A new algorithm for minimizing convex functions over convex sets
(extended abstract)

Pravin M. Vaidya,
AT&T Bell Laboratories,
Murray Hill, NJ 07974.

Abstract We describe a new algorithm for minimizing a convex function over a convex set. The notion of a volumetric center of a polytope and a related ellipsoid of maximum volume inscribable in the polytope are central to the algorithm. The algorithm has a much better rate of global convergence than the ellipsoid algorithm.

1. Introduction

Let $S \subseteq \mathbf{R}^n$ be a convex set for which there is an oracle with the following property. The oracle accepts as input any point in $\mathbf{R}^n$. If the input $z \in S$ then the oracle returns a "Yes"; whereas if $z \notin S$ then the oracle returns a "No" along with a vector $c \in \mathbf{R}^n$ such that $S \subseteq \{ x : c^T x \geq c^T z \}$. The *feasibility problem* is the problem of finding a point in S given an oracle for S. The *convex optimization problem* is the problem of minimizing a convex function over S. In this paper we shall describe a new algorithm for the feasibility problem. An easy modification to the algorithm for the feasibility problem will give an algorithm for the convex optimization problem. For simplicity we shall assume that S is contained in a ball of radius 2^L centered at the origin and that if S is nonempty then it contains a ball of radius 2^{-L}. Our algorithm easily adapts to the different versions of the feasibility and the optimization problems described in [4].

A generic iterative algorithm for the feasibility problem is as follows. We maintain a region R such that $S \subseteq R$. At each iteration we choose a test point z in R and call the oracle with z as input. We halt if $z \in S$. So suppose $z \notin S$. Then the oracle returns a vector c such that $\forall\, x \in S,\ c^T x \geq c^T z$. Let $\beta \leq c^T z$. Then $S \subseteq (R \cap \{ x : c^T x \geq \beta \})$ and R is reset to be the region $(R \cap \{ x : c^T x \geq \beta \})$. As the algorithm proceeds R shrinks and its volume decreases at a certain rate. If S is nonempty then it contains a ball of radius 2^{-L} and the algorithm halts with a point in S before the volume of R falls below 2^{-nL}. If S is empty then the algorithm halts the first time the volume of R falls below 2^{-nL} and since R contains S this gives a proof that S is empty. During the course of the algorithm the description of R can become complicated and choosing the test point can become expensive; so if the region R becomes too complicated we replace R by a simpler region that contains R; such a replacement trades volume for computational efficiency and the algorithm still converges.

The well-known ellipsoid algorithm [4, 5] falls in this generic scheme; in the ellipsoid algorithm the region R is an ellipsoid and the test point used is the center of the ellipsoid. Another algorithm due to Levin [5] uses simplices instead of ellipsoids. Our algorithm also follows the above scheme. In our case the region R is a bounded full-dimensional polytope $P = \{ x : Ax \geq b \}$ where $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^n$. The test point we use is the point that minimizes the determinant of the Hessian of the logarithmic barrier for P. Specifically, the logarithmic barrier is the function $-\sum_{i=1}^{m} \ln(a_i^T x - b_i)$ and its Hessian evaluated at x, denoted by $H(x)$, is given by

$$H(x) = \sum_{i=1}^{m} \frac{a_i a_i^T}{(a_i^T x - b_i)^2}$$

where a_i^T denotes the *ith* row of A. Let $F(x) = \frac{1}{2}\ln(\det(H(x)))$ where $\det(H(x))$ denotes the determinant of $H(x)$, and let ω be the point that minimizes $F(x)$ over P. The point ω will be called the *volumetric center* of P. We use ω (or a good approximation to ω) as our test point. The function $F(x)$ is strictly convex and a Newton-type method can be used to compute a good approximation to ω efficiently. The polytope P is also trimmed from time to time (i.e. some of the planes defining P are dropped) so that the number of planes in the description of P does not grow beyond $O(n)$.

The volume of P decreases by a fixed constant factor (independent of the dimension n) at each iteration on the average, and our algorithm halts with a point in S (or with the conclusion that S is empty) in $O(nL)$ iterations. During each iteration we have to invert an $n \times n$ matrix (and solve a system of linear equations), and possibly query the oracle once. Let T be the cost (in terms of number of arithmetic operations) of one query to the oracle. Then the total number of arithmetic operations performed by our algorithm is $O(TnL + n^4 L)$, and the total number of calls to the oracle is $O(nL)$. If we use fast matrix multiplication for performing the matrix inversion the total number of arithmetic operations reduces to $O(TnL + M(n)nL)$, where $M(n)$ is the number of operations for multiplying two $n \times n$ matrices. (It is known that $M(n) = O(n^{2.38})$[3].) The ellipsoid algorithm was previously the best known algorithm for the feasibility problem. In the ellipsoid algorithm the volume falls by a factor of about $(1 - \frac{1}{n})$ at each iteration, and the number of iterations is $O(n^2 L)$. The total number of arithmetic operations in the ellipsoid algorithm is $O(Tn^2 L + n^4 L)$, and the total number of calls to the oracle in $O(n^2 L)$. (Using fast matrix multiplication does not reduce the number of operations performed by the ellipsoid algorithm.) Thus our algorithm performs asymtotically fewer operations as well as fewer calls to the oracle. The reason for

CH2806-8/89/0000/0338/$01.00 © 1989 IEEE

stressing the number of calls to the oracle is that in many cases the cost of querying the oracle far exceeds the other costs in the algorithm [4].

