A Modified Frank-Wolfe Algorithm for Computing Minimum-Area Enclosing Ellipsoidal Cylinders: Theory and Algorithms [*]

S. Damla Ahipaşaoğlu Michael J. Todd

September 11, 2009

Abstract

We study a first-order method to find the minimum cross-sectional area ellipsoidal cylinder containing a finite set of points. This problem arises in optimal design in statistics when one is interested in a subset of the parameters. We provide convex formulations of this problem and its dual, and analyze a method based on the Frank-Wolfe algorithm for their solution. Under suitable conditions on the behavior of the method, we establish global and local convergence properties. However, difficulties may arise when a certain submatrix loses rank, and we describe a technique for dealing with this situation.

1 Introduction

We study the problem of finding an ellipsoidal cylinder containing a finite set of points in $\mathbf{R}^n$, such that its cross-section with a k-dimensional subspace has minimum area. This is a generalization of the minimum-volume enclosing ellipsoid (MVEE) problem, which has been much studied, with applications in data analysis and (via its dual) the D-optimal design problem in statistics.

The minimum-area enclosing ellipsoidal cylinder (MAEC) problem has also been widely studied, mainly because its dual is another optimal design problem in statistics, where now one is interested in estimating just k out of n parameters in a regression problem by choosing the design points optimally in some sense. See Fedorov [5], Silvey and Titterington [11], Atwood [2, 3], and Pukelsheim [8] for more details.

Our interest in this problem is mainly algorithmic: we study a first-order method based on the Frank-Wolfe method [6] with Wolfe's away steps [13], which was introduced in the context of optimal design by Atwood [3]. However,

[*] Supported in part by NSF through grant DMS-0513337 and ONR through grant N00014-02-1-0057.

no detailed analysis of this method has been performed, and in certain cases it breaks down unless modified to keep an appropriate matrix positive definite.

We show that under reasonable conditions on the iterates produced by the algorithm, global complexity estimates and local convergence properties can be established. These conditions require that a certain principal submatrix of a positive semidefinite matrix produced by the algorithm remain positive definite.

We also provide a technique that allows the iterations to proceed when rank deficiency occurs; although we have no guarantee of convergence in this case, the method appears to work in practice. Finally, some computational results for large random problems are given.

The paper is organized as follows. In the next section, we state convex formulations of the MAEC problem and its dual. Although previous papers have included formulations with some convexity properties, they have not been fully convex. Section 3 describes the basic algorithm of Atwood [3]. We prove global and local convergence results in Section 4. The case of rank-deficiency of a critical submatrix is discussed in Section 5, and Section 6 contains the results of our computational study. We conclude in Section 7 with some final remarks.

2 Problem formulation and duality

In this section we provide convex formulations of the MAEC problem and its dual. Previous formulations, such as that in Silvey and Titterington [11], have not been convex in all the variables (see problem (P') below), although they have some convexity properties, and the dual problem involved a Schur complement rather than our simpler formulation (D).

We also relate these formulations to earlier ones and prove duality results. The section ends by defining notions of approximate optimality, which will be used in our algorithms.

2.1 Problem definition, convex formulation

Suppose we are given a matrix $X = [x_1, x_2, \ldots, x_m] \in \mathbf{R}^{n\times m}$ whose columns, the points $x_1, \ldots, x_m$, span $\mathbf{R}^n$. Let $X = \begin{bmatrix} Z \\ Y \end{bmatrix}$ be a partition of X, where $Z \in \mathbf{R}^{(n-k)\times m}$ and $Y \in \mathbf{R}^{k\times m}$. If H' is a symmetric matrix of order k that is positive definite (we write $H' \succ 0$) and E is a matrix of order $k \times (n-k)$, then the set

$$\mathcal{C}(0; E; H') := \{[z; y] : z \in \mathbf{R}^{n-k}, y \in \mathbf{R}^k, (y + Ez)^T H'(y + Ez) \le k\}$$

is a central (i.e., centered at the origin) ellipsoidal cylinder whose intersection with the subspace

$$\Pi := \{[z; y] \in \mathbf{R}^n : z \in \mathbf{R}^{n-k}, y \in \mathbf{R}^k, z = 0\}$$

has volume $(\det H')^{-1/2}$ times that of a Euclidean ball in $\mathbf{R}^k$ of radius $\sqrt{k}$. H' determines the shape of the cross-section and E the “directions of the axes” of the cylinder. Hence, finding a central ellipsoidal cylinder, which contains the columns of X and has minimum-volume (area) intersection with Π, amounts to solving

$$\begin{array}{rrcl} & \min_{H' \succ 0}\ \bar{f}(H',E) & := & -\ln\det H' \\ (P') & (y_i + Ez_i)^T H' (y_i + Ez_i) & \leq & k, \quad i = 1,\ldots,m, \end{array}$$

where the variables are H' and E. This is called the minimum-area enclosing ellipsoidal cylinder (MAEC) problem.

This problem is nonconvex. The following lemma proves that it can be reformulated as a convex programming problem as follows:

$$\begin{array}{rrcl} & \min_{H_{YY} \succ 0}\ f(H) & := & -\ln\det H_{YY} \\ (P) & x_i^T H x_i & \leq & k, \quad i = 1,\ldots,m, \\ & H & \succeq & 0, \end{array}$$

where the variable H is partitioned as $\begin{bmatrix} H_{ZZ} & H_{ZY} \\ H_{ZY}^T & H_{YY} \end{bmatrix}$ and $H_{YY} \in \mathbf{R}^{k \times k}$.

Lemma 2.1 *Problems (P) and (P') are equivalent.*

Proof: To see this, consider any feasible solution H to (P). Note that given $H_{YY} \succ 0$, $H \succeq 0$ holds iff $H_{ZZ} \succeq H_{ZY} H_{YY}^{-1} H_{ZY}^T$. We can therefore assume that $H_{ZZ} = H_{ZY} H_{YY}^{-1} H_{ZY}^T$ without loss of generality, since replacing H_{ZZ} by the right-hand side will preserve feasibility and leave unchanged the objective value. Then

$$x_i^T H x_i \leq k$$

iff

$$y_i^T H_{YY} y_i + 2 y_i^T H_{ZY}^T z_i + z_i^T H_{ZY} H_{YY}^{-1} H_{ZY}^T z_i \leq k.$$

If we let $E = H_{YY}^{-1} H_{ZY}^T$, the latter inequality holds iff

$$y_i^T H_{YY} y_i + 2 y_i^T H_{YY} E z_i + z_i^T E^T H_{YY} E z_i \leq k$$

or

$$(y_i + Ez_i)^T H_{YY} (y_i + Ez_i) \leq k,$$

and we obtain a feasible solution to problem (P') with the same objective value. Conversely, given any feasible solution to (P'), we can set $H_{YY} := H'$, $H_{ZY} := E^T H_{YY}$, and $H_{ZZ} := H_{ZY} H_{YY}^{-1} H_{ZY}^T$ to get a feasible solution to (P) with the same objective value. Thus, problems (P) and (P') are equivalent. □

When $k = n$, the MAEC problem reduces to the minimum-volume enclosing ellipsoid (MVEE) problem. We note that searching for a central ellipsoidal cylinder containing the columns of X is without loss of generality: if an arbitrary ellipsoidal cylinder is sought, it can be obtained by finding a central ellipsoidal cylinder in $\mathbf{R}^{n+1}$ containing the points $(1; x_i)$, $i = 1, \ldots, m$, with the same value of k.

2.2 Weak and strong duality, and optimality conditions

Problem (P) is convex with linear inequality constraints. After some simplification, its Lagrangian dual can be written as

$$(D) \qquad \begin{array}{lrcl} \max_{u,K\succ 0} & g(u,K) := \ln\det K & & \\ & XUX^T - \overline{K} := XUX^T - \begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix} & \succeq & 0, \\ & e^T u & = & 1, \\ & u & \geq & 0, \end{array}$$

where matrix U is a diagonal matrix with the components of u on its diagonal and e is a vector of ones in $\mathbf{R}^m$. Problem (D) is also the statistical problem of finding a D_k-optimal design measure u on the columns of X, that is, one that maximizes the determinant of a $k \times k$ Schur complement of the Fisher information matrix $E[xx^T] = XUX^T$: see, e.g., Silvey and Titterington [11] and Fedorov [5].

We say H is feasible in (P) if it is positive semidefinite and satisfies the constraints of (P) and H_{YY} is positive definite; similarly, (u,K) is feasible in (D) if it satisfies the constraints of (D) and K is positive definite. We first establish weak duality:

Lemma 2.2 *$f(H) \geq g(u,K)$ for any H and (u,K) feasible in (P) and (D), respectively.*

Proof: Since H and $XUX^T - \overline{K}$ are positive semidefinite,

$$0 \leq H \bullet \left(XUX^T - \overline{K}\right) = \sum_i u_i x_i^T H x_i - H \bullet \overline{K} \leq k e^T u - H_{YY} \bullet K. \qquad (2.1)$$

(We use $\bullet$ to denote the trace inner product on symmetric matrices: $V \bullet W := \mathrm{Tr}(VW)$.) Hence we have

$$\begin{array}{rcl} -\ln\det H_{YY} - \ln\det K & = & -\ln\det H_{YY}K \\ & = & -k\ln(\Pi_{i=1}^k \lambda_i)^{1/k} \\ & \geq & -k\ln\left(\dfrac{\sum_{i=1}^k \lambda_i}{k}\right) \\ & \geq & -k\ln\left(\dfrac{k}{k}\right) \\ & = & 0, \end{array}$$

where the λ_i's are the positive eigenvalues of $H_{YY}K$ (or of ${H_{YY}}^{1/2}K{H_{YY}}^{1/2}$), and (2.1) shows their sum is at most k. □

Hence feasible solutions H and (u,K) are optimal if $f(H) = g(u,K)$. We next show that strong duality holds, so that this condition is necessary as well as sufficient.

Theorem 2.1 *There exist optimal solutions for problems (P) and (D). Furthermore, the following conditions, together with primal and dual feasibility, are necessary and sufficient for optimality in both (P) and (D):*

(a) $H \bullet (XUX^T - \overline{K}) = 0$,

(b) $u_i > 0$ *only if* $x_i^T H x_i = k$, *and*

(c) $H_{YY} = K^{-1}$.

Proof: Let H be a feasible solution for problem (P). Summing up the linear constraints, we must have $\sum x_i^T H x_i = H \bullet XX^T \leq km$. Since $XX^T \succ 0$ and $km > 0$, $\{H \succeq 0 : H \bullet XX^T \leq km\}$ is a compact set. Hence the feasible region for problem (P) is also a compact set (since it is the intersection of a compact set with a finite set of halfspaces). Moreover, $H = \epsilon I$ is feasible for (P) for sufficiently small positive ϵ, so we can add the constraint that $\ln\det H_{YY} \geq k \ln \epsilon$, and the feasible region remains compact and has the same set of optimal solutions. Now the objective function is continuous on the compact feasible region, so an optimal solution exists for problem (P). Existence of an optimal solution for (P) implies the existence of optimal solutions for (P') and also for (D) as will be discussed later.

Sufficiency follows from the previous lemma, since the conditions imply equality in the weak duality inequality. In order to prove necessity, let $\tilde{H} = \begin{pmatrix} \tilde{H}_{ZZ} & \tilde{H}_{ZY} \\ \tilde{H}_{ZY}^T & \tilde{H}_{YY} \end{pmatrix}$ be an optimal solution for (P), so that $(H' = \tilde{H}_{YY}, \tilde{E} = \tilde{H}_{YY}^{-1} \tilde{H}_{ZY}^T)$ is an optimal solution for the primal problem (P'). Hence the Karush-Fritz-John conditions must hold for this solution, i.e., there exist nonnegative multipliers $\tilde{v} \in \mathbf{R}$ and $\tilde{u} \in \mathbf{R}^m$ with at least one of the multipliers nonzero and the following equalities hold:

$$-\tilde{v} {H'}^{-1} + \sum_{i=1}^{m} \tilde{u}_i (y_i + \tilde{E} z_i)(y_i + \tilde{E} z_i)^T = 0, \tag{2.2}$$

$$2 \sum_{i=1}^{m} \tilde{u}_i H' y_i z_i^T + 2 \sum_{i=1}^{m} \tilde{u}_i H' \tilde{E} z_i z_i^T = 0, \text{ and} \tag{2.3}$$

$$\tilde{u}_i ((y_i + \tilde{E} z_i)^T H' (y_i + \tilde{E} z_i) - k) = 0, \forall i. \tag{2.4}$$

For one moment let's assume that $\tilde{v} = 0$. Then (2.2) becomes

$$\sum \tilde{u}_i (y_i + \tilde{E} z_i)(y_i + \tilde{E} z_i)^T = 0,$$

and hence, by taking the inner product with H', we see that (2.4) implies that $k \sum \tilde{u}_i = 0$. Since at least one of the $\tilde{u}_i$'s is positive, this is a contradiction and we must have $\tilde{v} > 0$. We can without loss of generality assume that $\tilde{v} = 1$, and conclude that any optimal solution $(H', \tilde{E})$ must satisfy

$$-{H'}^{-1} + \sum \tilde{u}_i (y_i + \tilde{E} z_i)(y_i + \tilde{E} z_i)^T = 0, \tag{2.5}$$

which together with (2.4) implies

$$\sum \tilde{u}_i = 1. \tag{2.6}$$

Since $H' \succ 0$, equation (2.3) can be written as $Y\tilde{U}Z^T + \tilde{E}Z\tilde{U}Z^T = 0$, and hence (2.5) becomes

$$\begin{aligned} {H'}^{-1} &= Y\tilde{U}Y^T + Y\tilde{U}Z^T\tilde{E}^T + \tilde{E}Z\tilde{U}Y^T + \tilde{E}Z\tilde{U}Z^T\tilde{E}^T \\ &= Y\tilde{U}Y^T - \tilde{E}Z\tilde{U}Z^T\tilde{E}^T. \end{aligned} \tag{2.7}$$

Let $\tilde{K} := Y\tilde{U}Y^T - \tilde{E}Z\tilde{U}Z^T\tilde{E}^T$. Then it is easy to check that $(\tilde{u}, \tilde{K})$ is a feasible solution for the dual problem, and that strong duality holds for the solution pair $\tilde{H}$ and $(\tilde{u}, \tilde{K})$. So strong duality must hold for any pair of optimal solutions H and (u, K). Hence conditions $(a) - (c)$ are both necessary and sufficient. □

Note that, by the strict convexity of the function $-\ln\det$, the H_{YY}-part of the optimal solution H of (P) is unique, and hence by the theorem, so is the K part of the dual solution. However, there may be several optimal u's, and there may be several optimal E's for (P') (and H's for (P)). Indeed, as we show below, for any nonnegative u there may be several associated matrices E, but there is a unique associated $K = K(u)$, which is defined as is $\tilde{K}$ in the proof above.

