Mathematical Programming 57 (1992) 163-192
North-Holland

A scaling technique for finding the weighted analytic center of a polytope

David S. Atkinson*
Department of Mathematics, University of Illinois at Urbana-Champaign, Urbana, USA

Pravin M. Vaidya**
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, USA

Received 15 June 1991
Revised manuscript received 12 November 1991

Let a bounded full dimensional polytope be defined by the system $Ax \geqslant b$ where A is an $m \times n$ matrix. Let a_i denote the ith row of the matrix A, and define the *weighted analytic center* of the polytope to be the point that minimizes the strictly convex barrier function $-\sum_{i=1}^{m} w_i \ln(a_i^{\mathrm{T}} x - b_i)$. The proper selection of weights w_i can make any desired point in the interior of the polytope become the weighted analytic center. As a result, the weighted analytic center has applications in both linear and general convex programming. For simplicity we assume that the weights are positive integers.

If some of the w_i's are much larger than others, then Newton's method for minimizing the resulting barrier function is very unstable and can be very slow. Previous methods for finding the weighted analytic center relied upon a rather direct application of Newton's method potentially resulting in very slow global convergence. We present a method for finding the weighted analytic center that is based on the scaling technique of Edmonds and Karp and is an enhancement of Newton's method. The scaling algorithm runs in $O(\sqrt{m} \log W)$ iterations, where m is the number of constraints defining the polytope and W is the largest weight given on any constraint. Each iteration involves taking a step in the Newton direction and its complexity is dominated by the time needed to solve a system of linear equations.

1. Introduction

Interior point algorithms for linear programming have been the focus of much research in recent years. Interior point methods for general convex optimization have also been developed, though less attention has been given them so far. A common feature of these algorithms is the use of some appropriately defined center of a bounded constraint polytope as a reference point for improving the value of the objective function. Typically, the linear programming algorithms involve a sequence of iterations in which the polytope is altered and the resulting centers converge toward an optimal facet of the polytope, while the general convex optimization algorithms involve a sequence of operations that use some test at the center

Correspondence to: Dr. David S. Atkinson, Department of Mathematics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.
* Supported by the Campus Research Board, University of Illinois at Urbana-Champaign.
** Supported by the National Science Foundation under Grants CCR-9057481 and CCR-9007195.

as a way to cut away part of the constraint region in which the test indicates the optimal solution can not lie. Usually, the center is taken as the minimum of some convex barrier function defined on the interior of the polytope.

Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. Define $P \subset \mathbb{R}^n$ to be the polytope given by $\{x\colon Ax \geqslant b\}$. We will make the assumption that P is bounded and full dimensional. Consider now a barrier function over the polytope P of the form

$$F(x) = -\sum_{i=1}^{m} w_i \ln(a_i^{\mathrm{T}}x - b_i)$$

where $w_i > 0$ for all i. We call F the *weighted logarithmic barrier* function. It is clear that F is strictly convex over P. The *weighted analytic center* is the unique minimizer of F over P.

The *analytic center*, namely a weighted analytic center with all weights set to 1, has been extensively used in interior point methods for linear programming [7, 9, 10, 12]. Clearly, the weighted analytic center is just a generalization of the analytic center. Renegar [9] used the analytic center to develop a linear programming algorithm in which the objective plane is advanced in the direction of increasing value and the sequence of resulting analytic centers converges toward an optimum. Vaidya [12] developed a similar algorithm for linear programming that also used the analytic center. The primary difference between the two algorithms was that Renegar used Euclidean distance as a measure of closeness to the analytic center while Vaidya used the barrier function itself as a measure of closeness, as we will do in what follows. The measure of closeness to the analytic center is an important factor in the convergence of center-following algorithms. A few other possible measures of closeness are discussed in [11].

In this paper, we consider finding the weighted analytic center for an arbitrary vector of positive integer weights $(w_1, \ldots, w_m)$. The weighted analytic center has numerous applications. One application is a very direct approach to solving linear programs. Suppose we wish to maximize an objective function $c^{\mathrm{T}}x$ over the polytope P. Suppose further that we have a suitable lower bound β on the objective function such that $P \subset \{x\colon c^{\mathrm{T}}x \geqslant \beta\}$. In this case, we can give the objective plane an "exponentially" large weight — setting the weights on the other planes to 1 — and it can be easily shown that the weighted analytic center is sufficiently close to the optimum to allow isolation of an optimum vertex of the polytope.

The weighted analytic center, and thus the technique for finding it, also has applications in general convex optimization, i.e., minimizing a convex function g over a convex set S. Vaidya developed an algorithm for convex optimization in [13]. One interpretation of his algorithm is that it maintains a polytope known to contain S and uses the gradient of g at a weighted analytic center to allow the cutting away of portions of the polytope that can not contain the minimizer over S. The process repeats with a new weighted analytic center over the resulting polytope. In this interpretation of Vaidya's algorithm, the weights are "implicit" and depend on the current center. If, using some heuristic criteria, the weights can be explicitly altered to lead to reasonable approximations of successive weighted centers, then the method we present would fit nicely into the minimization scheme.

Given a vector $(w_1, \ldots, w_m)$ of weights, by definition, finding the weighted analytic center simply involves minimizing the weighted barrier function. In this paper, we will do this by taking steps in the Newton direction. In general, the relative sizes of the weights can have a profound effect on the ease with which Newton's method can be applied. Furthermore, Newton's method must have a reasonably good starting point in order to give good convergence. If, for example, one weight is very large relative to the others, then the set of acceptable starting points can become prohibitively small.

A simple example will illustrate the difficulty. Suppose we want to find the weighted analytic center of the polytope $[0, 1] \subset \mathbb{R}$ where the facet $x \geqslant 0$ is given a weight W, while the facet $1-x \geqslant 0$ is given a weight of 1. In this case, we have $F(x) = -\ln(1-x) - W \ln x$. Then $F'(x) - 1/(1-x) - W/x$, $F''(x) = 1/(1-x)^2 + W/x^2$.

It is clear in this simple case that the weighted analytic center is $\omega = W/(1+W)$. Consider using Newton's method to find ω. The formula is

$$\begin{aligned} x_{k+1} &= x_k - \frac{F'(x_k)}{F''(x_k)} \\ &= x_k - \frac{x_k^2(1-x_k) - Wx_k(1-x_k)^2}{x_k^2 + W(1-x_k)^2} \\ &= \frac{2Wx_k(1-x_k)^2 + x_k^3 - x_k^2(1-x_k)^2}{W(1-x_k)^2 + x_k^2}. \end{aligned}$$

We consider now the set

$$\left\{ x \in (0, 1): \frac{2Wx(1-x)^2 + x^3 - x^2(1-x)^2}{W(1-x)^2 + x^2} \geqslant 1 \right\}.$$

This is the set of feasible points such that the next Newton step would lead to a point to the right of the feasible region. A little algebraic manipulation shows this is the same set as

$$\{x \in (0, 1): W(1-x)(2x-1) - 2x^2 \geqslant 0\}.$$

Notice now that at $x=0$ we have $W(1)(-1) - 2(0)^2 < 0$ and at $x=1$ we have $W(0)(1) - 2(1)^2 < 0$. However, at $x = \frac{3}{4}(W/(W+1))$, we get $(W^2 - 8W)/(8(1+W)) > 0$ (so long as $W > 8$). We can thus omit the condition that $x \in (0, 1)$ from the above set, because we have shown that all possible points must lie in $(0, 1)$ anyway. Now, using the quadratic formula, we can say

$$\begin{aligned} &\{x: W(1-x)(2x-1) - 2x^2 \geqslant 0\} \\ &= \left[\tfrac{3}{4}\left(\frac{W}{W+1}\right) - \tfrac{1}{4}\sqrt{\frac{W^2 - 8W}{(1+W)^2}}, \tfrac{3}{4}\left(\frac{W}{W+1}\right) + \tfrac{1}{4}\sqrt{\frac{W^2 - 8W}{(1+W)^2}} \right]. \end{aligned}$$

Denote this interval as $[a(W), b(W)]$; then

$$\lim_{W\to\infty} \text{length}[a(W), b(W)] = \lim_{W\to\infty} \tfrac{1}{2}\sqrt{\frac{W^2-8W}{(1+W)^2}} = \tfrac{1}{2}.$$

That is, the interval of points such that the next step leads to an infeasible point becomes nearly half the entire feasible set as $W\to\infty$. This result is not a complete analysis of which points could be good starting points for Newton's method — in general a difficult task — but it illustrates the difficulties that can be encountered when the weights differ greatly. The difficulties can be even more acute when working in n dimensions. A key issue, then, is to ensure that Newton's method will converge from whatever point we consider to be the *current* point in the algorithm.

Freund considered the weighted analytic center in [2]. Using projective transformations, he showed that beginning from an easily obtainable starting point, a sequence of points found by successive steps in Newton directions will eventually converge superlinearly to the weighted analytic center. Freund did not give any specific growth bounds on the number of such steps based on the weights or the number of constraints. The algorithm to be presented will give such bounds, and will benefit within each iteration from the same superlinear convergence as Freund's approach. The primary improvement of the algorithm is that it maintains the assurance of good convergence by Newton's method in a novel way: it uses the technique of scaling the weights.

Introduced by Edmonds and Karp [1], scaling has found wide applicability in weighted combinatorial optimization problems (see [4]). Suppose we have a vector of weights $w=(w_1,\ldots,w_m)$. The basic idea is that we recursively solve the given problem with all weights w_i replaced by weights $\lfloor\frac{1}{2}w_i\rfloor$. The solution with weights $\lfloor\frac{1}{2}w_i\rfloor$ can then be used to find a solution for the problem with the original weights. Typically, then, if producing a solution to the problem with full weights w_i using the solution with halved weights $\lfloor\frac{1}{2}w_i\rfloor$ takes $T(m,n)$ time, then the final solution will be produced in $T(m,n)(\lfloor\log_2 W\rfloor+1)$ time where W is the largest weight in the problem.

A slightly different point of view that is actually equivalent to the recursive process described in the last paragraph is to consider the binary representations of the weights in the problem

$$w_i = b_{i_0}b_{i_1}\cdots b_{i_p}$$

where $b_{i_j}\in\{0,1\}$ and $p=\lfloor\log_2 W\rfloor$ where W is the maximum weight in the problem. The algorithm begins by solving the given problem, but with each weight set to the most significant bit of the original weight. It then moves on to consider successively less significant bits in the weights. In particular, during the kth iteration, we double all current weight settings, so we have

$$w_{i_k} = b_{i_0}b_{i_1}\cdots b_{i_{k-1}}0.$$

If $b_{i_k}=0$, the current weight setting is correct for the kth iteration. Otherwise, we need to add 1 to the ith weight. Adding 1's where necessary and calculating the new solution is the same process as using the solution with weights $\lfloor \frac{1}{2}\alpha \rfloor$ to calculate the solution with weights α and will be accomplished in $T(m, n)$ time. The algorithm halts after $\lfloor \log_2 W \rfloor + 1$ iterations. This second point of view is more convenient for the algorithm to be presented.