A natural question that arises is : Is there a simple but intuitive explanation for why is the volumetric center ω a good test point ? The question may be anwered as follows. Let $E(H(x), x, r)$ denote the ellipsoid given by

$$E(H(x), x, r) = \{ y : (y-x)^T H(x)(y-x) \le r^2 \} .$$

$E(H(x), x, 1) \subseteq P$ and may be thought of as a local quadratic approximation to P. $E(H(\omega), \omega, 1)$ has the largest volume among all such ellipsoids $E(H(x), x, 1)$ and is hence a maximum volume quadratic approximation to P. A plane through ω divides $E(H(\omega), \omega, 1)$ into two parts of equal volume; so there is a good chance that a plane through ω divides P into two parts with approximately equal volume (loosely speaking). So if the process of cutting P through ω is iterated the volume would be expected to decrease at a good rate.

There is also a simple intuitive reason for why our algorithm has a faster rate of convergence than the ellipsoid algorithm. In the ellipsoid algorithm the half-ellipsoid to which the set S is localized after an oracle query is immediately enclosed in another smaller ellipsoid and the vector c generated by the oracle is not used in subsequent steps; as a result a considerable amount of information is given up at each step. Since our algorithm works with polytopes instead of ellipsoids the cutting planes generated by the oracle are maintained for several steps after they are generated and continue to directly influence the choice of the test point. Furthermore, hyperplanes are dropped and the polytope P is trimmed not at each step but whenever necessary. As a consequence more of the information generated by the oracle gets utilized and the volume of P shrinks at a geometric rate independent of n.

A byproduct of our algorithm is an algorithm for solving linear programming problems which performs a total of $O(mn^2L + M(n)nL)$ arithmetic operations in the worst case, where m is the number of constraints and n is the number of variables; this gives an improvement in the time complexity of linear programming for $m > n^2$[8]. We also note that if the polytope P is not trimmed in our algorithm (i.e. we do not discard any plane generated by the oracle) we still get a convergent algorithm that halts in $O(n^2L^2)$ iterations.

2. An Overview

In this section we shall describe the algorithm for the feasibility problem. But first we shall introduce some notation. Let P be the bounded full-dimensional polytope

$$P = \{ x : Ax \ge b \}$$

where $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$ and $x \in \mathbf{R}^n$. Let $H(x)$ be defined as

$$H(x) = \sum_{i=1}^{m} \frac{a_i a_i^T}{(a_i^T x - b_i)^2}$$

where a_i^T denotes the *ith* row of A. $H(x)$ is the Hessian of the logarithmic barrier function $\sum_{i=1}^{m} - \ln(a_i^T x - b_i)$ and is positive definite for all x in the interior of P. Let $F(x)$ be defined as

$$F(x) = \frac{1}{2}\ln(\det(H(x)))$$

where $\det(H(x))$ denotes the determinant of $H(x)$, and let ω be the point that minimizes $F(x)$ over the polytope P. The point ω will be called the volumetric center of P. Let $\nabla F(x)$ ($\nabla^2 F(x)$) denote the gradient (Hessian) of $F(x)$ evaluated at x. Let

$$\sigma_i(x) = \frac{a_i^T H(x)^{-1} a_i}{(a_i^T x - b_i)^2}, \quad 1 \le i \le m .$$

The gradient $\nabla F(x)$ may be written as

$$\nabla F(x) = -\sum_{i=1}^{m} \sigma_i(x) \frac{a_i}{a_i^T x - b_i} .$$

Let $Q(x)$ be defined as

$$Q(x) = \sum_{i=1}^{m} \sigma_i(x) \frac{a_i a_i^T}{(a_i^T x - b_i)^2} .$$

Note that $Q(x)$ is positive definite over the interior of P. $Q(x)$ is a good approximation to $\nabla^2 F(x)$; specifically, the quadratic forms $\xi^T \nabla^2 F(x) \xi$ and $\xi^T Q(x) \xi$ satisfy the condition

$$\forall\ \xi \in \mathbf{R}^n,\ 5\,\xi^T Q(x)\xi \ \ge\ \xi^T \nabla^2 F(x) \xi \ \ge\ \xi^T Q(x) \xi .$$

Since $Q(x)$ is positive definite this condition implies that $F(x)$ is a strictly convex function over the interior of P. Let $\mu(x)$ be the largest number λ satisfying the condition that

$$\forall\ \xi \in \mathbf{R}^n,\ \xi^T Q(x) \xi \ge \lambda\, \xi^T H(x) \xi .$$