Lemma 2.3 *Let u be dual feasible (nonnegative with components adding to one). Then:*

a) There exists E satisfying

$$EZUZ^T = -YUZ^T; \tag{2.8}$$

b) $K(u) := YUY^T - EZUZ^TE^T$ is independent of which E satisfying (2.8) is chosen; and

c) $XUX^T - \bar{K}$ is positive semidefinite iff $K \preceq K(u)$.

Proof: a) Suppose not. Then there is some q with $ZUZ^Tq = 0$ but YUZ^Tq nonzero. But then

$$\begin{aligned} 0 &\le \begin{pmatrix} q \\ p \end{pmatrix}^T \begin{bmatrix} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T \end{bmatrix} \begin{pmatrix} q \\ p \end{pmatrix} \\ &= q^TZUZ^Tq + 2p^TYUZ^Tq + p^TYUY^Tp, \end{aligned}$$

which is negative for p a sufficiently small negative multiple of YUZ^Tq, a contradiction.

b) Suppose E and E' satisfy (2.8). Then we have $(E-E')ZUZ^T = 0$ and hence

$$EZUZ^TE^T - E'ZUZ^T(E')^T = (E-E')ZUZ^TE' + EZUZ^T(E-E')^T = 0,$$

so that $K(u)$ is uniquely defined.

c) For any q and p, and any E satisfying the equation in (a),

$$\begin{array}{rcl} \left(\begin{array}{c} q \\ p \end{array}\right)^T (XUX^T - \bar{K}) \left(\begin{array}{c} q \\ p \end{array}\right)^T & = & \left(\begin{array}{c} q \\ p \end{array}\right)^T \left(\left[\begin{array}{cc} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T - K \end{array}\right]\right) \left(\begin{array}{c} q \\ p \end{array}\right)^T \\ & = & \left(\begin{array}{c} q \\ p \end{array}\right)^T \left[\begin{array}{cc} ZUZ^T & -ZUZ^TE^T \\ -EZUZ^T & EZUZ^TE \end{array}\right] \left(\begin{array}{c} q \\ p \end{array}\right)^T \\ & + & \left(\begin{array}{c} q \\ p \end{array}\right)^T \left[\begin{array}{cc} 0 & 0 \\ 0 & K(u) - K \end{array}\right] \left(\begin{array}{c} q \\ p \end{array}\right)^T \\ & = & (q - E^Tp)^T ZUZ^T (q - E^Tp) + p^T(K(u) - K)p, \end{array}$$

and the result follows. □

The lemma implies that, for any dual feasible u, we can assume without loss of generality that $K = K(u)$, because the latter is feasible and yields at least as good an objective value. We therefore write $g(u)$ for $g(u, K(u))$, abusing notation slightly. It is easy to see that g is concave on the set of feasible u's. If u and u' are dual feasible, and $\bar{u} := (1-\lambda)u + \lambda u'$ for $0 \leq \lambda \leq 1$, then $(\bar{u}, (1-\lambda)K(u) + \lambda K(u'))$ is feasible in (D), and hence $g(\bar{u}) \geq \ln\det((1-\lambda)K(u) + \lambda K(u'))$. The result now follows from the concavity of $\ln\det$. The problem of maximizing $g(u, K(u))$ subject to u lying in the unit simplex is the formulation used earlier for the D_k-optimal design problem: see, e.g., Atwood [3] or Silvey and Titterington [11].

We have seen above two definitions of the axis matrix E: one from the primal problem (P), $E = H_{YY}^{-1}H_{ZY}^T$, and one from a dual feasible solution as in (2.8). We now show that the first notion implies the second when feasible solutions H and u satisfy optimality condition (a): $H \bullet (XUX^T - \bar{K}) = 0$. (Here $\bar{K}$ is defined using $K = K(u)$.)

Indeed, if this equation holds, it also holds when H_{ZZ} is replaced by $E^TH_{YY}E$, where $E = H_{YY}^{-1}H_{ZY}^T$, which maintains positive semidefiniteness. But then

$$\begin{array}{rcl} 0 & = & (\left[\begin{array}{c} E^T \\ I \end{array}\right] H_{YY}[E \quad I]) \bullet \left[\begin{array}{cc} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T - K \end{array}\right] \\ & = & H_{YY} \bullet ([E \quad I] \left[\begin{array}{cc} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T - K \end{array}\right] \left[\begin{array}{c} E^T \\ I \end{array}\right]). \end{array}$$

Since H_{YY} is positive definite and the second matrix above is positive semidefinite, the latter must be zero. This then implies (e.g., by considering the positive semidefinite square root of $XUX^T - \bar{K}$) that

$$[E \quad I] \left[\begin{array}{cc} ZUZ^T & ZUY^T \\ YUZ^T & YUY^T - K \end{array}\right] = 0,$$

which implies (2.8).

Definition 2.1 *We call a dual feasible point (u, K, E), i.e., $u \geq 0$, $e^T u = 1$, $YUZ^T = -E(ZUZ^T)$ and $K = YUY^T - EZUZ^T E^T \succ 0$,* ϵ-primal feasible *if $(y_i + Ez_i)^T K^{-1}(y_i + Ez_i) \leq (1+\epsilon)k$ for all i, and say that it satisfies the* ϵ-approximate optimality conditions *or it is an* ϵ-approximate optimal solution *if moreover $(y_i + Ez_i)^T K^{-1}(y_i + Ez_i) \geq (1-\epsilon)k$ whenever $u_i > 0$.*

Lemma 2.4 *Given a dual feasible solution (u, K, E) which is ϵ-primal feasible, we have $0 \leq g^* - g(u) \leq k\ln(1+\epsilon)$, where g^* is the optimal objective function value of (D) and $g(u) := g(u, K)$.*

Proof: The ϵ-primal feasibility implies that $((1+\epsilon)^{-1}K^{-1}, E)$ is feasible for the primal problem (P'). Then by weak duality we have

$$0 \leq g^* - g(u) \leq \bar{f}((1+\epsilon)^{-1}K^{-1}, E) - \ln\det K = k\ln(1+\epsilon).$$

□

3 The algorithm

In the following two sections, we will assume that u and u_+ satisfy the following assumption:

Assumption 3.1 *The dual feasible variable u satisfies $ZUZ^T \succ 0$ where $U :=$* Diag (u).

(We remark that the rank deficiency of ZU^*Z^T for the optimal solution u^* does not contradict this assumption's holding at all iterations, although some numerical instability might be expected.)

We will show that this assumption is not too restrictive by proposing a method for dealing with the rank-deficient case in Section 5. (Note that Atwood [3] suggests just reducing u_i to a very small positive value if making it zero during the algorithm would lead to rank deficiency.)

Using this assumption and the discussion in the previous section, we see that each dual feasible solution u is associated with a unique dual feasible variable $K = K(u) = YUY^T - YUZ^T(ZUZ^T)^{-1}ZUY^T$ and a unique axis matrix $E = E(u) = -(YUZ^T)(ZUZ^T)^{-1}$. As above, we will use $g(u)$ instead of $g(u, K)$. We can now motivate and describe our algorithm.

3.1 Updating variances

As shown in Section 8.2 of the appendix, the objective function g of (D) has gradient

$$\omega(u) := \nabla g(u) = \left((y_i + Ez_i)^T K^{-1}(y_i + Ez_i)\right)_{i=1}^{m}. \tag{3.9}$$

If we make an update such as

$$u_+ := (1-\tau)u + \tau e_j, \tag{3.10}$$

where e_j denotes the jth unit vector, rank-one update formulae lead to cheap updates of the objective function value and the gradient as we will discuss. Please note that the algebraic details of the derivations in this section are presented in the appendix (see Section 8.2) for the sake of brevity; notation is listed in Section 8.1. Let $\xi_i(u) := x_i^T(XUX^T)^{-1}x_i$ and $\zeta_i(u) := z_i^T(ZUZ^T)^{-1}z_i$. First, as proved in Lemma 8.1, any dual feasible solution u satisfies the following equality

$$\xi_i(u) = \omega_i(u) + \zeta_i(u), \quad i = 1, \ldots, m. \tag{3.11}$$

(Those readers familiar with the MVEE problem will notice that the MVEE problem is a special case in which we have $k = n$, $\omega(u) = \xi(u)$ and $\zeta(u) = 0$.) Now let $\lambda := \frac{\tau}{1-\tau}$ and $\mu := \frac{\lambda}{1+\lambda\zeta_j(u)}$; then we have

$$(ZU_+Z^T)^{-1} = \frac{1}{1-\tau}((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_jz_j^T(ZUZ^T)^{-1}), \tag{3.12}$$

$$E_+ = E - \mu(y_j + Ez_j)((ZUZ^T)^{-1}z_j)^T, \tag{3.13}$$

and

$$K_+ = (1-\tau)(K + \mu(y_j + Ez_j)(y_j + Ez_j)^T). \tag{3.14}$$

This leads to

$$\ln\det K_+ = \ln\det K - k\ln(1+\lambda) + \ln(1 + \mu\omega_j(u)), \tag{3.15}$$

whose derivative is

$$\frac{\partial \ln\det K_+}{\partial \lambda} = \frac{-k}{(1+\lambda)(1+\lambda\zeta_j(u))(1+\lambda\xi_j(u))}\left(a\lambda^2 - 2b\lambda + c\right), \tag{3.16}$$

where $a := \zeta_j(u)\xi_j(u) \geq 0$, $b := -\zeta_j(u) - \frac{\omega_j(u)}{2} + \frac{\omega_j(u)}{2k} \leq 0$, and $c := 1 - \frac{\omega_j(u)}{k}$. The multiplier on the left of the quadratic is negative for all feasible values of λ. We can find the best step size τ^* (or λ^*) by investigating the roots of the quadratic equation $a\lambda^2 - 2b\lambda + c = 0$ and the boundary condition ($\lambda^* \geq -u_j$) arising from the nonnegativity of the dual feasible solutions. Once we determine the step size τ, we can find $\omega(u_+)$ and $\zeta(u_+)$ cheaply (in $O(mn)$ work) from

$$\omega_i(u_+) = \frac{1}{1-\tau}\left(\omega_i(u) - \eta\omega_{ij}^2(u) - 2\eta\zeta_{ij}(u)\omega_{ij}(u) + \eta\mu\omega_j(u)\zeta_{ij}^2(u)\right),$$

and

$$\zeta_i(u_+) = \frac{1}{1-\tau}\left(\zeta_i(u) - \mu\zeta_{ij}(u)^2\right), \tag{3.17}$$

where $\eta := \frac{\lambda}{1+\lambda\xi_j(u)}$, $\xi_{ij}(u) := x_i^T(XUX^T)^{-1}x_j$, $\zeta_{ij}(u) := z_i^T(ZUZ^T)^{-1}z_j$ and $\omega_{ij}(u) := (y_i + Ez_i)^TK^{-1}(y_j + Ez_j)$. For this, we need only compute inner products once we have $(XUX^T)^{-1}x_j$, $(ZUZ^T)^{-1}z_j$, and $K^{-1}(y_j + EZ_j)$, all of which are easy to obtain if we maintain a Cholesky factor $L = \begin{bmatrix} L_{ZZ} & 0 \\ L_{YZ} & L_{YY} \end{bmatrix}$ of XUX^T, since then L_{ZZ} is a Cholesky factor of ZUZ^T and L_{YY} of K.

Note that (3.17) gives in particular

$$\zeta_j(u_+) = \frac{(1-\mu\zeta_j(u))\zeta_j(u)}{1-\tau} = \frac{(1+\lambda)\zeta_j(u)}{1+\lambda\zeta_j(u)} \tag{3.18}$$

and it is easy to see that similar arguments lead to

$$\xi_j(u_+) = \frac{(1+\lambda)\xi_j(u)}{1+\lambda\xi_j(u)}. \tag{3.19}$$

3.2 Algorithm with away steps

Consider the following Frank-Wolfe type algorithm:

```
Algorithm 1: Computes an $\epsilon$-approximate optimal solution for $(D)$.
Input: $X \in \mathbf{R}^{n\times m}$, $k \in \{1,\ldots,n\}$ and $\epsilon > 0$.
Step 0. Initialize $u = e/m$. Compute $K$, $E$, $\omega(u)$, and $\zeta(u)$.
Step 1. Find $s := \arg\max\{\omega_i(u) - k\}$, $t := \arg\min\{\omega_i(u) - k : u_i > 0\}$.
    If $\omega_s(u) - k \le \epsilon k$ and $\omega_t(u) - k \ge -\epsilon k$,
        STOP: $u$ is an $\epsilon$-approximate optimal solution.
    Else,
        if $\omega_s(u) - k > k - \omega_t(u)$, go to Step 2;
        else, go to Step 3.
Step 2. Replace $u$ by $u_+ := (1-\tau)u + \tau e_s$, where $\tau > 0$ is chosen as in
        Section 3.1 to maximize $g$. Go to step 4.
Step 3. Replace $u$ by $u_+ := (1-\tau)u + \tau e_t$, where now $\tau$ is chosen from
        negative values to maximize $g$ subject to $u_+$ remaining feasible.
Step 4. Update $K$, $E$, $\omega(u)$, and $\zeta(u)$. Go to Step 1.
```

The algorithm starts with a feasible dual solution $u = e/m$ and stops when an ϵ-approximate optimal solution is found. Theoretically, we are able to prove that the initial solution described above is not far from the optimal one in some sense. In practice, using the procedure described by [7] provides better initial solutions but without any performance guarantee. Until an ϵ-approximate optimal solution is found, a Frank-Wolfe type step such as (3.10) is taken at each iteration (Steps 2,3, and 4 above). At each iteration one of the m vertices of the unit simplex is chosen and the new solution is obtained by either moving towards (called an increase or add step) or away from (called a decrease or drop step) this vertex. (Away steps were introduced by Wolfe [13].) There are alternative ways to choose the vertex. Algorithm 1 picks one of the two vertices which maximize or minimize a linear approximation to g. (Choosing one of these two vertices has both theoretical and practical advantages as we will discuss later.) Once this decision is made, the best step size is found by a line search on the line segment joining our current iterate to the optimal solution of the linearized problem. Alternatively, we can calculate the improvement in the objective function value g for each vertex and choose the best one. This version will be referred as Algorithm 1ALL. We will discuss the practical implications of these decisions in Section 6.

