Linear Convergence of a Modified Frank-Wolfe Algorithm for Computing Minimum Volume Enclosing Ellipsoids

S. Damla Ahipasaoglu* Peng Sun† Michael J. Todd‡

October 5, 2006

Dedicated to the memory of Naum Shor

Abstract

We show the linear convergence of a simple first-order algorithm for the minimum-volume enclosing ellipsoid problem and its dual, the D-optimal design problem of statistics. Computational tests confirm the attractive features of this method.

Keywords: Linear convergence, Frank-Wolfe algorithm, minimum-volume ellipsoids, optimizing on a simplex.

1 Introduction

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 $\mathbf{R}^n$. Since the volume of the ellipsoid

$$E(0, H) := \{x \in \mathbf{R}^n : x^T H x \leq n\},$$

*School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853, USA. This author was supported in part by NSF through grant DMS-0513337 and ONR through grant N00014-02-1-0057. e-mail: dse8@cornell.edu

†The Fuqua School of Business, Duke University, Durham, NC 27708, USA. email: psun@duke.edu

‡School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853, USA. This author was supported in part by NSF through grant DMS-0513337 and ONR through grant N00014-02-1-0057. e-mail: mjt7@cornell.edu

where $H \succ 0$, is $(\det H)^{-1/2}$ times that of a ball in $\mathbb{R}^n$ of radius $\sqrt{n}$, finding a minimum-volume central (i.e., centered at the origin) ellipsoid containing the columns of X amounts to solving

$$(P) \qquad \begin{array}{rl} \min_{H \succ 0} & f(H) := -\ln\det H \\ & x_i^T H x_i \ \leq \ n,\ i = 1, \ldots, m. \end{array} \tag{1}$$

We call this the minimum-volume enclosing ellipsoid (MVEE) problem.

Khachiyan in [8] showed that the seemingly more general problem of finding a not necessarily central ellipsoid of minimum volume containing a finite point set in $\mathbb{R}^n$ reduces to the MVEE problem for a related point set in $\mathbb{R}^{n+1}$, so we henceforth consider only the central case. Such ellipsoid problems arise in data analysis and computational geometry (see the references in [15, 10]) and as a subproblem in optimization, e.g., for each iteration of the ellipsoid method of Yudin and Nemirovskii [20] and Shor [13] (where a closed-form solution is available) or to initialize Lenstra's integer programming algorithm [11].

Problem (P) is convex, with linear constraints. After some simplification, its Lagrangian dual turns out to be

$$(D) \qquad \begin{array}{rl} \max_u & g(u) := \ln\det XUX^T \\ & e^T u \ = \ 1, \\ & u \ \geq \ 0, \end{array} \tag{2}$$

which is also the statistical problem of finding a D-optimal design measure on the columns of X, that is, one that maximizes the determinant of the Fisher information matrix $E(xx^T)$: see, e.g., Atwood [1, 2], Fedorov [3], John and Draper [6], Kiefer and Wolfowitz [9], Silvey [14], and Wynn [18, 19].

In [2], Atwood developed an algorithm for (D) that is a simple modification of those of Fedorov [3] and Wynn [18]. Indeed, we will see in Section 2 that Atwood's method is a specialization to (D) of the Frank-Wolfe algorithm [4] with Wolfe's "away steps" [17]. We will prove linear convergence of the objective function values for both (P) and (D) using the Wolfe-Atwood method.

Note that (D) is a convex problem, with feasible region the unit simplex. For such problems, Wolfe [17] sketched the proof of, and Guélat and Marcotte [5] proved in detail, linear convergence of the Frank-Wolfe algorithm with away steps. However, the objective function g of (D) does not satisfy the conditions assumed by Wolfe and by Guélat and Marcotte: it is neither

boundedly concave, nor strictly (let alone strongly) concave. Instead of using their approach, we prove our result by applying Robinson's analysis [12] of upper Lipschitzian continuity to a perturbation of problem (P).

In Section 3, we consider concave maximization over the simplex, and prove linear convergence of the method under assumptions a little weaker than those of Wolfe and Guélat-Marcotte. Instead of strong convexity, we suppose that Robinson's second-order sufficient condition holds at the maximizer. (We do assume twice differentiability, while Guélat and Marcotte require only a Lipschitz continuous gradient.)

Finally, Section 4 gives some computational results of the Wolfe-Atwood algorithm for the MVEE problem, showing its surprising efficiency and accuracy. However, for $m \gg n$, it can be much slower than the DRN method (a specialized interior-point algorithm) with an active set strategy in Sun and Freund [15].

From both a theoretical and computational viewpoint, the simple first-order method of Wolfe and Atwood for the MVEE problem appears to be an attractive algorithm, like many of those developed by N. Z. Shor.

2 Algorithms and analysis for the MVEE problem

Note that the objective function g of (D) is a concave function with gradient

$$w(u) := \nabla g(u) = (x_i (XUX^T)^{-1} x_i)_{i=1}^n, \tag{3}$$

and that, with

$$u_+ := (1-\tau)u + \tau e_i, \tag{4}$$

rank-one update formulae give

$$(XU_+X^T)^{-1} = \frac{1}{1-\tau}\left[(XUX^T)^{-1} - \frac{\tau (XUX^T)^{-1} x_i x_i^T (XUX^T)^{-1}}{1-\tau+\tau w_i(u)}\right] \tag{5}$$

and

$$\det XU_+X^T = (1-\tau)^{n-1}[1-\tau+\tau w_i(u)] \det XUX^T. \tag{6}$$

It is therefore computationally inexpensive to update ∇g after an update such as (4) and to perform a line search on g to determine the optimal τ.