We shall now describe the algorithm for the feasibility problem. The algorithm starts out with the simplex $P = \{ x : x_j \ge -2^L,\ 1 \le j \le n,\ \sum_{j=1}^{n} x_j \le n2^L \}$. (The algorithm could start with any polytope whose volumetric center is easy to compute, say for example a box.) Since S is contained in a ball of radius 2^L centered at the origin, initially $S \subseteq P$. Throughout the algorithm S and P satisfy the relation $S \subseteq P$. Let δ and ε be small constants such that $\delta \le 10^{-4}$, and $\varepsilon \le 10^{-3}\delta$. At the beginning of each iteration we have a point $z \in P$ such that

$$F(z) - F(\omega) \le \varepsilon^4 \mu(\omega) .$$

(Note that when the algorithm starts the polytope P is just a simplex and an explicit solution to $\nabla F(x) = 0$ is easily obtained for a simplex.) The computation performed during an iteration falls into two cases depending on the value of $\min_{1 \le i \le m} \{ \sigma_i(z) \}$.

Case 1. $\min_{1 \le i \le m} \{ \sigma_i(z) \} \ge \varepsilon$.

In this case we add a plane to the polytope P. First, the oracle is called with the current point z as input. The algorithm halts if $z \in S$; otherwise the oracle returns a vector c such that

$$\forall\ x \in S,\ c^T x \ge c^T z .$$

We choose β such that $c^T z \ge \beta$ and

$$\frac{c^T H(z)^{-1} c}{(c^T z - \beta)^2} = \frac{(\delta\varepsilon)^{1/2}}{2} .$$

Let $\tilde{A} = \begin{bmatrix} A \\ c \end{bmatrix}$ and $\tilde{b} = \begin{bmatrix} b \\ \beta \end{bmatrix}$. A and B are reset as

$$A \leftarrow \tilde{A} , \quad b \leftarrow \tilde{b} .$$

Since ω shifts due to the addition of a plane to P, we use a Newton-type method to move closer to ω as follows.

For $j=1$ to $\lceil 30\ln(2\varepsilon^{-4.5}) \rceil$ do $z \leftarrow z - 0.18\, Q(z)^{-1} \nabla F(z)$.

Case 2. $\min_{1 \le i \le m} \{ \sigma_i(z) \} < \varepsilon$.

In this case we remove a plane from the polytope P. Wlog suppose that $\sigma_m(z) = \min_{1 \le i \le m} \{ \sigma_i(z) \}$. Let $a_m = c$, $b_m = \beta$, $A = \begin{bmatrix} \tilde{A} \\ c \end{bmatrix}$, and $b = \begin{bmatrix} \tilde{b} \\ \beta \end{bmatrix}$. A and b are reset as

$$A \leftarrow \tilde{A} , \quad b \leftarrow \tilde{b} .$$

Since ω shifts due to the removal of a plane, we use a Newton-type method to move closer to ω as follows.

For $j=1$ to $\lceil 30\ln(4\varepsilon^{-3}) \rceil$ do $z \leftarrow z - 0.18\, Q(z)^{-1} \nabla F(z)$.

The convergence lemma below summarizes the behaviour of the algorithm; its proof will be given in the full paper.

Convergence Lemma. Let $\delta \le 10^{-4}$, let $\varepsilon \le 10^{-3}\delta$, and let ρ^k denote the value of $F(\omega)$ at the beginning of the *kth* iteration. Then at the beginning of each iteration z satisfies the condition

$$F(z) - F(\omega) \le \varepsilon^4 \mu(\omega) .$$

Futhermore, the following statements hold.

1. If Case 1 occurs during the *kth* iteration then
$$\rho^{k+1} - \rho^k \ge \frac{(\delta\varepsilon)^{1/2}}{5} .$$
2. If Case 2 occurs during the *kth* iteration then
$$\rho^k - \rho^{k+1} \le 5\varepsilon \quad \blacksquare$$

Bounding the number of iterations. Let π^k denote the volume of the polytope P at the beginning of the *kth* iteration. Using the Convergence Lemma we shall next obtain an upper bound on π^k, and show that the algorithm halts in $O(nL)$ iterations. An easy consequence of the Convergence Lemma is that

$$\rho^k \ge \rho^0 + \frac{k\varepsilon}{2} .$$

We bound π^k as follows. Note that if x^* is the point that maximizes the logarithmic barrier over P, then [see 6]

$$P \subseteq \{ x : (x - x^*)^T H(x^*)(x - x^*) \le m^2 \} .$$

Thus

$$\begin{aligned} volume(P) &\le (2m)^n (\det(H(x^*)))^{-1/2} \\ &\le (2m)^n (\det(H(\omega)))^{-1/2} \\ &\le (2m)^n e^{-F(\omega)} . \end{aligned}$$

Since $\sum_{i=1}^{m} \sigma_i(x) = n$ (Claim 3, section 7.1), m cannot exceed n/ε. Then as $\rho^0 \ge -(n(L+1) + \ln(n+1))$, we get that

$$\begin{aligned} \ln(\pi^k) &\le n\ln(2m) - \rho^k \\ &\le n\ln(2n/\varepsilon) - \rho^0 - \frac{k\varepsilon}{2} \\ &\le n(L + \ln(2n/\varepsilon) + 1) + \ln(n+1) - \frac{k\varepsilon}{2} . \end{aligned}$$

Thus the volume of P must fall below 2^{-nL} in $O(nL)$ iterations. Hence the algorithm must halt in $O(nL)$ iterations since $S \subseteq P$ and S contains a ball of radius 2^{-L} if it is nonempty.