$T(m, n)$ is the total work done in $O(\sqrt{m})$ *inner iterations* of our algorithm. The remaining factors of $T(m, n)$, other than $\sqrt{m}$, are dominated by the time required to calculate Newton directions, which involves solving a linear system. A detailed analysis of all such work could be carried out precisely as in [12] to show that $T(m, n) = O(m^{1.5}n + mn^2)$, or $O(mn + \sqrt{m}n^2)$ work within each inner iteration.

We will need one modification of the scaling technique. In finding weighted analytic centers, we can not allow any weights to be zero if we want to ensure that the barrier function has a unique minimum. In effect, a zero weight could fool the barrier function into seeing an unbounded polytope. So, we will always set every weight to a value of at least 1. This means that if the first $k-1$ significant bits of a weight are all 0, then we will set that weight to 1 and when we double this artificial 1 at the beginning of the kth iteration, we must *decrease* that weight by 1 during the iteration. Thus, increases of some weights and decreases of others may occur during the kth iteration. Of course, once we are far enough along to encounter non-zero bits in all weights, no further decreases will be necessary. We will show that the increases and decreases can be done simultaneously without hampering the efficiency of the algorithm.

2. The algorithm

The algorithm to be presented will work in outer and inner iterations. The outer iterations consist merely of doubling the weights on all planes, adding $-1, 0$, or 1 to the resulting weights as needed, calculating the new weighted analytic center, and checking to see whether all weights have reached their final values. The inner iterations are the workhorse of the algorithm. During the inner iterations, the weights that need adjustment change gradually while we keep track of the ever-changing weighted analytic center. The key is to change the weights as rapidly as possible while maintaining certain invariants that ensure Newton steps will be effective in maintaining the changing weighted analytic centers.

In our notation, we use subscripts to denote outer iterations and superscripts to denote inner iterations. Thus ω_k^j represents the weighted analytic center at the jth inner iteration of the kth outer iteration. The algorithm maintains an approximation x_k^j to ω_k^j. Rather than measuring closeness by a Euclidean distance, we will measure it using the weighted logarithmic barrier function F, i.e., x_k^j will satisfy

$$F(x_k^j) - F(\omega_k^j) < 0.00125. \tag{1}$$

We will show that this notion of closeness to ω_k^j will be sufficient. At the end of the inner iterations, we will denote the point obtained merely as x_k, which serves as an approximation to ω_k. The inner iterations begin with x_k^0, which is obtained from x_{k-1} at the beginning of an outer iteration, and end with the point $x_k = x_k^J$, where J denotes the last inner iteration.

Here is the algorithm.

Initialization: All weights begin at 1. Also set $k=1$.

During the kth outer iteration:

Step 1. Double all weights. For all i, we have

$$w_{i_k}^0 = \begin{cases} 10 & \text{if } b_{i_0} b_{i_1} \cdots b_{i_{k-1}} = 0, \\ b_{i_0} b_{i_1} \cdots b_{i_{k-1}} 0 & \text{otherwise.} \end{cases}$$

Step 2. Take O(1) Newton steps to obtain from x_{k-1} a point x_k^0 such that $F(x_k^0) - F(\omega_k^0) < 0.00125$ (note that $\omega_k^0 = \omega_{k-1}$).

Step 3. Add 1's and -1's to some weights.

Define the ith *goal weight* G_{i_k} by

$$G_{i_k} = \begin{cases} 1 & \text{if } b_{i_0} b_{i_1} \cdots b_{i_{k-1}} = 0, \\ b_{i_0} b_{i_1} \cdots b_{i_k} & \text{if } b_{i_0} b_{i_1} \cdots b_{i_{k-1}} \neq 0. \end{cases}$$

(i) Set $j=1$.

(ii) Partition the index set $I = \{1, \ldots, m\}$ into sets I_k^1, I_k^2, I_k^3, where

$$I_k^1 = \{i \colon b_{i_0} b_{i_1} \cdots b_{i_{k-1}} = 0\},$$

$$I_k^2 = \{i \colon b_{i_0} b_{i_1} \cdots b_{i_{k-1}} \neq 0 \text{ and } b_{i_k} = 0\},$$

$$I_k^3 = \{i \colon b_{i_0} b_{i_1} \cdots b_{i_{k-1}} \neq 0 \text{ and } b_{i_k} = 1\}.$$

(iii) Define γ_i, $1 \leq i \leq m$, by

$$\gamma_i = \begin{cases} 1 - \rho/\sqrt{m w_i}, & i \in I_k^1, \\ 1, & i \in I_k^2, \\ 1 + \rho/\sqrt{m w_i}, & i \in I_k^3. \end{cases}$$

For $i = 1, \ldots, m$, set $w_{i_k}^j = \gamma_i w_{i_k}^{j-1}$. If multiplying by γ_i causes over-shooting of the ith goal weight, then just set $w_i^j = G_{i_k}$. ρ is a constant to be specified later.

(iv) Take O(1) Newton steps to obtain point x_k^j such that $F(x_k^j) - F(\omega_k^j) < 0.00125$.

(v) If all goal weights have been met, set $x_k = x_k^j$ and stop inner iterations. Otherwise set $j = j+1$ and return to (iii).

Step 4. If $G_{i_k} = w_i$ for all i, then halt. Otherwise set $k = k+1$ and return to Step 1.

Note that the inner iterations are executed in Steps 3(iii) through 3(v). Furthermore, since each weight w_i is at least 1 and ρ is a constant, there can be at most $O(\sqrt{m})$ inner iterations. It is also clear from previous discussion of the method that there are $\lfloor \log_2 W \rfloor + 1$ outer iterations, where W is the largest of the weights on the planes. The algorithm starts from the *unweighted* analytic center in which all w_i's are equal to 1. This starting point may be found as described in [9] or [12]. That

O(1) Newton steps suffice in Steps 2 and 3(iv) will be proved in Section 5. So the algorithm executes a total of $O(\sqrt{m}\log_2 W)$ steps of Newton's method, and as discussed earlier the total number of arithmetic operations performed is $O((m^{1.5}n+mn^2)\log W)$.

3. The functions F and Ψ

Crucial to the development of the proofs will be a way to approximate the values $F(x)-F(\omega)$ from local information at x. It will also be convenient to use a generalized notion of nearness to x, or a ball about x. The next several lemmas will develop some useful relationships.

We should note here that the function F depends both on x and the weights $(w_1,\ldots,w_m)$. That is, each time a weight is changed, we have a different logarithmic barrier function. However, since great care will be taken to specify the weights at all stages of the algorithm, and since within each inner iteration the weights are held constant, we will refer to each member of this large family of barrier functions as $F(x)$. The reader should understand that the precise meaning of $F(x)$ depends upon the current weight settings. Furthermore, the results of this section are results on the function F that are independent of the weight settings (except where specified) and so no subscripts or superscripts on the arguments will be needed.

First, two definitions.

Definition.

$$\Psi(z):=\nabla F(z)^{\mathrm{T}}(\nabla^2F(z))^{-1}\nabla F(z).$$

Definition.

$$\Sigma(z,r):=\left\{x:\ \left|\frac{a_i^{\mathrm{T}}(x-z)}{a_i^{\mathrm{T}}z-b_i}\right|\leqslant r \text{ for all } i=1,\ldots,m\right\}.$$

The quantity $\Psi(z)$ has many useful properties that have been pointed out by several authors, including [3], [7] and [13]. Nesterov and Nemirovsky [8] called $\Psi(z)$ "Newton's decrement" of $F(z)$ at z. The reason for their terminology is that $\Psi(z)$ represents the decrease obtained in the quadratic approximation to F by taking a Newton step. It is also clear that $\Psi(z)$ is the square of the inverse Hessian norm of the gradient of F. $\Sigma(z,r)$ is a generalization of a ball about z. Its shape depends both on the planes that make up the polytope and on the distance of z from each facet of the polytope.

The following lemma allows us to relate the quadratic form determined by the Hessian evaluated at any point in $\Sigma(z,r)$ to the quadratic form in which the Hessian is evaluated at z. Creating a region in which the Hessian doesn't change too much is a useful idea in proving convergence of barrier minimization algorithms. Similar lemmas are proved in [3], [7] and [13].

Lemma 1. *If $x \in \Sigma(z, r)$ where $r < 1$, then for all $\xi \in \mathbb{R}^n$,*

$$\frac{\xi^T \nabla^2 F(z) \xi}{(1+r)^2} \leq \xi^T \nabla^2 F(x) \xi \leq \frac{\xi^T \nabla^2 F(z) \xi}{(1-r)^2}.$$

Proof. First note that

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

So,

$$\xi^T \nabla^2 F(x) \xi = \sum_{i=1}^{m} \frac{w_i (a_i^T \xi)^2}{(a_i^T x - b_i)^2} = \sum_{i=1}^{m} \frac{w_i (a_i^T \xi)^2}{(a_i^T z - b_i)^2} \cdot \frac{(a_i^T z - b_i)^2}{(a_i^T x - b_i)^2}.$$

Since $x \in \Sigma(z, r)$, we can say $1 - r \leq |(a_i^T x - b_i)/(a_i^T z - b_i)| \leq 1 + r$ for all i. As a result, we have

$$\frac{1}{(1+r)^2}\left[\sum_{i=1}^{m} \frac{w_i (a_i^T \xi)^2}{(a_i^T z - b_i)^2}\right] \leq \sum_{i=1}^{m} \frac{w_i (a_i^T \xi)^2}{(a_i^T x - b_i)^2} \leq \frac{1}{(1-r)^2}\left[\sum_{i=1}^{m} \frac{w_i (a_i^T \xi)^2}{(a_i^T z - b_i)^2}\right]. \qquad \square$$

The next lemma will convert Lemma 1 into a result on the quadratic form of the inverse of $\nabla^2 F(x)$. For a positive definite matrix A, define $E(A, x, r) := \{y: (y-x)^T A (y-x) \leq r^2\}$. Using the Karush-Kuhn-Tucker conditions, or just considering the properties of the elliptic norm, we can say

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

See [6] or [13] for details. Lemmas similar to the following appear in several standard texts on optimization, as well as in [11] and [13].