Indeed, the optimal stepsize is (see, e.g., (2.19) in Khachiyan [7])

$$\tau_* = \frac{w_i(u)/n - 1}{w_i(u) - 1}. \tag{7}$$

For these reasons, applying the Frank-Wolfe algorithm [4] to (D) is an attractive procedure, and this was proposed by Fedorov [3] and Wynn [18], the latter without the line search, in the context of optimal design. Without confusion, we can thus call this the FW algorithm.

We say H is feasible in (P) if it is positive definite and satisfies the constraints of (P); similarly, u is feasible in (D) if it satisfies the constraints of (D) and moreover has a finite objective function value, i.e., $XUX^T \succ 0$. Suppose H and u are feasible in (P) and (D) respectively. Then

$$n \geq \sum_i u_i x_i^T H x_i = \sum_i H \bullet x_i u_i x_i^T = H \bullet (XUX^T) = \mathrm{Tr}(H^{1/2} XUX^T H^{1/2}).$$

Hence

$$\begin{aligned}
-\ln\det H - \ln\det XUX^T &= -\ln\det HXUX^T \\
&= -\ln\det H^{1/2} XUX^T H^{1/2} \\
&= -n\ln(\Pi_{j=1}^n \lambda_j)^{1/n} \\
&\geq -n\ln\left(\frac{\sum_{j=1}^n \lambda_j}{n}\right) \\
&= -n\ln\left(\frac{\mathrm{Tr}(H^{1/2} XUX^T H^{1/2})}{n}\right) \\
&\geq 0,
\end{aligned}$$

where the λ_j's are the positive eigenvalues of $H^{1/2} XUX^T H^{1/2}$. This proves weak duality, and gives the following sufficient conditions for optimality in both (P) and (D):

(a) $u_i > 0$ only if $x_i^T H x_i = n$; and

(b) $H = (XUX^T)^{-1}$,

since we must have all eigenvalues equal, and hence $H^{1/2} XUX^T H^{1/2}$ a multiple of the identity, to have the geometric and arithmetic means coincide, and we must have the multiple equal unity to have the trace equal n. In

fact, it is easy to show using the Karush-Kuhn-Tucker conditions for (P) that these conditions are also necessary. Moreover, u provides a vector of Lagrange multipliers for (P).

We also have (e.g., Khachiyan [7]), that for any feasible u,

$$u^T w(u) = n,$$

so that, given (b), (a) above holds if $x_i^T H x_i \leq n$ for all i. Hence, for a feasible u, we need only check that $H = (XUX^T)^{-1}$ is feasible in (P) to check the optimality of u. Henceforth, we use $H(u)$ to denote this matrix:

$$H(u) := (XUX^T)^{-1}. \tag{8}$$

The FW algorithm thus starts with some feasible u, and then at each iteration finds the index i with maximal $w_i(u) = x_i^T H(u) x_i$, stops if this maximum value is at most $(1+\epsilon)n$, and otherwise replaces u with u_+ in (4), where τ is chosen to maximize $g(u_+)$.

We motivated the algorithm above using the optimality conditions, but note that e_i solves the problem

$$\max_{\bar{u}} \{g(u) + \nabla g(u)^T(\bar{u} - u) : e^T \bar{u} = 1, \bar{u} \geq 0\},$$

so at each iteration we maximize a linear approximation to g and do a line search on the line segment joining our current iterate to the optimal solution of the linearized problem: that is, we are performing the Frank-Wolfe algorithm on (D).

When the algorithm stops, we have $(1+\epsilon)^{-1}H$ feasible in (P), so that this and u are both optimal in their respective problems up to an additive constant of $n\ln(1+\epsilon) \leq n\epsilon$. Moreover, $\text{conv}\{\pm x_1, \ldots, \pm x_m\}$ is contained in $\{x \in \mathbf{R}^n : x^T H x \leq (1+\epsilon)n\}$, but also contains $\{x \in \mathbf{R}^n : x^T H x \leq 1\}$, since the maximum of $|v^T x|$ over the latter is $\sqrt{v^T H^{-1} v} = \sqrt{v^T X U X^T v} = \sqrt{\sum_i u_i (v^T x_i)^2} \leq \max_i |v^T x_i|$ for any v. Thus we have a $\sqrt{(1+\epsilon)n}$ rounding of this convex hull. Finally, for $0 < \eta \leq 1$, we have $n\ln(1+\epsilon) \leq 2\ln(1+\eta)$ for $\epsilon = \eta/n$, so for this value of ϵ we get an ellipsoid that has minimum volume up to the the factor $(1+\eta)$.

