taking $\bar{l}_i = l_i - \delta, \bar{u}_i = u_i + \delta$ and noting that f is a continuous function of l and u for any given d, we can choose small δ so that for the corresponding system we have $\mathrm{int}(P) \neq \emptyset$ and $f(d) < 0$, which contradicts Lemma 2.6.

"$\Longrightarrow$". First of all, we recall Helly's theorem which can be found, e.g., in Rockafellar [20]:

> Let $\{C_i : i \in \mathcal{I}\}$ be a finite collection of convex sets in R^n. If every subcollection consisting of $n+1$ or fewer sets has a nonempty intersection, then the entire collection has a nonempty intersection.

If $P = \emptyset$, by Helly's theorem there are $n+1$ constraints which are inconsistent. If the rank of the set of the corresponding a_i's is $k < n$, then, since Helly's theorem is invariant under affine transformation, we can find $k+1$ constraints which are inconsistent too. Moreover, since we assume that A is of full rank, we can expand these $k+1$ ones to $n+1$ constraints which are inconsistent and the corresponding a_i's are of rank n.

Therefore, if $P = \emptyset$, there are $n+1$ constraints, say the first $n+1$ ones, which are inconsistent and the corresponding a_i's are of rank n. Without loss of generality (by making a nonsingular affine transformation of the space), we assume that

$$\bar{A} = [I, a], \quad l_i = -1, u_i = 1 \ \forall i = 1, \ldots, n, \quad \text{and } \|a\|_1 = 1,$$

and the corresponding system

$$\begin{pmatrix} -e \\ l_{n+1} \end{pmatrix} \le \begin{pmatrix} I \\ a^T \end{pmatrix} x \le \begin{pmatrix} e \\ u_{n+1} \end{pmatrix} \tag{12}$$

is inconsistent. We claim that there exists a $d \in \{d \in \mathcal{D} : d_i = 0 \ \forall i = n+2, \ldots, m\}$ for which $f(d) < 0$. Equivalently, we show that the claim is true for system (12).

We denote $s := \frac{u-l}{2}$ for this reduced system as before. Since system (12) has no solution, if we let $\mathcal{X} := \{x \in R^n : -1 \le x_j \le 1, j = 1, \ldots, n\}$, then either $a^T x < l_{n+1} \ \forall x \in \mathcal{X}$, or $a^T x > u_{n+1} \ \forall x \in \mathcal{X}$. We assume the first case. Note that $a^T x$ is maximized over $\mathcal{X}$ by some vertex of $\mathcal{X}$, which gives its optimal value as $\sum_{j=1}^n |a_{[j]}| = \|a\|_1 = 1$, where $a = (a_{[1]}, \ldots, a_{[n]})^T$. Therefore, $l_{n+1} > 1$.

We now show how to define d so that $f(d) < 0$. Confirming this inequality requires some straightforward but messy manipulations; for details see [13, 15].

(i) If $a_{[j]} \ne 0 \ \forall j = 1, \ldots, n$, take $d_j = |a_{[j]}|$ for $j = 1, \ldots, n$ and

$$d_{n+1} = \frac{l_{n+1} u_{n+1} - \sum_{j=1}^n d_j \sum_{j=1}^n a_{[j]}^2 / d_j}{2 s_{n+1}^2 \sum_{j=1}^n a_{[j]}^2 / d_j}.$$

(ii) If some $a_{[j]}$'s are zeros, then we take

$$d_j = \begin{cases} |a_{[j]}| - \varepsilon & \text{if } a_{[j]} \ne 0, \\ \frac{n'}{n-n'} \varepsilon & \text{if } a_{[j]} = 0, \end{cases}$$

where n' is the number of $a_{[j]}$'s with $a_{[j]} \ne 0$ and d_{n+1} as above. By letting ε be sufficiently small, we get d so that $f(d) < 0$. $\square$

2.2. An algorithm for (P$_{f+h}$). We now consider how to use model (P$_{f+h}$) to solve the original feasibility problem and how to obtain the center $x_c^* := x_c(d^*)$ (by the uniqueness theorem, this is well defined).

Here is a coordinate descent algorithm.

ALGORITHM 2.1.

- Initialization. Choose $d^0 > 0$, and scale it so that $f(d^0) = h(d^0)$; set $k = 0$.
- For $k = 0, 1, \ldots,$ do

  If $x_c^k := x_c(d^k) \in P$, stop with the feasible solution x_c^k; otherwise, choose j with the jth constraint violated by x_c^k. Let

  $$\beta := \frac{s_j^2}{f(d^k) a_j^T (ADA^T)^{-1} a_j}, \quad \gamma := a_j^T (ADA^T)^{-1} a_j;$$

  and

  $$(l_0, u_0) = \begin{cases} (l_j, l_j + (f(d^k)\gamma)^{\frac{1}{2}}) & \text{if } a_j^T x_c^k < l_j, \\ (u_j - (f(d^k)\gamma)^{\frac{1}{2}}, u_j) & \text{if } a_j^T x_c^k > u_j. \end{cases}$$

  If $l_0 > u_j$ or $u_0 < l_j$, stop with the conclusion that the original system is not consistent.

(i) (canonical case) If $\beta \le \frac{1}{4}$, we take $\lambda^k = \frac{1}{n\gamma}$, and $d^{k+1} = d^k + \lambda^k e_j$.

(ii) If $\beta > \frac{1}{4}$, we take $\lambda^k = \frac{1}{n\gamma}$, and $d^{k+1} = d^k + \lambda^k e_j$, AND update the bounds as follows:

$$l_j \leftarrow \frac{d_j^k l_j + \tilde{d}_0^{k+1} l_0}{d_j^k + \tilde{d}_0^{k+1}} \quad \text{if } a_j^T x_c^k > u_j, \text{ or}$$

$$u_j \leftarrow \frac{d_j^k u_j + \tilde{d}_0^{k+1} u_0}{d_j^k + \tilde{d}_0^{k+1}} \quad \text{if } a_j^T x_c^k < l_j,$$

where $\tilde{d}_0^{k+1} = \frac{1}{n\gamma}$.

Then, scale d^{k+1} so that $f(d^{k+1}) = h(d^{k+1})$, where f is computed with the updated jth bounds; set $k \leftarrow k+1$, and repeat.

Let us explain the noncanonical case briefly. Without loss of generality, $a_j^T x_c^k < l_j$, and $\beta > \frac{1}{4}$ implies that $u_j > l_j + (f(d^k)\gamma)^{\frac{1}{2}} =: u_0 > a_j^T x_c^k + (f(d^k)\gamma)^{\frac{1}{2}}$ so that the hyperplanes $\{x : a_j^T x = u_j\}$ and $\{x : a_j^T x = u_0\}$ don't intersect the current ellipsoid. It follows that the upper bounds u_j and u_0 on $a_j^T x$ are implied by the remaining bounds and that the polytope P is unchanged if u_j is replaced by u_0 or by any convex combination of these. Case (ii) above corresponds to the following procedure: we add a new constraint $l_0 \le a_0^T x \le u_0$, where $a_0 := a_j$, with current weight $d_0^k = 0$, which doesn't change the current ellipsoid. The current center violates the lower bound of this new pair of constraints, for which the value of β is exactly $\frac{1}{4}$. Thus we perform the canonical update with this constraint so that d_0 becomes $\frac{1}{n\gamma}$. It turns out that the resulting ellipsoid can also be described with just the old constraints if we update the weights *and* u_j as described above; the two terms of the defining inequality corresponding to indices 0 and j are thus combined into one. The difference from the Liao and Todd algorithm [14] is that here the objective function is $F(d) := f(d) + h(d)$ instead of $v(d) := f(d) \cdot h(d)$ as used there.