4 Convergence Analysis of the Algorithm

We discuss the global and local convergence properties of the algorithm in the next two subsections.

4.1 Global convergence of the algorithm

The following lemma will be very useful in our analysis.

Lemma 4.1 *For any dual feasible solution u satisfying Assumption 3.1, we have*

$$\sum_{i=1}^{m} u_i \omega_i(u) = k. \tag{4.20}$$

Proof: Using (3.11)

$$\begin{aligned}
\sum_{i=1}^{m} u_i \omega_i(u) &= \sum_{i=1}^{m} u_i \left(\xi_i(u) - \zeta_i(u)\right) \\
&= \sum_{i=1}^{m} u_i \xi_i(u) - \sum_{i=1}^{m} u_i \zeta_i(u) \\
&= \sum_{i=1}^{m} u_i x_i^T (XUX^T)^{-1} x_i - \sum_{i=1}^{m} u_i z_i^T (ZUZ^T)^{-1} z_i \\
&= XUX^T \bullet (XUX^T)^{-1} - ZUZ^T \bullet (ZUZ^T)^{-1} \\
&= n - (n-k) = k. \quad \square
\end{aligned}$$

If we look closely at the algorithm described in the previous section, we can identify four different types of iterations. Let u^l be the dual feasible solution at hand at iteration number l, e_{j_l} be the vertex that we use in our update and τ_l be the step size associated with this update. We refer to iteration l as

- an *increase step* if $u^l_{j_l} > 0$ and $\tau_l > 0$,
- an *add step* if $u^l_{j_l} = 0$ and $\tau_l > 0$,
- a *decrease step* if $u^l_{j_l} > 0$ and $\frac{-u^l_{j_l}}{1-u^l_{j_l}} < \tau_l < 0$, and

- a *drop step* if $u^l_{j_l} > 0$ and $\tau_l = \frac{-u^l_{j_l}}{1-u^l_{j_l}}$.

Note that after a drop step we have $u^{l+1}_{j_l} = 0$. In a drop step, we may not be able to improve the objective function as much as we desire. Fortunately, the number of drop steps is bounded above by the number of add steps plus a constant (m, the number of positive components of the initial solution), and hence studying only the first three types of steps will be enough to obtain convergence results.

The following lemma gives a bound on the improvement obtained at each iteration, assuming that all the quantities $\xi_i(u^l)$ are uniformly bounded by some positive C at all iterations. The global convergence estimate then depends on this constant C. In practice it appears that these quantities are bounded by a reasonable constant; unfortunately, we have not been able to establish global convergence without this assumption.

Lemma 4.2 *Let u^l be the dual solution at the l^{th} iteration of the algorithm and $g_l := g(u^l)$. Assume that u^l satisfies Assumption 3.1 for all $l = 1, 2, \dots$ Let ϵ_l be the smallest number such that u^l satisfies the ϵ_l-approximate optimality conditions. If for some positive constant C*

$$\xi(u^l) \leq Ce \ for \ all \ l = 1, 2, \dots,$$

then we have

$$g_0 > -\infty, \ \epsilon_0 \leq m - 1, \tag{4.21}$$

$$\Delta_l = g_{l+1} - g_l \geq \ln(1 + \hat{c}\epsilon_l^2), \ l = 0, 1, \dots, \tag{4.22}$$

whenever l is not a drop step, where $\hat{c} = \hat{c}(C) > 0$, and

$$\delta_l = g^* - g_l \leq k \ln(1 + \epsilon_l), \ l = 0, 1, \dots. \tag{4.23}$$

Proof: From Lemma 4.1 we have

$$\frac{1}{m}\sum_{i=1}^{m} \omega_i(u^0) = k.$$

By definition of ϵ_0,

$$k(1 + \epsilon_0) = \max\{\omega_i(u^0) | i = 1, .., m\} \leq km,$$

or

$$k(1 - \epsilon_0) = \min\{\omega_i(u^0) | i = 1, .., m\} \geq 0,$$

which implies $\epsilon_0 \leq \max\{1, m - 1\}$, and hence (4.21) holds.

For simplicity of notation, let $j := j_l$. In order to prove (4.22), we can use (3.18) and (3.19) to get

$$\begin{aligned} \omega_j(u^{l+1}) &= \xi_j(u^{l+1}) - \zeta_j(u^{l+1}) \\ &= \frac{(1+\lambda_l)\xi_j(u^l)}{1 + \lambda_l \xi_j(u^l)} - \frac{(1+\lambda_l)\zeta_j(u^l)}{1 + \lambda_l \zeta_j(u^l)} \\ &= \frac{(1+\lambda_l)\omega_j(u^l)}{(1 + \lambda_l \xi_j(u^l))(1 + \lambda_l \zeta_j(u^l))}. \end{aligned} \tag{4.24}$$

Now, let's assume that iteration l is an add or increase step, i.e., $j = \arg\max_i \omega_i(u^l)$ and $\omega_j(u^l) = (1+\epsilon_l)k$, so that $\hat{\lambda}_l$ (the optimal step length) is positive. Then for any λ_l such that $0 \leq \lambda_l \leq \hat{\lambda}_l$,

$$\begin{aligned} k &\leq \frac{(1+\lambda_l)\omega_j(u^l)}{(1+\lambda_l\xi_j(u^l))(1+\lambda_l\zeta_j(u^l))} \\ &\leq \frac{(1+\lambda_l)\omega_j(u^l)}{(1+\lambda_l\zeta_j(u^l))^2}. \end{aligned} \tag{4.25}$$

Therefore we have

$$(1+\lambda_l\zeta_j(u^l))^2 \quad \leq \quad (1+\lambda_l)\omega_j(u^l)/k = (1+\lambda_l)(1+\epsilon_l),$$

which gives

$$1+\lambda_l\zeta_j(u^l) \quad \leq \quad (1+\lambda_l)\sqrt{1+\epsilon_l}. \tag{4.26}$$

Note that since ϵ_l is bounded above $((1+\epsilon_l)k = \omega_j(u^l) \leq \xi(u^l) \leq C)$ we can find a constant $c_1 > 0$ such that $\sqrt{1+\epsilon_l} \geq 1 + c_1\epsilon_l$, and using this and (3.15), we obtain (for $\lambda_l < 1/k$)

$$\begin{aligned} g_{l+1} - g_l &= \ln\{(1+\lambda_l)^{-k}\left(1+\frac{\lambda_l\omega_j(u^l)}{1+\lambda_l\zeta_j(u^l)}\right)\} \\ &\geq \ln\{(1-k\lambda_l)\left(1+\frac{\lambda_l k(1+\epsilon_l)}{(1+\lambda_l)\sqrt{1+\epsilon_l}}\right)\} \\ &= \ln\{(1-k\lambda_l)\left(1+\frac{\lambda_l k\sqrt{1+\epsilon_l}}{1+\lambda_l}\right)\} \\ &\geq \ln\{(1-k\lambda_l)\left(1+\frac{\lambda_l k(1+c_1\epsilon_l)}{1+\lambda_l}\right)\}. \end{aligned} \tag{4.27}$$

Now choose λ' and c_0 positive and small enough such that for all $0 \leq \lambda \leq \lambda'$ the following inequality holds:

$$(1-k\lambda)(1+\frac{\lambda k(1+c_1\epsilon_l)}{1+\lambda}) \geq 1 + \lambda c_0\epsilon_l. \tag{4.28}$$

For $\lambda_l = \min\{\hat{\lambda}, \lambda'\}$, this leads to the following bound:

$$g_{l+1} - g_l \quad \geq \quad \ln(1+\lambda c_0\epsilon_l). \tag{4.29}$$

Note that (4.28) can be rewritten as

$$\begin{aligned} (1-k\lambda)(1+\lambda+\lambda k+\lambda k c_1\epsilon_l) &\geq (1+\lambda c_0\epsilon_l)(1+\lambda), \text{ or} \\ \lambda k c_1\epsilon_l - k\lambda^2 - k^2\lambda^2 - k^2\lambda^2 c_1\epsilon_l &\geq \lambda c_0\epsilon_l + \lambda^2 c_0\epsilon_l, \text{ or} \\ \lambda^2(c_0\epsilon_l + k + k^2 + k^2c_1\epsilon_l) + \lambda(c_0\epsilon_l - kc_1\epsilon_l) &\leq 0. \end{aligned} \tag{4.30}$$

We can assume that $\lambda' = \frac{-c_0\epsilon_l + kc_1\epsilon_l}{c_0\epsilon_l + k + k^2 + k^2 c_1 \epsilon_l}$ and choose $c_0 < kc_1$, and have the following bound for some constant c_2:

$$\lambda' \geq c_2\epsilon_l > 0. \tag{4.31}$$

On the other hand, since $\hat{\lambda}$ is the optimal step length we have

$$\begin{aligned} k = \omega_j(u^{l+1}) &= \frac{(1+\hat{\lambda})\omega_j(u^l)}{(1+\hat{\lambda}\xi_j(u^l))(1+\hat{\lambda}\zeta_j(u^l))} \\ &\geq \frac{(1+\hat{\lambda})\omega_j(u^l)}{(1+\hat{\lambda}\xi_j(u^l))^2}. \end{aligned}$$

Therefore we have

$$(1+\hat{\lambda}\xi_j(u^l))^2 \geq \omega_j(u^l)/k = 1+\epsilon_l,$$

which leads to

$$1+\hat{\lambda}\xi_j(u^l) \geq 1 + c_1\epsilon_l$$

and hence

$$\hat{\lambda} \geq \frac{c_1}{\xi_j(u^l)}\epsilon_l \geq \frac{c_1}{C}\epsilon_l = c_3\epsilon_l. \tag{4.32}$$

Equations (4.31) and (4.32) can be combined to obtain

$$\lambda_l = \min\{\lambda', \hat{\lambda}\} \geq c_4\epsilon_l > 0, \tag{4.33}$$

where $c_4 = \min\{c_2, c_3\}$. This inequality together with (4.29) gives (4.22) whenever iteration l is an add or increase step. The proof of the case where iteration l is a decrease step is very similar and presented in Section 8.3 of the appendix.

Finally, (4.23) follows from Lemma 2.4. □

Lemma 4.3 *Let $\epsilon \in (0,1)$. Under the assumptions of Lemma 4.2, Algorithm 1 obtains an ϵ-approximate optimal solution in at most*

$$L(\epsilon) = \mathcal{O}(m + k(\epsilon^{-1} + \ln k + \ln\ln m)) \tag{4.34}$$

iterations.

Note that the "big oh" here and in Theorem 4.1 below contains constants that depend on C in Lemma 4.2, which relies on Assumption 3.1.
Proof: We will first show that

$$L(1) := \min\{l : \epsilon_l \leq 1\} = \mathcal{O}(k(\ln k + \ln\ln m)). \tag{4.35}$$

First note that as long as $\epsilon_l \geq 1$, the l^{th} iteration of Algorithm 1 can only be an add or increase step. Furthermore, $\epsilon_l \geq 1$, (4.22), and (4.23) imply that

$$\delta_l - \delta_{l+1} = \Delta_l \geq \ln(1 + \hat{c}\epsilon_l^2) \geq \ln(1 + \hat{c}\epsilon_l) \geq \tilde{c}\ln(1 + \epsilon_l) \geq \frac{\tilde{c}}{k}\delta_l$$

for some $\tilde{c} = \tilde{C}(c) = \min\{1, \log_2(1 + \hat{c})\} > 0$. Hence

$$\delta_{l+1} \leq (1 - \frac{\tilde{c}}{k})\delta_l,$$

which implies

$$\delta_l \leq \delta_0(1 - \frac{\tilde{c}}{k})^l \leq \delta_0 e^{-\tilde{c}l/k}. \tag{4.36}$$

From (4.21) and (4.23), we have

$$\delta_0 \leq k\ln(1 + \epsilon_0) \leq k\ln m. \tag{4.37}$$

$\epsilon_l \geq 1$ also implies that

$$\delta_l \geq \Delta_l \geq \bar{c} > 0, \tag{4.38}$$

where $\bar{c} = \ln(1 + \hat{c})$. Hence from (4.36), (4.37) and (4.38), we get

$$L(1) \leq \frac{k}{\tilde{c}}\ln\frac{k\ln m}{\bar{c}} \leq \mathcal{O}(k(\ln k + \ln\ln m)).$$

Now assume that $\epsilon_l \leq 1$, and let $h(\epsilon_l)$ be the number of add, increase, and decrease steps required to obtain an $\epsilon_l/2$-approximate optimal solution starting with an ϵ_l-approximate optimal solution. As long as $\epsilon_{l+h} \geq \epsilon_l/2$ and $l + h$ is an add, increase, or decrease step, we also have

$$\Delta_{l+h} \geq \ln(1 + \hat{c}(\epsilon_l/2)^2) \geq \frac{\hat{c}_1}{8}\epsilon_l^2 > 0,$$

where $\hat{c}_1 := \min\{4, \hat{c}\}$. Since we have $\delta_l \geq \delta_{l+1} \geq \ldots$, and

$$\delta_l \leq k\ln(1 + \epsilon_l) \leq k\epsilon_l,$$

we obtain the following bound:

$$h(\epsilon_l) \leq \frac{\delta_l}{\hat{c}_1\epsilon_l^2/8} \leq \frac{8k}{\hat{c}_1\epsilon_l}. \tag{4.39}$$

Applying this argument repeatedly, we conclude that we need at most

$$\begin{aligned} \mathcal{H}(\epsilon) &= h(\epsilon_l) + h(\epsilon_l/2) + \ldots + h(\epsilon_l/2^{\lceil\log\epsilon_l/\epsilon\rceil - 1}) \\ &\leq \frac{8k}{\hat{c}_1}\left[\frac{1}{\epsilon_l} + \frac{2}{\epsilon_l} + \ldots + \frac{2^{\lceil\log\epsilon_l/\epsilon\rceil - 1}}{\epsilon_l}\right] \leq \frac{16k}{\hat{c}_1\epsilon} = \mathcal{O}\left(\frac{k}{\epsilon}\right) \end{aligned} \tag{4.40}$$

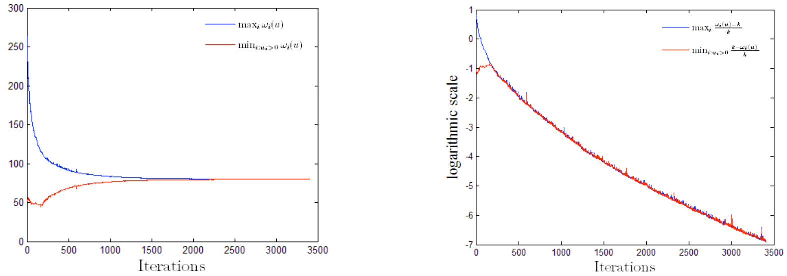

Figure 1: Behavior of the algorithm for $(m, n, k) = (10000, 100, 80)$.

add, increase, and decrease iterations to obtain an ϵ-approximate optimal solution starting with an ϵ_l-approximate optimal solution where $\epsilon_l \in (0, 1]$. Since the number of drop steps is bounded above by the number of add steps plus m (the number of positive components of the initial solution u^0), (4.34) is immediate. $\square$

We can implement this algorithm using rank-one update formulae so that each iteration takes $\mathcal{O}(nm)$ arithmetic operations and comparisons. Hence the following theorem follows from Lemma 4.3.