(Khachiyan [7] shows that to find a $(1+\epsilon)n$ rounding of the convex hull of m points $y_1, \ldots, y_m$ in $\mathbf{R}^n$, or to find a nearly minimum-volume not-necessarily-central ellipsoid containing these points, it suffices to find a good rounding or a nearly minimum-volume central ellipsoid for the set of the

previous paragraph, where $x_i = (y_i; 1) \in R^{n+1}$ for each i. So at the expense of increasing the dimension by one, we can confine our attention to the central case.)

We call a feasible u *ϵ-primal feasible* if $x_i^T H(u) x_i \leq (1+\epsilon)n$ for all i, and say that it *satisfies the (strong) ϵ-approximate optimality conditions* if moreover $x_i^T H(u) x_i \geq (1-\epsilon)n$ whenever $u_i > 0$. (In the next section, we will have both weak and strong ϵ-approximate optimality conditions, corresponding to these two properties.) The algorithms of Khachiyan [7] and Kumar and Yildirim [10] seek an ϵ-primal feasible u, while that of Todd and Yildirim [16] seeks one satisfying the ϵ-approximate optimality conditions. In fact, apart from the details of their initialization and termination, the first two methods coincide with that of Fedorov (and Wynn, although he didn't use an optimal line search) for the optimal design problem, and hence a specialization of that of Frank and Wolfe. We therefore denote them the FW-K method and the FW-KY method.

Let us now describe the method analyzed by Todd and Yildirim informally. At each iteration, we have a feasible u, and we compute the index i with maximum $w_i(u) - n$ as before. We also compute the index j with maximum $n - w_j(u)$ among those j with $u_j > 0$. If $w(i) - n$ is larger than $n - w_j(u)$, we proceed as in the FW algorithm, but otherwise, we replace u by

$$u_+ := (1-\tau)u + \tau e_j, \tag{9}$$

where now τ is chosen from negative values to maximize g subject to u_+ remaining feasible. (The optimal unconstrained τ is again given by (7), with j replacing i, as long as $w_j(u) > 1$: otherwise, τ is made as negative as feasible.) It is easily seen that e_j solves the problem of minimizing the linearization of g on a restriction of the feasible set, where zero components of u are fixed at zero, so this is the FW algorithm with away steps as in Wolfe [17] (u moves away from e_j), with specific initialization and termination details given. This algorithm was also proposed by the statistician Atwood [2] for the optimal design problem. We therefore call it the WA-TY method.

Observe that (P) can be reformulated as having a strictly convex continuous objective function and a compact feasible set, so that it has a unique optimal solution H_* with optimal value f_*, and (D) also has an optimal solution, possibly not unique, with optimal value $g^* = f_*$. The analyses of Khachiyan [7], Kumar-Yildirim [10], and Todd-Yildirim [16] bound the number of steps until an ϵ-primal feasible solution u is obtained

(or until one satisfying the ϵ-approximate optimality conditions is found), by bounding the improvement in $g(u)$ at each iteration.

Khachiyan starts with $u_0 = (1/m)e$, while Kumar-Yildirim and Todd-Yildirim start with a more complicated procedure to determine a u_0 with at most $2n$ positive components. Khachiyan shows that at most $4n(\ln n + \ln\ln m + 2)$ iterations are necessary from his initial solution until a 1-primal feasible solution is found, while Kumar and Yildirim show that no more than $16n(\ln n + 1)$ are needed from their start to obtain the same quality. The same is true for the WA-TY method, since until a 1-primal feasible solution is obtained, no away steps will be performed. We therefore concentrate on the algorithms after they produce a 1-primal feasible solution (which also satisfies the 1-approximate optimality conditions) until they reach an ϵ-primal feasible solution or one that satisfies the ϵ-approximate optimality conditions. For this analysis, we need the following results.

Lemma 2.1 *(Khachiyan [7], Lemma 2). If u is δ-primal feasible (and hence if it satisfies the δ-approximate optimality conditions),*

$$g^* - g(u) \leq n\delta. \tag{10}$$

□

For our analysis of away steps, it is convenient to characterize normal FW steps where u_i is increased from zero as *add*-iterations, and those where it is increased from a positive value as *increase*-iterations. Away steps are called *drop*-iterations if u_j is decreased to zero, and otherwise *decrease*-iterations. Note that every drop-iteration can be associated with either a previous add-iteration where that component of u was last increased from zero, or with one of the original at most $2n$ positive components of u_0.

Lemma 2.2 *Suppose $\delta \leq 1/2$.*

(a) If u is not δ-primal feasible, any add- or increase-iteration improves $g(u)$ by at least $2\delta^2/7$.

(b) If a feasible u does not satisfy the δ-approximate optimality conditions, any decrease-iteration improves $g(u)$ by at least $2\delta^2/7$.

Proof: Khachiyan [7] (Lemma 3, see also the proof of Lemma 4) proved (a), while Todd and Yildirim [16] (Lemma 4.2) proved (b). □

Because they are limited by remaining feasible, drop-iterations may not provide a certifiably large increase in g, but at least g does not decrease.

