The Educational Testing Problem Revisited

Moody T. Chu [1] and Joel W. Wright
Department of Mathematics
North Carolina State University
Raleigh, North Carolina 27695-8205.

September 21, 1995

[1] This research was supported in part by National Science Foundation under grant DMS-9123448.

Abstract

The educational testing problem is a convex non-smooth optimization problem. We recast the problem so that classical non-smooth optimization techniques such as the ellipsoid method can readily be applicable. Attention is paid to the Dikin's method where a special barrier function and interior ellipsoids for the feasible domain are explicitly formulated. The implementation is much easier than that in [8]. The convergence property is numerically demonstrated.

1. Introduction.

The educational testing problem (ETP) is a nonlinear programming problem which arises in statistics [7]. The problem is to determine how much can be subtracted from the diagonal of a given symmetric and positive definite matrix S such that the resulting matrix is positive semi-definite. The (ETP) can be formulated as follows:

$$\text{Maximize} \quad \operatorname{trace}(D), \tag{1}$$

$$\text{subject to} \quad S - D \geq 0, \; D \geq 0 \tag{2}$$

where $D = \operatorname{Diag}\{d_1, \ldots, d_n\}$ denotes a diagonal matrix and $M \geq 0$ means that the matrix M is positive semi-definite. It is easy to see that the (ETP) is a convex programming problem. A local solution that is also a global solution to the (ETP) always exists.

For convenience we shall denote henceforth the column vector formed from the diagonal entries of a matrix M by $\operatorname{diag}(M)$, and the diagonal matrix with diagonal entries from a column vector d by $D = \operatorname{Diag}(d)$. Quite often, the distinction between D and d is immaterial.

The structure of the feasible domain (2), including expressions for the normal cone, feasible directions and optimality conditions, has been carefully studied by Fletcher [8]. Based on this framework, Fletcher has proposed a quadratically convergent algorithm which involves solving a sequence of subproblems, each defined by a guess of the nullity of $S - D$ and an exact penalty function. Fletcher's key idea was to replace the constraint (2) by a set of nonlinear algebraic equations [8, Formula (3.3)].

In this paper we discuss how the (ETP) can be tackled differently. In particular, we discuss how the constraint (2) can be realized more easily. We suggest two channels of attack. Both are easy to be implemented and make many of the computational concerns involved in Fletcher's method [8] less significant. Furthermore, both of our approaches are globally convergent.

Our first approach is to directly reformulate the (ETP) into two new but mathematically equivalent convex programming problems. The reformulation is quite straightforward, but the constraint becomes more manageable. The advantage is that many standard methods, the ellipsoid method in particular, are immediately applicable.

Our second approach is to approximate the boundary of the feasible domain by level curves of a special barrier function. The (ETP), therefore, is approximated by a sequence of subproblems where linear objective functions are to be optimized over ellipsoids. The advantage is that the solution to each subproblem is readily obtainable.

It should be mentioned that recently Glunt has proposed another approach to the (ETP) on the basis of an alternating projection method [11]. A major component in Glunt's method is the use of Dkystra's algorithm [5] for computing projections onto the intersection of convex sets. It can be proved that Glunt's method converges globally at linear rate.

Discussion on the ellipsoid method is fairly rich in the literature. Far from being complete, we simply mention references [2, 3, 12, 14, 21]. The application of this method to the (ETP) is demonstrated in Section 2. Although the ellipsoid method is known to converge eventually, the iterates (the centers) quite often are unfeasible, and the so called *constraint iteration* has to take place to correct the points back to the feasible domain. In contrast, the Dikin's method is a variation of the interior point method. That is, all the iterates and the ellipsoids generated are interior to the feasible domain. These features are discussed in Section 3. Numerical experiments with comparison to existing results are presented in Section 4.

2. First Approach.

2.1. Reformulation.

We reformulate the (ETP) by taking into account the eigenvalues. We discuss two reformulations.

First, by the inertia theorem, $\lambda S - D$ is positive definite if and only if $\lambda I - S^{-1/2} D S^{-1/2}$ is positive definite. So for fixed $D \geq 0$ the smallest λ that makes $\lambda S - D$ positive semi-definite is the largest eigenvalue of $S^{-1/2} D S^{-1/2}$. For any symmetric matrix M, let $\lambda_1(M)$ denote the largest eigenvalue of M. Define

$$\mu(D) := \lambda_1(S^{-1/2} D S^{-1/2}). \tag{3}$$

The diagonal matrix $\frac{D}{\mu(D)}$ is invariant under scalar multiplication. So the (ETP) can be formulated as

$$\begin{aligned} &\text{Minimize} && \mu(D), && (4)\\ &\text{subject to} && \operatorname{trace}(D) = 1,\ -D \leq 0. && (5) \end{aligned}$$

We note from (3) that $\mu(D)$ is the composition of the convex function λ_1 and the linear function $S^{-1/2} D S^{-1/2}$, and hence is still convex. The equality constraint in (5) can easily be removed by defining, for example, $d_n = 1 - \sum_{i=1}^{n-1} d_i$. The (ETP) is equivalent to

$$\begin{aligned} &\text{Minimize} && \tilde{\mu}(d_1, \ldots, d_{n-1}), && (6)\\ &\text{subject to} && -d_i \leq 0,\ \sum_{i=1}^{n-1} d_i - 1 \leq 0 && (7) \end{aligned}$$

where

$$\tilde{\mu}(d_1, \ldots, d_{n-1}) := \mu\left(d_1, \ldots, d_{n-1}, 1 - \sum_{i=1}^{n-1} d_i\right). \tag{8}$$

It is worth mentioning that one may replace the square root matrix $S^{1/2}$ in the above discussion by the Cholesky factor L of S and form a similar problem.

Our second reformulation comes from the observation that $S - D \geq 0$ if and only if

$$\nu(D) := \lambda_1(D - S) \leq 0. \tag{9}$$

Thus the (ETP) may also be expressed as

$$\begin{aligned} &\text{Minimize} && -\operatorname{trace}(D), && (10)\\ &\text{subject to} && \nu(D) \leq 0,\ -D \leq 0. && (11) \end{aligned}$$

Once again, we note that $\nu(D)$ is a convex function.

Both reformulations involve some eigenvalue inequalities.