Theorem 4.1 *Let $\epsilon \in (0, 1)$. Under the assumptions of Lemma 4.2, Algorithm 1 finds an $\epsilon-$approximate optimal solution to the MAEC problem in*

$$N(\epsilon) = \mathcal{O}(knm(\epsilon^{-1} + \ln k + \ln \ln m) + nm^2) \tag{4.41}$$

arithmetic operations and comparisons.

4.2 Local convergence of the algorithm

In this section, we will show that Algorithm 1 is locally linearly convergent, i.e., the number of iterations grows with $\mathcal{O}(\ln \epsilon^{-1})$ not $\mathcal{O}(\epsilon^{-1})$ asymptotically under certain assumptions. The typical behavior of the algorithm as illustrated in Figure 1 illustrates this property. Unfortunately, this bound depends on the data of the problem as well as the dimensions and the constant C in Lemma 4.2, and so does not provide a global complexity bound better than that above.

We use the following perturbation of (P'):

$$\begin{array}{lrcl} & \min_{H' \succ 0} \ \bar{f}(H', E) & := & -\ln \det H' \\ (P'(\kappa)) & (y_i + Ez_i)^T H' (y_i + Ez_i) & \leq & k + \kappa_i, \quad i = 1, \ldots, m. \end{array}$$

Given a dual feasible u which with its associated $K = K(u)$ and $E = E(u)$ satisfies the γ-approximate optimality conditions, let $H' = K^{-1}$ and define $\kappa := \kappa(u, \gamma) \in \mathbf{R}^m$ by

$$\kappa_i := \begin{cases} \gamma k & \text{if } u_i = 0 \\ (y_i + Ez_i)^T K^{-1} (y_i + Ez_i) - k & \text{else.} \end{cases}$$

Observe that each component of κ has absolute value at most γk, and that this property may fail if we merely assume that (u, K, E) is γ-primal feasible. Moreover using (4.1),

$$u^T \kappa = \sum_{i:u_i>0} u_i \kappa_i = u^T \omega(u) - ke^T u = k - k = 0. \qquad (4.42)$$

Lemma 4.4 *Suppose (u, K, E) satisfies the γ-approximate optimality conditions. Then (H', E) is optimal in $(P'(\kappa(u, \gamma)))$.*

Proof: We note that (H', E) is feasible. Furthermore, u provides the required vector of Lagrange multipliers that satisfy the necessary and sufficient conditions for optimality given in the proof of Theorem 2.1 for $(P'(\kappa(u, \gamma)))$. $\square$

As discussed in Section 2 and proved in Lemma 2.1, there is an equivalent convex problem, say $(P(\kappa))$, to $(P'(\kappa))$. Let $\phi(\kappa)$ denote the value function, the optimal value of $(P(\kappa))$. Then ϕ is convex, and if u' is any vector of Lagrange multipliers for the optimal solution of $(P(\kappa))$, then u' is a subgradient of ϕ at κ. In particular, if u_* is any vector of Lagrange multipliers for the optimal solution of (P), then u_* is a subgradient of ϕ at 0.

For any u satisfying the γ-approximate optimality conditions and $\kappa := \kappa(u, \gamma)$,

$$\begin{array}{rcl} g(u) = \bar{f}(K^{-1}, E) = \phi(\kappa) & \geq & \phi(0) + u_*^T \kappa \\ & = & g^* + (u_* - u)^T \kappa \\ & \geq & g^* - \|u - u_*\| \|\kappa\|. \end{array} \qquad (4.43)$$

Here the last equality follows from (4.42). We have already noted that $\|\kappa\| \leq k\sqrt{m}\gamma$. To obtain an improvement on Lemma 4.2, we would like to prove that

$$\|u - u_*\| \leq L\|\kappa\| \leq Lk\sqrt{m}\gamma \qquad (4.44)$$

whenever $\|\kappa\|$ is sufficiently small. We will use the following assumption:

Assumption 4.1 *The strong second-order sufficient condition for local optimality and the linear independence of the active constraints hold for the optimal solution (H'^*, E^*) for problem (P') and the corresponding multipliers u_*.*

We have shown that the optimal solutions for problem $(P'(\kappa))$ and points that satisfy the corresponding KKT system coincide (see the proof of Theorem 2.1). If Assumption 4.1 holds, we know that u_*, the vector of multipliers for the optimal solution, is unique. Furthermore, the set-valued map S_{KKT}, which maps a perturbation vector κ to the set of optimal solutions for problem $(P'(\kappa))$ (and corresponding solutions for $(P(\kappa))$) and their corresponding multipliers, is locally single-valued and locally upper Lipschitz around $(0, (H'^*, E^*), u_*)$ (see Robinson [9] or Theorem 4.2 of Dontchev and Rocakfellar [4]). Also, the set-valued map X_{KKT}, which maps a perturbation vector κ to the set of optimal solutions is single-valued around $(0, (H'^*, E^*))$ (see Corollary 3.5 in Dontchev and Rocakfellar [4]). There exist neighborhoods V around 0 and $W_1 \times W_2$ around $((H'^*, E^*), u_*)$, such that the S_{KKT} map is single-valued in $W_1 \times W_2$ for all

$\kappa \in V$ and the X_{KKT} map is single-valued in W_1 for all $\kappa \in V$. Therefore, when γ is small enough so that κ lies in V, there must exist a pair of optimal solutions $(\hat{H}', \hat{E})$ for $(P'(\kappa))$ (and also $\hat{H}$ for $(P(\kappa))$) and multipliers $\hat{u}$ such that $(\hat{H}', \hat{E})$ is the only solution in W_1 for $(P'(\kappa))$ and $((\hat{H}', \hat{E}), \hat{u})$ is the only solution-multiplier pair in $W_1 \times W_2$. If $\hat{u} = u$, the local Lipschitz property of the S_{KKT} map provides that $\|u - u_*\| \le L\|\kappa\| \le Lk\sqrt{m}\gamma$ and we are done. Now assume that $\hat{u} \neq u$. There are two cases, $(H', E) = (\hat{H}', \hat{E})$ and $(H', E) \neq (\hat{H}', \hat{E})$. We will show that both cases lead to a contradiction and hence $\hat{u}$ must be equal to u. When $(H', E) \neq (\hat{H}', \hat{E})$, we can come up with two different solutions H and $\hat{H}$ for the convex problem $(P'(\kappa))$ which are both optimal. Then any convex combination of the points H and $\hat{H}$ must be optimal, too. We can find an optimal solution $\tilde{H}$ arbitrarily close to $\hat{H}$; hence we can find an optimal solution $(\tilde{H}', \tilde{E})$ for $(P'(\kappa))$, which is arbitrarily close to $(\hat{H}', \hat{E})$. This violates the local single-valuedness property of the X_{KKT} map at $(\hat{H}', \hat{E})$, so we must have $(H', E) = (\hat{H}', \hat{E})$. On the other hand, if we have two different vectors of multipliers, u and $\hat{u}$, both corresponding to the primal optimal solution (H', E), then we can find another vector of multipliers, say $\tilde{u}$, arbitrarily close to $\hat{u}$. This violates the local uniqueness property of the S_{KKT} map and hence leads to a contradiction. Hence, we have shown that $\hat{u} = u$ whenever Assumption 4.1 holds and (4.44) is valid.

From (4.44) and (4.43), we conclude

Proposition 4.1 *If Assumption 4.1 holds, there is some constant $M > 0$ (depending on the data of problem (P)) such that, whenever (u, K, E) is a γ-approximate optimal solution for some sufficiently small γ, we have*

$$g^* - g(u) \le M\gamma^2. \tag{4.45}$$

Let h and $\mathcal{H}$ be defined as in the proof of Lemma 4.3. Applying Proposition 4.1 instead of Lemma 4.2 in (4.39), we obtain

$$h(\gamma) \le \frac{M\gamma^2}{\hat{c}\gamma^2/8} = \frac{8M}{\hat{c}} \text{ for sufficiently small } \gamma, \tag{4.46}$$

and this yields, using the argument above (4.40), the existence of a constant $Q > 0$ with

$$\mathcal{H}(\epsilon) \le Q + \frac{16M}{\hat{c}} \ln(1/\epsilon) \text{ for sufficiently small } \epsilon.$$

We therefore have

Theorem 4.2 *If Assumption 4.1 holds, then there exist data-dependent constants $\tilde{Q}$ and $\tilde{M}$ such that Algorithm 1 requires at most $\tilde{Q} + \tilde{M}\ln(1/\epsilon)$ iterations to obtain an ϵ-approximate optimal solution.*

5 Rank-Deficient Case

As we have briefly discussed above, providing a complete algorithm for the MAEC problem is problematic. Assume we are at the l^{th} iteration with

$ZU^{l-1}Z^T$ nonsingular, and produce a new iterate u^l with ZU^lZ^T singular (we will show below that this can only occur at a drop iteration). It is then far from clear how we can continue the algorithm, because our update formulae in Section 3 assume nonsingularity of ZU^hZ^T for $h = 1, 2, \ldots$. In this section we show how the algorithm can be modified to deal with this case.

Lemma 5.1 *Given* $Z = [z_1, \ldots, z_m] \in \mathbf{R}^{r\times m}$*,* $\text{range}\,(ZUZ^T) = \text{span}\,(\{z_i : u_i > 0\})$*. Hence* $\text{rank}\,(ZUZ^T) = \dim\text{span}\,(\{z_i : u_i > 0\})$*.*

Proof: It is a direct consequence of the singular value decomposition that for any matrix $\tilde{Z}$ we have $\text{range}\,(\tilde{Z}) = \text{range}\,(\tilde{Z}\tilde{Z}^T)$. Let Z^+ be a matrix whose columns are the elements of the set $\{z_i : u_i > 0\}$ and U^+ ($U^{+,1/2}$) be a diagonal matrix with (the square roots of) the positive u_i's on the diagonal. Substituting $\tilde{Z} = Z^+U^{+,1/2}$ gives the desired result since

$$\begin{aligned} \text{range}\,(ZUZ^T) &= \text{range}\,(Z^+U^+{Z^+}^T) = \text{range}\,(Z^+U^{+,1/2}) \\ &= \text{range}\,(Z^+) = \text{span}\,(\{z_i : u_i > 0\}). \end{aligned} \tag{5.47}$$

□

The result above shows that difficulties only occur at drop iterations. To simplify the discussion, we rename u^{l-1} as $\hat{u}$ and let $u = \frac{1}{1-\hat{u}_d}(\hat{u} - \hat{u}_d e_d)$, so that x_d is dropped. We suppose ZUZ^T is singular, so that there are many solutions to $YUZ^T = -EZUZ^T$. However, one is particularly easy to find:

Lemma 5.2 *Let* $\hat{E}$ *satisfy* $Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T = 0$*. If* $\text{rank}\,(Z\hat{U}Z^T) = r$ *and* $\text{rank}\,(ZUZ^T) = r - 1$*, then we have the following equalities:*

i. $y_d + \hat{E}z_d = 0$*;*

ii. $YUZ^T + \hat{E}ZUZ^T = 0$*;*

iii. $K(u) = \frac{1}{1-\hat{u}_d}K(\hat{u})$*; and*

iv. $\hat{u}_d\zeta_d(\hat{u}) = 1$*.*

Proof: First observe that $\hat{u} = (1 - \hat{u}_d)u + \hat{u}_d e_d$. Since ZUZ^T is singular and $XUX^T \succeq 0$, there exists a vector $w \neq 0 \in \mathbf{R}^r$ such that $ZUZ^Tw = 0$ and $YUZ^Tw = 0$. Then we have

$$\begin{aligned} \hat{u}_d(y_d + \hat{E}z_d)z_d^Tw &= \hat{u}_d y_d z_d^T w + \hat{u}_d\hat{E}z_d z_d^T w \\ &= (1 - \hat{u}_d)YUZ^Tw + \hat{u}_d y_d z_d^T w + \ldots \\ &\quad\ (1 - \hat{u}_d)\hat{E}ZUZ^Tw + \hat{u}_d\hat{E}z_d z_d^T w \\ &= Y\hat{U}Z^Tw + \hat{E}Z\hat{U}Z^Tw = (Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T)w = 0. \end{aligned}$$