Let $k(\delta)$ (respectively, $\bar{k}(\delta)$) denote the number of iterations of the FW-K or FW-KY method (number of add-, increase-, and decrease-iterations of the WA-TY method) from the first iterate that is δ-primal feasible (satisfies the δ-approximate optimality conditions) until the first that is $\delta/2$-primal feasible (satisfies the $\delta/2$-approximate optimality conditions). Then Lemmas 2.1 and 2.2 show that

$$k(\delta) \leq n\delta/(2(\delta/2)^2/7) = 14n/\delta, \tag{11}$$

and similarly for $\bar{k}$. So if $K(\epsilon)$ (respectively, $\bar{K}(\epsilon)$) denotes the number of iterations (number of add-, increase-, and decrease-iterations) from the first iterate that is 1-primal feasible (satisfies the 1-approximate optimality conditions) until the first that is ϵ-primal feasible (satisfies the ϵ-approximate optimality conditions), we find

$$\begin{array}{rcl} K(\epsilon) & \leq & k(1) + k(1/2) + \cdots + k(1/2^{\lceil \ln 1/\epsilon \rceil - 1}) \\ & \leq & 14n(1 + 2 + \cdots + 2^{\lceil \ln 1/\epsilon \rceil - 1}) \leq 28n/\epsilon, \end{array} \tag{12}$$

and again similarly for $\bar{K}$. Hence we have the following

Theorem 2.1 *(a) The total number of iterations for the FW-K algorithm to obtain an ϵ-primal feasible solution is at most $28n/\epsilon + 4n(\ln n + \ln\ln m + 2)$, while for the FW-KY algorithm, it is at most $28n/\epsilon + 16n(\ln n + 1)$.*

(b) The total number of iterations for the WA-TY method to obtain a solution u which satisfies the ϵ-approximate optimality conditions is at most $56n/\epsilon + 32n(\ln n + 2)$.

(c) The total number of iterations for the FW-K algorithm to obtain an η-optimal solution (i.e., a solution u with $g^* - g(u) \leq \eta$) is at most $3.5n^2/\eta + 4n(\ln n + \ln\ln m + 6)$, while for the FW-KY algorithm, it is at most $3.5n^2/\eta + 16n(\ln n + 2)$ and for the WA-TY method it is at most $7n^2/\eta + 32n(\ln n + 3)$.

Proof: The argument for (a) is stated above the statement of the theorem, and for (b) we need only note that the number of drop-iterations is bounded by the number of add-iterations plus $2n$.

For part (c) we note first that if we have an η/n-primal feasible solution or one that satisfies the η/n-approximate optimality conditions, then we automatically have an η-optimal solution by Lemma 2.1. Thus (c) almost follows from (a) and (b). To obtain the improved coefficient for n^2/η, and to simplify the proof, we use the proof technique of Wolfe [17]. Let γ denote $g^* - g(u)$ and γ_+ denote $g^* - g(u_+)$. We obtain a 1/2-primal feasible solution or one satisfying the 1/2-approximate optimality conditions in $14n$ or $28n$

more steps than to find a 1-primal feasible solution or one satisfying the 1-approximate optimality conditions. From then on, $\gamma \leq n/2$ and u is not δ-primal feasible or does not satisfy the δ-approximate optimality conditions for all $\delta < \gamma/n \leq 1/2$. Then Lemmas 2.1 and 2.2 show that, at every add-, increase-, or decrease-iteration, $\gamma_+ \leq \gamma - 2\gamma^2/(7n^2)$, so if we set $\bar{\gamma}$ to be $\gamma/(3.5n^2)$ and similarly for $\bar{\gamma}_+$, we find

$$\frac{1}{\bar{\gamma}_+} \geq \frac{1}{\bar{\gamma}(1-\bar{\gamma})} \geq \frac{1+\bar{\gamma}}{\bar{\gamma}} = \frac{1}{\bar{\gamma}} + 1,$$

and so, from its initial positive value, $1/\bar{\gamma}$ will increase to at least k in k iterations; thus γ will be at most η in at most $3.5n^2/\eta$ iterations. For the WA-TY method, the bound must again be doubled for the drop-iterations. □

Observe that the more complicated analysis of Khachiyan leads to bounds on the number of iterations to be able to guarantee a certain quality solution, while the simpler argument for part (c) gives bounds on the number of iterations required to obtain a certain quality solution, but we may not know that this quality has been reached.

We now wish to show that the WA-TY algorithm modification, i.e., the inclusion of decrease- and drop-iterations, leads to an asymptotic bound that grows with $\ln(1/\epsilon)$ rather than $1/\epsilon$, that is linear convergence. Unfortunately, this bound depends on the data of the problem as well as the dimensions, and so does not provide a global complexity bound better than that above.

We use the following perturbation of (P):

$$(P(z)) \qquad \begin{array}{ll} \min_{H \succ 0} & -\ln\det H \\ & x_i^T H x_i \;\leq\; n + z_i,\ i = 1, \ldots, m. \end{array}$$

Given u satisfying the δ-approximate optimality conditions, let $H(u)$ be as in (8), and define $z := z(u, \delta) \in \mathbf{R}^m$ by

$$z_i := \begin{cases} \delta n & \text{if } u_i = 0 \\ x_i^T H(u) x_i - n & \text{else.} \end{cases}$$

Observe that each component of z has absolute value at most δn, and that this property fails if we merely assume that u is δ-primal feasible. Moreover,

$$u^T z = \sum_{i:u_i>0} u_i z_i = u^T w(u) - n e^T u = n - n = 0. \tag{13}$$

Lemma 2.3 *Suppose u satisfies the δ-approximate optimality conditions. Then $H(u)$ is optimal in $(P(z(u,\delta)))$.*