THEOREM 2.16. *Suppose that $\{d^k\}$ is the sequence generated by the algorithm. Then $\frac{F(d^{k+1})}{F(d^k)} \le 1 - \frac{1}{16n^2}$, as long as $x_c(d^k) \notin P$.*

Proof. Let $\tilde{d}^{k+1}$ be $d^k + \lambda^k e_j$ (the next iterate before scaling). Also let $\theta = \lambda\gamma$ and $\theta_0 := \frac{(a_j^T x_c - l_j)(a_j^T x_c - u_j)}{s_j^2} \ge 0$. Then the same analysis as in Liao and Todd [14] leads to

$$\begin{aligned}\frac{F(\tilde{d}^{k+1})}{F(d^k)} &= 1 + \frac{1}{F(d^k)}\left[f(d^k)\beta\frac{\theta}{1+\theta}(\theta-\theta_0) - (1-(1+\theta)^{-\frac{1}{n}})h(d^k)\right] \\ &= 1 - \frac{f(d^k)}{f(d^k)+h(d^k)}\left[1-(1+\theta)^{-\frac{1}{n}} - \beta\frac{\theta}{1+\theta}(\theta-\theta_0)\right] \\ &= 1 - \frac{1}{2}\left[1-(1+\theta)^{-\frac{1}{n}} - \beta\frac{\theta}{1+\theta}(\theta-\theta_0)\right].\end{aligned}$$

Let $R(\theta) := 1-(1+\theta)^{-\frac{1}{n}} - \beta\frac{\theta}{1+\theta}(\theta-\theta_0)$. As in [14], it is enough to deal with the canonical case, i.e., $\beta \le \frac{1}{4}$. Since $\theta_0 > 0$,

$$\begin{aligned}R(\theta) &\ge 1-(1+\theta)^{-\frac{1}{n}} - \beta\frac{\theta^2}{1+\theta} \\ &\ge 1-(1+\theta)^{-\frac{1}{n}} - \frac{1}{4}\cdot\frac{\theta^2}{1+\theta} =: r(\theta).\end{aligned}$$

The following inequality can be obtained from Taylor's theorem:

$$\left|(1+\theta)^{\frac{1}{n}} - \left(1+\frac{\theta}{n}\right)\right| \le \frac{n-1}{2n^2}\theta^2 \text{ for } 0 < \theta \le 1.$$

Hence, noting that $\theta > 0$,

$$\left|(1+\theta)^{-\frac{1}{n}} - \left(1+\frac{\theta}{n}\right)^{-1}\right| \le \frac{n-1}{2n^2}\theta^2.$$

Therefore, for $0 < \theta \le 1$,

$$\begin{aligned} r(\theta) &\ge 1 - \left(1+\frac{\theta}{n}\right)^{-1} - \frac{1}{4}\cdot\frac{\theta^2}{1+\theta} - \frac{n-1}{2n^2}\theta^2 \\ &= \frac{\theta}{n+\theta} - \frac{1}{4}\cdot\frac{\theta^2}{1+\theta} - \frac{n-1}{2n^2}\theta^2 \\ &= \theta\left[\frac{4-n\theta+4\theta-\theta^2}{4(n+\theta)(1+\theta)} - \frac{n-1}{2n^2}\theta\right]. \end{aligned}$$

If we take $\theta = \frac{1}{n}$, we have

$$\begin{aligned} r(\theta) &\ge \frac{1}{n}\left[\frac{3}{4(1+\theta)(n+1)} - \frac{n-1}{2n^3}\right] \ge \frac{1}{n}\left[\frac{3}{8(n+1)} - \frac{n-1}{2n^3}\right] \\ &\ge \frac{1}{n}\left[\frac{6n^3-8n^2+8}{16n^3(n+1)}\right] \ge \frac{1}{n}\cdot\frac{1}{8n} = \frac{1}{8n^2}. \end{aligned}$$

So

$$\frac{F(\tilde{d}^{k+1})}{F(d^k)} = 1 - \frac{1}{2}R(\theta) \le 1 - \frac{1}{2}r(\theta) \le 1 - \frac{1}{16n^2}. \qquad \square$$

In terms of the shrinkage of the volumes of the corresponding ellipsoids, we have

$$\frac{\text{volume}(E^+)}{\text{volume}(E)} \le \left(1 - \frac{1}{16n^2}\right)^n \le \exp\left(-\frac{1}{16n}\right).$$

Although, from the above results, the coordinate descent algorithm is polynomial, it converges slowly. In the next section we will propose another model where Newton's method can be employed completely. We also note that a partial Newton step algorithm is proposed in [13].

When we use Model I to solve the feasibility problem (1), there are three possible outcomes according to the values of $F_{\inf} := \inf\{F(d) : d \in \mathcal{D}\}$.

THEOREM 2.17.

(i) *If* $F_{\inf} > 0$, *then there is a* $d^* \in \mathcal{D}$ *such that* $F(d^*) = F_{\inf}$ *and* $x_c(d^*)$ *is feasible.*

(ii) *If* $F_{\inf} < 0$, *then* $P = \emptyset$.

(iii) *If* $F_{\inf} = 0$, *then* $\text{int}(P) = \emptyset$ *but* $P \ne \emptyset$.

Proof. It is easy to see that if we replace (A1) by $F_{\inf} > 0$ then f has no direction of recession and so the conclusions of Theorem 2.7 still hold; (i) thus follows. If $F_{\inf} < 0$, then there is a $d \in \mathcal{D}$ such that $F(d) < 0$ and therefore $f(d) < 0$. Hence, by

Theorem 2.15, $P = \emptyset$, which shows (ii). Outcome (iii) follows from Theorem 2.7 and Theorem 2.15. ∎

In case (iii), if there is a $d^* \in \mathcal{D}$ at which $f(d^*) = 0$, then $x_c(d^*)$ is feasible; however, in general, $x_c(d^k)$ does not approach the feasible set even if d^k is such that $F(d^k) \longrightarrow 0$ as shown in the following example.

Example. Let

$$\begin{pmatrix} 0 \\ 0 \\ 2 \\ 0 \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} x \leq \begin{pmatrix} 2 \\ 2 \\ 4 \\ 30 \end{pmatrix}.$$

By calculation, we have

$$\begin{aligned} x_c &= \left(\frac{d_1 + 3d_3}{d_1 + d_3}, \ \frac{d_2 + 15d_4}{d_2 + d_4} \right), \\ f(d) &= \frac{(d_1 + 3d_3)^2}{d_1 + d_3} + \frac{(d_2 + 15d_4)^2}{d_2 + d_4} - 8d_3, \\ h(d) &= \frac{1}{\sqrt{(d_1 + d_3)(d_2 + d_4)}}. \end{aligned}$$

If we take $d_1^k = d_3^k = \frac{1}{2}$ and $d_2^k = d_4^k = \frac{1}{k}$, then $v(d^k) \longrightarrow 0$ as $k \longrightarrow \infty$, but $x_c(d^k) \longrightarrow (2, 8)^T$ which is not feasible. By scaling, this sequence of d^k can be converted to $\bar{d}^k$ such that $F(\bar{d}^k) \longrightarrow 0$ as $k \longrightarrow \infty$ and $x_c(\bar{d}^k) = x_c(d^k) \longrightarrow (2, 8)^T$.

3. Model II and the associated center. Model I provides us with a kind of center for P, but this center is not so easy to obtain practically. In this section we propose a simpler model whose associated center can be obtained more efficiently.

By examining the previous model, we find that it is based on the homogeneity properties of f and h. Since f provides a tool for determining the feasibility of our problem we want to keep it, but the role of function h is really like a barrier: it balances the weights $d_i, i = 1, \ldots, m$, and forces the corresponding $x(d)$ to approach the center x_c^*. We will use a much simpler function to replace h. The new model is more efficient in practice.

As a matter of fact, any convex homogeneous function B of degree -1 defined on $\mathcal{D}$ such that

(i) $f + B$ has an optimal solution in $\mathcal{D}$,

(ii) $\nabla B(d) \leq 0$ for any $d \in \mathcal{D}$

can play the same role as h does. (The self-concordant barrier function $-\sum \ln d_i$ familiar in interior-point methodology fails to be homogeneous; the related multiplicative barrier $(\prod d_i)^{-1/m}$ is a possibility, but seems to be more complicated than the simple function we propose below and does not allow the convergence analysis below.)

One such function, which we choose for our new model, is as follows:

$$B(d) := e^T d^{-1} := \sum_{i=1}^{m} \frac{1}{d_i}.$$

We let $B(d) = e^T d^{-1} = +\infty$ if any $d_i = 0$.

The corresponding model, defined as Model II, is

$$(\mathrm{P}_{f+B}) \qquad \begin{array}{ll} \min & G(d) := f(d) + B(d) \\ \text{s.t.} & d \in \mathcal{D}. \end{array}$$

We note that (P_{f+B}) is equivalent to minimizing $G(d)$ over $R^m_{++} := \{d \in R^m : d > 0\}$.

3.1. Properties of $(\mathbf{P}_{f+B})$. Obviously, $B(d)$ is a strictly convex function, so $G(d)$ is strictly convex too. Accordingly, we have the following theorem.

THEOREM 3.1. *If* (A1) *holds, then* $G(d)$ *is a closed proper strictly convex function and* $G(d)$ *has no direction of recession. Thus there is an unique solution to* (P_{f+B}); *moreover, the level set*

$$\mathcal{D}_e := \{d \in \mathcal{D} : G(d) \le G(e)\}$$

is a closed bounded convex set.

Proof. The proof of Theorem 3.1 is similar to that of Theorem 2.7. We just note that, under assumption (A1), $f(d)$ has no recession direction in R^m_+; hence, $G(d)$ has no recession direction either. ∎

It is easy to prove that if d^* is the unique solution to (P_{f+B}), the associated $x_c(d^*)$ is an interior point of P. We thus define $x_c(d^*)$ as the *center* associated with Model II. In the following we show that (P_{f+B}) is equivalent to finding a minimum volume ellipsoid among those with a particular shape, and $x_c(d^*)$ is the center of this ellipsoid.