Since $\hat{u}_d > 0$, we must have either $y_d + \hat{E}z_d = 0$ or $z_d^Tw = 0$. Assume that $z_d^Tw = 0$: then we have $Z\hat{U}Z^Tw = ((1 - \hat{u}_d)ZUZ^T + \hat{u}_d z_d z_d^T)w = 0$, which contradicts the nonsingularity of $Z\hat{U}Z^T$. Hence we must have $y_d + \hat{E}z_d = 0$ as claimed in (i). $Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T = 0$ and (i) imply that

$$\begin{aligned} YUZ^T + \hat{E}ZUZ^T &= \frac{1}{1-\hat{u}_d}(Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T - \hat{u}_d y_d z_d^T - \hat{u}_d \hat{E} z_d z_d^T) \\ &= \frac{1}{1-\hat{u}_d}(Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T - \hat{u}_d(y_d + \hat{E}z_d)z_d^T) = 0, \end{aligned}$$

from which we get (ii). Similarly, we have

$$\begin{aligned} K(u) &= \frac{1}{1-\hat{u}_d}(Y\hat{U}Y^T + \hat{E}Z\hat{U}Y^T - u_d(y_d + \hat{E}z_d)y_d^T) \\ &= \frac{1}{1-\hat{u}_d}(Y\hat{U}Y^T + \hat{E}Z\hat{U}Y^T) = \frac{1}{1-\hat{u}_d}K(\hat{u}), \end{aligned}$$

which proves (iii). In order to prove (iv), we know that $\operatorname{rank}(ZUZ^T) = r-1$ implies that $\det(ZUZ^T) = 0$. We have

$$\begin{aligned} \det(ZUZ^T) &= (1-\hat{u}_d)^{-r}\det(Z\hat{U}Z^T - \hat{u}_d z_d z_d^T) \\ &= (1-\hat{u}_d)^{-r}\det(Z\hat{U}Z^T)(1 - \hat{u}_d\zeta_d(\hat{u})) = 0. \end{aligned}$$

Since $\det(Z\hat{U}Z^T) > 0$, we must have (iv). Note that the reverse claim is also correct, i.e., $\hat{u}_d\zeta_d(\hat{u}) = 1$ implies that ZUZ^T is singular. □

This result can be generalized as follows:

Lemma 5.3 *Let $\hat{u}$ be a dual feasible solution that satisfies* $\operatorname{rank}(Z\hat{U}Z^T) = r$ *with associated matrix $\hat{E}$ such that $Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T = 0$. Define $R(\hat{u}) := \{i : \operatorname{rank}(Z\hat{U}Z^T - \hat{u}_i z_i z_i^T) < r\}$ and suppose $R \subseteq R(\hat{u})$ is such that $\sigma := 1 - \sum_{i\in R}\hat{u}_i > 0$. Let $u = \frac{1}{\sigma}(\hat{u} - \sum_{i\in R}\hat{u}_i e_i)$; then*

i. $YUZ^T + \hat{E}ZUZ^T = 0$ *and*

ii. $K(u) = \frac{1}{\sigma}K(\hat{u})$.

Proof: The proof is very similar to that of the previous lemma. We have $y_i + \hat{E}z_i = 0$ for all $i \in R$ from Lemma 5.2, so that

$$\begin{aligned} YUZ^T + \hat{E}ZUZ^T &= \frac{1}{\sigma}(Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T - \sum_{i\in R}\hat{u}_i(y_i + \hat{E}z_i)z_i^T) \\ &= \frac{1}{\sigma}(Y\hat{U}Z^T + \hat{E}Z\hat{U}Z^T) = 0, \end{aligned}$$

which gives (i). Similarly,

$$\begin{aligned} K(u) &= \frac{1}{\sigma}(Y\hat{U}Y^T + \hat{E}Z\hat{U}Y^T - \sum_{i\in R}u_i(y_i + \hat{E}z_i)y_i^T) \\ &= \frac{1}{\sigma}(Y\hat{U}Y^T + \hat{E}Z\hat{U}Y^T) = \frac{1}{\sigma}K(\hat{u}), \end{aligned}$$

and hence we get (ii). $\square$

Let u be the current iterate, and assume that we have $\hat{u}$ such that $Z\hat{U}Z^T$ is nonsingular and $u = \frac{1}{\sigma}(\hat{u} - \sum_{i \in R} \hat{u}_i e_i)$ for some index set $R \subseteq R(\hat{u})$ and $\sigma = 1 - \sum_{i \in R} \hat{u}_i$. Also suppose E satisfies $Y\hat{U}Z^T + EZ\hat{U}Z^T = 0$. If $R \neq R(\hat{u})$, it is immediate from Lemma 5.3 that replacing u by $\frac{1}{\sigma}(\hat{u} - \sum_{i \in R(\hat{u})} \hat{u}_i e_i)$ increases $\ln\det(K(u))$, so we can assume that $R = R(\hat{u})$. The lemma above shows that much of the information required related to the current iterate u follows from information for the associated vector $\hat{u}$ for which $Z\hat{U}Z^T$ is nonsingular. We refer to iterations where we drop points x_d for $d \in R$ as *deferred* updates since we maintain information such as ω, ζ, K, etc., for the vector $\hat{u}$, which is the dual vector before dropping the points x_d, $d \in R$.

Now, assume that we make an update of the form $u_+ = (1-\tau)u + \tau e_j$. If we set $\hat{\tau} = \frac{\tau\sigma}{1-\tau+\tau\sigma}$, then the update $\hat{u}_+ = (1-\hat{\tau})\hat{u} + \hat{\tau} e_j$ will satisfy $u_+ = \frac{1}{\sigma_+}(\hat{u}_+ - \sum_{i \in R} \hat{u}_{+i} e_i)$ where $\sigma_+ = 1 - \sum_{i \in R} \hat{u}_{i+} = \frac{\sigma}{1-\tau+\tau\sigma}$. We can assume that $j \notin R$ without loss of generality because $j \in R$ implies that $y_j + Ez_j = 0$ which in turn implies $\omega_j(\hat{u}) = 0$, and such a choice of j would never be made by our algorithm. Let us assume for now that $R(\hat{u}_+) = R$.

Letting $\hat{\lambda} = \frac{\hat{\tau}}{1-\hat{\tau}}$ (which implies that $\hat{\lambda} = \sigma\lambda$), and using the update formulae in Section 3, we obtain

$$K(u_+) \quad = \quad \frac{(1-\hat{\tau})}{\sigma(1-\hat{\tau}) + \hat{\tau}} \left(K(\hat{u}) + \frac{\hat{\lambda}}{1 + \hat{\lambda}\zeta_j(\hat{u})} (y_j + Ez_j)(y_j + Ez_j)^T \right).$$

We have

$$\begin{aligned} \ln\det K(u_+) &= \ln\det K(\hat{u}) - k \ln\left(\frac{\sigma(1-\hat{\tau}) + \hat{\tau}}{1-\hat{\tau}} \right) + \ln\left(1 + \frac{\hat{\lambda}\omega_j(\hat{u})}{1 + \hat{\lambda}\zeta_j(\hat{u})} \right) \\ &= \ln\det K(\hat{u}) - k \ln\left(\sigma + \hat{\lambda} \right) + \ln\left(1 + \frac{\hat{\lambda}\omega_j(\hat{u})}{1 + \hat{\lambda}\zeta_j(\hat{u})} \right), \end{aligned}$$

whose derivative is

$$\begin{aligned} \frac{\partial \ln\det K(u_+)}{\partial \hat{\lambda}} &= -\frac{k}{\sigma + \hat{\lambda}} + \frac{1}{1 + \frac{\hat{\lambda}\omega_j(\hat{u})}{1 + \hat{\lambda}\zeta_j(\hat{u})}} \left(\frac{\omega_j(\hat{u})}{1 + \hat{\lambda}\zeta_j(\hat{u})} - \frac{\hat{\lambda}\zeta_j(\hat{u})\omega_j(\hat{u})}{(1 + \hat{\lambda}\zeta_j(\hat{u}))^2} \right) \\ &= -\frac{k}{\sigma + \hat{\lambda}} + \frac{\omega_j(\hat{u})}{(1 + \hat{\lambda}(\zeta_j(\hat{u}) + \omega_j(u)))(1 + \hat{\lambda}\zeta_j(\hat{u}))} \\ &= \frac{-k}{(\sigma + \hat{\lambda})(1 + \hat{\lambda}\zeta_j(\hat{u}))(1 + \hat{\lambda}(\zeta_j(\hat{u}) + \omega_j(\hat{u})))} \left(\hat{a}\hat{\lambda}^2 - 2\hat{b}\hat{\lambda} + \hat{c} \right), \end{aligned} \tag{5.48}$$

where $\hat{a} := \zeta_j(\hat{u})(\zeta(\hat{u}) + \omega(\hat{u})) \geq 0$, $\hat{b} := -\zeta_j(\hat{u}) - \frac{\omega_j(\hat{u})}{2} + \frac{\omega_j(\hat{u})}{2k} \leq 0$, and $\hat{c} := 1 - \frac{\sigma\omega_j(\hat{u})}{k}$. Note that this derivative at $\hat{\lambda} = 0$ is positive or negative according as $\hat{c}$ is negative or positive, or equivalently according as $\sigma\omega_j(\hat{u})$ (which is $\omega_j(u) = (y_j + Ez_j)^T K(u)^{-1}(y_j + Ez_j)$ by Lemma 5.3) is greater or less than k, as in the full-rank case.

Now we can try to find the optimal step length that maximizes the objective function value of the new iterate u_+ as before. We need to be careful at this point. If the choice of the vertex x_j used in the update and the corresponding optimal step length results in an increase, decrease, or drop iteration, then we must have $R(\hat{u}_+) = R(\hat{u}) = R$, and hence performing this update is the best option we have available. (Note that a drop iteration cannot lose rank, because we have already dropped all such points by our assumption that $R = R(\hat{u})$.) On the other hand, if the current update turns out to be an add iteration, we may have $R(\hat{u}_+) \neq R$ and we can perform one of the deferred updates in the list R at the same time as we perform the add iteration. As we discuss below, an appropriate choice of step lengths then results in a new iterate u_+ identical to the current iterate u, but we will find a new $\hat{u}_+$ and a new matrix E_+ which satisfies $YU_+Z^T + E_+ZU_+Z^T = 0$. This feature allows the continued progress of the algorithm.

Assume we would like to perform an add iteration in which $\hat{u}_j$ will be increased from 0 to a positive value. This may result in a new iterate for which at least one of the deferred updates, say that corresponding to dropping x_d, may be performed without violating the nonsingularity assumption. If this is the case, we will consider a combined update which simultaneously adds x_j and drops x_d, with d in the deferred update list R. Therefore updates of the following form will be of interest:

$$\begin{aligned} \hat{u}_+ &= (1-\hat{\tau})\left(\frac{\hat{u} - \hat{u}_d e_d}{1-\hat{u}_d}\right) + \hat{\tau} e_j \\ &= \left(\frac{1-\hat{\tau}}{1-\hat{u}_d}\right)\hat{u} - \frac{(1-\hat{\tau})\hat{u}_d}{1-\hat{u}_d} e_d + \hat{\tau} e_j. \end{aligned}$$

Let $\hat{\lambda} = \frac{\hat{\tau}(1-\hat{u}_d)}{1-\hat{\tau}}$, so that we can write the update above as

$$\hat{u}_+ = (1 - \hat{u}_d + \hat{\lambda})^{-1}(\hat{u} - \hat{u}_d e_d + \hat{\lambda} e_j). \tag{5.49}$$

Let us see when such a deferred drop can be made. This will be the case exactly when the corresponding matrix $Z\hat{U}Z^T + \hat{\lambda} z_j z_j^T - \hat{u}_d z_d z_d^T$ is nonsingular (since the factor $1 - \hat{u}_d + \hat{\lambda}$ is always positive). Now we use

$$(Z\hat{U}Z^T + \hat{\lambda} z_j z_j^T)^{-1} = (Z\hat{U}Z^T)^{-1} - \hat{\lambda}(1 + \hat{\lambda}\zeta_j(\hat{u}))^{-1}(Z\hat{U}Z^T)^{-1} z_j z_j^T (Z\hat{U}Z^T)^{-1}$$

to see that the above matrix is nonsingular iff $1 \neq \hat{u}_d z_d^T (Z\hat{U}Z^T + \hat{\lambda} z_j z_j^T)^{-1} z_d = \hat{u}_d \zeta_d(\hat{u}) - \hat{\lambda}\hat{u}_d \zeta_{dj}(\hat{u})^2 (1 + \hat{\lambda}\zeta_j(\hat{u}))^{-1}$. Thus, using $\hat{u}_d \zeta_d(\hat{u}) = 1$, we see that x_d can be dropped exactly when $\zeta_{dj}(\hat{u})$ is nonzero. If all $\zeta_{dj}(\hat{u})$, $d \in R$, are zero, we proceed with the add iteration without performing any deferred updates. Suppose now that one of these quantities is nonzero. Then we choose a corresponding d and update $\hat{u}$ as in (5.49). Note that Lemmas 5.1 and 5.3

show that $Z\hat{U}Z^T - \hat{u}_d z_d z_d^T - \hat{u}_{d'} z_{d'} z_{d'}^T$ has rank $r-2$ for any two indices d and d' in R, so we can only perform at most one deferred update.