Proof: We note that $H(u)$ is feasible and that u provides the required vector of Lagrange multipliers, which suffice because the problem is convex. □

Let $\phi(z)$ denote the value function, the optimal value of $(P(z))$. Then ϕ is convex, and if u' is any vector of Lagrange multipliers for the optimal solution of $(P(z))$, then u' is a subgradient of ϕ at z. 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, and we find for any u satisfying the δ-approximate optimality conditions and $z := z(u,\delta)$,

$$\begin{aligned} g(u) = f(H(u)) = \phi(z) &\geq \phi(0) + u_*^T z \\ &= g^* + (u_* - u)^T z \\ &\geq g^* - \|u - u_*\|\|z\|. \end{aligned} \tag{14}$$

Here the last equality follows from (13). We have already noted that $\|z\| \leq n\sqrt{m}\delta$. To obtain an improvement on Lemma 2.1, we need to bound $\|u-u_*\|$. Since f is strongly convex near any $H \succ 0$ and the constraints are linear, the second-order sufficient condition of Robinson [12] holds for (H, u') for any $(P(z))$, where H is the optimal solution and u' any vector of Lagrange multipliers. Moreover, since the constraints are linear and Slater's constraint qualification holds (when $\|z\| < 1$), the constraints are regular in the sense of Robinson at any feasible H. In addition, the constraints on H (besides the open convex set constraint that $H \succ 0$) are polyhedral, so that Robinson's Corollary 4.3 applies, which shows that, for some Lipschitz constant L, there is some u_* which is a vector of Lagrange multipliers for (P) such that

$$\|u - u_*\| \leq L\|z\| \leq Ln\sqrt{m}\delta$$

whenever $\|z\|$ is sufficiently small. From this and (14) we conclude

Proposition 2.1 *There is some constant $M > 0$ (depending on the data of problem (P)) such that, whenever u satisfies the δ-approximate optimality conditions for some sufficiently small δ, we have*

$$g^* - g(u) \leq M\delta^2. \tag{15}$$

□

Applying Proposition 2.1 instead of Lemma 2.1 in (11), we obtain

$$\bar{k}(\delta) \leq M\delta^2/(2(\delta/2)^2/7) = 14M \text{ for sufficiently small } \delta, \tag{16}$$

and this yields, using the argument above (12), the existence of a constant $Q > 0$ with

$$\bar{K}(\epsilon) \leq Q + 28M \ln(1/\epsilon) \text{ for sufficiently small } \epsilon.$$

We therefore have

Theorem 2.2 *There are data-dependent constants $\bar{Q}$ and $\hat{Q}$ such that:*

(a) The WA-TY algorithm for problem (P) requires at most $\bar{Q} + 56M \ln(1/\epsilon)$ iterations to obtain a solution that satisfies the ϵ-approximate optimality conditions; and

(b) The WA-TY algorithm for problem (P) gives a sequence of optimality gaps $g^* - g(u)$ that is nonincreasing and, asymptotically, at every add-, increase-, or decrease-iteration, decreases by the factor $1 - (3.5M)^{-1}$, so that at most $\hat{Q} + 7M \ln(1/\eta)$ iterations are required to obtain an η-optimal solution.

Here M is as in Proposition 2.1.

Proof: Part (a) follows directly from the analysis above, again allowing for the drop-iterations. For part (b), note that asymptotically, for every add-, increase- or decrease-iteration, Lemma 2.2 and Proposition 2.1 imply that

$$g^* - g(u_+) \leq (1 - \frac{2}{7M})(g^* - g(u)),$$

which gives the result. □

To conclude this section, we observe that Proposition 2.1 not only is used to help prove the convergence result above, but also implies that asymptotically, solutions u that satisfy the ϵ-approximate optimality conditions are likely to be much closer to optimality than those that are merely ϵ-primal feasible, even if no improved bound can be given because M is unknown.

3 The Frank-Wolfe algorithm with away steps on the simplex

We now prove linear convergence of the Frank-Wolfe algorithm with Wolfe's away steps [17] for the problem

$$(S) \qquad \begin{array}{rcl} \max_u \ g(u) & & \\ e^T u & = & 1, \\ u & \geq & 0, \end{array} \tag{17}$$

where g is a twice continuously differentiable concave function on the simplex, using arguments similar to those in the previous section. The assumptions we need are slightly weaker than those in Wolfe [17] and Guélat and Marcotte [5].

Let us write $w(u)$ for $\nabla g(u)$ to conform to the previous section. By simplifying a little, we arrive at the necessary and sufficient optimality conditions

(i) $w(u) \leq u^T w(u) e$, and

(ii) $w_i(u) = u^T w(u)$ if $u_i > 0$.

We say that u satisfies the weak ϵ-approximate optimality conditions if $w(u) \leq (u^T w(u) + \epsilon)e$, and the strong ϵ-approximate optimality conditions if moreover $w_i(u) \geq u^T w(u) - \epsilon$ if $u_i > 0$.