2.2. Subgradient.

A particular difficulty associated with eigenvalue optimization problems is that the eigenvalues of a differentiable matrix function are not themselves differentiable at points where they coalesce. Furthermore, it has been observed quite so often that at an optimal solution the eigenvalues coalesce [19]. To overcome this difficulty we can employ special techniques developed in, for example, [18, 19]. For convex programming problems, however, there are simple and effective algorithms that do not require smooth constraints or differentiable objectives. For the above (ETP) in particular, the notion of subdifferential is easy to be implemented.

Given a convex function $f : R^n \longrightarrow R$, any vector $g \in R^n$ such that

$$f(z) \geq f(x) + g^T(z-x) \text{ for all } z \tag{12}$$

is called a *subgradient* of f at x. A basic result of convex analysis is that every convex function always has at least one subgradient at every point. The notion of subgradients has an important implication. That is,

$$f(z) > f(x) \text{ whenever } g^T z > g^T x. \tag{13}$$

Thus if we want to reduce the values of f and if we know a subgradient g of f at x, then we only need to consider variables in the half-space

$$\mathcal{H}(x,g) := \left\{ z \in R^n | g^T(z-x) \leq 0 \right\}. \tag{14}$$

The difficulty associated with the constraint (2) is that it is not clear how to express the positive semi-definite constraints explicitly with m smooth and convex inequalities $\psi_i(D) \leq 0$ where m is small. Discussion for this class of constraints can be found, for example, in [1, 4, 8, 11, 19]. One naive way of representing (2) is that all its principal minors are non-negative. Such an expression, however, is very expensive. Fletcher has tried to depict the normal cone [8, Formula (4.4)] by approximated algebraic conditions. The implementation then involves some tailored SQP techniques. In contrast, by reformulating the problem the subgradients of either $\mu(D)$ or $\nu(D)$ are very easy to compute as will be illustrated below.

Let $A(D)$ denote either the matrix $S^{-1/2}DS^{-1/2}$ or the matrix $D-S$. The partial derivatives

$$A_k(D) := \frac{\partial A(D)}{\partial d_k} \tag{15}$$

are trivial to compute. The following result has been proved in [19, Theorem 3].

THEOREM 2.1. *Suppose $\lambda_1(A(D))$ has multiplicity t. Let columns of $Q_1(D) := [q_1(D), \ldots, q_t(D)]$ be a corresponding orthonormal basis of eigenvectors. Then the subgradients of $\lambda_1(A(D))$ form the set*

$$\begin{aligned} \partial\lambda_1(A(D)) = \{g \in R^n \mid \quad g_k = \quad & \langle U, Q_1(D)^T A_k(D) Q_1(D)\rangle, \\ & \textit{for some } U \in R^{t\times t}, U = U^T, U \geq 0, \mathrm{trace(U)} = 1 \right\}. \end{aligned} \tag{16}$$

In particular, regardless of the multiplicity, we have

COROLLARY 2.2. *Let $q(D)$ be any normalized eigenvector of $\lambda_1(A(D))$. Then the vector $g = [g_1, \ldots, g_n]^T$ with*

$$g_k := q(D)^T A_k(D) q(D) \tag{17}$$

is a subgradient of $\lambda_1(A(D))$.

Without causing any ambiguity, we use the same notation $\partial f(x)$ to denote any subgradient of f at x. Let $q(D)$ denote a normalized eigenvector of the corresponding $A(D)$, we have

$$\partial\mu(D) = \left(S^{-1/2}q(D)\right)\circ\left(S^{-1/2}q(D)\right), \tag{18}$$

$$\partial\nu(D) = q(D)\circ q(D), \tag{19}$$

$$\partial\tilde{\mu}(D) = T\partial\mu(D) \tag{20}$$

where $\circ$ denotes the Hadamard product and T is the $(n-1)\times n$ constant matrix

$$T := \begin{bmatrix} 1 & 0 & \dots & 0 & -1 \\ 0 & 1 & & & -1 \\ \vdots & & \ddots & & \vdots \\ 0 & & & 1 & -1 \end{bmatrix}.$$

2.3. Ellipsoid Method.

We briefly describe the ellipsoid method for a general convex programming problem

$$\text{Minimize} \quad \phi(x) \tag{21}$$

$$\text{subject to} \quad \psi(x)\le 0. \tag{22}$$

The method was first proposed by Shor [21] and is best known for being adapted by Khachiyan [14] to prove the polynomial time solvability of linear programming problem. More details can be found, for example, in [2, 12].

An ellipsoid $E\subset R^n$ can best be characterized by a vector $a\in R^n$ and a symmetric and positive definite matrix $B\in R^{n\times n}$ in such a way that

$$E = E(B,a) := \left\{x\in R^n | (x-a)^T B^{-1}(x-a)\le 1\right\}. \tag{23}$$

The ellipsoid method generates a sequence of ellipsoids $\{E^{(k)} = E(B^{(k)}, x^{(k)})\}$ with decreasing volumes such that

1. $E^{(1)}$ contains the feasible minimizer x^*.
2. $E^{(k+1)}$ is the ellipsoid of minimum volume (the Löwner-John ellipsoid) that contains the half-sliced ellipsoid $E^{(k)}\cap\mathcal{H}(x^{(k)}, g^{(k)})$ where

$$g^{(k)} := \begin{cases} \partial\phi(x^{(k)}), & \text{if } \psi(x^{(k)})\le 0; \\ \partial\psi(x^{(k)}), & \text{if } \psi(x^{(k)})>0. \end{cases} \tag{24}$$

The idea in (24) is to throw away points that are not helpful in determining the minimizer x^*. So using the property (13) of subgradients, if $x^{(k)}$ is feasible then we discard all points where objective values are greater than or equal to $\phi(x^{(k)})$, and if $x^{(k)}$ is not feasible then we discard all points which are further guaranteed to be infeasible.