Lemma 2. *Suppose A and B are positive definite $n \times n$ matrices such that $\xi^T A \xi \geq \theta \xi^T B \xi$ for some $\theta > 0$ and for all ξ in $\mathbb{R}^n$. Then $\xi^T A^{-1} \xi \leq (1/\theta) \xi^T B^{-1} \xi$ for all ξ in $\mathbb{R}^n$.*

Proof. If $\xi^T A \xi \leq 1$, then $\theta \xi^T B \xi \leq 1$ by hypothesis. Thus $E(A, 0, 1) \subset E(B, 0, 1/\sqrt{\theta})$. Thus, for any $\xi \in \mathbb{R}^n$,

$$\max_{y \in E(B,0,1/\sqrt{\theta})} \xi^T y \geq \max_{z \in E(A,0,1)} \xi^T z$$

or, equivalently,

$$(1/\sqrt{\theta})\sqrt{\xi^T B^{-1} \xi} \geq \sqrt{\xi^T A^{-1} \xi}.$$

So $\xi^T A^{-1} \xi \leq (1/\theta) \xi^T B^{-1} \xi$. $\square$

Corollary. *If $x \in \Sigma(z, r)$, then for all $\xi \in \mathbb{R}^n$,*

$$(1+r)^2 \xi^T (\nabla^2 F(z))^{-1} \xi \geq \xi^T (\nabla^2 F(x))^{-1} \xi \geq (1-r)^2 \xi^T (\nabla^2 F(z))^{-1} \xi.$$

Proof. Follows from Lemma 1 and Lemma 2. □

In several of the proofs to follow, we will need to express function values in terms of integrals. We will use the implicit function theorem applied to the function $\Phi:\mathbb{R}^{n+1}\to\mathbb{R}^n$ given by $\Phi(x,t)=\nabla F(x)-tw$ where w is some fixed vector in $\mathbb{R}^n$. Setting $\Phi(x,t)\equiv 0$ implicitly defines x as a function of t, and differentiating both sides with respect to t gives

$$\nabla^2 F(x)\dot{x}(t)-w=0$$

or

$$\dot{x}(t)=(\nabla^2 F(x))^{-1}w. \tag{3}$$

In most of the applications that follow, we will set $w=\nabla F(z)$, for some fixed z. It is not difficult to prove that x is in fact an analytic function of t, the importance of which is that we will have independence of path in the integrals we shall consider. Some regularity conditions on F are required in order to use the implicit function theorem. A detailed verification of the required properties along with a proof of the analyticity of $x(t)$ can be found in [13].

The next two lemmas and following theorem will begin to give us information on the behavior of $\Psi(x)$ inside (and outside) the set $\Sigma(z,r)$.

Lemma 3. *For any x in the polytope P,*

$$\frac{a_i^{\mathrm{T}}(\nabla^2(F(x))^{-1}a_i}{(a_i^{\mathrm{T}}x-b_i)^2}\leqslant 1$$

for all i.

Proof. First we show that $E(\nabla^2 F(x),x,1)\subset P$. By definition,

$$E(\nabla^2 F(x),x,1)=\left\{y:\sum_{i=1}^{m}\frac{w_i(a_i^{\mathrm{T}}(y-x))^2}{(a_i^{\mathrm{T}}x-b_i)^2}\leqslant 1\right\}. \tag{4}$$

In our application of the scaling algorithm, we have $w_1,\ldots,w_m\geqslant 1$ throughout. Therefore,

$$(a_i^{\mathrm{T}}(y-x))^2\leqslant(a_i^{\mathrm{T}}x-b_i)^2\quad\text{for all } i.$$

As a result, from (4),

$$|a_i^{\mathrm{T}}(y-x)|\leqslant a_i^{\mathrm{T}}x-b_i\quad\text{for all } i. \tag{5}$$

If $y\notin P$, then there exists some index j such that $a_j^{\mathrm{T}}y-b_j<0$, hence $a_j^{\mathrm{T}}(y-x)<-a_j^{\mathrm{T}}x+b_j=-(a_j^{\mathrm{T}}x-b_j)$, which contradicts (5). Thus $E(\nabla^2 F(x),x,1)\subset P$, and this along with (2) implies

$$a_i^{\mathrm{T}}(\nabla^2 F(x))^{-1}a_i=\max_{y\in E(\nabla^2 F(x),x,1)}[a_i^{\mathrm{T}}(y-x)]^2\leqslant[a_i^{\mathrm{T}}x-b_i]^2.$$

Therefore,

$$\frac{a_i^{\mathrm{T}}(\nabla^2 F(x))^{-1}a_i}{(a_i^{\mathrm{T}}x-b_i)^2}\leqslant 1. \qquad \square$$

Lemma 4. *Let $r<1$. Let w be a fixed vector in $\mathbb{R}^n$. Consider the trajectory implicitly defined by $\nabla F(x)=tw$ and let $\hat{x}$ be such that $\nabla F(\hat{x})=\hat{t}w$ for some scalar $\hat{t}$. Let $x=x(t)$ be a point on the trajectory such that $x\in\Sigma(\hat{x},r)$. Then for $1\leqslant i\leqslant m$,*

$$\left|\frac{a_i^{\mathrm{T}}\dot{x}(t)}{a_i^{\mathrm{T}}x-b_i}\right|\leqslant(1+r)\sqrt{w^{\mathrm{T}}(\nabla^2F(\hat{x}))^{-1}w}.$$

Proof. Since $\dot{x}(t)=(\nabla^2F(x))^{-1}w$, we have

$$\left(\frac{a_i^{\mathrm{T}}\dot{x}(t)}{a_i^{\mathrm{T}}x-b_i}\right)^2=\left(\frac{a_i^{\mathrm{T}}(\nabla^2F(x))^{-1}w}{a_i^{\mathrm{T}}x-b_i}\right)^2.$$

Now considering

$$u=\frac{(\nabla^2F(x))^{-1/2}a_i}{\sqrt{a_i^{\mathrm{T}}x-b_i}}\quad\text{and}\quad v=\frac{(\nabla^2F(x))^{-1/2}w}{\sqrt{a_i^{\mathrm{T}}x-b_i}}$$

and using $(u^{\mathrm{T}}v)^2\leqslant\|u\|_2^2\|v\|_2^2$, we get

$$\begin{aligned}\left(\frac{a_i^{\mathrm{T}}\dot{x}(t)}{a_i^{\mathrm{T}}x-b_i}\right)^2&\leqslant\frac{a_i^{\mathrm{T}}(\nabla^2F(x))^{-1}a_i}{a_i^{\mathrm{T}}x-b_i}\cdot\frac{w^{\mathrm{T}}(\nabla^2F(x))^{-1}w}{a_i^{\mathrm{T}}x-b_i}\\&\leqslant\frac{a_i^{\mathrm{T}}(\nabla^2F(x))^{-1}a_i}{(a_i^{\mathrm{T}}x-b_i)^2}\cdot(1+r)^2w^{\mathrm{T}}(\nabla^2F(\hat{x}))^{-1}w.\end{aligned}$$

The last inequality follows from the hypothesis $x\in\Sigma(\hat{x},r)$. Now, by Lemma 3, we know the first term on the right is less than or equal 1. The result follows. □

Theorem 1. *Let $r<1$ and let $w\in\mathbb{R}^n$ be fixed. Let $\hat{x}$ be a point in $\mathbb{R}^n$ such that $\nabla F(\hat{x})=\hat{t}w$ for some scalar $\hat{t}$. Let $x(\bar{t})$ be a point on the trajectory defined by $\nabla F(x)=tw$ such that $x(\bar{t})$ is not contained in the interior of $\Sigma(\hat{x},r)$. Then*

$$|\bar{t}-\hat{t}|\geqslant\frac{r-\frac{1}{2}r^2}{(1+r)\sqrt{w^{\mathrm{T}}(\nabla^2F(\hat{x}))^{-1}w}}.$$

Proof. Define $\tilde{x}=x(\tilde{t})$ to be the first point on the trajectory in moving from $\hat{x}$ to $\bar{x}=x(\bar{t})$ that lies on the boundary of $\Sigma(\hat{x},r)$ (see Figure 1). Then there is an index k such that $(a_k^{\mathrm{T}}\tilde{x}-b_k)/(a_k^{\mathrm{T}}\hat{x}-b_k)=1+r$ or $1-r$. In either case, since $r<1$, we have

$$\left|\ln\left(\frac{a_k^{\mathrm{T}}\tilde{x}-b_k}{a_k^{\mathrm{T}}\hat{x}-b_k}\right)\right|\geqslant r-\tfrac{1}{2}r^2.$$

Now, we also have

$$\left|\ln\left(\frac{a_k^{\mathrm{T}}\tilde{x}-b_k}{a_k^{\mathrm{T}}\hat{x}-b_k}\right)\right|=\left|\int_{\hat{t}}^{\tilde{t}}\frac{a_k^{\mathrm{T}}\dot{x}(t)}{a_k^{\mathrm{T}}x(t)-b_k}\,\mathrm{d}t\right|.$$

By Lemma 4,

$$\begin{aligned}\left|\int_{\hat{t}}^{\tilde{t}}\frac{a_k^{\mathrm{T}}\dot{x}(t)}{a_k^{\mathrm{T}}x(t)-b_k}\,\mathrm{d}t\right|&\leqslant\int_{\hat{t}}^{\tilde{t}}\left|\frac{a_k^{\mathrm{T}}\dot{x}(t)}{a_k^{\mathrm{T}}x(t)-b_k}\right|\mathrm{d}t\\&\leqslant|\tilde{t}-\hat{t}|(1+r)\sqrt{w^{\mathrm{T}}(\nabla^2F(\hat{x}))^{-1}w}.\end{aligned}$$

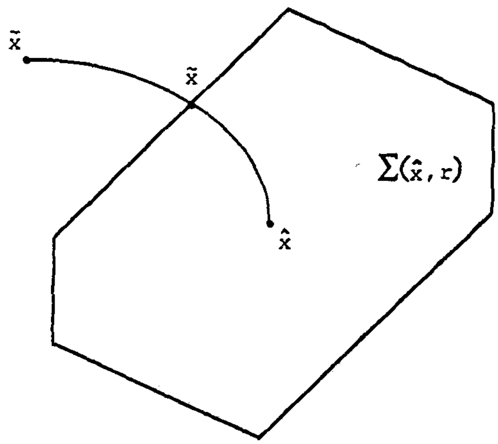

Fig. 1.

It follows, then, that

$$|\bar{t}-\hat{t}| \geq |\tilde{t}-\hat{t}| \geq \frac{r-\frac{1}{2}r^2}{(1+r)\sqrt{w^{\mathrm{T}}(\nabla^2 F(\hat{x}))^{-1}w}}. \quad \square$$

Lemma 5. *Consider the trajectory $x(t)$ implicitly defined by $\nabla F(x) = t\nabla F(z)$. If $\mu \leq 0.16$ and $\Psi(z) \leq 2\mu^2$, then $x(t) \in \Sigma(z, \sqrt{\mu})$ for all t in $[0, 1]$. In particular, $\omega \in \Sigma(z, \sqrt{\mu})$.*

Proof. Suppose not. That is, suppose there exists $\bar{t} \in [0, 1]$ such that $x(\bar{t}) \notin \Sigma(z, \sqrt{\mu})$. Then by Theorem 1, with $w = \nabla F(z)$,

$$1 \geq |\bar{t}-1| \geq \frac{\sqrt{\mu}-\frac{1}{2}\mu}{(1+\sqrt{\mu})\sqrt{\Psi(z)}}.$$

Thus

$$\Psi(z) \geq \left(\frac{\sqrt{\mu}-\frac{1}{2}\mu}{1+\sqrt{\mu}}\right)^2 > 2\mu^2 \quad (\text{since } \mu \leq 0.16).$$

Contradiction. Therefore $x(t) \in \Sigma(z, \sqrt{\mu})$ for all $t \in [0, 1]$, and so $\omega = x(0) \in \Sigma(z, \sqrt{\mu})$. $\square$

The next theorem is fundamental. Given only information about $\Psi(z)$, the theorem will allow us to draw conclusions about the distance of a point z from the weighted analytic center ω, measured in terms of values of F.