Number of arithmetic operations. Since $m = O(n)$, the weights $\sigma_i(x)$ may be computed in $O(n^3)$ operations, and then $Q(z)$, $Q(z)^{-1}$ and $\nabla F(z)$ may be evaluated in $O(n^3)$ additional operations. Furthermore, the oracle is called at most once per iteration and one such call costs T operations. It then follows that the number of operations per iteration is $O(T + n^3)$. Using fast matrix multiplication the number of operations per iteration may be reduced to $O(T + M(n))$ where $M(n)$ is the number of operations for multiplying two $n \times n$ matrices. (It is known that $M(n) = O(n^{2.38})$ [3].) Since the number of iterations is $O(nL)$, the total number of operations is $O(TnL + n^4 L)$ without fast matrix multiplication and $O(TnL + M(n)nL)$ with fast matrix multiplication. The total number of calls to the oracle is $O(nL)$.

3. Adding/deleting a plane and moving closer to the volumetric center ω

In this section we shall discuss three theorems on which the proof of the Convergence Lemma in section 2 is based; their proofs will appear in the full paper.

Theorem 1. Let $\delta \le 10^{-4}$, let $\eta = Q(z)^{-1}\nabla F(z)$, and let r be a scalar such that $0 \le r \le 0.2$. Let $z' = z - \lambda\eta$ where λ is defined as follows.

If $F(z) - F(\omega) \le \delta\sqrt{\mu(\omega)}$ then $\lambda = r$ else $\lambda = \dfrac{r\delta^{1/2}(\mu(z))^{1/4}}{(\nabla F(z)^T \eta)^{1/2}}$.

Then the following statements hold.

1. If $F(z) - F(\omega) \le \delta\sqrt{\mu(\omega)}$ then
$F(z') - F(\omega) \le (1 - 0.71r + 1.9r^2)(F(z) - F(\omega))$.
2. If $F(z) - F(\omega) > \delta\sqrt{\mu(\omega)}$ then
$$F(z) - F(z') \ge \frac{(r - 2.65r^2)\,\delta}{2\sqrt{m}} \quad \blacksquare$$

Theorem 1 states that a Newton-type algorithm for minimizing $F(x)$ will converge linearly if started from a point z such that $F(z) - F(\omega) \le \delta\sqrt{\mu(\omega)}$, $\delta \le 10^{-4}$. It also states that taking a Newton-like step from a point z such that $F(z) - F(\omega) > \delta\sqrt{\mu(\omega)}$ will decrease F by at least $\Omega(1/\sqrt{m})$.

The next two theorems address the following question: by how much does the minimum value of $F(\omega)$ increase (decrease) when we add (remove) the constraint $c^T x \ge \beta$ to (from) the set of constraints defining the polytope P? We shall require some additional notation to denote the polytope obtained by adding (removing) a plane to (from) P, and the related functions and matrices. Let $\tilde{A} \in \mathbf{R}^{\tilde{m} \times n}$, $\tilde{b} \in \mathbf{R}^{\tilde{m}}$, and let $\tilde{P}$ be the polytope $\tilde{P} = \{ x : \tilde{A}x \geq \tilde{b} \}$. Let $\tilde{H}(x)$ be defined as

$$\tilde{H}(x) = \sum_{i=1}^{\tilde{m}} \frac{\tilde{a}_i \tilde{a}_i^T}{(\tilde{a}_i^T x - \tilde{b}_i)^2},$$

let $\tilde{F}(x) = \frac{1}{2} \ln(\det(\tilde{H}(x)))$, and let $\tilde{\omega}$ be the point that minimizes $\tilde{F}(x)$ over the polytope $\tilde{P}$. Let $\tilde{\sigma}_i(x) = \frac{\tilde{a}_i^T \tilde{H}(x)^{-1} \tilde{a}_i}{(\tilde{a}_i^T x - \tilde{b}_i)^2}$, $1 \leq i \leq \tilde{m}$, let

$$\tilde{Q}(x) = \sum_{i=1}^{\tilde{m}} \tilde{\sigma}_i(x) \frac{\tilde{a}_i \tilde{a}_i^T}{(\tilde{a}_i^T x - \tilde{b}_i)^2}$$

and let $\tilde{\mu}(x)$ be the largest number λ such that $\forall \xi \in \mathbf{R}^n$, $\xi^T \tilde{Q}(x) \xi \geq \lambda\, \xi^T \tilde{H}(x) \xi$.