It turns out that $B^{(k+1)}$ and $x^{(k+1)}$ can be explicitly described in terms of $B^{(k)}$, $x^{(k)}$ and $g^{(k)}$. See, for example, [12, Formulas (3.1.11-12)] or [2, Appendix B]. Thus a basic ellipsoid algorithm for problem (21) and (22) can be summarized as follows:

ALGORITHM 2.1. *(Basic Ellipsoid Method)*

Given $B^{(1)}$ *and* $x^{(1)}$ *so that* $E^{(1)}$ *contains a feasible minimizer, do:*

Compute $\psi(x^{(k)})$,

If $\psi(x^{(k)}) > 0$,

Compute any $g^{(k)} \in \partial\psi(x^{(k)})$;

$\gamma := \sqrt{g^{(k)^T} B^{(k)} g^{(k)}}$;

$g := \frac{g^{(k)}}{\gamma}$;

If $\psi(x^{(k)}) - \gamma > 0$, *quit.*

Else,

Compute any $g^{(k)} \in \partial\phi(x^{(k)})$;

$\gamma := \sqrt{g^{(k)^T} B^{(k)} g^{(k)}}$;

$g := \frac{g^{(k)}}{\gamma}$;

$b := B^{(k)} g$;

$x^{(k+1)} := x^{(k)} - \frac{b}{n+1}$;

$B^{(k+1)} := \frac{n^2}{n^2-1}\left(B^{(k)} - \frac{2}{n+1} bb^T\right)$;

If $\psi(x^{(k)}) \le 0$ *and* $\gamma \le \epsilon$, *stop.*

A nice feature of the ellipsoid method is that

$$\frac{\mathrm{vol}(E^{(k+1)})}{\mathrm{vol}(E^{(k)})} = \left(\left(\frac{n}{n+1}\right)^{n+1}\left(\frac{n}{n-1}\right)^{n-1}\right)^{\frac{1}{2}} < e^{\frac{-1}{2n}}. \tag{25}$$

Thus the ellipsoid method always converges (but slowly). It should be noted, however, that in finite precision arithmetic roundoff error will almost invariably cause the computed matrix $B^{(k)}$ to become indefinite. Consequently, the quantity γ may not be a real number. Fortunately, this numerical unstability can be remedied by updating, instead of $B^{(k)}$, the factor of $B^{(k)} = J^{(k)} J^{(k)^T}$. In this way, the square root is avoided. The modified algorithm is as follows:

ALGORITHM 2.2. *(Modified Ellipsoid Method)*

Given $B^{(1)} = J^{(1)} J^{(1)^T}$ *and* $x^{(1)}$ *so that* $E^{(1)}$ *contains a feasible minimizer, do:*

Compute $\psi(x^{(k)})$,

If $\psi(x^{(k)}) > 0$,

Compute any $g^{(k)} \in \partial\psi(x^{(k)})$;

$\gamma := \|J^{(k)^T} g^{(k)}\|$;

$g := \frac{J^{(k)^T} g^{(k)}}{\gamma}$;

If $\psi(x^{(k)}) - \gamma > 0$, *quit.*

Else,

Compute any $g^{(k)} \in \partial\phi(x^{(k)})$;

$\gamma := \|J^{(k)^T} g^{(k)}\|$;

$g := \frac{J^{(k)^T} g^{(k)}}{\gamma}$;

$b := J^{(k)} g$;

$x^{(k+1)} := x^{(k)} - \frac{b}{n+1}$;

$J^{(k+1)} := \frac{n}{\sqrt{n^2-1}} J^{(k)}\left(I - (1 \pm \sqrt{\frac{n-1}{n+1}}) gg^T\right)$;

If $\psi(x^{(k)}) \le 0$ *and* $\gamma \le \epsilon$, *stop.*

Furthermore, the $J^{(k)}$ can be taken to a lower triangular matrix (the Cholesky factor) and hence a lower triangular $J^{(k+1)}$ can be obtained at the mild cost of $O(n^2)$ operations. More details can be found in [2, Section 6].

As is seen, the ellipsoid method requires only the evaluation of function values and any one (of the possibly many) subgradients of functions. On the other hand, Corollary 2.2 shows how convenient a subgradient for either $\mu(D)$ or $\nu(D)$ can be calculated. Thus the ellipsoid method is readily applicable to the (ETP) in either the form (6) and (7) or the form (10) and (11).

3. Second Approach.

3.1. Barrier Function. Let $\lambda_i(D)$ denote the i^{th} eigenvalue of $S-D$. Consider the function

$$\phi(D) := \sum_{i=1}^{n} \ln \frac{1}{\lambda_i(D)} + \sum_{i=1}^{n} \ln \frac{1}{d_i} \tag{26}$$

Since the logarithm is undefined for non-positive arguments, the function ϕ is defined only for strictly feasible D in (2). For computational purpose, we may write ϕ as

$$\phi(D) = \ln \det(S-D)^{-1} + \ln \det D^{-1}. \tag{27}$$

The idea of introducing the so called barrier function ϕ is that its level curves should be reasonable approximations to the boundary of the feasible domain (2). For a 2-dimensional example where $S = \begin{bmatrix} 4 & \sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}$, the boundary of the feasible domain and the level curves of ϕ are plotted in Figure 1.

We first derive formulas for the gradient $\nabla \phi(D)$ and the Hessian $\nabla^2 \phi(D)$. More general results can be found in [4, 16].

LEMMA 3.1. *The gradient vector of $\phi(D)$ is given by*

$$\nabla \phi(D) = diag\left((S-D)^{-1} - D^{-1}\right). \tag{28}$$

Proof. The derivatives of the second term in (27) is trivial. So the only concern is the partial derivative of the first term $\psi(D) = \ln \det(S-D)^{-1}$. It is a well known fact that if a matrix M has columns $[m_1, \ldots, m_n]$, then

$$\frac{d}{dx} \det M = \det \left[\frac{d}{dx} m_1, m_2, \ldots, m_n\right] + \ldots + \det \left[m_1, m_2, \ldots, \frac{d}{dx} m_n\right].$$

It follows that

$$\begin{aligned} \frac{\partial}{\partial d_i} \psi(D) &= -\frac{1}{\det(S-D)} \frac{\partial}{\partial d_i} \det(S-D) \\ &= \frac{\sigma_{ii}(S-D)}{\det(S-D)} \end{aligned}$$

where $\sigma_{ij}(M)$ denotes the cofactor of the elements m_{ij} of the matrix M. Recall the fact that