If $\hat{u}$ is updated as in (5.49), then we have

$$Y\hat{U}_+Y^T = (1-\hat{u}_d+\hat{\lambda})^{-1}(Y\hat{U}Y^T - \hat{u}_d y_d y_d^T + \hat{\lambda} y_j y_j^T), \tag{5.50}$$

$$Y\hat{U}_+Z^T = (1-\hat{u}_d+\hat{\lambda})^{-1}(Y\hat{U}Z^T - \hat{u}_d y_d z_d^T + \hat{\lambda} y_j z_j^T), \tag{5.51}$$

and

$$Z\hat{U}_+Z^T = (1-\hat{u}_d+\hat{\lambda})^{-1}(Z\hat{U}Z^T - \hat{u}_d z_d z_d^T + \hat{\lambda} z_j z_j^T). \tag{5.52}$$

We would like to find a matrix E_+ that satisfies $E_+Z\hat{U}_+Z^T + Y\hat{U}_+Z^T = 0$. Let us assume that it can be chosen of the form $E_+ = E + pq^T$ for some $p \in \mathbf{R}^k$ and $q \in \mathbf{R}^r$. Keeping in mind that $EZ\hat{U}Z^T = -Y\hat{U}Z^T$ and $y_d + Ez_d = 0$ (see Lemmas 5.3 and 5.2), we can find appropriate vectors p and q as follows. We would like to satisfy:

$$\begin{aligned} E_+Z\hat{U}_+Z^T &= -Y\hat{U}_+Z^T, \text{ or} \\ (E+pq^T)(Z\hat{U}Z^T - \hat{u}_d z_d z_d^T + \hat{\lambda} z_j z_j^T) &= -(Y\hat{U}Z^T - \hat{u}_d y_d z_d^T + \hat{\lambda} y_j z_j^T), \text{ or} \\ \hat{\lambda}(y_j + Ez_j)z_j^T &= -p(q^T(Z\hat{U}Z^T - \hat{u}_d z_d z_d^T + \hat{\lambda} z_j z_j^T)). \end{aligned}$$

Setting $p = y_j + Ez_j$, and $q = \pi(Z\hat{U}Z^T)^{-1}z_d$, this holds as long as

$$-\hat{\lambda} z_j = \pi z_d - \hat{u}_d \pi \zeta_d(\hat{u}) z_d + \hat{\lambda}\pi\zeta_{dj}(\hat{u})z_j = \hat{\lambda}\pi\zeta_{dj}(\hat{u})z_j$$

where we have used $\hat{u}_d\zeta_d(\hat{u}) = 1$ since $d \in R$. This requires that $\pi = -\frac{1}{\zeta_{dj}(\hat{u})}$, and we have therefore found a matrix

$$\begin{aligned} E_+ &= E + pq^T \\ &= E - \frac{1}{\zeta_{dj}(\hat{u})}(y_j + Ez_j)z_d^T(Z\hat{U}Z^T)^{-1}, \end{aligned}$$

which can be used in further arguments. Using again the fact that $\hat{u}_d\zeta_d(\hat{u}) = 1$ and also that $y_d + Ez_d = 0$, we get

$$\begin{aligned} K(\hat{u}_+) &= Y\hat{U}_+Y^T + E_+Z\hat{U}Y^T \\ &= (1-\hat{u}_d+\hat{\lambda})^{-1}(Y\hat{U}Y^T - \hat{u}_d y_d y_d^T + \hat{\lambda} y_j y_j^T + \ldots \\ &\qquad (E+pq^T)(Z\hat{U}Y^T - \hat{u}_d z_d y_d^T + \hat{\lambda} z_j y_j^T)) \\ &= (1-\hat{u}_d+\hat{\lambda})^{-1}(K(\hat{u}) + \hat{\lambda}(y_j+Ez_j)y_j^T - \ldots \\ &\qquad \frac{1}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)(-z_d^TE - \hat{u}_d\zeta_d(\hat{u})y_d^T + \hat{\lambda}\zeta_{dj}(\hat{u})y_j^T)) \\ &= (1-\hat{u}_d+\hat{\lambda})^{-1}(K(\hat{u}) + \hat{\lambda}(y_j+Ez_j)y_j^T - \ldots \\ &\qquad \frac{1}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)((1-\hat{u}_d\zeta_d(\hat{u}))y_d^T + \hat{\lambda}\zeta_{dj}(\hat{u})y_j^T)) \\ &= (1-\hat{u}_d+\hat{\lambda})^{-1}K(\hat{u}). \end{aligned} \tag{5.53}$$

Since x_d will be dropped, it appears that we will have $R_+ = R - \{d\}$ which yields $\sigma_+ = 1 - \sum_{i \in R_+} \hat{u}_{i+} = \frac{1-\hat{\tau}}{1-\hat{u}_d}\sigma + \hat{\tau} = \frac{\sigma+\hat{\lambda}}{1-\hat{u}_d+\hat{\lambda}}$. Therefore we can write the new objective function value as

$$\begin{aligned} \ln\det K(u_+) &= \ln\det \frac{1}{\sigma+\hat{\lambda}} K(\hat{u}) \\ &= \ln\det K(\hat{u}) - k\ln(\sigma+\hat{\lambda}), \end{aligned} \tag{5.54}$$

which is maximized at $\hat{\lambda} = \hat{\tau} = 0$. However, if we choose $\hat{\tau}$ to be zero, we find that $Z\hat{U}_+Z^T$ is singular. We deal with this situation by setting $\hat{\tau}$ positive, but then as we will see letting x_j be a deferred drop instead of x_d.

As before, we can update $\omega(\hat{u}_+)$ cheaply:

$$\begin{aligned} \omega_i(\hat{u}_+) &= (y_i + E_+z_i)^T K(\hat{u}_+)^{-1}(y_i + E_+z_i) \\ &= (1-\hat{u}_d+\hat{\lambda})\left(y_i + (E - \frac{1}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)z_d^T(ZUZ^T)^{-1})z_i\right)^T K(\hat{u})^{-1} \\ &\quad \ldots \left(y_i + (E - \frac{1}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)z_d^T(ZUZ^T)^{-1})z_i\right) \\ &= (1-\hat{u}_d+\hat{\lambda})\left(y_i + Ez_i - \frac{\zeta_{di}(\hat{u})}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)\right)^T K(\hat{u})^{-1} \\ &\quad \ldots \left(y_i + Ez_i - \frac{\zeta_{di}(\hat{u})}{\zeta_{dj}(\hat{u})}(y_j+Ez_j)\right) \\ &= (1-\hat{u}_d+\hat{\lambda})\left(\omega_i(\hat{u}) - \frac{2\zeta_{di}(\hat{u})}{\zeta_{dj}(\hat{u})}\omega_{ij}(\hat{u}) + \frac{\zeta_{di}^2(\hat{u})}{\zeta_{dj}^2(\hat{u})}\omega_j(\hat{u})\right). \end{aligned} \tag{5.55}$$

Furthermore we have

$$\begin{aligned} (Z\hat{U}_+Z^T)^{-1} &= (1-\hat{u}_d+\hat{\lambda})(Z\hat{U}Z^T - \hat{u}_d z_d z_d^T + \hat{\lambda} z_j z_j^T)^{-1} \\ &= (1-\hat{u}_d+\hat{\lambda})((Z\hat{U}Z^T)^{-1} + \frac{1+\hat{\lambda}\zeta_j(\hat{u})}{\hat{\lambda}\zeta_{dj}^2(\hat{u})}(Z\hat{U}Z^T)^{-1}z_d z_d^T(Z\hat{U}Z^T)^{-1} \\ &\quad \ldots - \frac{1}{\zeta_{dj}(\hat{u})}(Z\hat{U}Z^T)^{-1}(z_d z_j^T + z_j z_d^T)(Z\hat{U}Z^T)^{-1}), \end{aligned} \tag{5.56}$$

which leads to

$$\zeta_i(\hat{u}_+) = (1-\hat{u}_d+\hat{\lambda})\left(\zeta_i(\hat{u}) + \frac{(1+\hat{\lambda}\zeta_j(\hat{u}))\zeta_{di}^2(\hat{u})}{\hat{\lambda}\zeta_{dj}^2(\hat{u})} - \frac{2\zeta_{di}(\hat{u})\zeta_{ij}(\hat{u})}{\zeta_{dj}(\hat{u})}\right) \tag{5.57}$$

We have $\hat{u}_{+j}\zeta_j(\hat{u}_+) = \hat{\tau}\frac{(1-\hat{u}_d+\hat{\lambda})}{\hat{\lambda}} = 1$, which implies that $j \in R(\hat{u}_+)$. Therefore we can perform a combined step in which x_j is added while x_d is dropped and add j to the deferred update list. The new deferred update list $R_+ = R - \{d\} \cup \{j\}$ satisfies $R_+ = R(\hat{u}_+)$ since we cannot have any more new deferred updates. The choice of $\hat{\lambda}$ is irrelevant because the new feasible point u_+ has as its components not in R_+ scalings of those of $\hat{u} - \hat{u}_d e_d + \hat{\lambda} e_j$ and so is independent of $\hat{\lambda}$ and in fact equal to u. We prefer to use $\hat{\lambda} = \hat{u}_d$ since it simplifies the calculations (see (5.49)) and σ is unchanged.

Based on this discussion we can modify Algorithm 1 as follows:

```
Algorithm 2: The modified algorithm that computes an
             $\epsilon$-approximate optimal solution for (D).
Input: $X \in \mathrm{R}^{n \times m}$, $k \in \{1, \ldots, n\}$ and $\epsilon > 0$.
Step 0. Initialize $u = e/m$, $\hat{u} = u$, $R = \emptyset$ and $\sigma = 1$,
        compute $K(\hat{u})$, $E(\hat{u})$, $\omega(\hat{u})$, and $\zeta(\hat{u})$,
        and set $\omega(u) = \omega(\hat{u})$, and $\zeta(u) = \zeta(\hat{u})$.
Step 1. Find $s := \arg\max_i\{\omega_i(u) - k\}$, $t := \arg\min_i\{\omega_i(u) - k : u_i > 0\}$.
        If $\omega_s(u) - k \leq \epsilon k$ and $\omega_t(u) - k \geq -\epsilon k$,
            STOP: $u$ satisfies the $\epsilon$-approximate optimality conditions.
Step 2. If $\omega_t(u) = 0$, let $D = \{i : \hat{u}_i \zeta_i(\hat{u}) = 1\}$,
        and set $R = R \cup D$, $\sigma = 1 - \sum_{i \in R} \hat{u}_i$, $u = \sigma^{-1}\hat{u}$, $u(R) = 0$,
        $\omega(u) = \sigma\omega(\hat{u})$, and $\zeta(u) = \sigma\zeta(\hat{u})$. Go to Step 1.
Step 3. If $\omega_s(u) - k < k - \omega_t(u)$, set $j = t$ and go to Step 6. Else $j = s$.
Step 4. If $\hat{u}_j = 0$ and $R \neq \emptyset$, calculate $\zeta_{d'j}(\hat{u})$ for $d' \in R$,
        and find $d := \arg\max_{d' \in R}\{|\zeta_{d'j}(\hat{u})|\}$.
        If $\zeta_{dj}(\hat{u}) = 0$, go to Step 5.
        Otherwise, replace $\hat{u}$ by $\hat{u} - \hat{u}_d e_d + \hat{u}_d e_j$,
        set $R = R - \{d\} \cup \{j\}$, update $E(\hat{u})$, $\omega(\hat{u})$, and $\zeta(\hat{u})$,
        and set $\omega(u) = \sigma\omega(\hat{u})$ and $\zeta(u) = \sigma\zeta(\hat{u})$. Go to Step 1.
Step 5. Replace $\hat{u}$ by $\hat{u}_+ := (1 - \hat{\tau})\hat{u} + \hat{\tau} e_j$, where $\hat{\tau} > 0$ is chosen as in
        Section 5 to maximize $g(u)$. Go to Step 7.
Step 6. Replace $\hat{u}$ by $\hat{u}_+ := (1 - \hat{\tau})\hat{u} + \hat{\tau} e_j$, where now $\hat{\tau}$ is chosen from
        negative values to maximize $g(u)$ subject to $\hat{u}_+$ remaining feasible.
Step 7. Update $K(\hat{u})$, $E(\hat{u})$, $\omega(\hat{u})$, and $\zeta(\hat{u})$,
        set $\sigma = 1 - \sum_{i \in R} \hat{u}_i$, $u = \sigma^{-1}\hat{u}$, $u(R) = 0$,
        $\omega(u) = \sigma\omega(\hat{u})$ and $\zeta(u) = \sigma\zeta(\hat{u})$. Go to Step 1.
```

This algorithm also starts with a dual feasible solution. We can use any initial solution u as long as $ZUZ^T \succ 0$. A good candidate is $u = e/m$ since the z_i's span R^r. Each feasible solution vector u is associated with a lifted solution vector $\hat{u}$ satisfying $u = \sigma^{-1}\hat{u}$ and $u_R = 0$. R is the set of deferred updates accumulated since the first iteration and σ is the weight of the remaining components of $\hat{u}$. We work with $\hat{u}$ for which $Z\hat{U}Z^T \succ 0$ by construction, although the solution that we are interested in is u for which ZUZ^T may be singular. As proved in Lemma 5.3, $\hat{u}$ provides all the information we need in order to calculate the variance and the objective function associated with u, so that we can figure out the next iterate and step length which maximizes $g(u)$ while taking a Frank-Wolfe step such as $u_+ = (1 - \tau)u + e_j$ (or equivalently, $\hat{u}_+ = (1 - \hat{\tau})\hat{u} + \hat{\tau} e_j$). At each iteration, we check whether we can add more deferred drop iterations, since each will increase $g(u)$. In some of the iterations, we are able to make a step where we exchange one of the points in the deferred set R with a new point. Although this step does not improve the objective function, it changes the axis of the cylinder at hand which may lead to better iterates. The following example demonstrates how the algorithm works on a toy problem. The cylinders generated at each iteration of the algorithm are illustrated below in Figure 2. Note that Algorithm 1 would fail at the first iteration for this example if we chose $j = 2$.