Theorem 2. Let $\tilde{A} = \begin{bmatrix} A \\ c \end{bmatrix}$, $\tilde{b} = \begin{bmatrix} b \\ \beta \end{bmatrix}$ and $\tilde{P} = P \cap \{ x : c^T x \geq \beta \}$. Let $z \in P$, let $\alpha \leq \delta \leq 10^{-4}$ and let $\alpha \leq \delta\, \mu(z)$. Suppose that $F(z) - F(\omega) \leq \alpha^2\, \mu(\omega)$ and that $\frac{c^T H(z)^{-1} c}{(c^T z - \beta)^2} = \frac{\alpha^{1/2}}{2}$. Then

$$\tilde{F}(z) - \tilde{F}(\tilde{\omega}) \leq 0.33(\delta\alpha)^{1/2} \leq 0.43\delta \sqrt{\tilde{\mu}(\tilde{\omega})}$$

and

$$\tilde{F}(\tilde{\omega}) - F(\omega) \geq \frac{\alpha^{1/2}}{5} \quad \blacksquare$$

Theorem 3. Let $A = \begin{bmatrix} \tilde{A} \\ c \end{bmatrix}$, $b = \begin{bmatrix} \tilde{b} \\ \beta \end{bmatrix}$ and $P = \tilde{P} \cap \{ x : c^T x \geq \beta \}$. Let $z \in P$, let $\alpha \leq \delta \leq 10^{-4}$, and let $F(z) - F(\omega) \leq \alpha^2\, \mu(\omega)$. Suppose that $\frac{c^T H(z)^{-1} c}{(c^T z - \beta)^2} \leq \min \{ \alpha, \mu(z) \}$. Then the polytope $\tilde{P}$ is bounded,

$$\tilde{F}(z) - \tilde{F}(\tilde{\omega}) \leq \min \{ 4\alpha\, \tilde{\mu}(\tilde{\omega}), \delta \sqrt{\tilde{\mu}(\tilde{\omega})} \},$$

and

$$F(\omega) - \tilde{F}(\tilde{\omega}) \leq 5\alpha \quad \blacksquare$$

4. Variants of the algorithm

The algorithm in section 2 is designed to obtain the best worst case time complexity. But an algorithm that has best the worst case running time may not necessarily be the one that gives the best performance in practice. Building on the ideas in the basic algorithm we can construct a wide variety of algorithms for the solution of convex programming problems. This will give us the flexibility of being able to design algorithms that suit the given problem and to exploit any additional information about or any special structure in the set of constraints describing S if any. Several variants of the basic algorithm are possible. One possibility is to keep on adding planes generated by the oracle without ever removing any plane (i.e. discard Case 2 from the algorithm); such an algorithm would converge in $O(n^2 L^2)$ iterations since by Theorem 2 (section 3, with $\alpha = \delta / m$) the value of $F(\omega)$ would increase by $\Omega(1/\sqrt{m})$ at each iteration. As a sample we shall describe two more ways of obtaining variants of the basic algorithm.

The volumetric center as a weighted analytic center. The weighted analytic center $\pi(w)$ of the polytope P is the point that minimizes the weighted logarithmic barrier function

$$logbar(w, x) = - \sum_{i=1}^{m} w_i \ln(a_i^T x - b_i)$$

over P where w_i, $1 \leq i \leq m$, are positive weights. (w_i is the weight on the plane $a_i^T x = b_i$.) The gradient of the weighted logarithmic barrier is given by

$$\nabla logbar(w, x) = - \sum_{i=1}^{m} w_i \frac{a_i}{a_i^T x - b_i}.$$

Comparing this with

$$\nabla F(x) = - \sum_{i=i}^{m} \sigma_i(x) \frac{a_i}{a_i^T x - b_i}$$

we get that the volumetric center ω is the minimizer of the weighted logarithmic barrier $- \sum_{i=1}^{m} \sigma_i(\omega) \ln(a_i^T x - b_i)$.

In the basic algorithm in section 2 the volumetric center of P is used as a test point. The idea is to use a weighted analytic center as a test point instead of the volumetric center. The weights $\sigma_i(x)$ would guide the choice of the weights w_i. One possibility is to use a weighted analytic center $\pi(w)$ such that the weights w_i satisfy the condition $\alpha_1\, \sigma_i(\pi(w)) \leq w_i \leq \alpha_2\, \sigma_i(\pi(w))$, $1 \leq i \leq m$, where α_1, α_2 are some constants. The key point is to ensure that the weighted analytic center $\pi(w)$ does not lie close to a plane with a small weight on it. One important reason for looking for variants along these lines is as follows. The main computational effort in the basic algorithm (except for querying the oracle) is in computing the weights $\sigma_i(x)$; so if one can design an algorithm where it suffices to compute coarse approximations to the weights $\sigma_i(x)$ then it could lead to a better running time in theory and/or practice. Even better would be an algorithm that somehow uses these weights implicitly and does not require their explicit computation.