$$(\mathrm{adj} M) M = (\det M) I$$

where

$$\mathrm{adj} M := [\sigma_{ij}(M)]^T.$$

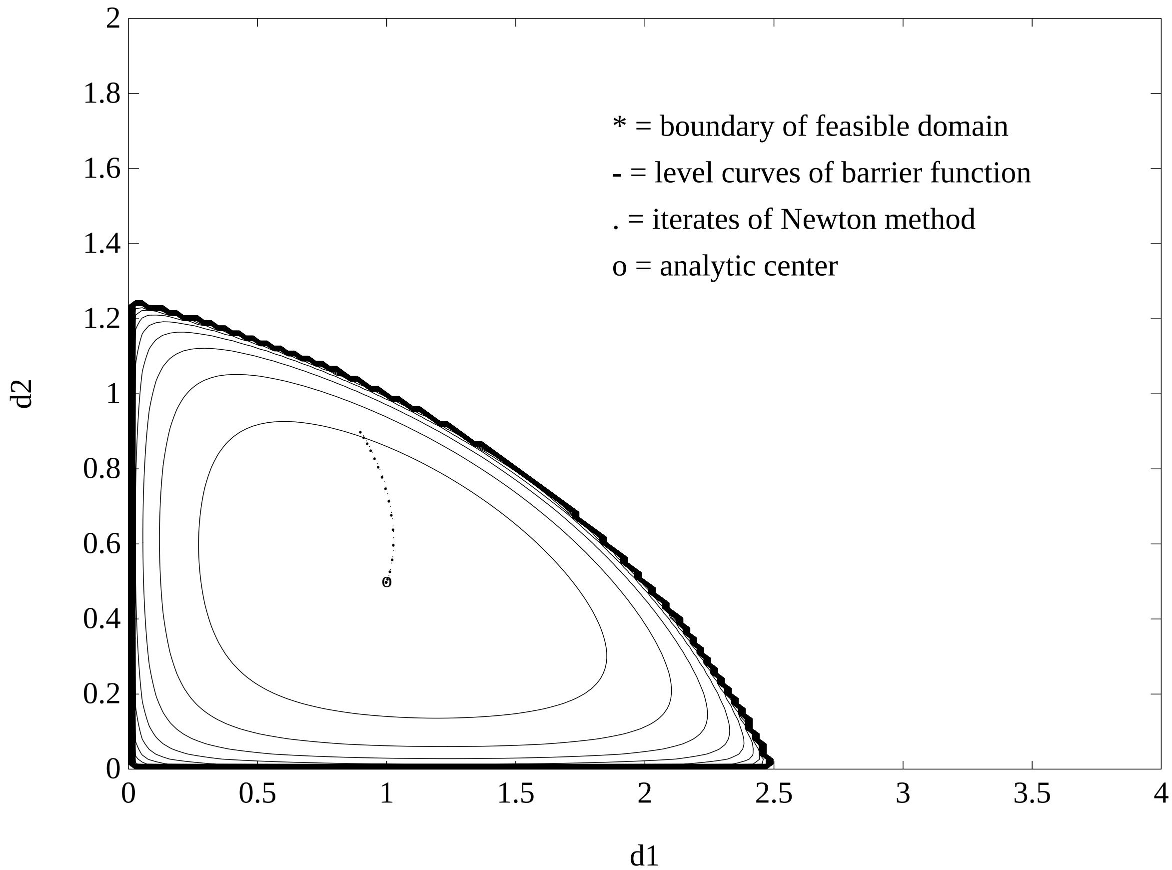

FIG. 1. *Boundary of feasible domain, level curves of barrier function and analytic center.*

The assertion follows. □

LEMMA 3.2. *The Hessian matrix $H(D)$ of $\phi(D)$ is given by*

$$H(D) = (S-D)^{-1} \circ (S-D)^{-1} + D^{-1} \circ D^{-1}. \tag{29}$$

Proof. Let Ξ_i denote the square unit matrix whose only non-zero entry with value 1 is at the (i,i) position. From Lemma 3.1, we may rewrite

$$\frac{\partial}{\partial d_i}\psi(D) = \langle (S-D)^{-1}, \Xi_i \rangle$$

where

$$< M, N >:= \operatorname{trace} M N^T = \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij} n_{ij}$$

denotes the Frobenius inner product of two matrices M and N. The second order derivative can now be conveniently calculated as follows:

$$\begin{aligned}
\frac{\partial^2}{\partial d_j \partial d_i}\psi(D) &= \langle \frac{\partial}{\partial d_j}(S-D)^{-1}, \Xi_i \rangle \\
&= \langle (S-D)^{-1}\Xi_j (S-D)^{-1}, \Xi_i \rangle \\
&= \langle \Xi_j (S-D)^{-1}, (S-D)^{-1}\Xi_i \rangle.
\end{aligned}$$

In the last equality above, we have used the facts that

$$\langle PM, N\rangle = \langle M, P^T N\rangle$$

and that $S - D$ is symmetric. □

We note from the well known Schur product theorem [13, Theorem 7.5.3] that $H(D)$ is positive definite if D is feasible, which also shows that the function ϕ is strictly convex over the feasible domain.

3.2. Inner Ellipsoid.

We have mentioned that the boundary of the feasible domain (2) can be approximated by the level curves of ϕ. In this section we describe how the latter can be approximated by inscribed ellipsoids determined by the Hessians of ϕ. More precisely, we have the following theorem where, in relation to the iteration that will be referred to in section 3.4, we denote the current iteration by the superscript $^{(c)}$ and the next iteration by $^{(+)}$.

THEOREM 3.3. *Suppose $D^{(c)}$ is a strictly feasible point with respect to (2). Then every diagonal matrix $D^{(+)}$ with $d^{(+)}$ from the ellipsoid $E(H(D^{(c)})^{-1}, d^{(c)})$ is also strictly feasible.*

Proof. Denote $\Delta := D^{(+)} - D^{(c)}$ and $\delta = \operatorname{diag}(\Delta)$. By Lemma 3.2 and the definition in (23), we have

$$\delta^T \left((S - D^{(c)})^{-1} \circ (S - D^{(c)})^{-1}\right)\delta + \delta^T \left({D^{(c)}}^{-1} \circ {D^{(c)}}^{-1}\right)\delta \le 1. \tag{30}$$

Consider the second term in (30) first. Since both $(S - D^{(c)})^{-1} \circ (S - D^{(c)})^{-1}$ and ${D^{(c)}}^{-1} \circ {D^{(c)}}^{-1}$ are positive definite, we have

$$\delta^T \left({D^{(c)}}^{-1} \circ {D^{(c)}}^{-1}\right)\delta < 1.$$

It follows that

$$|\delta_i| < d_i^{(c)}$$

for each i. This shows that $D^{(+)} > 0$.