Corresponding to Lemma 2.1, we can show

Lemma 3.1 *If u satisfies the weak or strong δ-approximate optimality conditions, then*

$$g^* - g(u) \leq \delta. \tag{18}$$

Proof: We note that $g(\bar{u}) \leq g(u) + w(u)^T(\bar{u} - u)$ for all $\bar{u}$ by concavity, and the maximum value of the latter over the simplex is $g(u) + \max_i(w_i(u) - u^T w(u)) \leq g(u) + \delta$. $\square$

Given a feasible u, the Frank-Wolfe algorithm for (S) stops if u satisfies the weak ϵ-approximate optimality conditions. If not, it solves the problem of maximizing the linearization above over the simplex, or equivalently finds the index i maximizing $w_i(u) - u^T w(u) > \epsilon$, and replaces u by u_+ in (4) where again τ is chosen to maximize g. To bound the improvement this yields, we need a bound L on the norm of $\nabla^2 g(u)$ over the simplex (we use the operator norm, i.e., the largest absolute value of an eigenvalue of this matrix). We find

$$\begin{aligned} g(u_+) &= g(u) + \tau w(u)^T(e_i - u) + \tfrac{1}{2}\tau^2(e_i - u)^T \nabla^2 g(u')(e_i - u) \\ &\geq g(u) + \tau w(u)^T(e_i - u) - \tfrac{1}{2}\tau^2 L(e_i - u)^T(e_i - u) \\ &\geq g(u) + \tau w(u)^T(e_i - u) - L\tau^2, \end{aligned} \tag{19}$$

where u' is some point between u and e_i. Here we have used the fact that any pair of points in the unit simplex are at a distance of at most $\sqrt{2}$.

Away steps modify the algorithm as follows. We now stop if u satisfies the strong ϵ-approximate optimality conditions. If not, we solve the problem

of maximizing the linearization over the simplex, or find the index i as above. We also minimize the linearization over the face of the simplex with u in its relative interior, or equivalently find the index j maximizing $u^T w(u) - w_j(u)$ among those j's with $u_j > 0$. If $w_i(u) - u^T w(u) \geq u^T w(u) - w_j(u)$, we perform a usual Frank-Wolfe step as above. Otherwise, we make an away step: we replace u by u_+ as in (9), where τ is chosen from negative values to maximize g subject to u_+ remaining feasible.

We characterize iterations as add-, increase-, decrease-, or drop-iterations exactly as in the previous section.

We can now prove a result analogous to Lemma 2.2:

Lemma 3.2 *(a) If a feasible u does not satisfy the weak δ-approximate optimality conditions, any add- or increase-iteration improves $g(u)$ by at least* $\min\{\delta/2, \delta^2/(4L)\}$.

(b) If a feasible u does not satisfy the strong δ-approximate optimality conditions, any decrease-iteration improves $g(u)$ by at least $\delta^2/(4L)$.

Proof: (a) We use (19). The value of τ maximizing the right-hand side is $(w_i(u) - u^T w(u))/(2L)$. If this is less than 1, we use this stepsize and guarantee an improvement in g of at least $(w_i(u) - u^T w(u))^2/(4L) \geq \delta^2/(4L)$. Otherwise, $L < (w_i(u) - u^T w(u))/2$, so choosing τ equal to 1 assures an improvement in g of at least $(w_i(u) - u^T w(u)) - L \geq (w_i(u) - u^T w(u))/2 \geq \delta/2$.

(b) We use the corresponding form of (19) with j replacing i. The value of τ maximizing the right-hand side is $(w_j(u) - u^T w(u))/(2L) < 0$. If this leads to a feasible u_+, we similarly obtain an improvement in g of at least $(w_j(u) - u^T w(u))^2/(4L) \geq \delta^2/(4L)$. If not, we have a drop-iteration, and there is nothing to prove (g does not decrease). □

This analysis leads to a global bound as in Khachiyan's analysis. Indeed, while u does not satisfy the weak $2L$-approximate optimality conditions, it is easy to see that the gap $g^* - g(u)$ decreases by a factor of two at every add-, increase-, or decrease-iteration. If u satisfies the weak or strong δ-approximate optimality conditions, with $\delta \leq 2L$, we can obtain a solution satisfying the weak or strong $\delta/2$-approximate optimality conditions in at most $\delta/(\delta^2/(16L)) = 16L/\delta$ iterations, so a total of $32L/\epsilon$ suffice to obtain one satisfying the weak or strong ϵ-approximate optimality conditions. These bounds are similar to (in fact, slightly weaker than) those in equation (6.3) in Wolfe [17]. The reasons are the same as those given below the proof of

Theorem 2.1: our bounds are on the number of iterations before a certain quality solution is guaranteed.

We now again improve Lemma 3.1 to obtain linear convergence. Since we have no simple closed-form dual problem, we work with the following perturbation of the problem (S):

$$(S(z)) \qquad \begin{array}{rcl} \max_u \quad g(u) - z^T u & & \\ e^T u & = & 1, \\ u & \geq & 0, \end{array} \tag{20}$$

where $z \in \mathbf{R}^m$ is a perturbation vector. Given u satisfying the strong δ-approximate optimality conditions, we define $z := z(u, \delta) \in \mathbf{R}^m$ by

$$z_i := \begin{cases} \delta & \text{if } u_i = 0 \\ w_i(u) - u^T w(u) & \text{else.} \end{cases}$$

We note that each component of z is at most δ in absolute value, and that this property fails if we assume only the weak δ-approximate optimality conditions. Moreover,

$$u^T z = \sum_{i:u_i>0} u_i z_i = u^T w(u) - (u^T w(u)) u^T e = 0.$$

The gradient of the objective function of $(S(z))$ is $\bar{w}(u) = w(u) - z$, with $u^T \bar{w}(u) = u^T w(u)$ by the equation above.