Combination of determinant barrier and logarithmic barrier. Suppose we want to solve

$$\begin{aligned} \min \quad & g(x) \\ \text{s.t.} \quad & x \in P \end{aligned}$$

where $g(x)$ is a differentiable convex function. An iterative algorithm for the solution of this problem is as follows. During the kth iteration we choose a test point $z(k)$ in P and compute a vector $c(k)$ (by differentiating $g(x)$ at $z(k)$) such that

$$\{ x : g(x) \leq g(z(k)) \} \subseteq \{ x : c(k)^T x \geq c(k)^T z(k) \}$$

and compute a suitable $\beta(k)$ such that $c(k)^T z(k) > \beta(k)$. Let

$$B(k, x)) = r\, I + \sum_{i=1}^{k-1} \frac{c(k) c(k)^T}{(c(k)^T x - \beta(k))^2}$$

where $r > 0$ is a suitable fixed scale factor, and let

$$\psi(k, x) = \ln(\det(B(k, x))) - \sum_{i=1}^{m} \ln(a_i^T x - b_i) .$$

The test point $z(k)$ is chosen to be the minimizer (or a good approximation to the minimizer) of $\psi(k, x)$ over the polytope $P \cap \{ x : c(j)^T x \geq \beta(j), 1 \leq j \leq k-1 \}$. $\psi(k, x)$ consists of a determinant barrier together with the logarithmic barrier for P; the determinant component pushes $z(k)$ towards decreasing values of $g(x)$ and the logarithmic barrier keeps $z(k)$ away from the boundaries of P.

5. Linear programming via path of volumetric centers

Consider the linear programming problem

$$\max \ c^T x$$
$$\text{s.t.} \quad x \in P .$$

Various known interior point algorithms for this problem follow the path of analytic centers to the optimum [6,7]. (The analytic center is the weighted analytic center with each of the weights w_i equal to 1.) Instead we can design an algorithm that follows the path of volumetric centers. The path of volumetric centers is defined by the equation

$$\nabla F(x) = tc , \quad t \in \mathbf{R}, \ t \geq 0 .$$

It is the set of all points in the polytope P where the gradient of $F(x)$ is a non-negative multiple of the cost vector c. Such an algorithm would start from the volumetric center and follow the path of volumetric centers using Newton-Raphson steps in a manner similar to the algorithms that follow the path of analytic centers [6,7].

Another possibility is to follow a path of hybrid centers. The path of hybrid centers is defined by

$$\nabla F(x) + r \, \nabla logbar(e, x) = tc , \quad t \in \mathbf{R}, \ t \geq 0 .$$

where $e \in \mathbf{R}^m$ is the vector of all ones, and r is a fixed positive constant. Note that $logbar(e, x)$ is just the logarithmic barrier for P; so the hybrid center may be thought of as a combination of the analytic center and the volumetric center. The author has obtained an algorithm that follows a path of hybrid centers (with $r = n/m$) and solves linear programming problems in $O((mn)^{1/4} L)$ iterations; each iteration is a Newton-Raphson step and involves inverting a matrix and solving a system of linear equations. (Here L is a standard parameter; for a definition of L see [7].) This improves on the previously best known bound of $O(\sqrt{m}\, L)$ iterations [6] when $n = o(m)$. Details and a complete presentation will be given in a subsequent paper.

6. Properties of $F(x)$

In this section we shall study the function $F(x)$. Let the polytope P, $H(x)$, $F(x)$, the volumetric center ω, $\sigma_i(x)$, $Q(x)$, and $\mu(x)$ be as defined in section 2. Let $\Sigma(x, r)$ be the region

$$\Sigma(x, r) = \{ y : \forall \ i, \ 1 \leq i \leq m, \ \left| \frac{a_i^T (y - x)}{a_i^T x - b_i} \right| \leq r \} .$$

Note that if $r \leq 1$ then $\Sigma(x, r) \subseteq P$. Lemmas 1 through 10 below summarize some of the properties of $F(x)$. Proofs of these lemmas will be given in a full version of the paper. Lemmas 1 and 2 give explicit formulae for the gradient and the Hessian of $F(x)$ respectively.

Lemma 1.

$$\nabla F(x) = - \sum_{i=1}^{m} \sigma_i(x) \ \frac{a_i}{a_i^T x - b_i} \quad \blacksquare$$

Lemma 2. Let $u_{ij} = \dfrac{a_i}{a_i^T x - b_i} - \dfrac{a_j}{a_j^T x - b_j}$. Then

$$\nabla^2 F(x) = Q(x) + 2 \sum_{1 \leq i < j \leq m} \frac{(a_i^T H(x)^{-1} a_j)^2}{(a_i^T x - b_i)^2 (a_j^T x - b_j)^2} \, u_{ij} u_{ij}^T \quad \blacksquare$$

Lemma 3 states that $Q(x)$ serves as a good approximation to $\nabla^2 F(x)$.