To show that $S - D^{(+)} > 0$, we observe that

$$(S - D^{(c)})^{-\frac{1}{2}}(S - D^{(+)})(S - D^{(c)})^{-\frac{1}{2}} = I - (S - D^{(c)})^{-\frac{1}{2}}\Delta(S - D^{(c)})^{-\frac{1}{2}}.$$

Thus it suffices to show that

$$\|(S - D^{(c)})^{-\frac{1}{2}}\Delta(S - D^{(c)})^{-\frac{1}{2}}\|_F^2 < 1. \tag{31}$$

But (31) follows from the observation that

$$\begin{aligned}
& \|(S - D^{(c)})^{-\frac{1}{2}}\Delta(S - D^{(c)})^{-\frac{1}{2}}\|_F^2 \\
= \; & \langle (S - D^{(c)})^{-\frac{1}{2}}\Delta(S - D^{(c)})^{-\frac{1}{2}}, (S - D^{(c)})^{-\frac{1}{2}}\Delta(S - D^{(c)})^{-\frac{1}{2}}\rangle \\
= \; & \langle (S - D^{(c)})^{-1}\Delta, \Delta(S - D^{(c)})^{-1}\rangle \\
= \; & \delta^T \left((S - D^{(c)})^{-1} \circ (S - D^{(c)})^{-1}\right)\delta < 1
\end{aligned}$$

whereas the last inequality follows from the first term in (30). □

3.3. Analytic Center.

Being strictly convex over the the feasible domain (2), the barrier function $\phi(D)$ has a unique minimizer $\hat{D}$. Such a point is called the analytic center of ϕ. From Lemma 3.1 we see that $\hat{D}$ must satisfy the equation

$$\nabla\phi(\hat{D}) = \mathrm{diag}(S - \hat{D})^{-1} - \hat{D}^{-1} = 0. \tag{32}$$

The analytic center can be computed by a Newton method with damping [4, 17]:

ALGORITHM 3.1. *(Nesterov and Nemirovsky's Method)*
Given any initial point $D^{(0)}$ that is feasible, do:
Compute $\nabla\phi(D^{(k)})$ and $H(D^{(k)})$;
Solve $H(D^{(k)})^{\frac{1}{2}}\xi = \nabla\phi(D^{(k)})$;
$\rho := \|\xi\|$,
  If $\rho \le \frac{1}{4}$,
    $\alpha^{(k)} := 1$;
  Else,
    $\alpha^{(k)} := \frac{1}{1+\rho}$.
Solve $H(D^{(k)})\delta^{(k)} = -\alpha^{(k)}\nabla\phi(D^{(k)})$;
$D^{(k+1)} := D^{(k)} + Diag(\delta^{(k)})$.

It can be proved that the damping factor $\alpha^{(k)}$ results in $D^{(k+1)}$ being feasible [17]. Iterations illustrating the above iteration for the matrix $S = \begin{bmatrix} 4 & \sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}$ are plotted in Figure 1. Together with the notion of inner ellipsoids mentioned in Theorem 3.3, the analytic center serves as a good starting point for the so called Dikin's method.

3.4. Dikin's Method.

Suppose $D^{(c)}$ is a feasible point. Dikin's method [6] amounts to approximating the (ETP) locally by the following subproblem

$$\text{Maximize} \quad e^T d, \tag{33}$$

$$\text{subject to} \quad d \in E(H(D^{(c)})^{-1}, d^{(c)}) \tag{34}$$

where $d^{(c)} = diag(D^{(c)})$ and $e := [1, \ldots, 1]^T$.

Optimizing linear objective function over ellipsoids is easy. In fact, it can be proved that [12, Page 68]

LEMMA 3.4. *For $p \neq 0$, the maximal value of $p^T x$ subject to the condition $x \in E(B, a)$ occurs at*

$$x^* := a + \frac{1}{\sqrt{p^T B p}} Bp. \tag{35}$$

Thus a basic Dikin's algorithm can be stated as follows:

ALGORITHM 3.2. *(Basic Dikin's Method)*
Given $x^{(1)} \in R^n$ strictly feasible, do:
If $S - Diag(x^{(k)})$ is singular,
  Stop;
Else,
  Solve $H(Diag(x^{(k)}))b = e$ for b;
  $x^{(k+1)} := x^{(k)} + \frac{1}{\sqrt{e^T b}} b$.

Clearly, using (30), Dikin's method is extremely easy to be implemented. It should be cautioned, however, that the afore-mentioned stopping criterion is not sufficient for ensuring that trace(D) is maximized when the algorithm stops. Theorem 3.3 guarantees that $x^{(k)}$ is strictly feasible and hence $S - \mathrm{Diag}(x^{(k)})$ is never singular in exact arithmetic. However, in floating point arithmetic, one has to settle the singularity (as well as the rank) of a matrix for an eigenvalue (or a singular value) that is less than a prescribed tolerance. A usual choice of tolerance for zero is $\epsilon||S||$ where ϵ is the machine dependent floating point relative accuracy. For this reason it is possible that the algorithm may stop at a point where $S - D$ is *numerically* semi-definite yet trace(D) may have not reached its maximal value. Indeed, one difficulty in implementing Dikin's method is that, in contrast to the ellipsoid methods, there is no general stopping criterion [15]. To reduce the risk of hitting boundaries of the feasible domain too soon, we find it is a good idea to start out the Dikin's method from a point that is most interior to the feasible domain. Our numerical experiences seem to indicate the analytic center, for example, is always a good starting point.

4. Numerical Experiment.

We have applied the algorithms discussed in this paper to solve the set of educational testing problems given by Woodhouse [22]. The Woodhouse data set is in general an $N \times m$ matrix $X = [x_{ij}]$ where x_{ij} gives the score of student i on subject j. Test problems are generated by selecting various subsets of columns for form the matrix S. More precisely, let $v = \{v_1, \ldots, v_n\}$ with $n \leq m$ denote the subset of column indices being considered. Then $S = S_v = [s_{ij}]$ is the $n \times n$ matrix generated by

$$s_{jk} := \frac{1}{N-1} \sum_{i=1}^{N} \left(x_{iv_j} - \overline{x}_{v_j}\right)\left(x_{iv_k} - \overline{x}_{v_k}\right) \tag{36}$$

where $\overline{x}_j = \sum_{i=1}^{N} x_{ij}/N$ is the column mean.

Our results are compared against those given by Fletcher [8] where a specific 64×20 matrix X is used [7, 22] . The computations have been carried out by MATLAB on a DECstation 5000/200. We should point out that the efficiency of the basic algorithms described in the paper can be greatly improved by taking into account more careful programming details. Nevertheless, even with the simple version, our results show that the methods are effective and reliable.