Theorem 2. *If $\mu \leq 0.16$ and $\Psi(z) \leq 2\mu^2$, then*

$$\tfrac{1}{2}(1-\sqrt{\mu})^2\Psi(z) \leq F(z)-F(\omega) \leq \tfrac{1}{2}(1+\sqrt{\mu})^2\Psi(z).$$

Proof. Considering the trajectory implicitly defined by $\nabla F(x) = t\nabla F(z)$, we get

$$F(z) - F(\omega) = \int_{x=\omega}^{x=z} \nabla F(x) \cdot \mathrm{d}x = \int_{t=0}^{t=1} \nabla F(x(t)) \cdot \dot{x}(t)\, \mathrm{d}t$$

$$= \int_0^1 t\nabla F(z)^{\mathrm{T}}(\nabla^2 F(x(t)))^{-1}\nabla F(z)\, \mathrm{d}t \quad \text{(see equation (3))}.$$

By Lemma 5, we know the entire trajectory over which we integrate lies within $\Sigma(z, \sqrt{\mu})$. Thus, by the corollary to Lemma 2, we can say

$$\int_0^1 t\nabla F(z)^{\mathrm{T}}(\nabla^2 F(x(t)))^{-1}\nabla F(z)\, \mathrm{d}t$$

$$\leqslant \int_0^1 t(1+\sqrt{\mu})^2 \nabla F(z)^{\mathrm{T}}(\nabla^2 F(z))^{-1}\nabla F(z)\, \mathrm{d}t$$

$$= \tfrac{1}{2}(1+\sqrt{\mu})^2 \Psi(z).$$

The lower bound follows in a completely similar fashion. □

Lemma 5 and Theorem 2 give us a bound on $F(z) - F(\omega)$ from a known bound on $\Psi(z)$. The next lemma and theorem turn this around and give us a bound on $\Psi(z)$ from a known bound on $F(z) - F(\omega)$. It is interesting to note that our knowledge of a bound on $F(z) - F(\omega)$ derives from information about $\Psi(z)$. So in a sense, this result seems circular. However, the proof of convergence of Newton's method will be expressed in terms of the quantity $F(x_k^j) - F(\omega_k^j)$, as is typically the case in convergence proofs for iterative methods. So, to be able to prove that $\Psi(z)$ also decreases, we will need a bound on $\Psi(z)$ when $F(x_k^j) - F(\omega_k^j)$ is made small. A direct proof of acceptable decrease in $\Psi(z)$ due to Newton steps would be more difficult.

Lemma 6. *Consider the trajectory implicitly defined by* $\nabla F(x) = t\nabla F(z)$ *where* z *is fixed. If* $\nu \leqslant 0.008$ *and* $F(z) - F(\omega) \leqslant \nu$, *then* $x(t) \in \Sigma(z, 5\sqrt{\nu})$ *for all* $t \in [0, 1]$. *In particular, then,* $x(0) = \omega \in \Sigma(z, 5\sqrt{\nu})$.

Proof. Suppose not. Let $x(\bar{t})$ be the first point on the trajectory from z to ω that lies on the boundary of $\Sigma(z, 5\sqrt{\nu})$. Then by Theorem 1,

$$1 - \bar{t} \geqslant \frac{5\sqrt{\nu} - \frac{25}{2}\nu}{(1+5\sqrt{\nu})\sqrt{\Psi(z)}}.$$

Thus,

$$(1-\bar{t})^2 \Psi(z) \geqslant \left(\frac{5\sqrt{\nu} - \frac{25}{2}\nu}{1+5\sqrt{\nu}}\right)^2. \qquad (6)$$

So now,

$$\begin{aligned}
F(z)-F(\omega) &= F(x(1))-F(x(0)) \\
&\geqslant F(x(1))-F(x(\bar{t})) \quad \text{(since } \omega \text{ is the minimizer)} \\
&= \int_{\bar{t}}^{1} t\nabla F(z)^{\mathrm{T}}(\nabla^2 F(x(t)))^{-1}\nabla F(z)\,\mathrm{d}t \\
&\geqslant (1-5\sqrt{\nu})^2\Psi(z)\int_{\bar{t}}^{1} t\,\mathrm{d}t \quad \text{(by the corollary to Lemma 2)} \\
&= \tfrac{1}{2}(1-\bar{t}^2)(1-5\sqrt{\nu})^2\Psi(z) \\
&\geqslant \tfrac{1}{2}(1-\bar{t})^2(1-5\sqrt{\nu})^2\Psi(z) \\
&\geqslant \tfrac{1}{2}(1-5\sqrt{\nu})^2\left[\frac{5\sqrt{\nu}-\frac{25}{2}\nu}{1+5\sqrt{\nu}}\right]^2 \quad \text{(by equation (6))} \\
&> \nu \quad \text{(since } \nu\leqslant 0.008).
\end{aligned}$$

Contradiction. Thus $x(t)\in\Sigma(z, 5\sqrt{\nu})$ for all $t\in[0, 1]$. □

Now the promised analog to Theorem 2.

Theorem 3. *If $\nu\leqslant 0.008$ and $F(z)-F(\omega)\leqslant\nu$, then*

$$\tfrac{1}{2}(1-5\sqrt{\nu})^2\Psi(z)\leqslant F(z)-F(\omega)\leqslant\tfrac{1}{2}(1+5\sqrt{\nu})^2\Psi(z).$$

Proof. As in the proof of Theorem 2,

$$F(z)-F(\omega)=\int_0^1 t\nabla F(z)^{\mathrm{T}}(\nabla^2 F(x(t)))^{-1}\nabla F(z)\,\mathrm{d}t.$$

By Lemma 6, the entire trajectory over which the integral runs lies within $\Sigma(z, 5\sqrt{\nu})$. Both inequalities follow by the corollary to Lemma 2. □

4. Taking Newton steps

In this section, we begin to prove the convergence of the Newton iterations in Steps 2 and 3(iv) of the algorithm. The relationships between F and Ψ stated in Theorem 2 and Theorem 3 are used extensively. We develop some rather general bounds on the improvement we can expect from taking Newton steps. In the next section, we will consider the most advantageous numerical parameters for the algorithm, and the general bounds will become specific.

We will consider the behavior of F along the Newton direction $-\eta$, where $\eta:=(\nabla^2 F(z))^{-1}\nabla F(z)$, beginning at a point z. This will, of course, give a function from $\mathbb{R}$ to $\mathbb{R}$. We will get a guaranteed decrease in this function by upper bounding its first derivative by a function guaranteed to be "sufficiently negative". By this we mean a function that will be sufficiently negative over enough of the interval of integration to yield a negative value for the integral. The upper bound on the first derivative will be expressed in terms of $\Psi(z)$, the fraction λ of a full Newton step we choose to take, and the quantity δ that bounds $F(z)-F(\omega)$. We will upper bound the first derivative in terms of these quantities quite simply by upper bounding the second derivative in terms of these quantities.

The following lemma will prove useful in bounding the second derivative.

Lemma 7. *Let $y(t)=z-t\eta$, for $t\in[0,\lambda]$ where η is defined above. If $F(z)-F(\omega)<\delta$, where $\delta\leqslant 0.008$, then $y(t)\in\Sigma(z,\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))$ for all $t\in[0,\lambda]$.*

Proof. By definition of y and η,

$$\begin{aligned}(y-z)^{\mathrm{T}}\nabla^2F(z)(y-z)&=t^2\eta^{\mathrm{T}}\nabla^2F(z)\eta\\&=t^2\nabla F(z)^{\mathrm{T}}(\nabla^2F(z))^{-1}\nabla F(z)\\&=t^2\Psi(z)\\&\leqslant\lambda^2\Psi(z)\quad\text{for all } t\in[0,\lambda].\end{aligned}$$

Now, since $\delta\leqslant 0.008$, Theorem 3 implies that

$$\Psi(z)\leqslant\frac{2(F(z)-F(\omega))}{(1-5\sqrt{\delta})^2}<\frac{2\delta}{(1-5\sqrt{\delta})^2}.$$

Hence $(y-z)^{\mathrm{T}}\nabla^2F(z)(y-z)\leqslant(2\lambda^2\delta)/(1-5\sqrt{\delta})^2$, and so for all $t\in[0,\lambda]$,

$$y(t)\in E^*:=E(\nabla^2F(z),z,\lambda\sqrt{2\delta}/(1-5\sqrt{\delta})).$$

Thus for all pairs (a_i,b_i), applying equation (2) and Lemma 3, we have

$$\max_{y\in E^*}(a_i^{\mathrm{T}}(y-z))^2=\frac{2\lambda^2\delta}{(1-5\sqrt{\delta})^2}a_i^{\mathrm{T}}(\nabla^2F(z))^{-1}a_i\leqslant\frac{2\lambda^2\delta}{(1-5\sqrt{\delta})^2}(a_i^{\mathrm{T}}z-b_i)^2.$$

Therefore,

$$y\in E^*\quad\text{implies}\quad\left|\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right|\leqslant\frac{\lambda\sqrt{2\delta}}{(1-5\sqrt{\delta})}.$$

So, $E(\nabla^2F(z),z,(\lambda\sqrt{2\delta})/(1-5\sqrt{\delta}))\subset\Sigma(z,\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))$. It follows that $y(t)\in\Sigma(z,\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))$ for all $t\in[0,\lambda]$. $\square$

We will now use this fact to get a bound on the second derivative, as promised.

Lemma 8. *Under the hypotheses of Lemma 7,*

$$\frac{\mathrm{d}^2F(y(t))}{\mathrm{d}t^2}\leqslant\frac{\Psi(z)}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}$$

for all $t\in[0,\lambda]$.

Proof. Since $F(y(t)) = F(z - t\eta)$, using the chain rule gives

$$\frac{\mathrm{d}F(y(t))}{\mathrm{d}t} = -\nabla F(y(t))^{\mathrm{T}}\eta$$

and

$$\frac{\mathrm{d}^2F(y(t))}{\mathrm{d}t^2} = \eta^{\mathrm{T}}\nabla^2 F(y(t))\eta.$$

According to Lemma 7, $y(t) \in \Sigma(z, \lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))$. Consequently, using Lemma 1, we have

$$\begin{aligned}\frac{\mathrm{d}^2F(y(t))}{\mathrm{d}t^2} &= \eta^{\mathrm{T}}\nabla^2 F(y(t))\eta \\ &\leqslant \frac{\eta^{\mathrm{T}}\nabla^2 F(z)\eta}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2} \\ &= \frac{\Psi(z)}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2} \quad \text{for all } t \in [0, \lambda]. \qquad \square\end{aligned}$$

Now we are ready to state the general result about how much decrease we can expect from moving in the Newton direction.