Lemma 3. The matrices $Q(x)$ and $\nabla^2 F(x)$ satisfy the condition

$$\forall \ \xi \in \mathbf{R}^n, \quad 5 \, \xi^T Q(x) \xi \ \geq \ \xi^T \nabla^2 F(x) \xi \ \geq \ \xi^T Q(x) \xi .$$

Hence, $\nabla^2 F(x)$ is positive definite and $F(x)$ is strictly convex over the interior of P. ■

Lemma 4 states that the value of the quadratic form $\xi^T Q(x) \xi$ does not deviate too far from the value of the quadratic form $\xi^T H(x) \xi$ and gives bounds on $\mu(x)$.

Lemma 4.

$$\forall \ \xi \in \mathbf{R}^n, \quad \xi^T H(x) \xi \ \geq \ \xi^T Q(x) \xi \geq \frac{1}{4m} \, \xi^T H(x) \xi ,$$

and thus

$$1 \geq \mu(x) \geq \max \{ \min_{1 \leq i \leq m} \{\sigma_i(x)\}, \ \frac{1}{4m} \} .$$

Lemma 5 formalizes the observation that for all x in $\Sigma(\hat{x}, r)$ the quadratic form $\xi^T Q(x) \xi$ does not deviate too far from the quadratic form $\xi^T Q(\hat{x}) \xi$ if r is less than some small constant.

Lemma 5. Suppose that $r < 1$ and that $x \in \Sigma(\hat{x}, r)$. Then for all $\xi \in \mathbf{R}^n$,

$$\frac{(1-r)^2}{(1+r)^4} \, \xi^T Q(\hat{x}) \xi \ \leq \ \xi^T Q(x) \xi \ \leq \ \frac{(1+r)^2}{(1-r)^4} \, \xi^T Q(\hat{x}) \xi$$

and

$$\frac{(1-r)^4}{(1+r)^4} \, \mu(\hat{x}) \leq \mu(x) \leq \frac{(1+r)^4}{(1-r)^4} \, \mu(\hat{x}) \quad \blacksquare$$

Consider the equation $\nabla F(x) = t\, w$ where t is a scalar and w is a fixed n-dimensional vector. This equation implicitly defines x as a function of t and Lemma 6 summarizes some of the properties of this implicitly defined function that can be derived from the implicit function theorem [1, 2].

Lemma 6. Let $\Phi(x, t) = \nabla F(x) - t\,w$ where $t \in \mathbf{R}$ and w is a fixed vector in $\mathbf{R}^n$. Then the equation $\Phi(x, t) = 0$ implicitly defines x as a function of t, and we may write $x = x(t)$. Moreover, $x(t)$ is an analytic function of t, and $\frac{dx(t)}{dt}$, the derivative of $x(t)$ w.r.t t evaluated at t, may be written as

$$\frac{dx(t)}{dt} = \nabla^2 F(x)^{-1} w,$$

and if $0 \le t_1 \le t_2$ then

$$\frac{1}{5}\int_{t_1}^{t_2} t\, w^T Q(x)^{-1} w \, dt \le F(x(t_2)) - F(x(t_1)) \le \int_{t_1}^{t_2} t\, w^T Q(x)^{-1} w \, dt \quad \blacksquare$$

Consider the trajectory $\nabla F(x) = t\,w$, where $t \in \mathbf{R}$ and w is fixed, that passes through $\hat{x}$. Lemma 7 gives an upper bound on the derivative of $\ln(a_i^T x(t) - b_i)$ w.r.t t for the portion of this trajectory in $\Sigma(\hat{x}, r)$ in terms of quantities evaluated at $\hat{x}$. Lemma 8 gives a lower bound on how much t must change before the trajectory reaches the boundary of $\Sigma(\hat{x}, r)$.

Lemma 7. Let w be a fixed vector in $\mathbf{R}^n$, and let $\hat{x}$ be such that $\nabla F(\hat{x}) = \hat{t}\,w$ for some scalar $\hat{t}$. Let $t \in R$, and let $x = x(t)$ be a point on the trajectory $\nabla F(x) = t\,w$ such that $x \in \Sigma(\hat{x}, r)$, $r < 1$. Then for $1 \le i \le m$,

$$\left| \frac{a_i^T \frac{dx(t)}{dt}}{a_i^T x - b_i} \right| \le \frac{(1+r)^3}{(1-r)^2} \frac{(w^T Q(\hat{x})^{-1} w)^{1/2}}{(\mu(\hat{x}))^{1/4}} \quad \blacksquare$$

Lemma 8. Let $r < 1$, let w be a fixed vector in $\mathbf{R}^n$, and let $\hat{x}$ be such that $\nabla F(\hat{x}) = \hat{t}\,w$ for some scalar $\hat{t}$. Let $x(\bar{t})$ be a point on the trajectory $\nabla F(x) = t\,w$, $t \in \mathbf{R}$, such that $x(\bar{t})$ does not lie in the interior of $\Sigma(\hat{x}, r)$. Then

$$|\hat{t} - \bar{t}| > \frac{(r - \frac{r^2}{2})(1-r)^2 (\mu(\hat{x}))^{1/4}}{(1+r)^3 \sqrt{w^T Q(\hat{x})^{-1} w}} \quad \blacksquare$$

Lemma 9 gives a sufficient condition in terms of $\nabla F(z)^T Q(z)^{-1} \nabla F(z)$ and $\mu(z)$ for the point z to be in the region $\Sigma(\omega, r)$.