Lemma 3.3 *If u satisfies the strong δ-approximate optimality conditions, then u is optimal in $(S(z(u, \delta))$.*

Proof: Observe that u satisfies the optimality conditions for $(S(z(u, \delta))$ by the remarks above. □

Let u^* denote any optimal solution of (S). Then u^* is feasible for $(S(z))$ with $z = z(u, \delta)$, and we conclude that $g(u^*) - z^T u^* \leq g(u) - z^T u$, so that

$$g^* - g(u) \leq z^T (u^* - u) \leq \|z\| \|u - u^*\|.$$

We have already observed that $\|z\| \leq \delta\sqrt{m}$, so it suffices to bound $\|u - u^*\|$. To use Robinson's Corollary 4.3 again, we need to assume that

there is an optimal solution u^* of (S) satisfying the strong sufficient condition of Robinson [12].

Of course, the condition should be adapted to the maximization problem (S) instead of a minimization problem. We do not need to prescribe the Lagrange multipliers because the constraints are linear. This condition certainly holds if g is strongly concave as in Wolfe [17] and Guélat and Marcotte [5], but is slightly weaker. The above condition implies that u^* is a strict local maximizer, and since the problem is convex, it is the unique maximizer.

Since the constraints are linear and polyhedral, and the Slater condition holds, the other conditions required for Robinson's Corollary 4.3 hold, and we conclude that there is some Lipschitz constant N such that

$$\|u - u^*\| \leq N\|z\| \leq N\sqrt{m}\delta$$

whenever δ is sufficiently small. From this we obtain

Proposition 3.1 *There is some constant M depending on the data of problem (S) such that, whenever u satisfies the strong δ-approximate optimality conditions for sufficiently small δ, we have*

$$g^* - g(u) \leq M\delta^2.$$

□

From this we obtain, exactly as in the last section, the following linear convergence result.

Theorem 3.1 *There are constants $\bar{Q}$ and $\hat{Q}$ such that:*

(a) The Frank-Wolfe algorithm with away steps for problem (S) requires at most $\bar{Q} + 64LM\ln(1/\epsilon)$ iterations to obtain a solution that satisfies the strong ϵ-approximate optimality conditions; and

(b) The Frank-Wolfe algorithm with away steps for problem (S) gives a sequence of optimality gaps $g^* - g(u)$ that is nonincreasing and, asymptotically, at every add-, increase-, or decrease-iteration, decreases by the factor $1 - (4LM)^{-1}$, so that at most $\hat{Q} + 8LM\ln(1/\eta)$ iterations are required to obtain an η-optimal solution.

Here L is a bound on the norm of the Hessian matrix of g on the simplex, and M is as in Proposition 3.1.

4 Computational Study

In this section we present some computational results for the Wolfe-Atwood-Todd-Yildirim (WA-TY) modified FW algorithm, using different initializa-

tion strategies. Specifically, we test the original Khachiyan initialization strategy, where the initial feasible u is set to be the center of the simplex in $\mathbb{R}^m$, that is, $u_i = 1/m$ for all $i = 1 \dots m$. We also test the Kumar-Yildirim initialization strategy, see [10].

We compare the above Frank-Wolfe-type first-order algorithms with a second-order interior-point algorithm, the DRN algorithm proposed in Sun and Freund [15]. For better illustration, we use the same test data sets as in Sun and Freund [15]. All computations were conducted on a Dell Xeon with 3GHz CPU, running Linux and Matlab version 7 (R14).

In Table 1, we compare the computation time of the DRN algorithm and the WA-TY algorithm with the two initialization strategies on small- to medium-sized data sets. We set $\epsilon = 10^{-7}$ for the WA-TY algorithm, and $\epsilon_1 = \epsilon_2 = 10^{-7}$ for the DRN algorithm (see Sun and Freund [15]). It is clear from the results that, while the computation time for algorithm DRN increases dramatically with the increase in the number of data points m, the running time for the WA-TY algorithm increases more or less linearly. Therefore while DRN is slightly faster for small-sized problems, the WA-TY algorithm shows a decisive advantage in large-scale problems compared to the DRN algorithm not combined with active set strategies. Another observation is that the Kumar-Yildirim initialization strategy demonstrates a considerable advantage over the original Khachiyan initialization strategy, especially for problems with large m.

We also tested the original FW-K and FW-KY algorithms. We stopped the algorithms after 100,000 iterations, which took from 300 to 450 seconds, at which point the optimality gap ϵ was only around 10^{-4}. It is striking that the away steps enable the FW algorithm to achieve a high degree of accuracy.