First we note that (P_{f+B}) is equivalent to

$$(P_{fB}) \qquad \begin{array}{ll} \min & v^{\mathrm{II}}(d) := f(d) \cdot B(d) \\ \text{s.t.} & d \in \mathcal{D} \end{array}$$

in the following sense: if d^* is the solution to (P_{f+B}), it is also a solution to (P_{fB}); and if d^* is a solution to (P_{fB}), then $\sqrt{B(d^*)/f(d^*)}d^*$ is the unique solution to (P_{f+B}).

LEMMA 3.2. *Suppose that* $d_i \ge 0, i = 1, \dots, m$. *Then*

$$\sum_{i=1}^{m} d_i w_i^2 \le 1 \tag{13}$$

implies

$$\sum_{i=1}^{m} w_i^2 \le e^T d^{-1}. \tag{14}$$

Proof. If there is i_0 such that $d_{i_0} = 0$, then (14) is always true, since we have set $e^T d^{-1}$ to ∞ in this case; otherwise we suppose that $d_{i_0} = \min\{d_i : i = 1, \dots, m\}$. Then $d_{i_0} \sum_{i=1}^m w_i^2 \le \sum_{i=1}^m d_i w_i^2 \le 1$, which leads to $\sum_{i=1}^m w_i^2 \le \frac{1}{d_{i_0}} \le e^T d^{-1}$. ∎

We define

$$E^{\mathrm{II}}(d) := \left\{ x \in R^n : \frac{1}{e^T d^{-1}} (x - x_c)^T A A^T (x - x_c) \le f(d) \right\} \tag{15}$$

where $x_c = x_c(d) = (ADA^T)^{-1} ADr$ and $f(d) = x_c^T ADA^T x_c - l^T Du$. Then (P_{fB}) is equivalent to minimizing the volume of the ellipsoid $E^{\mathrm{II}}(d)$.

The relationship between this ellipsoid and the one defined by (4) is described in the next proposition.

PROPOSITION 3.3. *If* $E(d)$ *is the ellipsoid defined by* (4), *then* $E(d) \subset E^{\mathrm{II}}(d)$.

Proof.

$$
\begin{aligned}
E(d) &= \{x \in R^n : (x - x_c)^T ADA^T (x - x_c) \le f(d)\} \\
&= \{x \in R^n : (A^T(x - x_c))^T D(A^T(x - x_c)) \le f(d)\}.
\end{aligned}
$$

Let $w := A^T(x - x_c)$. Then, by Lemma 3.2,

$$
\begin{aligned}
E(d) &= \{x \in R^n : w^T D w \le f(d)\} = \left\{ x \in R^n : \sum_{i=1}^m d_i w_i^2 \le f(d) \right\} \\
&\subseteq \left\{ x \in R^n : \sum_{i=1}^m w_i^2 \le f(d) e^T d^{-1} \right\} \\
&= \left\{ x \in R^n : \frac{1}{e^T d^{-1}} (x - x_c)^T AA^T (x - x_c) \le f(d) \right\} = E^{\mathrm{II}}(d). \qquad \square
\end{aligned}
$$

Since $P \subset E(d)$, the above result also shows that $P \subset E^{\mathrm{II}}(d)$.

We note that since ADA^T can be nonsingular if one component of d is zero, the volume of $E(d)$ may be finite, but for such a d the volume of $E^{\mathrm{II}}(d)$ is infinite; hence E^{II} and E are not equivalent, i.e., there is no constant $\lambda > 0$ such that

$$
E(d) \subset E^{\mathrm{II}}(d) \subset \lambda E(d) \text{ for all } d \in \mathcal{D}.
$$

3.2. Algorithms for $(\mathbf{P}_{f+B})$. We describe first a coordinate descent algorithm to solve the feasibility problem (P) by using Model II.

ALGORITHM 3.1.

- Initialization. Normalize the constraints so that $\|a_i\| = 1$ $\forall i = 1, \ldots, m$. Choose $d^0 > 0$ and scale it so that $f(d^0) = B(d^0)$; set $k = 0$.
- For $k = 0, 1, \ldots,$ do
  If $x_c^k := x_c(d^k) \in P$, stop with the feasible solution x_c^k; otherwise, choose j with the jth constraint violated by x_c^k. Then take $\lambda^k = \arg\min\{G(d^k + \lambda e_j) : \lambda \ge 0\}$, and $d^{k+1} = d^k + \lambda^k e_j$; scale d^{k+1} so that $f(d^{k+1}) = B(d^{k+1})$; set $k \leftarrow k + 1$, and repeat.

To analyze this algorithm we first need the following result.

LEMMA 3.4. *Suppose that the feasible region P of* (1) *is "fat" so that there is a ball, say $\mathcal{B}$, of radius δ_0 contained in P, i.e., there is some $\hat{x}$ such that $\mathcal{B} := \mathcal{B}(\hat{x}; \delta_0) \subset P$, and suppose that $\|a_i\| = 1$ for all $i = 1, \ldots, m$. Then, for all $i = 1, \ldots, m$, and $d \in \mathcal{D}_+$ with $f(d) = B(d)$ and $G(d) \le G(e)$,*

$$
\begin{aligned}
\kappa_i(d) &:= \frac{f(d)}{a_i^T (ADA^T)^{-1} a_i} \le \frac{(G_e)^2}{4\delta_0^2}, \\
\beta_i(d) &:= \frac{s_i^2}{f(d) a_i^T (ADA^T)^{-1} a_i} \le \frac{s_{\max}^2}{\delta_0^2},
\end{aligned}
$$

where

$$
\begin{aligned}
G_e &:= G(e) = r^T A^T (AA^T)^{-1} A r - l^T u + m, \\
s_{\max} &:= \max\{s_i : i = 1, \ldots, m\}.
\end{aligned}
$$

Proof. Since $2\sqrt{f(d)a_i^T(ADA^T)^{-1}a_i}$ is the width of the ellipsoid $E(d)$ in the direction a_i, we have $\sqrt{f(d)a_i^T(ADA^T)^{-1}a_i} \geq \delta_0$. Thus

$$\begin{aligned}\kappa_i(d) &= \frac{f(d)}{a_i^T(ADA^T)^{-1}a_i} = \frac{f(d)^2}{f(d)a_i^T(ADA^T)^{-1}a_i} \\ &= \frac{G(d)^2}{4f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{(G(e))^2}{4f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{(G_e)^2}{4\delta_0^2}.\end{aligned}$$

On the other hand,

$$\beta_i(d) = \frac{s_i^2}{f(d)a_i^T(ADA^T)^{-1}a_i} \leq \frac{s_{\max}^2}{\delta_0^2}. \qquad \square$$

THEOREM 3.5. *Assume the hypotheses of Lemma* 3.4 *and let* $\omega := \max\{\frac{(G_e)^2}{4\delta_0^2}, \frac{s_{\max}^2}{\delta_0^2}, 1\}$. *Then as long as* $x_c(d^k) \notin P$, *we have*

$$\frac{G(d^{k+1})}{G(d^k)} \leq 1 - \frac{1}{16n^6\omega^5}.$$

Proof. For simplicity we write d for d^k, λ for λ^k, and d^+ for $d + \lambda^k e_j$. Then

$$B(d^+) = B(d) - \frac{\lambda}{d_j(d_j+\lambda)},$$

and, from [14],

$$f(d^+) = f(d) + f(d)\frac{\beta_j\theta}{1+\theta}(\theta - \theta_0) \leq f(d)\left(1 + \frac{\beta_j\theta^2}{1+\theta}\right),$$

where $\beta_j = \beta_j(d)$, $\theta = \lambda\gamma_j$, $\gamma_j = a_j^T(ADA^T)^{-1}a_j$, and $\theta_0 := \frac{(a_j^T x_c - l_j)(a_j^T x_c - u_j)}{s_j^2} \geq 0$. Also,

$$\begin{aligned}\sum d_i a_i^T(ADA^T)^{-1}a_i &= \operatorname{tr}(DA^T(ADA^T)^{-1}A) \\ &= \operatorname{tr}(ADA^T(ADA^T)^{-1}) = n,\end{aligned}$$

so $d_j\gamma_j \leq n$. Thus, noting that $f(d) = B(d)$,

$$\begin{aligned}\frac{G(d^+)}{G(d)} &\leq 1 - \frac{1}{2}\left[\frac{\lambda}{f(d)d_j(d_j+\lambda)} - \beta_j\frac{\theta^2}{1+\theta}\right] \\ &\leq 1 - \frac{1}{2}\left[\frac{\theta}{nf(d)(d_j+\lambda)} - \beta_j\frac{\theta^2}{1+\theta}\right] \\ &\leq 1 - \frac{1}{2}\left[\frac{\theta}{n\kappa_j(d)(n+\theta)} - \beta_j\frac{\theta^2}{1+\theta}\right] \\ &\leq 1 - \frac{1}{2}\left[\frac{\theta}{n\omega(n+\theta)} - \omega\frac{\theta^2}{1+\theta}\right].\end{aligned}$$