Theorem 4. *Assume $F(z) - F(\omega) < \delta$ where $\delta < 0.008$. Define $\eta = (\nabla^2 F(z))^{-1}\nabla F(z)$, and let $y = z - \lambda\eta$ where $\lambda \in [0, 1]$. Then*

$$F(z) - F(y) \geqslant \frac{1}{(1+5\sqrt{\delta})^2}\left[2\lambda - \frac{\lambda^2}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right](F(z) - F(\omega)).$$

Thus,

$$\begin{aligned}F(y) - F(\omega) \leqslant &\left(1 - \frac{1}{(1+5\sqrt{\delta})^2}\left[2\lambda - \frac{\lambda^2}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right]\right) \\ &\times (F(z) - F(\omega)).\end{aligned}$$

Proof. By Lemma 8, for $t \in [0, \lambda]$,

$$\begin{aligned}\frac{\mathrm{d}F(y(t))}{\mathrm{d}t} &= \frac{\mathrm{d}F(y(0))}{\mathrm{d}t} + \int_0^t \frac{\mathrm{d}^2F(y(s))}{\mathrm{d}s^2}\,\mathrm{d}s \\ &= -\Psi(z) + \int_0^t \frac{\mathrm{d}^2F(y(s))}{\mathrm{d}s^2}\,\mathrm{d}s \\ &\leqslant -\Psi(z) + \frac{\Psi(z)}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\int_0^t \mathrm{d}s \\ &= -\left(1 - \frac{t}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right)\Psi(z).\end{aligned}$$

The last expression bounds the first derivative of the restriction of F to the Newton direction by a function that is 'sufficiently negative' for t in $[0, \lambda]$.

To finish the proof, we integrate again:

$$\begin{aligned} F(z)-F(y) &= F(z)-F(z-\lambda\eta) \\ &= -\int_0^\lambda \frac{\mathrm{d}F(y(t))}{\mathrm{d}t}\,\mathrm{d}t \\ &\geqslant \Psi(z)\int_0^\lambda \left[1-\frac{t}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right]\mathrm{d}t \\ &= \Psi(z)\left[\lambda-\tfrac{1}{2}\lambda^2\frac{1}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right]. \end{aligned} \tag{7}$$

Bringing in the result of Theorem 3, which says that since $F(z)-F(\omega)<\delta$ and $\delta \leqslant 0.008$, we have

$$\Psi(z) \geqslant \frac{2}{(1+5\sqrt{\delta})^2}(F(z)-F(\omega)),$$

it follows, using equation (7), that

$$F(z)-F(y) \geqslant \frac{1}{(1+5\sqrt{\delta})^2}\left[2\lambda-\frac{\lambda^2}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right](F(z)-F(\omega)).$$

This proves the first statement in the conclusion of the theorem. The second statement follows simply by rearrangement. □

5. Selection of numerical parameters and a line search alternative for inner iterations

The first goal of this section is to transform Theorem 4 into a result with a guaranteed numerical factor of decrease rather than an abstract decrease involving a complicated formula. Achieving this goal clearly will require us to specify a value for λ and to show that the condition $\delta \leqslant 0.008$ can be maintained, as required by Theorem 4. Recall, however, that our knowledge of δ derives from our knowledge of $\Psi(z)$, so that ultimately the bound α that we place on $\Psi(z)$ is the crucial factor. We also must consider the effect of the parameter ρ that occurs in Step 3(iii) of the algorithm. We will see that both α and ρ must be selected to maintain a small enough bound on $\Psi(z)$ so that the theorems of Sections 3 and 4 apply. The second goal of this section is to prove that if the common technique of line search is implemented in place of the inner iterations, then we are still assured of polynomial convergence. This fact is of practical significance, although it will not help our complexity bounds. In applications, line search often accelerates convergence.

To get a concrete form of Theorem 4, we will show that if we choose $\alpha = 0.03$ and $\rho = 0.03$ then the conditions $\Psi(z) \leqslant 0.01$ and $F(z)-F(\omega) \leqslant 0.008$ are maintained throughout the algorithm. As a result the conditions of Theorem 4 are satisfied and $F(z)-F(\omega)$ decreases by a multiplicative factor in every Newton step executed in Steps 2 and 3(iv) of the algorithm. Hence the value of $F(z)-F(\omega)$ falls below the desired bound of 0.00125 in O(1) steps; in other words O(1) Newton steps suffice in Steps 2 and 3(iv) of the algorithm.

We begin by considering the effects of the steps of the algorithm that cause Newton steps to be necessary, i.e., Steps 1 and 3(iii). All our knowledge of accuracy is contained in our bound on $\Psi(z)$, so we must consider how the specified changes affect $\Psi(z)$. Note that just as with the function F, changing weights induces a new function Ψ. For the same reasons stated earlier for F, we simply denote each member of this large family as Ψ. At the end of this section, when we consider line search, we will specify which F and which Ψ by using subscripts corresponding to the particular inner iteration. Such additional notation is not particularly useful in the proofs concerning the original algorithm, and so will not be employed there.

Theorem 5. *If $\Psi(z)\leqslant 2\alpha^2$, then after all weights are doubled, $\Psi(z)\leqslant 4\alpha^2$.*

Proof. Trivial. Doubling all weights is tantamount to multiplying the function F by 2. But then

$$(2\nabla F(z))^{\mathrm{T}}(2\nabla^2 F(z))^{-1}(2\nabla F(z))=2\nabla F(z)^{\mathrm{T}}(\nabla^2 F(z))^{-1}\nabla F(z). \qquad \square$$

The type of adjustment in Step 3(iii) of the algorithm is not so trivial to handle. It will help us to develop some new notation for considering $\nabla F(x)$ and $\nabla^2 F(x)$. We define the $m\times m$ matrix D to be the diagonal matrix whose ith diagonal entry is $\sqrt{w_i}/(a_i^{\mathrm{T}}x-b_i)$ for $1\leqslant i\leqslant m$. We note first that

$$\nabla^2 F(x)=\sum_{i=1}^{m}\frac{w_i a_i a_i^{\mathrm{T}}}{(a_i^{\mathrm{T}}x-b_i)^2}=A^{\mathrm{T}}D^2A. \tag{8}$$

To avoid confusion with the transposes in (8), we note that the vector a_i is considered a column vector, although it is the ith *row* of the matrix A. Alternatively, we could define a_i to be the ith column of the matrix A^{T}.

Define the vector θ by

$$\theta^{\mathrm{T}}=[\sqrt{w_1},\ldots,\sqrt{w_m}].$$

Then

$$\nabla F(x)=-\sum_{i=1}^{m}\frac{w_i a_i}{a_i^{\mathrm{T}}x-b_i}=-A^{\mathrm{T}}D\theta. \tag{9}$$

It is interesting to observe that

$$\Psi(x)=\theta^{\mathrm{T}}(DA)[A^{\mathrm{T}}D^2A]^{-1}(DA)^{\mathrm{T}}\theta,$$

although we will not use this fact. We will use the notion of a projection matrix. The projection matrix arises in seeking the solution of the minimization problem

$\min_{x\in\mathbb{R}^n}\|b-Bx\|_2$. In general, the solution to this problem is $x=B^\dagger b$, where $B^\dagger$ is an $n\times m$ matrix called the pseudo-inverse of the $m\times n$ matrix B. In our application, we will set $B=DA$. When B has full column rank, as it will in the current case since the polytope is bounded, we have

$$B^\dagger=[B^{\mathrm{T}}B]^{-1}B^{\mathrm{T}}.$$

As a result, the closest approximation to b in the column space of the matrix B is $BB^\dagger b=B[B^{\mathrm{T}}B]^{-1}B^{\mathrm{T}}b$. Thus, the matrix $B[B^{\mathrm{T}}B]^{-1}B^{\mathrm{T}}$ is called the *projection matrix onto the column space of B.*

Lemma 9. *Let γ_i be defined as in Step* 3(iii) *of the algorithm. Then*

$$\left(\sum_{i=1}^m \frac{(1-\gamma_i)w_ia_i}{a_i^{\mathrm{T}}x-b_i}\right)^{\mathrm{T}}(\nabla^2F(x))^{-1}\left(\sum_{i=1}^m \frac{(1-\gamma_i)w_ia_i}{a_i^{\mathrm{T}}x-b_i}\right)\leqslant\rho^2$$

for any x in the polytope P.

Proof. Define the vector $\vartheta\in\mathbb{R}^m$ by

$$\vartheta^{\mathrm{T}}=[(1-\gamma_1)\sqrt{w_1},\ldots,(1-\gamma_m)\sqrt{w_m}].$$

Straightforward calculation yields

$$\begin{aligned}
&\left(\sum_{i=1}^m \frac{(1-\gamma_i)w_ia_i}{a_i^{\mathrm{T}}x-b_i}\right)^{\mathrm{T}}(\nabla^2F(x))^{-1}\left(\sum_{i=1}^m \frac{(1-\gamma_i)w_ia_i}{a_i^{\mathrm{T}}x-b_i}\right)\\
&=\vartheta^{\mathrm{T}}(DA)[\nabla^2F(x)]^{-1}(DA)^{\mathrm{T}}\vartheta\\
&=\vartheta^{\mathrm{T}}(DA)[A^{\mathrm{T}}D^2A]^{-1}(DA)^{\mathrm{T}}\vartheta\\
&=\vartheta^{\mathrm{T}}(DA)[(DA)^{\mathrm{T}}(DA)]^{-1}(DA)^{\mathrm{T}}\vartheta\\
&\leqslant\vartheta^{\mathrm{T}}\vartheta\\
&=\sum_{i=1}^m(1-\gamma_i)^2w_i\\
&=\sum_{i=1}^m\left(\frac{\rho}{\sqrt{mw_i}}\right)^2w_i\\
&=\sum_{i=1}^m\frac{\rho^2}{m}\\
&=\rho^2.
\end{aligned}$$

The inequality above follows because the inner product of a vector with its projection onto a subspace is certainly less than or equal to the inner product with itself. □

Theorem 6. *Suppose $\Psi(z)\leqslant 2\alpha^2$. If the weights $w_1,\ldots,w_m$ are all multiplied by the factors γ_i defined in Step* 3(iii) *of the algorithm, then the new Ψ, call it $\Psi_\gamma(z)$, satisfies*

$$\Psi_\gamma(z)\leqslant\frac{3}{1-\rho}[2\alpha^2+\rho^2].$$

Proof. Let $\tilde{\gamma} = \min_{1 \leq i \leq m} \gamma_i$. Temporarily let F_γ be the F that results from the multiplications by γ_i's. For all ξ in $\mathbb{R}^n$,

$$\xi^T \nabla^2 F_\gamma(z) \xi = \sum_{i=1}^m \frac{\gamma_i w_i (a_i^T \xi)^2}{(a_i^T z - b_i)^2} \geq \tilde{\gamma} \sum_{i=1}^m \frac{w_i (a_i^T \xi)^2}{(a_i^T z - b_i)^2} = \tilde{\gamma} \xi^T \nabla^2 F(z) \xi.$$