We first compute the analytic center. For simplicity, the starting point $D^{(0)}$ of Algorithm 3.1 for every test case is taken to be

$$D^{(0)} := 0.9\lambda_n(S)I \tag{37}$$

where λ_n is the smallest eigenvalue of the underlying S. The iteration stops when the 2-norm of the difference between consecutive iterates is less than $\epsilon||S||$ where ϵ is the machine dependent floating point relative accuracy and is $\approx 2.2204 \times 10^{-16}$ in our case. In Table 1 we list the column indices used to generate the matrix S, the analytic centers and the number of iterations needed for convergence. So as to fit the data comfortably in the running text, we display all the numbers with only four decimal digits. It should be pointed out that Algorithm 3.1 eventually become quadratically convergent. As a matter of fact, Nesterov and Nemirovsky even provide a sharp bound on the number of iterations required to compute the analytic center within a given accuracy.

Using the analytic center as the starting point, we tabulate in Table 2 the optimal solutions computed from Algorithm 3.2 and the number of iterations required for

v	Analytic Center				Iteration
1,2,5,6	130.3018	137.0515	64.2004	23.8803	9
1,3,4,5	103.3142	126.0694	85.1781	65.0063	9
1,2,3,6,	28.6016	50.3854	43.1713	21.6472	
8,10	40.8325	37.7731			9
1,2,4,5,	47.6804	114.1003	71.4372	63.9306	
6,8	24.5235	48.0063			9
1:6	101.3008	49.7498	53.2610	74.4367	
	64.3381	22.7019			9
1:8	37.5094	48.7373	49.0590	78.7444	
	63.2342	22.5123	57.4478	46.7986	9
1:10	17.0703	48.0679	45.1975	70.8547	
	63.9691	21.2662	55.9627	40.4653	
	46.9100	37.4983			10
1:12	13.4577	47.5093	44.9369	70.8163	
	58.6676	20.9336	54.2917	39.1942	
	36.6014	37.1797	49.9820	42.3908	11
1:14	6.5662	42.5043	46.6774	62.6464	
	54.0793	18.5147	41.3755	32.8890	
	37.5776	35.9830	33.5691	43.5142	
	13.4953	10.3959			13
1:16	3.8189	42.3981	43.5428	59.6363	
	51.9807	18.2824	43.8108	30.0555	
	34.6286	34.5487	30.5031	39.2808	
	14.7268	10.4285	5.0935	18.8981	15
1:18	2.2427	42.1140	44.0036	59.5748	
	46.6077	17.4513	42.9404	27.7346	
	36.2674	35.6251	29.0568	36.6165	
	13.5875	10.0714	5.1555	17.1592	
	37.3088	26.8049			18
1:20	2.3241	40.5550	45.2906	55.3856	
	41.5906	16.2843	31.7069	21.6311	
	34.1435	30.8012	27.2964	30.9536	
	13.7232	8.6853	4.8949	15.1030	
	31.8938	27.3978	43.3779	69.2924	20

TABLE 1
Analytic Centers

Dikin's method. The matrix $S - \mathrm{Diag}(x^{(k)})$ is assumed to be singular if the minimum eigenvalue of $S - \mathrm{Diag}(x^{(k)})$ is less than $\epsilon\|S\|$. We indicate earlier that quite often at the optimal solution the eigenvalues coalesce. This is evidenced in Table 2 by the multiplicity of the eigenvalue 0 at the optimal solution. Using the same analytic center as the starting point, we also have applied Algorithm 2.2 to solve the (ETP) in the form of (6) and (7). Table 3 provides information similar to Table 2. We observe that the ellipsoid method is notably slow in convergence.

In certain cases, we find our results are different from Fletcher's results [8] by a substantial discrepancy that is beyond what should be if the correct answer is rounded to four or five digits. The 2-norm of the discrepancy is also recorded in Table 2 and 3 where the number inside the parentheses is the exponent in base 10. We note particularly the case $v = \{1, \ldots, 10\}$ where Fletcher's result is wrong in that d_7 and d_8 were transposed. Since it has been noted that Glunt [11] was able to reproduce Fletcher's results by using the alternating projection method, it seems to imply that our algorithms are not reliable. However, we should point out that Glunt has only reported on the relative discrepancy. When comparing the absolute discrepancy, Glunt's results seem to have the worst accuracy among the four numerical methods as will be exemplified below. On the other hand, since our two methods agree more closely with each other than with Fletcher's results, it also seems to imply that our results should be trustworthy. This paradox is even more perplexing when one examine the result for the case $v = \{1, 2, 5, 6\}$ carefully — The first result in Table 3 agrees closely with Fletcher's result, and the one in Table 2 does not. This suggests one of the methods may have failed.

In an attempt to resolve the above enigma, we list in Table 4 the computed solution D in all available digits (Fletcher's and Glunt's results are available only up to 6 digits from the literature.) We then calculate all eigenvalues of the corresponding matrix $S - D$ in Table 5.

As can be seen, Fletcher's result gives rise to a small negative eigenvalue which should be theoretically zero. The magnitude in the order of 10^{-6} is expectable given the fact Fletcher has only reported 6 digits of accuracy. The ellipsoid method produces a result that is close to Fletcher's, except that a smaller negative number in the order of 10^{-13} is taken to be the zero eigenvalue. The threshold for determining singularity in this case is $\epsilon\|S\| \approx 1.2038 \times 10^{-13}$. So in our view the ellipsoid method has carried out its best possible accomplishment.

Dikin's method, on the other hand, produces a substantially different result in this case. We first observe that the feasibility is maintained since the zero eigenvalue of $S - D$ is approximated by a small positive quantity or order 10^{-14}. If we assume that the true solution to the (ETP) is better approximated by the ellipsoid method than by Fletcher's method, then it is rather surprising to see that the component-wise maximal discrepancy between the matrices D generated by the ellipsoid method and by Dikin's method is as large as ≈ 0.3473. While the ellipsoid method gives a slightly larger objective value $\mathrm{trace}(D) \approx 5.427735615069689 \times 10^2$ by violating the feasible constraints within the machine precision, Dikin's method ensures feasibility by returning a slightly smaller objective value $\mathrm{trace}(D) \approx 5.427730183170040 \times 10^2$. From this viewpoint, it is truely difficult to judge which method is most satisfactory. Apparently this proves that finding the exact solution to the (ETP) is a very delicate task. Fortunately, as far as its application in statistics is concerned, the objective value $\mathrm{trace}(D)$ usually does not require very high accuracy [7].