If we take $\theta = \frac{1}{\omega^2 n^2}$, then

$$\frac{G(d^+)}{G(d)} \leq 1 - \frac{1}{16n^6\omega^5}. \qquad \square$$

In terms of the shrinkage of the volumes of the corresponding ellipsoids we have

$$\frac{\text{volume}(E^{\text{II}}(d^+))}{\text{volume}(E^{\text{II}}(d))} \leq \exp\left(-\frac{1}{16n^5\omega^5}\right).$$

This shrinkage rate depends on ω and hence on the bound on $\kappa_i(d)$. We note that if the initial point is chosen as $d^0 = \lambda e$ such that $G(d^0) = 2f(d^0) = 2B(d^0)$, then we can obtain a simpler bound. We denote

$$\text{Lev}_0^{\text{II}} := \{d : G(d) \leq G(d^0)\}.$$

Then, for $d \in \text{Lev}_0^{\text{II}}$ with $f(d) = B(d)$,

$$\begin{aligned}\kappa_i(d) &= \frac{f(d)^2}{f(d)a_i^T(ADA^T)^{-1}a_i} = \frac{(\frac{G(d)}{2})^2}{f(d)a_i^T(ADA^T)^{-1}a_i}\\ &\leq \frac{(\frac{G(d^0)}{2})^2}{f(d)a_i^T(ADA^T)^{-1}a_i} = \frac{f(d^0)^2}{f(d)a_i^T(ADA^T)^{-1}a_i}\\ &\leq \frac{f(d^0)B(d^0)}{\delta_0^2} \leq \zeta^{\frac{2}{n}} \cdot \det(AA^T)^{\frac{1}{n}},\end{aligned}$$

where $\zeta = \frac{\text{volume}(E_0^{\text{II}})}{\text{volume}(\mathcal{B})}$ which is the ratio of the volume of the initial ellipsoid over the volume of the ball contained in the feasible region P.

By scaling and changing variables we can assume, without loss of generality, $AA^T = I$. Then Algorithm 3.1 is a kind of ball method [2] for which Todd [25] and Goffin [5] show that an exponential number of iterations may be required. But here, unlike the ball method, we can use Newton's method to solve (P_{f+B}), thus obtaining a quadratic convergence rate; this follows from Theorem 3.1, since the Hessian of G is bounded in the compact level set.

We now state a Newton scaling algorithm for solving (P_{f+B}).

ALGORITHM 3.2.

- Initialization. Take $\tilde{d}^0 = e$ and scale it to d^0, i.e., $d^0 = \mu^0\tilde{d}^0$ with $\mu^0 = \sqrt{B(\tilde{d}^0)/f(\tilde{d}^0)}$; set $k = 0$.
- For $k = 0, 1, \ldots,$ do
  If a convergence condition holds, stop. Otherwise, perform a line search along the Newton direction, that is: take $\tilde{d}^{k+1} = d^k + \lambda^k d_{nt}$ where $d_{nt} := -(\nabla^2 G(d^k))^{-1} \cdot \nabla G(d^k)$, and $\lambda^k = \arg\min\{G(d^k + \lambda d_{nt}) : \lambda \geq 0\}$. Let $d^{k+1} = \mu^{k+1}\tilde{d}^{k+1}$ with $\mu^{k+1} = \sqrt{B(\tilde{d}^{k+1})/f(\tilde{d}^{k+1})}$. Set $k = k+1$ and repeat.

The following theorem describes the convergence properties of this algorithm.

THEOREM 3.6. *Suppose* (A1) *holds, that is,* $\text{int}(P) \neq \emptyset$, *and* $\{d^k\}$ *is the sequence of points generated by Algorithm* 3.2. *Then* d^k *converges to* d^* *q-quadratically*; *further, Algorithm* 3.2 *solves the feasibility problem* (1) *in finitely many iterations.*

Proof. Since Algorithm 3.2 is a descent method, by Theorem 3.1 we can restrict our attention to the compact set $\mathcal{D}_e$. Let

$$\begin{aligned}\lambda_u &:= \max\{\text{largest eigenvalue of } H(d) : d \in \mathcal{D}_e\},\\ \lambda_l &:= \min\{\text{smallest eigenvalue of } H(d) : d \in \mathcal{D}_e\},\end{aligned}$$

where $H(d)$ is the Hessian of G. Let H denote $H(d^k)$ and ∇G denote $\nabla G(d^k)$. Then for any $t > 0$ there is $d(\xi) := d^k + \xi d_{nt}$ with $0 < \xi < t$ such that

$$\begin{aligned} G(d^{k+1}) &\le G(\tilde{d}^{k+1}) \le G(d^k + t d_{nt}) \\ &= G(d^k) + t\nabla G(d^k)^T d_{nt} + \frac{t^2}{2} d_{nt}^T H(d(\xi)) d_{nt} \\ &\le G(d^k) + t\nabla G(d^k)^T d_{nt} + \frac{t^2}{2}\lambda_u \|d_{nt}\|^2 \\ &\le G(d^k) - t\nabla G(d^k)^T H^{-1} \nabla G(d^k)^T + \frac{t^2}{2}\lambda_u \|H^{-1}\nabla G\|^2 \\ &\le G(d^k) - t\lambda_u^{-1}\|\nabla G\|^2 + \frac{t^2}{2}\lambda_u \lambda_l^{-2}\|\nabla G\|^2 \\ &= G(d^k) - \sigma\|\nabla G\|^2 \quad (\text{for } t = \lambda_u^{-2}\lambda_l^2), \end{aligned} \tag{16}$$

where $\sigma := \frac{1}{2}\lambda_u^{-3}\lambda_l^2$. Thus d^k must converge to d^* otherwise $G(d^k) \longrightarrow -\infty$, contradicting (A1). On the other hand, from Theorem 3.1, there is $\bar{d} > 0$ such that

$$d_i \le \bar{d} \text{ for all } d \in \mathcal{D}_e \text{ and all } i.$$

If $\|\nabla G(d)\| \le \bar{d}^{-2}$, then, for $i = 1, 2, \ldots, m$,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2} \ge -\frac{1}{\bar{d}^2}.$$

Thus,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) \ge \frac{1}{d_i^2} - \frac{1}{\bar{d}^2} \ge 0,$$

which in turn shows that $l_i^k \le a_i^T x_c(d) \le u_i^k \ \forall i = 1, 2, \ldots, m$, and thus $x_c(d)$ is feasible for (FP). Therefore, by (16), $x_c(d^k)$ must be feasible to (FP) for $k \ge \frac{G(d^0)\bar{d}^4}{\sigma} - 1$. Hence, the algorithm solves the feasibility problem (1) in finitely many iterations.

Since for any k, $\tilde{d}^k$ is the result of one step of Newton's method, it is easy to see that there is a constant, say $\gamma > 0$, such that

$$\|\tilde{d}^{k+1} - d^*\| \le \gamma \|d^k - d^*\|^2. \tag{17}$$

On the other hand, it can be shown by Taylor's theorem that, for all $d \in \mathcal{D}_e$,

$$\frac{1}{2}\lambda_l \|d - d^*\|^2 \le G(d) - G(d^*) \le \frac{1}{2}\lambda_u \|d - d^*\|^2.$$

Thus,

$$\frac{\lambda_l}{2}\|d^{k+1} - d^*\|^2 \le G(d^{k+1}) - G(d^*) \le G(\tilde{d}^{k+1}) - G(d^*) \le \frac{\lambda_u}{2}\|\tilde{d}^{k+1} - d^*\|^2,$$

which implies

$$\|d^{k+1} - d^*\|^2 \le \frac{\lambda_u}{\lambda_l}\|\tilde{d}^{k+1} - d^*\|^2.$$

The theorem thus follows by (17). ∎

As in Model I, when Model II is used to solve the feasibility problem (1), there are three possible outcomes according to the values of $G_{\inf} := \inf\{G(d) : d \in R^m_{++}\}$.

THEOREM 3.7.

(i) *If* $G_{\inf} > 0$*, then there is a* $d^* \in R^m_{++}$ *such that* $G(d^*) = G_{\inf}$ *and* $x_c(d^*)$ *is feasible.*

(ii) *If* $G_{\inf} < 0$*, then* $P = \emptyset$*.*

(iii) *If* $G_{\inf} = 0$*, then* $\mathrm{int}(P) = \emptyset$ *but* $P \neq \emptyset$*.*

Proof. The proof of this theorem is similar to that of Theorem 2.17. □

The example in the last part of the previous section can be adapted here to show that $x_c(d^k)$ may not approach the feasible region even if d^k is such that $G(d^k) \longrightarrow 0$. We should mention that this phenomenon did not occur in our computational experiments; in particular, our computational results always gave a feasible solution for this example.