By Lemma 2,

$$\frac{1}{\tilde{\gamma}} \xi^T (\nabla^2 F(z))^{-1} \xi \geq \xi^T (\nabla^2 F_\gamma(z))^{-1} \xi \quad \text{for all } \xi \in \mathbb{R}^n.$$

Specifically, for all z in P,

$$\nabla F_\gamma(z)^T (\nabla^2 F_\gamma(z))^{-1} \nabla F_\gamma(z) \leq \frac{1}{\tilde{\gamma}} \nabla F_\gamma(z)^T (\nabla^2 F(z))^{-1} \nabla F_\gamma(z). \tag{10}$$

Now, define $e := \sum_{i=1}^m ((1-\gamma_i) w_i a_i)/(a_i^T z - b_i)$ and $h := \nabla F(z)$. A quick calculation reveals that $\nabla F_\gamma(z) = h - e$. Using this fact along with equation (10), we get

$$\begin{aligned}
\Psi_\gamma(z) &= \nabla F_\gamma(z)^T (\nabla^2 F_\gamma(z))^{-1} \nabla F_\gamma(z) \\
&\leq \frac{1}{\tilde{\gamma}} \nabla F_\gamma(z)^T (\nabla^2 F(z))^{-1} \nabla F_\gamma(z) \\
&= \frac{1}{\tilde{\gamma}} [h - e]^T (\nabla^2 F(z))^{-1} [h - e] \\
&= \frac{1}{\tilde{\gamma}} \{\Psi(z) - 2e^T (\nabla^2 F(z))^{-1} h + e^T (\nabla^2 F(z))^{-1} e\}.
\end{aligned} \tag{11}$$

By hypothesis, $\Psi(z) \leq 2\alpha^2$. We must bound the other two terms in the braces of equation (11). By Lemma 9, we know that $e^T (\nabla^2 F(z))^{-1} e \leq \rho^2$. To bound the remaining term, we use the fact that

$$|u^T v| \leq \|u\|_2 \|v\|_2 \leq \max\{\|u\|_2^2, \|v\|_2^2\}$$

with $u = (\nabla^2 F(z))^{-1/2} e$, and $v = (\nabla^2 F(z))^{-1/2} h$. It follows readily that

$$\begin{aligned}
|e^T (\nabla^2 F(z))^{-1} h| &\leq \max\{e^T (\nabla^2 F(z))^{-1} e, \Psi(z)\} \\
&\leq e^T (\nabla^2 F(z))^{-1} e + \Psi(z) \\
&\leq \rho^2 + 2\alpha^2.
\end{aligned} \tag{12}$$

Now, returning to equation (11), we can say

$$\begin{aligned}
&\frac{1}{\tilde{\gamma}} \{\Psi(z) - 2e^T (\nabla^2 F(z))^{-1} h + e^T (\nabla^2 F(z))^{-1} e\} \\
&\quad \leq \frac{1}{\tilde{\gamma}} \{2\alpha^2 + 2[\rho^2 + 2\alpha^2] + \rho^2\} \\
&\quad = \frac{6\alpha^2 + 3\rho^2}{\tilde{\gamma}} \\
&\quad \leq \frac{6\alpha^2 + 3\rho^2}{1 - \rho}. \qquad \square
\end{aligned}$$

Now return to the result of Theorem 4. When $\delta \leqslant 0.008$, $F(z)-F(\omega)<\delta$, and $y=z-\lambda\eta$ where $\eta=(\nabla^2 F(z))^{-1}\nabla F(z)$,

$$F(y)-F(\omega) \leqslant \left(1-\frac{1}{(1+5\sqrt{\delta})^2}\left[2\lambda-\frac{\lambda^2}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right]\right) \times (F(z)-F(\omega))$$

for all λ in $[0, 1]$. This result guarantees that Newton steps give a decrease in the function values. According to Theorem 2, we need $\Psi(z)<0.01$ (actually, about 0.00998 — we use 0.01 for convenience) in order to ensure that $F(z)-F(\omega)\leqslant 0.008$. So, we need to know that $\Psi(z)<0.01$ at *all* times. By Theorem 5 and Theorem 6, to ensure that $\Psi(z)<0.01$ at all times, we must have

$$\max\left\{4\alpha^2, \frac{6\alpha^2+3\rho^2}{1-\rho}\right\}<0.01$$

at all times. The second term is clearly the more restrictive. In choosing α and ρ, we have a trade-off between the two desirable qualities of having large factors of change in the Steps 3(iii), and maintaining less accuracy in x_k^j. We will split the difference and set $\alpha=0.03$, $\rho=0.03$. The following theorem summarizes these results.

Theorem 7. *If* $\alpha=0.03$, $\rho=0.03$, *and* $\Psi(z)\leqslant 2\alpha^2=0.0018$, *then after the weight changes in Step* 1 *or Step* 3(iii) *of the algorithm, the new* $\Psi(z)$ *is less than* 0.01. *Furthermore, at the start of Step* 2 *or Step* 3(iv), $F(z)-F(\omega)\leqslant 0.008$.

Proof. The bound on the new $\Psi(z)$ follows from Theorem 5, Theorem 6, and the above discussion. The bound on $F(z)-F(\omega)$ in turn follows from the bound on the new $\Psi(z)$ and Theorem 3. □

Having now ensured that the hypotheses of Theorem 4 hold in Steps 2 and 3(iv), we note that from empirical observation, for $\lambda\in[0, 1]$ and $\delta\in[0, 0.008]$, the function

$$g(\lambda, \delta) := \left(1-\frac{1}{(1+5\sqrt{\delta})^2}\left[2\lambda-\frac{\lambda^2}{(1-\lambda\sqrt{2\delta}/(1-5\sqrt{\delta}))^2}\right]\right)$$

is a convex function in λ. When δ is fixed, g is minimized at around $\lambda=0.75$. Assuming λ is set at 0.75, we find that $g(0.75, \delta)$ is a strictly increasing function of δ for $\delta\in[0, 0.008]$. We also see that $g(0.75, 0.008)<0.68$.

From the above discussion, we can now re-state Theorem 4 in the following more concrete form.

Theorem 8. *Assume* $F(z)-F(\omega)<\delta$ *where* $\delta\leqslant 0.008$. *Define* $\eta=(\nabla^2 F(z))^{-1}\nabla F(z)$ *and let* $y=z-0.75\eta$. *Then*

$$F(z)-F(y)\geqslant 0.32(F(z)-F(\omega)).$$

Thus,

$$F(y)-F(\omega)\leqslant 0.68(F(z)-F(\omega)).$$

Proof. Follows immediately from Theorem 4 and the above discussion. □

Observe that since we have set $\alpha = 0.03$, our approximation x_k^j to ω_k^j must satisfy

$$\Psi(x_k^j) \leqslant 2(0.03)^2 = 0.0018.$$

It follows, by Theorem 2, that in this case

$$F(x_k^j) - F(\omega) < 0.00125.$$

This result explains the occurrence of 0.00125 in equation (1). The following theorem justifies the statement that O(1) Newton steps suffice in Steps 2 and 3(iv) of the algorithm.

Theorem 9. *After doubling of the weights in Step 1 of the algorithm, or multiplication of the weights by the factors γ_i, at most ten Newton steps produce a point x_k^j satisfying $\Psi(x_k^j) \leqslant 2(0.03)^2$.*

Proof. Let us refer to the general iterate in moving from x_k^j to x_k^{j+1} (or from x_{k-1} to x_k^0 in the doubling case) simply as y_i, where $y_0 = x_k^0$ (resp., $y_0 = x_{k-1}$). We have shown in the discussion preceding Theorem 8 that we *always* maintain the condition $\Psi(x_k^j) < 0.01$. As a result, Theorem 2 implies

$$F(y_0) - F(\omega_k^{j+1}) = F(x_k^j) - F(\omega_k^{j+1}) < 0.008.$$

Theorem 8 thus implies that

$$F(y_i) - F(\omega_k^{j+1}) < 0.68(F(y_{i-1}) - F(\omega_k^{j+1})) \quad \text{for all } i \geqslant 1.$$

Since the difference in function values only decreases, it follows via Theorem 3, using the best available ν, i.e., $\nu = F(y_i) - F(\omega_k^{j+1})$, that

$$\begin{aligned}\Psi(y_i) &\leqslant \frac{2[F(y_i) - F(\omega_k^{j+1})]}{(1 - 5\sqrt{F(y_i) - F(\omega_k^{j+1})})^2} \\ &\leqslant \frac{2[F(y_i) - F(\omega_k^{j+1})]}{(1 - 5\sqrt{0.008})^2} \\ &\leqslant \frac{2}{(1 - 5\sqrt{0.008})^2}(0.68)^i[F(y_0) - F(\omega_k^{j+1})] \\ &< \frac{2}{(1 - 5\sqrt{0.008})^2}(0.68)^i(0.008) \\ &< (0.03)(0.68)^i.\end{aligned}$$

In $i = 10$ steps, we are thus guaranteed that $\Psi(y_i) < 2(0.03)^2$. □

For the remainder of this section, we take a different tack. Frequently in applications, one-dimensional minimization in the Newton direction is found to accelerate convergence. That is, instead of just taking Newton steps, we minimize the function in the Newton direction. We want to prove that line search can be used in the current algorithm with assured polynomial convergence.

Within some of the following proofs, it is important to identify the weighted barrier function in effect during a particular inner iteration. We introduce subscripts on the functions F and Ψ corresponding to the inner iteration number. Our plan is to substitute the line search process for the inner iterations of the algorithm in Section 2. Thus, in the kth outer iteration, we shall be adding Δ_i to the ith weight, where $\Delta_i \in \{-1, 0, 1\}$, and minimizing the weighted barrier function

$$F_J(x) := -\sum_{i=1}^{m} (w_{i_k}^0 + \Delta_i)\ln(a_i^T x - b_i).$$

Of course, the minimizer of F_J is ω_k, the same minimizer that we would be dealing with in the last inner iteration of the technique described in Section 2. F_J is just a convenient way of denoting the barrier function that results from the full unit weight changes in the inner iterations. To make the distinction obvious, we will denote the barrier function we have at the beginning of the inner iterations as F_0. That is,

$$F_0(x) := -\sum_{i=1}^{m} w_{i_k}^0 \ln(a_i^T x - b_i).$$

Convergence proofs for algorithms that minimize logarithmic barrier functions nearly always fall into the same general pattern. They prove that there is an ellipsoid about the minimizer such that if the current point lies within the ellipsoid, then Newton steps lead to linear convergence, while if the current point lies outside the ellipsoid, then Newton steps give a guaranteed constant decrease in the barrier. We will not here prove that this fact holds for logarithmic barrier functions, but will refer the reader to [7], [8] and [12]. Note that this same result applies to a weighted logarithmic barrier function since the integer weights may be thought of as multiple copies of a constraint plane in an unweighted logarithmic barrier function. From this fact about logarithmic barriers, it follows that to prove polynomial convergence of line search in place of the inner iterations, we just need to prove a polynomial bound on $F_J(x_k^0) - F_J(\omega_k)$, i.e., we want a polynomial bound on the distance from optimality of the logarithmic barrier function in effect at the end of the inner iterations as measured at the current point x_k^0 we have at the beginning of inner iterations.