In fact, we have checked all 12 test cases and observed that Dikin's method did

v	Optimal Solution				Iteration	Multiplicity	Discrepancy
1,2,5,6	173.4639	236.6797	103.7673	28.8621	34	1	0.4122(0)
1,3,4,5	156.2324	240.9354	128.7423	107.2478	17	2	5.1485(-5)
1,2,3,6,	0.0000	102.0203	19.8772	31.4606			
8,10	82.2832	69.8404			20	1	2.7509(-4)
1,2,4,5,	59.6233	214.0318	69.8054	115.7325			
6,8	47.0397	58.2305			28	2	0.0212(0)
1:6	152.7058	54.4757	82.9314	99.6415			
	104.6550	40.9529			20	2	9.1933(-5)
1:8	14.0323	38.5418	95.0990	158.9009			
	120.3823	28.3713	106.7753	79.7356	29	2	5.8057(-5)
1:10	0.0000	43.8923	80.7165	132.8874			
	126.8620	28.0302	92.6100	56.6200			
	61.3363	67.8258			39	2	50.8976(0)
1:12	18.6332	61.8632	63.4274	127.5681			
	99.9735	30.7704	96.5349	45.2875			
	41.6016	45.3291	64.0408	52.4596	38	3	7.1738(-5)
1:14	0.0000	59.4989	62.9123	109.9237			
	99.9491	32.7194	79.0728	31.7381			
	47.4210	33.7888	41.9528	63.5956			
	4.2517	4.4508			38	2	6.3563(-5)
1:16	0.0000	63.4868	52.3890	108.1923			
	92.3952	34.5616	85.7551	21.9573			
	37.5494	32.9670	28.5112	54.5709			
	12.9296	4.1035	6.7064	27.3866	37	2	0.0032(0)
1:18	0.0000	58.3802	62.1620	107.2306			
	80.2873	25.3833	70.7034	24.3173			
	52.4379	41.6948	24.2924	39.1760			
	15.7610	6.8615	3.2590	14.5931			
	68.8044	52.1616			44	3	4.2067(-5)
1:20	0.0000	47.3728	76.5817	101.0016			
	63.4500	13.3822	41.4830	4.3003			
	56.3649	33.9832	33.7699	29.9598			
	17.5971	0.0000	4.3281	13.6903			
	45.5872	51.5863	57.2066	128.6977	45	2	6.4933(-5)

TABLE 2
Results for Algorithm 3.2.

v	Optimal Solution				Iteration	Multiplicity	Discrepancy
1,2,5,6	173.1170	236.8681	103.8767	28.9118	209	1	4.5646(-4)
1,3,4,5	156.2324	240.9354	128.7423	107.2478	351	2	5.0141(-5)
1,2,3,6,	0.0000	102.0203	19.8771	31.4606			
8,10	82.2833	69.8404			702	1	2.2068(-4)
1,2,4,5,	59.6236	214.0323	69.8051	115.7325			
6,8	47.0397	58.2302			797	2	0.0206(0)
1:6	152.7057	54.4758	82.9313	99.6415			
	104.6550	40.9529			805	2	2.4789(-4)
1:8	14.0325	38.5418	95.0989	158.9009			
	120.3821	28.3714	106.7753	79.7356	1511	2	3.4259(-4)
1:10	0.0000	43.8922	80.7168	132.8876			
	126.8620	28.0300	92.6101	56.6198			
	61.3362	67.8258			2499	2	50.8974(0)
1:12	18.6332	61.8634	63.4275	127.5681			
	99.9734	30.7703	96.5348	45.2876			
	41.6014	45.3290	64.0409	52.4597	3957	3	3.2475(-4)
1:14	0.0000	59.4991	62.9122	109.9236			
	99.9492	32.7194	79.0728	31.7385			
	47.4209	33.7889	41.9526	63.5956			
	4.2515	4.4508			4714	2	6.3563(-5)
1:16	0.0000	63.4867	52.3890	108.1923			
	92.3953	34.5616	85.7551	21.9572			
	37.5494	32.9671	28.5111	54.5709			
	12.9297	4.1036	6.7065	27.3866	6110	2	0.0032(0)
1:18	0.0000	58.3802	62.1620	107.2305			
	80.2874	25.3833	70.7033	24.3174			
	52.4379	41.6949	24.2924	39.1760			
	15.7607	6.8615	3.2590	14.5931			
	68.8044	52.1617			8575	3	3.6876(-4)
1:20	0.0000	47.3731	76.5815	101.0017			
	63.4505	13.3822	41.4829	4.3001			
	56.3649	33.9834	33.7696	29.9599			
	17.5970	0.0000	4.3280	13.6901			
	45.5872	51.5863	57.2065	128.6977	9526	2	7.9265(-4)

TABLE 3
Results for Algorithm 2.2.

Fletcher	Ellipsoid	Dikin	Glunt
1.731174(2)	1.731165897531778(2)	1.734639095485107(2)	1.731324(2)
2.368681(2)	2.368683462604490(2)	2.366797403246821(2)	2.368578(2)
1.038765(2)	1.038767739770922(2)	1.037672671785132(2)	1.038729(2)
2.891159(1)	2.891185151624995(1)	2.886210126529794(1)	2.89103(1)

TABLE 4
Computed solution D for $v = \{1, 2, 5, 6\}$.

Fletcher	Ellipsoid	Dikin	Glunt
4.2694(1)	4.2694(1)	4.2809(1)	4.2700(1)
2.7741(1)	2.7741(1)	2.7796(1)	2.7743(1)
-7.1236(-6)	-2.7356(-13)	5.6843(-14)	4.0113(-05)
3.4974(2)	3.4975(2)	3.4958(2)	3.4974(2)

TABLE 5
Eigenvalues of $S - D$ for $v = \{1, 2, 5, 6\}$.

Fletcher	Ellipsoid	Dikin	Glunt
2.4226(-7)	7.9784(-14)	3.4497(-13)	4.7439(-5)
1.9311(-6)	3.1757(-10)	2.4975(-13)	5.7266(-5)
-8.1763(-6)	5.7458(-9)	2.2659(-12)	1.9844(-5)
2.5072(1)	2.5072(1)	2.5072(1)	2.5072(1)
9.2006(0)	9.2006(0)	9.2006(0)	9.2009(0)
4.0672(1)	4.0672(1)	4.0672(1)	4.0672(1)
6.3500(1)	6.3500(1)	6.3500(1)	6.3501(1)
8.7266(1)	8.7267(1)	8.7266(1)	8.7266(1)
1.1460(2)	1.1460(2)	1.1460(2)	1.1460(2)
1.7187(2)	1.7187(2)	1.7187(2)	1.7187(2)
4.0826(2)	4.0826(2)	4.0826(2)	4.0826(2)
1.5977(3)	1.5977(3)	1.5977(3)	1.5977(3)

TABLE 6
Eigenvalues of $S - D$ for $v = \{1, \ldots, 12\}$.

not give negative eigenvalues for any of the test problems while the other two method do sometimes give small negative eigenvalues with Fletcher's in the order of 10^{-5} to 10^{-7} and the ellipsoid method in the order of 10^{-10} to 10^{-14}. Another interesting observation, as is demonstrated in Table 6 for the case $v = \{1, \ldots, 12\}$, is that the coalescent zero eigenvalues resulted from Dikin's method usually cluster together while those from the ellipsoid method spread over a wider range.