Example 5.1 *For* $X = \begin{bmatrix} 3 & 2 & 0 & 0 & 6 \\ 1 & 2 & 3 & 4 & 0 \end{bmatrix}$ *and* $k = 1$*, Algorithm 2 obtains a* 10^{-7}*-approximate optimal solution in 3 iterations as shown below:*

Iteration 0: Initialization $\hat{u}_0 = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \end{bmatrix}$ $\quad u_0 = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \end{bmatrix}$ $\omega(\hat{u}_0) = \begin{bmatrix} 0.889 & 0 & 2 & 3.556 & 8 \end{bmatrix}$ $\quad \omega(u_0) = \begin{bmatrix} 0.889 & 0 & 2 & 3.556 & 8 \end{bmatrix}$ $R_0 = \emptyset \quad E_0 = -1 \quad \sigma_0 = 1 \quad \hat{\tau}_0 = 0 \quad \tau_0 = -1 \quad j_0 = 2$
Iteration 1: Dropping x_2 is deferred $\hat{u}_1 = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \end{bmatrix}$ $\quad u_1 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ $\omega(\hat{u}_1) = \begin{bmatrix} 0.889 & 0 & 2 & 3.556 & 8 \end{bmatrix}$ $\quad \omega(u_1) = \begin{bmatrix} 0.445 & 0 & 1 & 1.778 & 4 \end{bmatrix}$ $R_1 = \{2\} \quad E_1 = -1 \quad \sigma_1 = 0.5 \quad \hat{\tau}_1 = N/A \quad \tau_1 = N/A \quad j_1 = 5 \quad d_1 = 2$
Iteration 2: x_2 is dropped and x_5 is deferred $\hat{u}_2 = \begin{bmatrix} 0 & 0 & 0.5 & 0 & 0.5 \end{bmatrix}$ $\quad u_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$ $\omega(\hat{u}_2) = \begin{bmatrix} 0.222 & 0.889 & 2 & 3.556 & 0 \end{bmatrix}$ $\quad \omega(u_2) = \begin{bmatrix} 0.111 & 0.445 & 1 & 1.778 & 0 \end{bmatrix}$ $R_2 = \{5\} \quad E_2 = 0 \quad \sigma_2 = 0.5 \quad \hat{\tau}_2 = 1 \quad \tau_2 = 1 \quad j_2 = 4$
Iteration 3: x_4 is added $\hat{u}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ $\quad u_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ $\omega(\hat{u}_3) = \begin{bmatrix} 0.063 & 0.25 & 0.563 & 1 & 0 \end{bmatrix}$ $\quad \omega(u_3) = \begin{bmatrix} 0.063 & 0.25 & 0.563 & 1 & 0 \end{bmatrix}$ $R_3 = \{5\} \quad E_3 = 0 \quad \sigma_3 = 1 \quad \hat{\tau}_3 = N/A \quad \tau_3 = N/A \quad j_3 = N/A$

Table 1: Execution of Algorithm 2 for a 2-D Toy Example

We also consider another version of this algorithm in which Step 3 is modified to calculate the possible improvement in the objective function for each vertex (not just the two with the maximal and minimal version as in Algorithm 2). In this version, the vertex which gives the best improvement is chosen. This algorithm will be referred to as Algorithm 2ALL. Both versions have attractive features as will be discussed in the next section.

6 Computational Study

In order to illustrate the efficiency of the algorithm, we have carried out computational tests with Algorithm 2 and Algorithm 2ALL. The computational experiments were carried out on a 3.40 GHz Pentium IV processor with 1.0 GB RAM using MATLAB version R2006b. The first data set was randomly

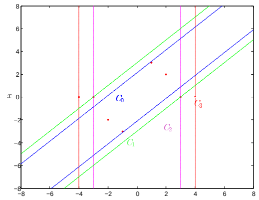

Figure 2: Cylinders generated by Algorithm 2 for Example 5.1.

generated as in [12] with sizes (n, m) varying from $(10, 500)$ to $(100, 100000)$. For each (n, m) value, we have solved the problem for three different values of k $(k = 0.2n, 0.5n, 0.8n)$. We have set $\epsilon = 10^{-4}$. For each fixed (n, k, m) ten different problem instances were generated for each data set. The computational results are reported in terms of averages over these instances in Table 2, which is divided into three sets of columns. The first set of columns reports the size (n, k, m). The second set of columns presents the results regarding the CPU time and is further divided into two parts, the first of which is devoted to the CPU time in seconds in order to obtain an $\epsilon-$approximate optimal solution using Algorithm 2 while the second one displays the CPU time using Algorithm 2ALL. The last set of columns reports the number of iterations and also further divided into two columns, displaying the number of iterations needed by Algorithm 2 and Algorithm 2ALL in this order. We have used the initialization algorithm in [7] to find the initial feasible solutions in our experiments. It was observed that these initial solutions decrease the number of iterations needed by the algorithms in practice. Finding a provably better initialization method would improve the theoretical complexity results in Section 4 and devising such a method remains as a challenge for us.

As demonstrated by Table 2, both algorithms are capable of solving very large instances of the problem in a reasonable amount of time. Although using Algorithm 2ALL may decrease the number of iterations needed, the total CPU time spent by Algorithm 2ALL is always worse than that by Algorithm 2 due to the large amount of calculation needed at each iteration.

Although the theoretical results suggest that the number of iterations and the total time spent by the algorithms are increasing with k, the experimental results demonstrate that in practice the problem becomes easier as the number

Table 2: Mean of the numbers of iterations and the solution times of two versions of Algorithm 2 for random samples of 21 problems, using data sets for Table 2 of Sun and Freund [12].

			CPU Time (Seconds)		No. of Iterations	
n	k	m	Alg 2	Alg 2ALL	Alg 2	Alg 2ALL
10	2	500	0.5531	0.8141	1047.9	880
10	5	500	0.325	0.5609	624.1	623
10	8	500	0.2062	0.3344	385.2	372.3
10	2	1000	0.5562	1.1703	1008.6	985.4
10	5	1000	0.5688	1.1047	979.1	908.4
10	8	1000	0.3812	0.8672	685	680.4
20	4	5000	4.2844	13.2078	3300.2	3058.6
20	10	5000	2.5156	9.175	2147.1	2189.4
20	16	5000	1.3719	5.4859	1260.1	1346.7
20	4	10000	8.375	31.2656	3670.7	3917.8
20	10	10000	4.1594	16.0328	1988.2	2056.3
20	16	10000	3.1875	12.6797	1577.9	1595.4
30	6	30000	40.4906	136.6312	5685.7	5771.4
30	15	30000	22.6766	78.5375	3355.4	3372.2
30	24	30000	17.75	65.8656	2812.9	2863.4
50	10	50000	156.025	412.7125	9477.5	9159.1
50	25	50000	99.8188	287.0797	6537.3	6593.7
50	40	50000	68.1203	207.375	4983.4	4937.6
100	20	100000	1120.3438	2302.5906	16826.9	17090.6
100	50	100000	783.3844	1700.3109	13125.3	13289.5
100	80	100000	555.7641	1279.6516	10535.4	10540.8

of parameters to be estimated are increased. This may be explained by the fact that as k increases the MAEC problem becomes increasingly closer in structure to the much easier MVEE problem. It is observed that for a problem instance with fixed $(n, 0.8n, m)$, the number of iterations required to solve the corresponding MVEE problem to the same level of precision is around 90 percent of the number of iterations required to solve the MAEC problem. In order to take a closer look at this phenomenon, we have conducted experiments on 10 data sets, where $(n, m) = (100, 10000)$. The instances are randomly generated as above and k is chosen from the set $\{5, 10, \ldots, 95, 100\}$. The results are summarized in Figure 3. In both figures the horizontal axis corresponds to the various values of the parameter k. The vertical axis corresponds to the average CPU time (left) and the average number of iterations (right) spent by the algorithm to solve 10 instances of the problem with parameter set $(100, k, 10000)$ in Figure 3. The problem instances are generated as in Table 2.

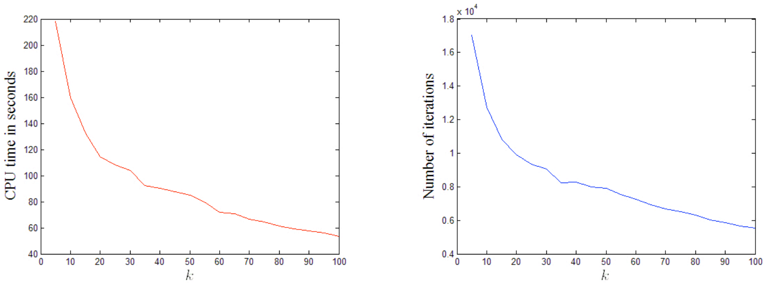

Figure 3: Average CPU times (left) and average number of iterations required (right) to obtain a 10^{-4}-approximate optimal solution using Algorithm 2 for randomly generated data sets with $(n, m) = (100, 10000)$ and various values of k

7 Discussion and Conclusions

In this paper, we have developed and analyzed an algorithm for the Minimum-Area Enclosing Ellipsoidal Cylinder problem. We have shown that this problem is a generalization of the Minimum-Volume Enclosing Ellipsoid problem. Our theoretical discussion and computational results show that the more general and harder MAEC problem can be solved by a Frank-Wolfe type algorithm just as the MVEE problem. Unfortunately developing this algorithm and analyzing its properties is not as straightforward as it is for the MVEE problem. We have illustrated this fact using a simple example and suggested a modification of the algorithm. One may suspect that the modified algorithm can cycle for some instances and fail. In our experience this never happens. We are not yet able to prove but we strongly believe that the modified algorithm does not cycle.

Our work shows how to use first-order methods to solve this problem. It is obvious that large instances of the problem can only be attacked by these type of techniques. The addition of Wolfe's away steps makes a huge difference in the number of iterations required to obtain a certain accuracy level in practice. The local convergence result in Section 4.2 provides a theoretical explanation to this phenomenon. Similar behavior is observed for the MVEE problem as discussed in [1].

We claim that this paper has three important contributions to the literature. First, we provide the first convex formulation of the MAEC problem. This formulation provides a clear characterization of the optimal solutions. Second, we analyze the convergence properties of an efficient algorithm which works for all practical instances. Our last contribution is computational. We solve very large instances of the problem and illustrate the behavior of the algorithm graphically. We hope that our computational results will provide a benchmark for future researchers.

8 Appendix

8.1 Notation

$$\begin{array}{rcl}
\omega_i(u) & := & (y_i+Ez_i)^TK^{-1}(y_i+Ez_i);\\
\xi_i(u) & := & x_i^T(XUX^T)^{-1}x_i;\\
\zeta_i(u) & := & z_i^T(ZUZ^T)^{-1}z_i;\\
\omega_{ij}(u) & := & (y_i+Ez_i)^TK^{-1}(y_j+Ez_j);\\
\xi_{ij}(u) & := & x_i^T(XUX^T)^{-1}x_j;\\
\zeta_{ij}(u) & := & z_i^T(ZUZ^T)^{-1}z_j;\\
\lambda & := & \dfrac{\tau}{1-\tau};\\
\mu & := & \dfrac{\lambda}{1+\lambda\zeta_j(u)};\text{ and}\\
\eta & := & \dfrac{\lambda}{1+\lambda\xi_j(u)}=\dfrac{\mu}{1+\mu\omega_j(u)}.
\end{array}$$

8.2 Algebraic details of Section 3.1

As in Section 3.1, we will assume that all dual solutions u and u_+ used in this section satisfy Assumption 3.1. The following lemma will be useful in our analysis.

Lemma 8.1 *$\xi_i(u)=\zeta_i(u)+\omega_i(u)$ for all i.*

Proof: Note first that

$$XUX^T=\left[\begin{array}{cc} ZUZ^T & ZUY^T\\ YUZ^T & YUY^T\end{array}\right]=\left[\begin{array}{cc} I & 0\\ -E & I\end{array}\right]\left[\begin{array}{cc} ZUZ^T & 0\\ 0 & K\end{array}\right]\left[\begin{array}{cc} I & -E^T\\ 0 & I\end{array}\right]. \tag{8.58}$$

Hence we have

$$\begin{array}{rcl}
\xi_i(u) & = & x_i^T(XUX^T)^{-1}x_i\\
& = & \left(\begin{array}{cc} z_i^T & y_i^T\end{array}\right)\left(\begin{array}{cc} ZUZ^T & ZUY^T\\ YUZ^T & YUY^T\end{array}\right)^{-1}\left(\begin{array}{c} z_i\\ y_i\end{array}\right)\\
& = & \left(\begin{array}{cc} z_i^T & y_i^T\end{array}\right)\left(\begin{array}{cc} I & E^T\\ 0 & I\end{array}\right)\left(\begin{array}{cc} (ZUZ^T)^{-1} & 0\\ 0 & K^{-1}\end{array}\right)\left(\begin{array}{cc} I & 0\\ E & I\end{array}\right)\left(\begin{array}{c} z_i\\ y_i\end{array}\right)\\
& = & \left(\begin{array}{cc} z_i^T & y_i^T+z_i^TE^T\end{array}\right)\left(\begin{array}{cc} (ZUZ^T)^{-1} & 0\\ 0 & K^{-1}\end{array}\right)\left(\begin{array}{c} z_i\\ y_i+Ez_i\end{array}\right)\\
& = & z_i^T(ZUZ^T)^{-1}z_i+(y_i+Ez_i)^TK^{-1}(y_i+Ez_i)\\
& = & \zeta_i(u)+\omega_i(u).\quad\Box
\end{array}$$

First note that assuming $ZUZ^T \succ 0$, from (8.58) we get $\det K = \frac{\det XUX^T}{\det ZUZ^T}$. Hence

$$\begin{aligned}\nabla g(u) &= \nabla(\ln\det XUX^T - \ln\det ZUZ^T)\\ &= \xi(u) - \zeta(u) = \omega(u). \end{aligned} \tag{8.59}$$

Let $u_+ = (1-\tau)u + \tau e_j$ where $e_j = (0,\dots,1,\dots,0)^T$ is the j^{th} unit vector. Then

$$ZU_+Z^T = (1-\tau)\left(ZUZ^T + \frac{\tau}{1-\tau} z_j z_j^T\right) = (1-\tau)(ZUZ^T + \lambda z_j z_j^T),$$

and using the Sherman-Morrison formula, we have

$$\begin{aligned}(ZU_+Z^T)^{-1} &= \frac{1}{1-\tau}\left((ZUZ^T)^{-1} - \frac{\lambda}{1+\lambda\zeta_j(u)}(ZUZ^T)^{-1}z_j z_j^T(ZUZ^T)^{-1}\right)\\ &= \frac{1}{1-\tau}\left((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_j z_j^T(ZUZ^T)^{-1}\right). \end{aligned} \tag{8.60}$$

Next,

$$YU_+Z^T = (1-\tau)(YUZ^T + \lambda y_j z_j^T), \tag{8.61}$$

and so

$$\begin{aligned}E_+ &= -(YU_+Z^T)(ZU_+Z^T)^{-1}\\ &= -(YUZ^T + \lambda y_j z_j^T)\Big((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_j z_j^T(ZUZ^T)^{-1}\Big)\\ &= E - \lambda y_j z_j^T(ZUZ^T)^{-1} - \mu E z_j z_j^T(ZUZ^T)^{-1} + \mu\lambda\zeta_j(u) y_j z_j^T(ZUZ^T)^{-1}\\ &= E - \Big(\lambda(1-\mu\zeta_j(u))y_j + \mu E z_j\Big) z_j^T(ZUZ^T)^{-1}\\ &= E - \mu(y_j + Ez_j)z_j^T(ZUZ^T)^{-1}. \end{aligned} \tag{8.62}$$