A clear proof of the polynomial bound on $F_J(x_k^0) - F_J(\omega_k)$ requires a few further lemmas about the behavior of the weighted logarithmic barrier function. The first such result gives us a bound on the error in using the second degree Taylor approximation of F about z at points known to be in a small ellipsoid centered at z. Substantially similar results are found in [5] and [14]. We present a proof here for completeness.

Lemma 10. *Suppose F is the weighted logarithmic barrier function and y in $\mathbb{R}^n$ is a point satisfying*

$$(y-z)^T \nabla^2 F(z)(y-z) \leq r^2$$

for some $r<1$. *Then*

$$F(y)-F(z)=\nabla F(z)^{\mathrm{T}}(y-z)+\tfrac{1}{2}(y-z)^{\mathrm{T}}\nabla^2 F(z)(y-z)+\varepsilon$$

where

$$|\varepsilon|\leqslant\frac{r^3}{3(1-r)}.$$

Proof. The expression for $F(y)-F(z)$ is just the second degree Taylor approximation of F with error. We need only prove the claimed error bound. Since

$$(y-z)^{\mathrm{T}}\nabla^2 F(z)(y-z)=\sum_{i=1}^{m} w_i\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^2\leqslant r^2 \tag{13}$$

and $w_i\geqslant 1$ for all i, we know

$$\left|\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right|\leqslant r<1 \quad \text{for all } i.$$

This fact implies that $\ln((a_i^{\mathrm{T}}y-b_i)/(a_i^{\mathrm{T}}z-b_i))=\ln(1+(a_i^{\mathrm{T}}(y-z))/(a_i^{\mathrm{T}}z-b_i))$ has the power series representation

$$\sum_{j=1}^{\infty}\frac{(-1)^{j+1}}{j}\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^j.$$

Thus

$$\begin{aligned}F(y)-F(z)&=-\sum_{i=1}^{m} w_i\ln\left(\frac{a_i^{\mathrm{T}}y-b_i}{a_i^{\mathrm{T}}z-b_i}\right)\\&=-\sum_{i=1}^{m} w_i\sum_{j=1}^{\infty}\frac{(-1)^{j+1}}{j}\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^j.\end{aligned}$$

Or, splitting off the first two terms of the expansion,

$$\begin{aligned}F(y)-F(z)=&\nabla F(z)^{\mathrm{T}}(y-z)+\tfrac{1}{2}(y-z)^{\mathrm{T}}\nabla^2 F(z)(y-z)\\&-\sum_{i=1}^{m} w_i\sum_{j=3}^{\infty}\frac{(-1)^{j+1}}{j}\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^j.\end{aligned}$$

So it suffices now to bound the absolute value of

$$\varepsilon=-\sum_{i=1}^{m} w_i\sum_{j=3}^{\infty}\frac{(-1)^{j+1}}{j}\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^j.$$

From equation (13), it follows that for $j\geqslant 3$,

$$\begin{aligned}r^j&>\left[\sum_{i=1}^{m} w_i\left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right)^2\right]^{j/2}\\&\geqslant\sum_{i=1}^{m} w_i^{j/2}\left|\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right|^j \quad \text{(Jensen's inequality)}\\&\geqslant\sum_{i=1}^{m} w_i\left|\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}}z-b_i}\right|^j \quad \text{(recall } w_i\geqslant 1 \text{ for all } i).\end{aligned}$$

Thus

$$\left| -\sum_{i=1}^{m} w_i \sum_{j=3}^{\infty} \frac{(-1)^{j+1}}{j} \left(\frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}} z - b_i} \right)^j \right| \leq \sum_{j=3}^{\infty} \sum_{i=1}^{m} \frac{w_i}{j} \left| \frac{a_i^{\mathrm{T}}(y-z)}{a_i^{\mathrm{T}} z - b_i} \right|^j$$

$$\leq \sum_{j=3}^{\infty} \frac{r^j}{j}$$

$$\leq \frac{r^3}{3(1-r)}. \qquad \square$$

Lemma 11. *Suppose that for* $i = 1, \ldots, m$, $|x_i| \leq \theta < \frac{1}{2}$ *and that* $\sum_{i=1}^{m} |x_i| \leq \theta\sqrt{m}$. *Then*

$$\sum_{i=1}^{m} |\ln(1+x_i)| \leq 3\theta\sqrt{m}.$$

Proof. By a simpler version of the same power series expansion used in Lemma 10, we can show that for all i,

$$\ln(1+x_i) = x_i - \tfrac{1}{2}x_i^2 + \varepsilon_i$$

where $|\varepsilon_i| \leq |x_i|^3/(3(1-|x_i|)$. Thus,

$$\sum_{i=1}^{m} |\ln(1+x_i)| = \sum_{i=1}^{m} |x_i - \tfrac{1}{2}x_i^2 + \varepsilon_i|$$

$$\leq \sum_{i=1}^{m} |x_i| + \tfrac{1}{2} \sum_{i=1}^{m} x_i^2 + \tfrac{1}{3} \sum_{i=1}^{m} \frac{|x_i|^3}{1-|x_i|}$$

$$\leq \tfrac{3}{2} \sum_{i=1}^{m} |x_i| + \tfrac{1}{3} \sum_{i=1}^{m} \frac{|x_i|^3}{1-|x_i|} \quad (\text{since } x_i^2 < |x_i|)$$

$$\leq \tfrac{3}{2} \sum_{i=1}^{m} |x_i| + \tfrac{2}{3} \sum_{i=1}^{m} |x_i|^3 \quad (\text{since } |x_i| < \tfrac{1}{2})$$

$$< \tfrac{13}{6} \sum_{i=1}^{m} |x_i|$$

$$< 3\theta\sqrt{m}. \qquad \square$$

We also need to show that a small value of $\Psi(z)$ ensures that ω is in a certain ellipsoid about z. The next lemma can be viewed as a companion lemma to Lemma 5. In the theorem to follow, the fact that ω is known to be in an ellipsoid about z, rather than in a set of the form $\Sigma(z, r)$ as implied by Lemma 5, gives us a stronger result.

Lemma 12. *Define the ellipsoid* $E(\nabla^2 F(z), z, 0.35)$ *by* $E(\nabla^2 F(z), z, 0.35) := \{y \in \mathbb{R}^n \colon (y-z)^{\mathrm{T}} \nabla^2 F(z)(y-z) \leq (0.35)^2\}$ *and suppose* $\Psi(z) < 0.01$. *Then* $\omega \in E(\nabla^2 F(z), z, 0.35)$.

Proof. Suppose not. Then consider the line segment connecting z to ω and let x' be the point where the segment intersects $E(\nabla^2 F(z), z, 0.35)$. Since F is convex and is minimized at ω, we know

$$F(x') - F(z) < 0. \tag{14}$$

By Lemma 10 and the fact that x' is on the boundary of $E(\nabla^2 F(z), z, 0.35)$,

$$\begin{aligned} F(x') - F(z) &= \nabla F(z)^{\mathrm{T}}(x'-z) + \tfrac{1}{2}(x'-z)^{\mathrm{T}}\nabla^2 F(z)(x'-z) + \varepsilon \\ &\geq \nabla F(z)^{\mathrm{T}}(x'-z) + \tfrac{1}{2}(0.35)^2 - \frac{(0.35)^3}{3(0.65)} \\ &\geq -(0.35)\sqrt{\Psi(z)} + \tfrac{1}{2}(0.35)^2 - \frac{(0.35)^3}{3(0.65)} \quad \text{(see equation (2))} \\ &\geq -(0.35)\sqrt{0.01} + \tfrac{1}{2}(0.35)^2 - \frac{(0.35)^3}{3(0.65)} \\ &> 0, \end{aligned}$$

contradicting (14). Thus $\omega \in E(\nabla^2 F(z), z, 0.35)$. □

We can now prove the desired result.

Theorem 10. *Let x_k^0 be the approximation to ω_{k-1} $(=\omega_k^0)$ in effect at the beginning of the inner iterations of the algorithm, where ω_{k-1} is the minimizer of $F_0(x)$. Let ω_k $(=\omega_k^J)$ be the last minimizer considered in the inner iterations, i.e., ω_k is the minimizer of $F_J(x)$. Then*

$$F_J(x_k^0) - F_J(\omega_k) = \mathrm{O}(m).$$

Proof. Consider the sequence of analytic centers ω_k^j within the inner iterations indexed by $j = 1, 2, 3, \ldots$ (k should be considered held fixed throughout this proof). For convenience of notation, for $j = -1$ define $\omega_k^{-1} := x_k^0$. Also, for $j \geq 0$, define a sequence of vectors $\boldsymbol{\delta}_j = (\delta_{1j}, \ldots, \delta_{mj})$, where $\boldsymbol{\delta}_0 = (0, \ldots, 0)$ and, for $j \geq 1$, $\delta_{ij} = w_{i_k}^j - w_{i_k}$. That is, δ_{ij} is the increment or decrement put on the original weights brought in from the outer iteration during the jth inner iteration. Now,

$$\begin{aligned} F_J(x_k^0) - F_J(\omega_k) &= -\sum_{i=1}^{m} (w_{i_k} + \Delta_i) \ln\left(\frac{a_i^{\mathrm{T}} x_k^0 - b_i}{a_i^{\mathrm{T}} \omega_k - b_i}\right) \\ &= -\sum_{i=1}^{m} (w_{i_k} + \Delta_i) \sum_{j=0}^{J} \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i}\right) \\ &= -\sum_{j=0}^{J} \sum_{i=1}^{m} (w_{i_k} + \delta_{ij}) \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i}\right) \\ &\quad + \sum_{j=0}^{J} \sum_{i=1}^{m} (\delta_{ij} - \Delta_i) \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i}\right). \end{aligned} \tag{15}$$

With the notations F_j and Ψ_j corresponding to the weight settings $w_{i_k} + \delta_{ij}$, $i = 1, \ldots, m$, i.e., the specific definitions of F and Ψ in effect during the jth inner iteration, the first sum on the right side of (15) can be written

$$\sum_{j=0}^{J} F_j(\omega_k^{j-1}) - F_j(\omega_k^j).$$

But we have seen (see the discussion after Theorem 6) that the weight changes in the inner iterations are such that we maintain the condition $\Psi_j(\omega_k^{j-1}) < 0.01$, and thus it follows via Theorem 2 that $F_j(\omega_k^{j-1}) - F_j(\omega_k^j) \leqslant 0.008$. Continuing the argument from the point of equation (15), we can now say

$$\begin{aligned} F_J(x_k^0) - F_J(\omega_k) &\leqslant \sum_{j=0}^{J} (0.008) + \sum_{j=0}^{J} \sum_{i=1}^{m} |\delta_{ij} - \Delta_i| \left| \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right) \right| \\ &\leqslant \sum_{j=0}^{J} (0.008) + \sum_{j=0}^{J} \sum_{i=1}^{m} \left| \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right) \right|. \end{aligned}$$