We note that [15] did not take advantage of the rank-one updating formulae for the FW-K method (called there the conditional gradient method) in the complexity analysis in the end of Section 4. The pessimistic view regarding its computation time in practice (see the end of Section 7 in [15]) is also partly due to the same error in the implementation. Our experience in this paper confirms that the correctly implemented FW-KY method is able to reach low accuracy (10^{-3}) in a reasonable time for small instances, but not high accuracy (10^{-7}).

Table 2 demonstrates the performance of the WA-TY algorithm on larger data sets, compared with the DRN algorithm combined with an active set strategy as in Sun and Freund [15]. For the Kumar-Yildirim initialization strategy, it seems that the computation time grows linearly in n

Table 1: Geometric mean of solution times of algorithms DRN and the WA-TY algorithm with the Kumar-Yildirim initialization (KY Init.) versus the Khachiyan initialization (Kha Init.), for random samples of 10 problems, using data sets for Table 2 of Sun and Freund [15].

		Geometric Mean of Time (Seconds)		
n	m	DRN	WA-TY (KY Init.)	WA-TY (Kha. Init.)
10	50	0.025	0.101	0.103
10	100	0.103	0.197	0.214
10	200	0.613	0.204	0.254
10	400	4.727	0.355	0.525
10	600	15.435	0.557	0.897
10	800	38.112	0.603	1.045
20	200	0.576	0.321	0.384
20	300	1.876	0.498	0.634
20	400	4.523	0.757	0.936
20	600	14.155	0.879	1.172
20	800	34.370	1.307	1.779
20	1000	71.292	1.289	1.982
20	1200	141.178	1.424	2.433
30	450	6.041	0.906	1.043
30	900	49.573	1.764	2.395
30	1350	187.907	2.529	3.794
30	1800	453.820	3.268	5.327

Table 2: Geometric mean of solution times and number of iterations (plus, minus and drop) of the WA-TY algorithm with the Kumar-Yildirim initialization versus the Khachiyan initialization and the DRN algorithm with an active set strategy, for random samples of 10 problems, using data sets in Table 3 of Sun and Freund [15].

Dimensions		WA-TY (KY Init.)		WA-TY (Kha. Init.)		DRN/Act. Set
n	m	Time (sec.)	# Iterations	Time (sec.)	# Iterations	Time (sec.)
20	1,000	1.24	1885.97	2.16	2974.86	0.77
10	10,000	6.06	2108.53	45.59	11943.62	0.55
20	10,000	12.84	4055.55	56.10	13828.65	2.13
20	20,000	20.07	3714.98	177.66	23755.99	2.71
20	30,000	42.87	5403.83	394.78	35328.57	3.35
30	10,000	19.60	5479.05	66.89	15137.82	7.29
30	20,000	38.32	5839.51	222.60	25941.59	8.73
30	30,000	57.98	6085.83	458.17	36032.44	9.47

and m. The results also indicate that the Kumar-Yildirim initialization is not only advantageous in theory, but also in practice, especially for large-scale problems. The superiority in the active set strategies suggests the potential of speeding up the computations by combining the WA-TY algorithm with some active set heuristics.

References

[1] C. L. Atwood. Optimal and efficient designs of experiments. *The Annals of Mathematical Statistics*, 40:1570–1602, 1969.

[2] C. L. Atwood. Sequences converging to D-optimal designs of experiments. *The Annals of Statistics*, 1:342–352, 1973.

[3] V. V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.

[4] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Nav. Res. Log. Quart.*, 3:95–110, 1956.

[5] J. Guélat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35:110–119, 1986.

[6] St. R. C. John and N. R. Draper. D-optimality for regression designs: A review. *Technometrics*, 17:15–23, 1975.

[7] L. G. Khachiyan. Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21:307–320, 1996.

[8] L. G. Khachiyan and M. J. Todd. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159, 1993.

[9] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Can. J. Math.*, 12:363–366, 1960.

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

[11] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[12] S. M. Robinson. Generalized equations and their solutions, part II: Applications to nonlinear programming. *Math. Prog. Study*, 19:200–221, 1982.

[13] N. Z. Shor. Cut-off method with space extension in convex programming problems. *Kibernetika*, 13(1):94–95, 1977. English translation: *Cybernetics* 13(1), 94–96.

[14] S. D. Silvey. *Optimal Design: An Introduction to the Theory for Parameter Estimation*. Chapman and Hall, New York, 1980.

[15] P. Sun and R. M. Freund. Computation of minimum volume covering ellipsoids. *Operations Research*, 52:690–706, 2004.

[16] M. J. Todd and E. A. Yıldırım. On Khachiyan's algorithm for the computation of minimum volume enclosing ellipsoids. Technical Report TR 1435, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York, 2005.

[17] P. Wolfe. Convergence theory in nonlinear programming. In J. Abadie, editor, *Integer and Nonlinear Programming*, pages 1–36. North-Holland, Amsterdam, 1970.

[18] H. P. Wynn. The sequential generation of D-optimum experimental design. *Annals of Mathematical Statistics*, 41:1655–1664, 1970.

[19] H. P. Wynn. Results in the theory and construction of D-optimum experimental designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34:133–147, 1972.

[20] D. B. Yudin and A. S. Nemirovskii. Informational complexity and efficient methods for the solution of convex extremal problems. *Ékonomika i Matematicheskie metody*, 12:357–369, 1976. English translation: *Matekon* 13(2), 3–25.