Finally, we point out that Glunt's method converges linearly and usually returns values in the order of 10^{-5} as the zero eigenvalue. It is not clear how long Glunt's method will take to reach the same accuracy as that of the ellipsoid method or Dikin's method.

It is remarkable that Dikin's method can obtain convergence quite rapidly, even for the 20×20 test case. We do not completely understand the theory of convergence for Dikin's method. For the time being, we can only refer readers to the recent review article [15] and the many references contained therein. In particular, we are aware of the long step version of the Dikin's method in which the next iterate is determined by taking a fixed fraction $\lambda \in (0, 2/3]$ of the whole step to the boundary of the inner ellipsoid. (See (35) and Algorithm 3.2.) Global convergence for the long step version of Dikin's method applied to degenerate linear programming problems can be proved [15]. The proof probably needs substantial modification for our problem. On the other hand, by using the long step version, i.e.,

$$x^{(k+1)} := x^{(k)} + \frac{2/3}{\sqrt{e^T b}} b,$$

we find that the inconsistency mentioned above between Algorithm 2.2 and 3.2 for the case $v = \{1, 2, 5, 6\}$ is fixed, confirming that the result from the ellipsoid method is better; nevertheless, the substantial discrepancy between Fletcher's results and ours for the cases $v = \{1, 2, 4, 5, 6, 8\}$ and $v = \{1, \ldots, 16\}$ still prevails.

We mention earlier that Dikin's method maintains the feasibility throughout the iteration. We demonstrate the convergence behavior of the ellipsoid method in Figure 2 and Figure 3. Figure 2 demonstrates the history of the first 400 iterations for the case $v = \{1, 2, 3, 6, 8, 10\}$. Since the optimal solution occurs at the boundary $d_1 = 0$, conceivably the centers of the ellipsoids will often fall outside the feasible domain. When this happens, a constraint iteration where the subgradient is taken from the constraints (rather then the objective function) must take place. This is recorded in Figure 2 by the symbol $+$. Figure 3 demonstrates the history of the first 400 iterations for the case $v = \{1, 2, 4, 5, 6, 8\}$. Since the optimal solution is strictly interior to the feasible domain, we see that the constraint iteration occurs only at the beginning. It is clear that the ellipsoid method is not necessarily a descent method for the objective function .

5. Acknowledgements.

The first author learned of the ideas presented in this paper from lectures given by Professor Stephen Boyd while attending the NATO ASI at Leuven, Belgium. The aftermath discussion which Professor Boyd has tirelessly provided is especially appreciated.

REFERENCES

[1] F. Alizadeh, Optimization over the positive semi-definite cone: Interior point methods and combinatorial applications, in Advances in Optimization and Parallel Computing, ed., P. Pardalos, North-Holland, Amsterdam, 1992, 1-25.

[2] R. G. Bland, D. Goldfarb and M. J. Todd, The ellipsoid method: A survey, Operations Research, 29(1981), 1039-1091.

[3] S. Boyd and C. Barrat, Linear Controller Design: Limits of Performance, Information and System Sciences Series, Prentice-Hall, New Jersey, 1990.

[4] S. Boyd and L. E. Ghaoui, Method of centers for minimizing generalized eigenvalues, preprint, Stanford University, 1992.

[5] J. P. Boyle and R. L. Dykstra, A method for finding projections onto the intersection of convex sets in Hilbert space, in Advances in Order Restricted Statistical Inference, ed. R. Dykstra, T. Robertson and F. T. Wright, Lecture Notes in Statistics, 37, Springer-Verlag, New York, 1986, 28-47.

[6] I. Dikin, Iterative solution of problems of linear and quadratic programming, Soviet Math. Dokl., 8(1967), 674-675.

[7] R. Fletcher, A nonlinear programming problem in statistics (Educational testing), SIAM J. Sci. Stat. Comput., 2(1981), 257-267.

[8] R. Fletcher, Semi-definite matrix constraints in optimization, SIAM J. Control Optim., 23(1985), 493-513.

[9] R. Fletcher, Practical Methods of Optimization, 2nd ed., John Wiley and Sons, Chichester, 1987.

[10] S. Friedland, J. Nocedal and M. L. Overton, The formulation and analysis of numerical methods for inverse eigenvalue problems, SIAM J. Numer. Anal. 24(1987), 634-667.

[11] W. K. Glunt, An alternating projections method for certain linear problems in a hilbert space, University of Kentucky, preprint, 1991.

[12] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer-Verlag, Berlin, 1988.

[13] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1991.

[14] L. G. Khachiyan, Polynomial algorithms in linear programming, USSR Computational Mathematics and Mathematical Physics, 20(1980), 53-72.

[15] R. D. C. Monteiro, T. Tsuchiya and Y. Wang, A simplified global convergence proof of the affine scaling algorithm, University of Arizona, preprint, 1992.

[16] Yu. E. Nesterov and A. S. Nemirovsky, Self-concordant functions and polynomial time methods in convex programming, Technical report, Centr. Econ. & Math. Inst. USSR Adad. Sci., Moscow, USSR, 1989.

[17] Yu. E. Nesterov and A. S. Nemirovsky, Interior Point Polynomial Methods in Convex Programming: Theory and Applications, SIAM, to be published.

[18] M. L. Overton, On minimizing the maximum eigenvalue of a symmetric matrix, SIAM J. Matrix Anal. Appl., 9(1988), 256-268.

[19] M. L. Overton, Large-scale optimization of eigenvalues, SIAM J. Optimization, 2(1992), 88-120.

[20] R. T. Rockafellar, The Theory of Subgradients and Its Applications to Problems of Optimization: Convex and Nonconvex Functions, Research and Education in Mathematics 1, Heldermann, Berlin, 1981.

[21] N. Z. Shor, Cut-off method with space extension in convex programming problems, Cybernetics 13(1977), 94-96.

[22] B. Woodhouse, Lower bounds for the reliability of a test, M. S. Thesis, Dept. Statistics, University Wales, Aberystwyth, 1976.