3.3. Computational results. Here we briefly describe some preliminary computational experiments we made with Algorithm 3.2. We generated 81 random problems, of sizes ranging from 40×45 to 200×250 for the matrix A, with various ways to choose l and u that guaranteed a feasible solution to (FP). The instances chosen used a Graeco-Latin square design so that we could make statistical tests from a relatively small number of observations. Solving these problems using MATLAB [16] on a Sun Sparcstation 2 required between 0 and 10 Newton iterations and between 0.2 and 390.6 seconds (much of the latter, for an 80×125 problem, arose from the line search, since evaluation of G requires the solution of a linear system). Details may be found in [13, 15].

Our statistical analyses (see [13]) show that (i) the number of iterations for solving the feasibility problems is very stable, almost independent of the sizes of problems; and (ii) the running time needed for solving the feasibility problems depends only on n and m, i.e., the size of the problem; it does not depend on the lower and upper bounds so much.

4. Solving linear programming problems via weighted centers. In the previous sections we introduced two kinds of centers. We discuss in this section how to use them to design algorithms for solving linear programming problems.

After defining a center, there are several possible ways to use it to solve linear programming problems. For example, we can cut the current polytope by a hyperplane through the current center and throw away the part that does not contain the optimal solution and repeat this procedure; we thus get a sequence of shrinking polytopes which contain the optimal solution, as do Levin [12] and Newman [18]. Alternatively, we can push the center toward the optimum by adjusting the appropriate bound as done by Renegar [19]. In §§4.1 and 4.2 we will use the same strategies to develop our algorithms via Models I and II, respectively. Since the center is usually a solution to some linear system, one natural way to solve the corresponding linear programming problem is to solve a linear system that combines the primal and dual in R^{m+n} and an inequality linking their objective functions. By duality theory, the solution of this system provides solutions of both the primal and the dual problems. However, from a practical viewpoint there are several disadvantages to this approach. First, the combined system is a higher-dimensional problem and thus needs more storage and more computational effort; next, since all solutions are within the hyperplane defined by letting the objectives of the primal and the dual be equal, the feasible region has zero volume, and thus some perturbation is necessary to ensure that the interior is not empty and this makes the computation rather difficult. In §4.3 we will propose a pulling technique so that, empirically, solving linear programming problems can be reduced to solving a linear system without the above disadvantages.

4.1. Sliding objective method via Model I. The sliding objective function method was first proposed by Yudin and Nemirovskii [30] and Shor [22]. The idea is to reduce the linear programming problem to a sequence of feasibility problems formed by letting the objective be an extra constraint and decreasing the bound corresponding to the objective function as long as it is possible.

Suppose we want to solve (LP). The sliding objective algorithm based on model I is as follows.

ALGORITHM 4.1.

- Initialization. Use Model I to find an approximate center x_c of the system:

$$l \leq A^T x \leq u, \tag{18}$$

  and let d be the corresponding weights. The corresponding ellipsoid is E^0. Let $u_0^0 := c^T x_c$ and $l_0^0 := c^T x_c - (f(d)(c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$, so that l_0^0 is tight for the current ellipsoid E^0. Set $A \leftarrow (c, A)$, i.e., let c be the 0th column, and $l^0 := (l_0^0, l^T)^T$, $u^0 := (u_0^0, u^T)^T$. Set $k = 0$.
- For $k = 0, 1, \ldots$, do
  Use Model I to solve system

$$l^k \leq A^T x \leq u^k \tag{FP$_k$}$$

  and get the (approximate) center x_c^k, weights d^k, and the current ellipsoid E^k. Let $u_0^{k+1} := c^T x_c^k$, $l_0^{k+1} := c^T x_c^k - (f(d^k)(c^T(AD^kA^T)^{-1}c)^{\frac{1}{2}}$, and $u^{k+1} := (u_0^{k+1}, u^T)^T$; $l^{k+1} := (l_0^{k+1}, l^T)^T$. Set $k \leftarrow k+1$, and repeat.

The convergence property of the above algorithm is similar to that of the standard sliding objective approach of the ellipsoid method. We refer to [2] and [6] for details.

We now discuss a practical stopping criterion. The dual of (LP) is

$$(DP) \qquad \begin{aligned} &\max\ l^T y_1 - u^T y_2 \\ &Ay_1 - Ay_2 = c, \\ &y_1,\ y_2 \geq 0. \end{aligned}$$

By the results of Burrell and Todd [3], (DP) is equivalent to

$$(\mathrm{DP}_1) \qquad \begin{aligned} \max\ \psi(y) &:= l^T y_+ - u^T y_- \\ Ay &= c, \end{aligned}$$

where $y_- := (\max\{0, -y_i\})$, $y_+ := (\max\{0, y_i\})$, so $y = y_+ - y_-$. By taking

$$y = (c^T(ADA^T)^{-1}c)^{\frac{1}{2}} D(A^T z - r),$$

where $z = x_c - (c^T(ADA^T)^{-1}c)^{-\frac{1}{2}}(ADA^T)^{-1}c$ and d is such that $f(d) = 1$, Burrell and Todd [3] show that

$$\psi(y) \geq c^T z = c^T x_c - (c^T(ADA^T)^{-1}c)^{\frac{1}{2}}.$$

We can use $\psi(y)$ as a lower bound on $c^T x$. Thus the duality gap associated with x_c^k and y is

$$gap(x_c^k, y) = c^T x_c^k - \psi(y) \leq (c^T(ADA^T)^{-1}c)^{\frac{1}{2}}.$$

We note that $(c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$ or, equivalently, $(f(d)c^T(ADA^T)^{-1}c)^{\frac{1}{2}}$ is half the width of the current ellipsoid along direction c. The above theorems ensure that this quantity goes to zero as the ellipsoids shrink. We thus can use it, or equivalently $f(d)c^T(ADA^T)^{-1}c$, as the stopping criterion.

4.2. Obtaining an ε-optimal solution via Model II. In this section we describe how to get an ε-optimal solution via Model II. Here we define an ε-optimal solution x^ε as a feasible solution with $c^T x^\varepsilon - z^* \le \varepsilon$. For convenience, we suppose that $\|a_i\| = 1$ for all $i = 1, 2, \ldots, m$ and $\|c\| = 1$. The idea is that we use the sliding objective function method via Model II and show that if the feasible region of the expanded system (with the objective as its 0th constraint) contains a ball $\mathcal{B}$ with radius ε, then the sliding method converges linearly. Thus within finitely many iterations an ε-optimal solution can be obtained. In the following we first state the algorithm, then prove some convergence results, and finally describe some relaxed versions. For convenience, we use H for the Hessian of G.

ALGORITHM 4.2.

- Initialization. Use Model II to find the (approximate) center x_c of the system:

$$l \le A^T x \le u, \tag{19}$$

and let d be the corresponding weights. The corresponding ellipsoid is E^0. Let $u_0^0 := c^T x_c$ and

$$l_0^{-1} := c^T x_c - (f(d) c^T (ADA^T)^{-1} c)^{\frac{1}{2}},$$
$$l_0^0 := l_0^{-1} - q,$$

where $q \ge \frac{3n^2(u_0^0 - l_0^{-1})}{4}$. Set $A \leftarrow (c, A)$, i.e., let c be the 0th column, and

$$l^0 := (l_0^0, l^T)^T, \; u^0 := (u_0^0, u^T)^T.$$

Set $k = 0$.

- For $k = 0, 1, \ldots,$ do
  (1) Use Algorithm 3.2 to solve system

$$l^k \le A^T x \le u^k, \tag{FP$_k$}$$

  i.e., find $d^k = \arg\min\{G^k(d) : d \in R_+^{m+1}\}$, where the quantities with superscripts are those corresponding to (FP$_k$). So $G^k(d)$ denotes $G(d)$ using l^k and u^k.

  (2) Let x_c^k be the center of the current ellipsoid E^k (= $E^{\mathrm{II}}(d^k)$ using l^k and u^k) and set $u_0^{k+1} := c^T x_c^k$ and $u^{k+1} := (u_0^{k+1}, u^T)^T$; and $l^{k+1} := (l_0^0, l^T)^T$. Set $k \leftarrow k + 1$, and repeat.

Note that after setting the lower bound for the 0th constraint at the initialization step the lower bound for system (FP$_k$) will remain the same and the distance between the optimal value and the 0th lower bound is at least q.