Finally,

$$YU_+Y^T = (1-\tau)(YUY^T + \lambda y_j y_j^T), \tag{8.63}$$

so

$$\begin{aligned}K_+ &= YU_+Y^T + YU_+Z^TE_+^T\\ &= (1-\tau)\Big(YUY^T + \lambda y_j y_j^T + (YUZ^T + \lambda y_j z_j^T)(E^T - \mu(ZUZ^T)^{-1}z_j(y_j+Ez_j)^T)\Big)\\ &= (1-\tau)\Big(K + \lambda y_j(y_j+Ez_j)^T + \mu E z_j(y_j+Ez_j)^T - \lambda\mu\zeta_j(u)y_j(y_j+Ez_j)^T\Big)\\ &= (1-\tau)\Big(K + (\lambda(1-\mu\zeta_j(u))y_j + \mu E z_j)(y_j+Ez_j)^T\Big)\\ &= (1-\tau)\Big(K + \mu(y_j+Ez_j)(y_j+Ez_j)^T\Big). \end{aligned} \tag{8.64}$$

We have

$$\ln\det K_+ = \ln\det K - k\ln(1+\lambda) + \ln(1+\mu\omega_j(u)),$$

whose derivative is

$$\begin{aligned}
\frac{\partial \ln\det K_+}{\partial\lambda} &= \frac{-k}{1+\lambda} + \frac{1}{1+\frac{\lambda\omega_j(u)}{1+\lambda\zeta_j(u)}}\left(\frac{\omega_j(u)}{1+\lambda\zeta_j(u)} - \frac{\lambda\zeta_j(u)\omega_j(u)}{(1+\lambda\zeta_j(u))^2}\right) \\
&= \frac{-k}{1+\lambda} + \frac{\omega_j(u)}{(1+\lambda(\zeta_j(u)+\omega_j(u)))(1+\lambda\zeta_j(u))} \\
&= \frac{-k}{(1+\lambda)(1+\lambda\zeta_j(u))(1+\lambda\xi_j(u))}\left(a\lambda^2 - 2b\lambda + c\right),
\end{aligned} \tag{8.65}$$

where $a := \zeta_j(u)\xi_j(u) \geq 0$, $b := -\zeta_j(u) - \frac{\omega_j(u)}{2} + \frac{\omega_j(u)}{2k} \leq 0$, and $c := 1 - \frac{\omega_j(u)}{k}$. The multiplier on the left is negative.

- If $c > 0$ (i.e., $\omega_j(u) < k$), the derivative is negative for $\lambda = 0$, so we want to decrease λ.
  - If $a = 0$ and $b < 0$, set $\lambda = \max\{\frac{c}{2b}, -u_j\}$; if $a = b = 0$ set $\lambda = -u_j$;
  - if $a > 0$ and $-b \leq \sqrt{ac}$, then the derivative is negative for all $\lambda < 0$ (possibly zero at one point), so set $\lambda = -u_j$ and hence $\tau = \frac{-u_j}{1-u_j}$ so that $(u_+)_j = 0$; and
  - if $a > 0$ and $-b > \sqrt{ac}$, then the derivative is negative up to the root of the quadratic, so set $\lambda = \max\{\frac{c}{b-\sqrt{b^2-ac}}, -u_j\}$.
- If $c < 0$ (i.e., $\omega_j(u) > k$), the derivative is positive for $\lambda = 0$, so we want to increase λ.
  - If $a = 0$ and $b < 0$, set $\lambda = \frac{c}{2b}$; if $a = b = 0$, set $\lambda = \infty$ and $\tau = 1$ (this can nly happen if $k = 1$); and
  - if $a > 0$, so $ac < 0$ and hence the quadratic has a positive root, set $\lambda = \frac{c}{b-\sqrt{b^2-ac}}$.

We can update $\omega(u_+)$ and $\zeta(u_+)$ cheaply. First, note that

$$\begin{aligned}
K_+^{-1} &= \frac{1}{1-\tau}(K^{-1} - \frac{\lambda}{1+\lambda\xi_j(u)}K^{-1}(y_j+Ez_j)(y_j+Ez_j)^T K^{-1}) \\
&= \frac{1}{1-\tau}(K^{-1} - \eta K^{-1}(y_j+Ez_j)(y_j+Ez_j)^T K^{-1}),
\end{aligned} \tag{8.66}$$

where $\eta := \frac{\lambda}{1+\lambda\xi_j(u)}$. Then

$$\omega_i(u_+) \quad = \quad (y_i + E_+z_i)^T K_+^{-1}(y_i + E_+z_i)$$

$$\begin{aligned}
&= \frac{1}{1-\tau}(y_i + (E - \mu(y_j + Ez_j)z_j^T(ZUZ^T)^{-1})z_i)^T(K^{-1} - \eta K^{-1}(y_j + Ez_j) \\
&\quad \ldots (y_j + Ez_j)^T K^{-1})(y_i + (E - \mu(y_j + Ez_j)z_j^T(ZUZ^T)^{-1})z_i) \\
&= \frac{1}{1-\tau}(y_i + Ez_i - \mu\zeta_{ij}(u)(y_j + Ez_j))^T(K^{-1} - \eta K^{-1}(y_j + Ez_j)(y_j + Ez_j)^T K^{-1}) \\
&\quad \ldots (y_i + Ez_i - \mu\zeta_{ij}(u)(y_j + Ez_j)) \\
&= \frac{1}{1-\tau}(y_i + Ez_i - \mu\zeta_{ij}(u)(y_j + Ez_j))^T(K^{-1}(y_i + Ez_i) - \eta\xi_{ij}(u)K^{-1}(y_j + Ez_j)) \\
&= \frac{1}{1-\tau}(\omega_i(u) - \eta\omega_{ij}^2(u) - 2\eta\zeta_{ij}(u)\omega_{ij}(u) + \eta\mu\omega_j(u)\zeta_{ij}^2(u)),
\end{aligned}$$

and

$$\begin{aligned}
\zeta_i(u_+) &= z_i^T(ZU_+Z^T)^{-1}z_i \\
&= \frac{1}{1-\tau}\left(z_i^T\left((ZUZ^T)^{-1} - \mu(ZUZ^T)^{-1}z_jz_j^T(ZUZ^T)^{-1}\right)z_i\right) \\
&= \frac{1}{1-\tau}\left(\zeta_i(u) - \mu\zeta_{ij}(u)^2\right), \qquad (8.67)
\end{aligned}$$

where $\xi_{ij}(u) := x_i^T(XUX^T)^{-1}x_j$, $\zeta_{ij}(u) := z_i^T(ZUZ^T)^{-1}z_j$ and $\omega_{ij}(u) := (y_i + Ez_i)^T K^{-1}(y_j + Ez_j)$. Note that, the following lemma has been used in the previous derivations and can be proved with arguments similar to those used in the proof of Lemma 8.1.

Lemma 8.2 *$\xi_{ij}(u) = \zeta_{ij}(u) + \omega_{ij}(u)$, for all i and j.*

8.3 Proof of Lemma 4.2 for decrease steps

Proof: (continued) Let's assume that iteration l is a decrease step, i.e., $j = \arg\min\{\omega_i(u^l) : u_i^l > 0\}$, $\omega_j(u^l) = (1-\epsilon_l)k$ and $\hat{\lambda}_l$ (the optimal step length) satisfies $-u_j^l < \hat{\lambda}_l < 0$. For any λ_l such that $0 \geq \lambda_l \geq \hat{\lambda}_l$, we have

$$k \;\geq\; \tfrac{(1+\lambda_l)\omega_j(u_l)}{(1+\lambda_l\zeta_j(u_l))^2}. \qquad (8.68)$$

Therefore we get

$$(1 + \lambda_l\zeta_j(u_l))^2 \;\geq\; (1+\lambda_l)\omega_j(u_l)/k \geq (1+\lambda_l)^2(1-\epsilon_l),$$

which in turn gives

$$1 + \lambda_l\zeta_j(u_l) \;\geq (1+\lambda_l)\sqrt{1-\epsilon_l}. \qquad (8.69)$$

Hence we have

$$\begin{aligned}
g_{l+1} - g_l &= \ln{(1+\lambda_l)^{-k}}(1 + \frac{\lambda_l\omega_j(u_l)}{1+\lambda_l\zeta_j(u_l)}) \\
&\geq \ln{(1 - k\lambda_l)}(1 + \frac{k\lambda_l(1-\epsilon_l)}{(1+\lambda_l)\sqrt{1-\epsilon_l}})
\end{aligned}$$

$$\begin{aligned} &\geq \ln{(1-k\lambda_l)}(1+\frac{k\lambda_l\sqrt{1-\epsilon_l}}{(1+\lambda_l)}) \\ &\geq \ln{(1-k\lambda_l)}(1+\frac{k\lambda_l(1-\epsilon_l/2)}{(1+\lambda_l)}). \qquad (8.70) \end{aligned}$$

Now choose $\lambda' < 0$ large enough such that for all $0 \geq \lambda \geq \lambda'$ the following inequality holds:

$$(1-k\lambda)(1+\frac{k\lambda(1-\epsilon_l/2)}{(1+\lambda)}) \geq 1 - \frac{\lambda k\epsilon_l}{3}. \qquad (8.71)$$

For $\lambda_l = \max{(\hat{\lambda}, \lambda')}$, we have the following bound:

$$g_{l+1} - g_l \quad \geq \ln\left(1 - \tfrac{\lambda_l k\epsilon_l}{3}\right). \qquad (8.72)$$

Note that (8.71) can be rewritten as

$$\begin{aligned} (1-k\lambda)(1+\lambda+\lambda k-\lambda k\epsilon_l/2) &\geq 1+\lambda-\lambda k\epsilon_l/3-\lambda^2 k\epsilon_l/3, \text{or} \\ -\lambda k\epsilon_l/2-\lambda^2 k-\lambda^2 k^2+\lambda^2 k^2\epsilon_l/2 &\geq -\lambda k\epsilon_l/3-\lambda^2 k\epsilon_l/3, \text{ or} \\ \lambda^2(k^2+k-k\epsilon_l/3-k^2\epsilon_l/2)+\lambda(k\epsilon_l/2-k\epsilon_l/3) &\leq 0. \qquad (8.73) \end{aligned}$$

Hence we can choose $\lambda' = \frac{\epsilon_l/3-\epsilon_l/2}{k+1-\epsilon_l/3-k\epsilon_l/2}$, and we have

$$|\lambda'| \geq c_5\epsilon_l. \qquad (8.74)$$

Since $\hat{\lambda}$ is the optimal step length we have

$$\begin{aligned} k = \omega_j(u_{l+1}) = &\frac{(1+\hat{\lambda})\omega_j(u_l)}{(1+\hat{\lambda}\xi_j(u_l))(1+\hat{\lambda}\zeta_j(u_l))} \\ &\leq \frac{(1+\hat{\lambda})\omega_j(u_l)}{(1+\hat{\lambda}\xi_j(u_l))^2}, \end{aligned}$$

which gives

$$(1+\hat{\lambda}\xi_j(u_l))^2 \leq \omega_j(u_l)/k = 1-\epsilon_l. \qquad (8.75)$$

From this we get $1+\hat{\lambda}\xi_j(u_l) \leq \sqrt{1-\epsilon_l} \leq 1-\epsilon_l/2$. Hence $\hat{\lambda} \leq \frac{-\epsilon_l}{2\xi_j(u_l)} \leq \frac{-\epsilon_l}{2C}$ and

$$|\hat{\lambda}| \geq c_6\epsilon_l. \qquad (8.76)$$

Equations (8.74) and (8.76) lead to

$$|\lambda_l| \geq \min\{c_5, c_6\}\epsilon_l = c_7\epsilon_l, \qquad (8.77)$$

which combined with (8.72) gives (4.22) for decrease steps and hence completes the proof of Lemma 4.2. □

References

[1] S. D. Ahipaşaoğlu and P. Sun and M. J. Todd, Linear convergence of a Modified Frank-Wolfe algorithm for computing minimum volume enclosing ellipsoids, Optimization Methods and Software, 23,5-19 (2008)

[2] C. L. Atwood, Optimal and Efficient Designs of Experiments, The Annals of Mathematical Statistics, 40, 1570-1602 (1969)

[3] C. L. Atwood, Sequences converging to D-optimal designs of experiments, The Annals of Statistics, 1,342-352 (1973)

[4] A.L. Dontchev and R.T. Rockafellar, Characterizations of Lipschitzian Stability in Nonlinear Programming. In: A.V. Fiacco (ed.) Mathematical Programming with Data Perturbations, 65-82. Marcel Dekker, Inc., New York (1998)

[5] Fedorov, V. V., Theory of Optimal Experiments. Academic Press, New York (1972)

[6] M. Frank and P. Wolfe, An algorithm for quadratic programming, Nav. Res. Log. Quart., 3, 95-110 (1956)

[7] P. Kumar and E. A. Yıldırım, Minimum volume enclosing ellipsoids and core sets, Journal of Optimization Theory and Applications, 126, 1-21 (2005)

[8] Pukelsheim, F., Optimal Design of Experiments. John Wiley and Sons, Inc., New York (1993)

[9] S. M. Robinson, Strongly regular generalized equations, Math. Oper. Res., 5, 43-62 (1980)

[10] S. M. Robinson, Generalized equations and their solutions, part II: Applications to nonlinear programming , Math. Prog. Study, 19, 200-221 (1982)

[11] S. Silvey and D. Titterington, A geometric approach to optimal design theory, Biometrika, 62, 21-32 (1973)

[12] P. Sun and R. M. Freund, Computation of Minimum Volume Covering Ellipsoids, Operations Research, 52, 690-706 (2004)

[13] P. Wolfe, Convergence theory in nonlinear programming. In: J. Abadie (ed.) Integer and Nonlinear Programming, 1-36. North-Holland, Amsterdam (1970)