The preceding inequality follows because $|\delta_{ij} - \Delta_i| \leqslant 1$. Since $\Psi_j(\omega_k^{j-1}) < 0.01$, Lemma 12 implies $\omega_k^j \in E(\nabla^2 F(\omega_k^{j-1}), \omega_k^{j-1}, 0.35)$. Hence,

$$\sum_{i=1}^{m} (w_{i_k} + \delta_{ij}) \left(\frac{a_i^{\mathrm{T}}(\omega_k^j - \omega_k^{j-1})}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right)^2 \leqslant (0.35)^2,$$

and since each weight is always at least 1,

$$\sum_{i=1}^{m} \left(\frac{a_i^{\mathrm{T}}(\omega_k^j - \omega_k^{j-1})}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right)^2 \leqslant (0.35)^2. \tag{16}$$

The relationship between the 2-norm and the 1-norm in $\mathbb{R}^m$ thus implies that

$$\sum_{i=1}^{m} \left| \frac{a_i^{\mathrm{T}}(\omega_k^j - \omega_k^{j-1})}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right| \leqslant (0.35)\sqrt{m}. \tag{17}$$

The inequalities in (16) and (17) show that we have met the hypotheses of Lemma 11, and we may conclude that

$$\sum_{i=1}^{m} \left| \ln\left(\frac{a_i^{\mathrm{T}} \omega_k^{j-1} - b_i}{a_i^{\mathrm{T}} \omega_k^j - b_i} \right) \right| \leqslant 3(0.35)\sqrt{m}.$$

So, we have

$$F_J(x_k^0) - F_J(\omega_k) \leqslant \sum_{j=0}^{J} (0.008) + \sum_{j=0}^{J} (1.05)\sqrt{m} = \sum_{j=0}^{J} (0.008 + 1.05\sqrt{m}).$$

Finally, since the number of inner iterations J is $\mathrm{O}(\sqrt{m})$, we have $F_J(x_k^0) - F_J(\omega_k) = \mathrm{O}(m)$. $\square$

The results through Theorem 9 in this section have codified the results of the previous section concerning the algorithm presented in Section 2. We have shown that by setting $\alpha = 0.03$ and $\rho = 0.03$ the weight changes don't change $\Psi(z)$ too much. We have also indicated that the restriction $\Psi(x_k^j) \leq 2(0.03)^2$ suffices to keep $\Psi(z) < 0.01$ at *all* times and thus keep $F(z) - F(\omega) < 0.008$ at *all* times. Then, for the estimates used, we have shown that moving $\lambda = 0.75$ times a full Newton step gives a guaranteed decrease in function value. As a result, we have guaranteed the algorithm remains "stable" and converges as claimed.

In the remainder of the section, following Theorem 9, we have shown that line search may be substituted for the inner iterations of our algorithm without giving up polynomial convergence. We are assured that the barrier function with full unit weight changes evaluated at the previous minimizer is only $O(m)$ from its value at the new minimizer.

6. Applications

The general procedure for finding a weighted analytic center has many applications both as a subroutine and as a complete algorithm. For each application, the weights need to be set properly to accomplish the desired goal.

One application alluded to earlier is linear programming. Suppose we wish to solve

$$\begin{aligned} \max \quad & c^{\mathrm{T}}x \\ \text{s.t.} \quad & Ax \geq b \end{aligned}$$

where A is $m \times n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, and we know some value β such that $\{x\colon Ax \geq b\} \subset \{x\colon c^{\mathrm{T}}x \geq \beta\}$ and $c^{\mathrm{T}}x^{\mathrm{opt}} - \beta = 2^{\mathrm{O}(L)}$, where

$$\begin{aligned} L = {} & \log_2(\text{largest absolute value of the determinant of any square submatrix of } A) \\ & + \log_2\left(\max_i c_i\right) + \log_2\left(\max_i b_i\right) + \log_2(m+n). \end{aligned}$$

As is customary, we assume the polytope $P = \{x\colon Ax \geq b\}$ is bounded and of full dimension. Then the weighted barrier function

$$F(x) = -\sum_{i=1}^{m} \ln(a_i^{\mathrm{T}}x - b_i) - \lambda \ln(c^{\mathrm{T}}x - \beta)$$

has a minimum value that becomes closer to the optimum facet of the polytope as λ grows larger. It can be shown that for $\lambda = 2^{\mathrm{O}(L)}$, the weighted analytic center is close enough to the optimal facet that we can isolate an optimum vertex of the polytope (see [12]). From this fact, it is clear that the weighted analytic center technique can solve such a linear programming problem in $O(\sqrt{m}\, L)$ iterations.

We can generalize this result a little. Suppose we want to maximize a product of linear functions

$$\prod_{i=1}^{k} (c_i^{\mathrm{T}} x - d_i)$$

over a polytope $P = \{x\colon Ax \geqslant b\}$, where each linear term in the product is non-negative over P. This objective function is a (rather specialized) polynomial. We can use the weighted logarithmic barrier function

$$F(x) = -\lambda \sum_{i=1}^{k} \ln(c_i^{\mathrm{T}} x - d_i) - \sum_{i=1}^{m} \ln(a_i^{\mathrm{T}} x - b_i)$$

on the polytope $Q = P \cap \{x\colon c_i^{\mathrm{T}} x - d_i \geqslant 0, 1 \leqslant i \leqslant k\}$. Although the product $\prod_{i=1}^{k} (c_i^{\mathrm{T}} x - d_i)$ is in general neither convex nor concave over P, it has a special enough form to give a strictly convex logarithmic barrier function. As λ increases, the weighted analytic center is pushed away from the hyperplanes $c_i^{\mathrm{T}} x - d_i = 0$, and toward an optimal solution. Less work has been done on this problem than on the linear programming problem, but it is to be expected that an exponentially large value of λ should again lead to a bound of $\mathrm{O}(\sqrt{m}\, L)$ on the number of iterations to reach a point sufficiently close to the optimum to allow isolation of an exact solution.

We mentioned in the introduction that general convex programming is a potential application of our technique. Vaidya's algorithm in [13] actually used a *log-determinant* barrier function to locate what was called a *volumetric center* of the polytope. A plane passing through the volumetric center divides the polytope into two pieces of approximately equal volumes (at least in an average sense). Thus, in throwing away part of the polytope that can not contain the optimum, it is possible to throw out about half at each iteration. This fact leads to a better asymptotic time complexity than would be true of the older ellipsoid method applied in the same way.

Although it arises as the minimizer of the log-determinant function, the volumetric center *is* a weighted analytic center. In fact, we can produce the weights resulting in any volumetric center. The difficulty is that these weights are functions of x; the weights are easily obtainable, but we don't know them until we have the center. If some method can be developed to approximate these weights efficiently, the technique of this paper should greatly facilitate the process of reaching the volumetric center.

7. Conclusions

We have demonstrated a new technique for locating the weighted analytic center of a polytope. The scaling technique has allowed us to show a bound of $\mathrm{O}(\sqrt{m} \log W)$ on the number of Newton steps, and a bound of $\mathrm{O}((m^{1.5} n + mn^2) \log W)$ on the growth of the required work, where m is the number of constraints, n is the dimension of the space, and W is the largest of the weights. This work consists of $\mathrm{O}(\sqrt{m} \log W)$

Newton steps with $O(mn+\sqrt{m}\,n^2)$ arithmetic operations on the average per step. Freund's technique in [2] required $O(mW)$ steps with the same amount of work per step as our method. A proper selection of weights can move the weighted analytic center to any point in the polytope, so this technique is very general. Several applications have been mentioned in the preceding section.

It is an open question how far this scaling technique can be extended. We have mentioned that potential applications exist for convex optimization with polyhedral constraint regions. If weights can be changed dynamically so that the weighted analytic center is a good approximation to the volumetric center, then a good algorithm for convex optimization would result. The fact that we have a good algorithm for finding the weighted analytic center would clearly be beneficial. It may be that scaling the weights can lead to significant improvements in interior point algorithms for optimization problems with special structure, such as network optimization problems. We can also extend the notion of weighted analytic center to regions more general than polytopes. It remains to be seen if the scaling technique will prove as useful in those cases.

Acknowledgement

We would like to thank Michael Loui for close readings of early drafts of this paper and for many helpful suggestions to improve the presentation.

References

[1] J. Edmonds and R.M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the ACM* 19 (1972) 248–264.

[2] R. Freund, "Projective transformations for interior-point algorithms, and a superlinearly convergent algorithm for the w-center problem," Working Paper, Alfred P. Sloan School of Management, MIT (Cambridge, MA, 1989).

[3] R. Freund and K. Tan, "A method for the parametric center problem, with a strictly monotone polynomial-time algorithm for linear programming," to appear in: *Mathematics of Operations Research.*

[4] H.N. Gabow, "Scaling algorithms for network problems," *Journal of Computer and System Sciences* 31 (1985) 148–168.

[5] C. Gonzaga, "An algorithm for solving linear programming problems in $O(n^3L)$ operations," in: N. Megiddo, ed., *Progress in Mathematical Programming: Interior Point and Related Methods* (Springer, Berlin, 1989) pp. 1–28.

[6] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization* (Springer, Berlin, 1988).

[7] N. Karmarkar, "A new polynomial time algorithm for linear programming," *Combinatorica* 4 (1984) 373–395.

[8] Y. Nesterov and A. Nemirovsky, "Self-concordant functions and polynomial time methods in convex programming," manuscript, USSR Academy of Sciences (Moscow, 1989).

[9] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming," *Mathematical Programming* 40 (1988) 59–93.

[10] G. Sonnevend, "An analytic center for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming," preprint, Department of Numerical Analysis, Institute of Mathematics, Eötvös University (Budapest, 1989).

[11] K. Tan and R. Freund, "Newton's method for the general parametric center problem with applications," Technical Report No. 457, Department of Mathematics, National University of Singapore (Singapore, 1991).

[12] P. Vaidya, "An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations," *Mathematical Programming* 47 (1990) 175–201.

[13] P. Vaidya, "A new algorithm for minimizing convex functions over convex sets," in: *Proceedings of 30th Annual IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society Press, Los Alamitos, CA, 1989) pp. 338–343, to appear in: *Mathematical Programming.*

[14] P. Vaidya, "A locally well-behaved potential function and a simple Newton-type method for finding the center of a polytope," in: N. Megiddo, ed., *Progress in Mathematical Programming: Interior Point and Related Methods* (Springer, Berlin, 1989) pp. 81–90.