In the following we discuss the convergence properties of this algorithm. We assume that there is a ball, say $\mathcal{B}$, with radius δ_0 contained in P^k, the feasible region of system (FP$_k$). We first prove some lemmas about the subiterations in Step 1 in the algorithm.

LEMMA 4.1. *Suppose there is a ball $\mathcal{B}$ with radius δ_0 contained in P^k, and let*

$$\mathrm{Lev}_k := \{d \in R_+^{m+1} : G^k(d) \le G^k(e)\}$$

be the level set. Then, for any $d \in \mathrm{Lev}_k$,

$$\underline{d}^k := \frac{1}{G_0^k} \le d_i \le \bar{d}^k := \frac{nG_0^k}{\delta_0^2} \;\forall i = 0, 1, \ldots, m, \tag{20}$$

where $G_0^k := G^k(e) = (r^k)^T A^T (AA^T)^{-1} A r^k - (l^k)^T u^k + (m+1)$.

Proof. Since $B(d) \le G^k(d) \le G_0^k$ for all $d \in \mathrm{Lev}_k$, $\frac{1}{d_i} \le G_0^k$ for all i, which leads to the first inequality of (20). As for the second inequality we note that $2\sqrt{f^k(d) a_i^T (ADA^T)^{-1} a_i}$ is the width of ellipsoid $E(d)$ (not $E^{\mathrm{II}}(d)$) along a_i, so

$$\sqrt{f^k(d) a_i^T (ADA^T)^{-1} a_i} \ge \delta_0.$$

Thus,

$$f^k(d) a_i^T (ADA^T)^{-1} a_i \ge \delta_0^2. \tag{21}$$

From $\sum_{i=0}^m d_i a_i^T (ADA^T)^{-1} a_i = n$ and (21), we have, for any i,

$$d_i \le \frac{n}{a_i^T (ADA^T)^{-1} a_i} \le \frac{n f^k(d)}{\delta_0^2} \le \frac{n G_0^k}{\delta_0^2}.$$

Thus the proof is complete. □

Without the assumption of (FP_k) being consistent, the level set might be unbounded. This lemma provides us bounds for the level set in terms of the inscribing ball $\mathcal{B}$. The bounds $\underline{d}^k$ and $\bar{d}^k$ will play important roles in the convergence analysis below.

LEMMA 4.2. *Under the same assumptions as in Lemma* 4.1*, if there is* $d \in \mathrm{Lev}_k$ *with*

$$\|\nabla G^k(d)\| \le \frac{1}{(\bar{d}^k)^2}, \tag{22}$$

then $x_c(d)$ *is feasible for* (FP_k).

Proof. Note that $\nabla G_i^k(d) = -(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2}$ for $i = 0, 1, \ldots, m$. Now (22) implies $|\nabla G_i^k(d)| \le (\bar{d}^k)^{-2}$. Therefore,

$$-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) - \frac{1}{d_i^2} \ge -\frac{1}{(\bar{d}^k)^2},$$

whence $-(a_i^T x_c(d) - l_i^k)(a_i^T x_c(d) - u_i^k) \ge d_i^{-2} - (\bar{d}^k)^{-2} \ge 0$, which in turn shows that $l_i^k \le a_i^T x_c(d) \le u_i^k$, $i = 0, 1, \ldots, m$, and thus $x_c(d)$ is feasible for (FP_k). □

Now we show that $G^k(d^k)$ converges to zero linearly if there is a ball $\mathcal{B}$ with radius $\delta_\infty > 0$ contained in P^k for all k. Let

$$\underline{d}^\infty := \frac{1}{G_0^\infty}, \ \bar{d}^\infty := \frac{n G_0^\infty}{\delta_\infty^2},$$

where $G_0^\infty := \max\{G(e) : u_0 \in [z_0, \ u_0^0]\}$ with $z_0 := \max\{c^T x : x \in \mathcal{B}\}$; note that the function G depends on the bounds l and u—here we take the maximum as u_0 varies. It is obvious that

$$\underline{d}^\infty \le \underline{d}^k, \ \bar{d}^\infty \ge \bar{d}^k \quad \text{for all } k.$$

Thus, from Lemma 4.1, for all $d \in \mathrm{Lev}_\infty := \cup \mathrm{Lev}_k$,

$$\underline{d}^\infty \le d_i \le \bar{d}^\infty \ \forall i = 0, 1, \ldots, m, \tag{23}$$

and Lemma 4.2 still holds for these bounds. We define

$$\lambda_u := \max\{\text{largest eigenvalue of } H : u_0 \in [z_0,\ u_0^0], d \in \text{Lev}_\infty\},$$
$$\lambda_l := \min\{\text{smallest eigenvalue of } H : u_0 \in [z_0,\ u_0^0], d \in \text{Lev}_\infty\}.$$

Since $\underline{d}^\infty > 0$, $\bar{d}^\infty < \infty$, and $\underline{d}^\infty \le d_i \le \bar{d}^\infty \ \forall i$, both λ_u and λ_l are finite and positive.

We now suppose that the output of Step 1 of the algorithm d^k is such that

$$\|\nabla G^k(d^k)\| \le \varepsilon \quad \text{with } 0 < \varepsilon \le \frac{1}{2(\bar{d}^\infty)^2} < \frac{1}{(\bar{d}^\infty)^2}, \tag{24}$$

$f^k(d^k) = B(d^k)$, and $G^k(d^k) \le G^k(d^{k-1})$ (this holds if we take the initial point as $d^0 = e$ if $G^k(e) \le G^k(d^{k-1})$ and $d^0 = d^{k-1}$ otherwise).

Then from Lemma 4.2, $x_c(d^k)$ is feasible to P^k and for each i

$$(a_i^T x_c - l_i)(a_i^T x_c - u_i) \in \left(-\frac{1}{2(\bar{d}^\infty)^2} - \frac{1}{d_i^2},\ \frac{1}{2(\bar{d}^\infty)^2} - \frac{1}{d_i^2}\right). \tag{25}$$

THEOREM 4.3. *There is a number $\delta = \delta(m, n, \underline{d}^\infty, \bar{d}^\infty)$ with $0 < \delta < 1$ such that*

$$\frac{f^{k+1}(d^{k+1}) \cdot B^{k+1}(d^{k+1})}{f^k(d^k) \cdot B^k(d^k)} \le \delta.$$

Proof. By (23), we have, for all $d \in \text{Lev}_\infty$,

$$d_0 \cdot B(d) = d_0 \cdot e^T d^{-1} = 1 + \frac{d_0}{d_1} + \cdots + \frac{d_0}{d_m} \le 1 + \frac{m\bar{d}^\infty}{\underline{d}^\infty} =: \bar{\delta}.$$

On the other hand, if we let d be d^k,

$$\begin{aligned}
\frac{1}{2}(u_0^k - c^T x_c^k)^2 d_0^3 c^T (ADA^T)^{-1} c &\le \frac{1}{2}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k}\left(\frac{1}{2(\bar{d}^\infty)^2} + \frac{1}{d_0^2}\right) d_0^3 c^T (ADA^T)^{-1} c \\
&\quad \text{(using (25))} \\
&\le \frac{n}{2}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k}\left(\frac{d_0^2}{2(\bar{d}^\infty)^2} + 1\right) \\
&\quad \text{(using } d_0 c^T (ADA^T)^{-1} c \le n) \\
&\le \frac{3n}{4}\frac{u_0^k - c^T x_c^k}{c^T x_c^k - l_0^k} \\
&\le \frac{3n}{4}\frac{u_0^0 - l_0^{-1}}{q} \le \frac{1}{n}.
\end{aligned}$$

Thus, letting $\Delta u := u_0^k - c^T x_c^k$ and f^{k+1} be the function f with $u = u^{k+1} = u^k - \Delta u \cdot e_0$,

$$\begin{aligned}
f^{k+1}(d) &= (r^k)^T DA^T (ADA^T)^{-1} ADr^k - (l^k)^T Du^k - \Delta u d_0 c^T x_c^k \\
&\quad + \Delta u d_0 l_0^k + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c \\
&= f^k(d) + d_0 \Delta u (l_0^k - c^T x_c^k) + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c \\
&= f^k(d) + d_0 (u_0^k - c^T x_c^k)(l_0^k - c^T x_c^k) + \frac{1}{4}(\Delta u)^2 d_0^2 c^T (ADA^T)^{-1} c
\end{aligned}$$

$$\leq f^k(d) - \frac{1}{d_0}\left(1 - \frac{d_0^2}{2(\bar{d}^\infty)^2}\right) + \frac{1}{4}(u_0^k - c^T x_c^k)^2 d_0^2 c^T (ADA^T)^{-1} c \text{ (using (25))}$$
$$\leq f^k(d) - \frac{1}{2d_0}\left(1 - \frac{1}{2}(u_0^k - c^T x_c^k)^2 d_0^3 c^T (ADA^T)^{-1} c\right)$$
$$\leq f^k(d) - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right).$$

Suppose the $(k+1)$st output of Step 1 is d^{k+1}. Then

$$2\sqrt{(f^{k+1}(d^{k+1})B(d^{k+1})} = f^{k+1}(d^{k+1}) + B(d^{k+1}) \leq f^{k+1}(d) + B(d)$$
$$\leq f^k(d) + B(d) - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right) = 2\sqrt{f^k(d)B(d)} - \frac{1}{2d_0}\left(1 - \frac{1}{n}\right).$$

Thus,

$$\frac{f^{k+1}(d^{k+1})B(d^{k+1})}{f^k(d)B(d)} \leq \left(1 - \frac{n-1}{4n}\frac{1}{d_0 B(d)}\right)^2 \leq \left(1 - \frac{n-1}{4n\bar{\delta}}\right)^2.$$

Letting $\delta := (1 - \frac{n-1}{4n\bar{\delta}})^2$, the theorem thus follows. □

In the following we consider how to use Model II to get a d^k satisfying (24) in Step 1. For this purpose we let k' denote the iteration index within Step 1.