Lemma 9. Let $\delta \le 10^{-4}$, let $z \in P$ and suppose that $\nabla F(z)^T Q(z)^{-1} \nabla F(z) \le \delta \sqrt{\mu(z)}$. Then $\omega \in \Sigma(z, 1.1\sqrt{\delta})$, $\mu(z) \le 1.1\,\mu(\omega)$, and

$$F(z) - F(\omega) \le 0.55\, \nabla F(z)^T Q(z)^{-1} \nabla F(z) \quad \blacksquare$$

Lemma 10 states that if $F(z) - F(\omega)$ is small then the quantities $F(z) - F(\omega)$ and $\nabla F(z)^T Q(z)^{-1} \nabla F(z)$ closely track each other.

Lemma 10. Let $\delta \le 10^{-4}$ and let z be a point in P such that $F(z) - F(\omega) \le \delta \sqrt{\mu(\omega)}$. Then $z \in \Sigma(\omega, 5\sqrt{\delta})$, $\mu(\omega) \le 1.5\,\mu(z)$, and

$$0.14\, \nabla F(z)^T Q(z)^{-1} \nabla F(z) \le F(z) - F(\omega) \le 1.4\, \nabla F(z)^T Q(z)^{-1} \nabla F(z) \quad \blacksquare$$

6.1. Some useful claims

We shall state four claims which are used in the proofs of the Lemmas and Theorems. For a symmetric positive definite $n \times n$ matrix B, we shall let $E(B, x, r)$ denote the ellipsoid given by

$$E(B, x, r) = \{ y : (y-x)^T B (y-x) \le r^2 \}.$$

Claim 1. Let B be a positive definite matrix, and let w be an arbitrary fixed vector in $\mathbf{R}^n$. Then

$$\max_{y \in E(B, x, r)} \{ (w^T (y-x))^2 \} = r^2\, w^T B^{-1} w .$$

Claim 2. Let $\theta > 0$, and let B_1, B_2 be $n \times n$ positive definite matrices. Suppose that $\forall\, \xi \in \mathbf{R}^n$, $\xi^T B_1 \xi \ge \theta\, \xi^T B_2 \xi$. Then $\forall\, \xi \in \mathbf{R}^n$, $\xi^T B_1^{-1} \xi \le \frac{1}{\theta} \xi^T B_2^{-1} \xi$.

Claim 3. $\sigma_i(x) = \sum_{j=1}^{m} \frac{(a_i^T H(x)^{-1} a_j)^2}{(a_i^T x - b_i)^2 (a_j^T x - b_j)^2}$, and $\sigma_i(x) \le 1$, $1 \le i \le m$. Moreover, $\sum_{i=1}^{m} \sigma_i(x) = n$.

Claim 4. $\frac{a_i^T Q(x)^{-1} a_i}{(a_i^T x - b_i)^2} \le \frac{1}{\sqrt{\mu(x)}}$, $1 \le i \le m$, and thus

$$E(Q(x), x, (\mu(x))^{1/4} r) \subseteq \Sigma(x, r) .$$

References

1. G. A. Bliss, Lectures on the calculus of variations, Phoenix Science Series, The University of Chicago Press, Chicago 37, 1946.
2. S. Bochner, and W. T Martin, Several Complex variables, Princeton University Press, Princeton, 1948.
3. D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *Proc. 19th Annual ACM Symp. Theory of Computing*, (May 1987) 1-6.
4. M. Grotschel, L. Lovasz, and A. Schrijver, Geometric algorithms and combinatorial optimization, Springer-Verlag Berlin Heidelberg, 1988.
5. M. Minoux, Mathematical Programming: Theory and Algorithms, John Wiley & Sons Ltd., New York, 1986.
6. J. Renegar, A polynomial-time algorithm based on Newton's method for linear programming, *Mathematical Programming*, 40, (1988), 59-93.
7. P. M. Vaidya, An algorithm for linear programming that requires $O(((m+n)n^2 + (m+n)^{1.5} n) L)$ arithmetic operations, *Proceedings 19th Annual ACM Symposium Theory of Computing*, (May 1987), pp. 29-38.
8. P. M. Vaidya, Speeding-up linear programming using fast matrix multiplication, Technical Memorandum, AT&T Bell Laboratories, Murray Hill, NJ, 1989; also to appear in *Proceedings 30th Annual IEEE Symposium Foundations of Computer Science*.