LEMMA 4.4. *Suppose d^* is the optimal solution of Model II for G^k. Then for all $d \in \mathrm{Lev}_\infty$,*

$$\frac{\lambda_l}{2}\|d - d^*\|^2 \leq G^k(d) - G^k(d^*) \leq \frac{\lambda_u}{2}\|d - d^*\|^2.$$

Proof. It is easy to prove this lemma by using Taylor's theorem. □

From (16), we have, where G denotes G^k,

$$G(d^{k'+1}) \leq G(d^{k'}) - \sigma\|\nabla G(d^{k'})\|^2,$$

where $\sigma := \frac{1}{2}\lambda_u^{-3}\lambda_l^2$. From the above result, after at most $\frac{G(d^0)}{\sigma\varepsilon} - 1$ iterations, we have $\|\nabla G(d^{k'})\| \leq \varepsilon$.

Therefore, to get a point satisfying (24) in Step 1, we only need to take

$$\varepsilon = \frac{1}{2(\bar{d}^\infty)^2} \tag{26}$$

and then use Algorithm 3.2 to get a d with $\|\nabla G(d)\| \leq \varepsilon$ (this can be done in finitely many iterations as shown above). The above results also give us a relaxed version of Algorithm 4.2 (we suppose that both G_0^∞ and δ_∞ are known): in Step 1 we need only find an approximate solution d^k to (FP_k) such that $\|\nabla G^k(d^k)\| \leq \varepsilon$ with ε defined by (26) and then scale d^k so that $f(d^k) = B(d^k)$.

THEOREM 4.5. *Under assumption* (A1)*, Algorithm* 4.2 *(or its relaxed version) provides an ε-optimal solution of* (LP) *in finitely many iterations.*

Proof. Suppose $\{x_c^k\}$ is a sequence of centers generated by Algorithm 4.2 (or its relaxed version) and suppose that for all $k = 0, 1, \ldots,$

$$|c^T x_c^k - c^T x^*| > \varepsilon.$$

Then, there is a ball $\mathcal{B}$ with some positive number, say δ_∞, as its radius contained in P^k for all k. From Theorem 4.3, $\frac{v^{k+1}}{v^k} \leq \delta$, i.e.,

$$\frac{\text{volume}(E^{k+1})}{\text{volume}(E^k)} \leq \delta^{\frac{n}{2}},$$

for some $\delta < 1$. Since $\mathcal{B} \subset P^k \subset E^k$ for all k, we thus have

$$0 < \kappa_n \cdot \delta_\infty^n = \text{volume}(\mathcal{B}) \leq \text{volume}(E^k) \longrightarrow 0$$

as $k \longrightarrow \infty$, which is a contradiction. Thus, noting that the number of subiterations within Step 1 is finite, Algorithm 4.2 (or its relaxed version) provides an ε-optimal solution of (LP) in finitely many iterations. $\square$

The above results can be summarized as follows: if the algorithm does not converge, then a linear convergence rate for the volume of the corresponding ellipsoid can be proved; thus, the algorithm does converge and the rate of shrinking of the volume of the ellipsoid is reduced as x_c^k approaches x^*. It can be shown, by Lemma 4.4, that if $G^k \longrightarrow 0$ linearly, $x_c^k \longrightarrow x^*$ linearly. The linear convergence rate is most likely the best. An extreme example is when $m + 1 = n$ and, after putting in the objective vector, the system becomes an $n \times n$ square system; thus, $x_c^k = A^{-1}r^k$ and $x_c^k \longrightarrow x^*$ at most linearly.

4.3. Computational techniques and numerical results. In this section, we report preliminary numerical results with Algorithm 4.2 when applied to problem (LP).

For given $d \in R_+^{m+1}$, consider $x_c(d)$ as a function of l_0, the lower bound imposed on the objective function $c^T x$. Then we have

$$c^T x_c(d) \longrightarrow -\infty \text{ as } l_0 \longrightarrow -\infty.$$

This suggests that if l_0 is chosen to be a very large negative number, finding an approximate solution of (LP) might need only one iteration, i.e., be reduced to a feasibility problem (FP). We call this technique the pulling technique, since it pulls the current center to the optimal solution. Note that most bound-update methods, e.g., Levin [12], Newman [18], and Renegar [19], decrease an appropriate bound so as to push the current center to approach the optimal solution. Our Algorithm 4.2 is a combination of these techniques, since we choose l_0^0 to be a very large negative number (pulling) but then update u_0^k (pushing). Our limited test results are reported in Table 1. The test problems we use are those in Avis and Chvatal [1] but with upper bounds on each component of x, namely,

$$\begin{array}{l} \max\ e^T x \\ Nx \leq 10^4 e, \\ 0 \leq x \leq 10e. \end{array} \tag{27}$$

N is an $n \times n$ matrix with integer elements chosen randomly in the range $1, \ldots, 1000$. We reduce (27) to our form (LP):

$$\begin{array}{l} \max\ e^T x \\ l \leq \tilde{N}x \leq u \end{array} \tag{28}$$

with

$$l = 0,\ \tilde{N} = \begin{pmatrix} N \\ I \end{pmatrix},\ u = (10^4 e^T, 10e^T)^T.$$

TABLE 1
Computational results of Algorithm 4.2.

Dimension	Simplex	Algorithm 4.2
10	iter=12	iter=3 (9+40+6), $\Delta = 2.9 \times 10^{-4}$
20	iter=30	iter=3 (9+58+5), $\Delta = 9.1 \times 10^{-4}$
30	iter=37	iter=4 (11+71+7+11), $\Delta = 6.6 \times 10^{-6}$
40	iter=85	iter=3 (16+152+12), $\Delta = 5.4 \times 10^{-4}$
50	iter=128	iter=4 (9+39+16+7), $\Delta = 6.9 \times 10^{-4}$

TABLE 2
The effect of q on the number of iterations.

	Dimension=30	Dimension=40	Dimension=50
$q = 10$	> 35	> 35	> 35
$q = 10^2$	> 35	> 35	> 35
$q = 10^3$	$> 35(> 114)$	$> 35(> 121)$	> 35
$q = 10^4$	$27(11 + 8 + 4 \cdot 5 + 21 \cdot 4 = 87)$	$18(16 + 13 + 2 \cdot 5 + 7 \cdot 4 + 7 \cdot 3 = 88)$	$> 35(> 134)$
$q = 10^5$	$9(11 + 16 + 8 + 17 + 5 \cdot 6 = 82)$	$6(16 + 17 + 13 + 12 + 6 + 6 = 70)$	$11(9 + 16 + 7 + 9 + 2 \cdot 6 + 5 \cdot 5 = 102)$
$q = 10^6$	$5(11 + 27 + 12 + 7 + 9 = 66)$	$4(16 + 50 + 8 + 7 = 81)$	$6(9 + 20 + 8 + 7 + 8 + 8 = 60)$
$q = 10^7$	$4(11 + 43 + 8 + 12 = 74)$	$3(16 + 76 + 12 = 104)$	$4(9 + 38 + 15 + 8 = 70)$
$q = 10^8$	$4(11 + 71 + 7 + 11 = 100)$	$3(16 + 152 + 12 = 180)$	$4(9 + 71 + 14 + 11 = 105)$

We choose $q = 10^8$. The data in the column corresponding to the simplex method are from Nazareth [17]; these are average numbers of iterations for a number of random problems. $\Delta := f(d)c^T(ADA^T)^{-1}c$ is the width of the ellipsoid in the direction c after the final iteration and it serves here as the stopping criterion (we terminate if $\Delta \leq 10^{-3}$). The stopping criterion for Model II is $\|\nabla G(d)\| \leq 10^{-5}$. The numbers in the parentheses are the numbers of iterations of Algorithm 3.2 for obtaining each approximate center. The solution times for the last three problems of dimensions 30, 40, and 50 were 43, 146, and 124 seconds, respectively. Table 2 shows the effects of q on the number of iterations. It also shows that the effect of pulling is much more significant than that of pushing. Thus, choosing an appropriate q will save computational effort. Again, the runs were performed using MATLAB on a Sun Sparcstation 2.

5. Summary and conclusions. In this paper we have analyzed and tested some generalized models for linear inequality systems and then proposed some algorithms for solving linear programming problems via these models. Foremost among these is the idea of generalizing Burrell and Todd's approach [3] which is closely related to the coordinate descent method so that Newton's method can be employed. We develop our models in §§2 and 3. The key result is that the function $v(d) = f(d)h(d)$, where v is such that $\kappa_n v^{\frac{n}{2}}$ is the volume of the ellipsoid, can be, equivalently for our purposes, replaced by $F(d) = f(d) + h(d)$ because $f(d)$ and $h(d)$ are homogeneous functions of degrees 1 and -1, respectively. The first model, Model I, is thus established by using $F(d) = f(d) + h(d)$ as the objective function. The advantage of this model is that it is a convex program and it preserves the advantages of Burrell and Todd's approach [3]. We prove the existence and uniqueness of the smallest ellipsoid of the form (3) which contains the feasible region P. The center of this smallest ellipsoid is proved to be an interior point of P; it is our first center. For solving Model I we propose a coordinate descent method obtained by modifying the Liao and Todd algorithm [14]. It is proved to have a polynomial-time bound. Since the objective function of Model I might not be strictly convex, it thus prevents the use of a full version of Newton's method, but a partial Newton step algorithm is described in [13]. To overcome this disadvantage we replace $h(d)$ by a strictly convex function $B(d)$

which forms our second model, Model II. Similarly, we propose two algorithms for solving this model: a coordinate descent algorithm and a Newton algorithm. The first one is actually a ball-like method [2] which has been shown not to be a polynomial-time algorithm. The second algorithm is a combination of Newton's algorithm and a scaling technique, which works well in practice. The statistical analysis shows that this algorithm is numerically robust. We also provide a rather detailed analysis of the outputs of these models according to their minimal values over their effective domains.

We note that while line search techniques are used in our algorithms, trust region methods can also be employed. Since our models are convex, these two techniques can be expected to perform similarly.

In §4 we develop methods for solving linear programming problems via these models. They are basically sliding objective function methods. The first one is based on Model I and enjoys geometrical convergence rates as measured by the volume of the corresponding ellipsoid and by the objective value of the linear program, respectively. The second algorithm, based on Model II, provides an ε-optimal solution of (LP) in finitely many iterations. One property of this approach is that if $c^T x_c$, the current objective value, is ε far away from z^*, the optimal value, a constant decrease (depending on ε) can be obtained. In other words, if x_c does not approach the optimal set, x_c approaches the optimal set linearly. In addition to the pushing technique similar to that used in the sliding objective function method [2] and Renegar [19], we also propose a pulling technique: we let the lower bound corresponding to the objective be a very large negative number, i.e., $l_0 \sim -\infty$ (or $u_0 \sim \infty$ for a maximization linear program). With the help of this pulling technique, solving (LP) is almost equivalent to solving a feasibility problem as shown by the numerical tests.

Finally, we comment on the major unfinished tasks that we believe are evident from this paper. The biggest one is, as mentioned above, to analyze our approach to see if Newton's algorithm is polynomial for linear programming problems within our framework. We list below some suggestions for further work in this direction:

(1) We showed that if $f(d) < 0$ for some $d \geq 0$, then the system (FP) is inconsistent. Further research might investigate the number of iterations needed to find the infeasibility if the system is not consistent.

(2) We gave artificial examples showing that, in the case that $\text{int}(P) = \emptyset$ but $P \neq \emptyset$, for both Model I and Model II, there exist sequences $\{x_c(d^k)\}$ which fail to approach the feasible region although d^k is such that $F(d^k)$ (or $G(d^k)$) approaches zero. But Algorithm 3.2 always gave feasible solutions in our computational tests. Investigation is thus needed for this kind of behavior of the algorithm.

(3) From the numerical tests, we find that the pulling technique is an important factor with regard to the number of major iterations, i.e., number of applications of Model II. The bigger q is, the fewer major iterations are needed. We thus hope to prove that solving (LP) is equivalent to finding the center of (FP_0) with a large q.

Acknowledgments. We would like to thank Yinyu Ye and two anonymous referees for very helpful comments.

REFERENCES

[1] D. Avis and V. Chvatal, *Notes on Bland's pivoting rule*, Math. Programming Study, 8 (1978), pp. 24–34.

[2] R. G. Bland, D. Goldfarb, and M. J. Todd, *The ellipsoid method: A survey*, Oper. Res., 29 (1981), pp. 1039–1091.
[3] B. P. Burrell and M. J. Todd, *The ellipsoid method generates dual variables*, Math. Oper. Res., 10 (1985), pp. 688–700.
[4] V. Chvátal, *Linear Programming*, W. H. Freeman and Company, San Francisco, CA, 1980.
[5] J. L. Goffin, *On the non-polynomiality of the relaxation method for systems of linear inequalities*, Math. Programming, 22 (1982), pp. 93–103.
[6] D. Goldfarb and M. J. Todd, *Modifications and implementation of the ellipsoid algorithm for linear programming*, Math. Programming, 23 (1982), pp. 1–19.
[7] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1990.
[8] F. John, *Extremum problems with inequalities as subsidiary conditions*, in Studies and Essays (Courant Anniversary Volume), Interscience, New York, 1948.
[9] N. K. Karmarkar, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.
[10] L. G. Khachiyan, *A polynomial algorithm for linear programming*, Dokl. Akad. Nauk USSR, 244 (1979), pp. 1093–1096.
[11] L. G. Khachiyan and M. J. Todd, *On the complexity of approximating the maximal inscribed ellipsoid for a polytope*, Math. Programming, 61 (1993), pp. 137–159.
[12] A. Y. Levin, *On an algorithm for the minimization of convex functions*, Soviet Math. Dokl., 6 (1965), pp. 286–290.
[13] A. Liao, *Algorithms for Linear Programming via Weighted Centers*, Ph.D. thesis, Cornell University, Ithaca, NY, 1992.
[14] A. Liao and M. J. Todd, *The ellipsoid algorithm using parallel cuts*, Comput. Optim. Appl., 2 (1993), pp. 299–316.
[15] ———, *Solving LP Problems via Weighted Centers*, CCOP Report 93-10, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1993.
[16] C. B. Moler, J. Little, S. Bangert, and S. Kleiman, *Pro-Matlab User's Guide*, The MathWorks, Inc., 1987.
[17] J. L. Nazareth, *Pricing criteria in linear programming*, in Progress in Mathematical Programming, N. Meggiddo, ed., Springer-Verlag, New York, 1989, pp. 105–130.
[18] D. J. Newman, *Location of the maximum on unimodal surfaces*, J. Assoc. Comput. Mach., 12 (1965), pp. 395–398.
[19] J. Renegar, *A polynomial-time algorithm based on Newton's method for linear programming*, Math. Programming, 40 (1988), pp. 59–93.
[20] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[21] N. Z. Shor, *Utilization of the operation of space dilatation in the minimization of convex functions*, Cybernetics, 6 (1970), pp. 7–15.
[22] ———, *Cut-off method with space extension in convex programming problems*, Cybernetics, 13 (1977), pp. 94–96.
[23] G. Sonnevend, *An analytical center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming*, in Lecture Notes in Control and Information Sciences, No. 84, Springer-Verlag, Berlin, 1986, pp. 866–875.
[24] S. P. Tarasov, L. G. Khachiyan, and I. I. Erlich, *The method of inscribed ellipsoids*, Sov. Math. Dokl., 37 (1988), pp. 226–230.
[25] M. J. Todd, *Some Remarks on the Relaxation Method for Linear Inequalities*, Technical Report 419, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1979.
[26] ———, *On minimum volume ellipsoids containing part of a given ellipsoid*, Math. Oper. Res., 7 (1980), pp. 253–261.
[27] ———, *Improved bounds and containing ellipsoids in Karmarkar's linear programming algorithm*, Math. Oper. Res., 13 (1988), pp. 650–659.
[28] P. M. Vaidya, *A new algorithm for minimizing convex functions over convex sets*, Math. Programming, 73 (1996), pp. 291–341.
[29] Y. Ye, *Karmarkar's algorithm and the ellipsoid method*, Oper. Res. Lett., 6 (1987), pp. 177–182.
[30] D. B. Yudin and A. S. Nemirovskii, *Informational complexity and efficient methods for the solution of convex extremal problems*, Matekon, 13 (1976), pp. 3–25.