

# A POLYNOMIAL-TIME PRIMAL-DUAL AFFINE SCALING ALGORITHM FOR LINEAR AND CONVEX QUADRATIC PROGRAMMING AND ITS POWER SERIES EXTENSION\*†

RENATO D. C. MONTEIRO,<sup>‡</sup> ILAN ADLER<sup>§</sup> AND  
MAURICIO G. C. RESENDE\*\*

We describe an algorithm for linear and convex quadratic programming problems that uses power series approximation of the weighted barrier path that passes through the current iterate in order to find the next iterate. If  $r \geq 1$  is the order of approximation used, we show that our algorithm has time complexity  $O(n^{\frac{1}{2}(1+1/r)}L^{(1+1/r)})$  iterations and  $O(n^3 + n^2r)$  arithmetic operations per iteration, where  $n$  is the dimension of the problem and  $L$  is the size of the input data. When  $r = 1$ , we show that the algorithm can be interpreted as an affine scaling algorithm in the primal-dual setup.

**1. Introduction.** After the presentation of the new polynomial-time algorithm for linear programming by Karmarkar in his landmark paper [15], several so-called interior point algorithms for linear and convex quadratic programming have been proposed. These algorithms can be classified into three main groups:

- (a) Projective algorithms, e.g. [3], [4], [8], [14], [15], [29] and [34].
  - (b) Affine scaling algorithms, originally proposed by Dikin [9]. See also [1], [5], [10] and [33].
  - (c) Path following algorithms, e.g. [13], [18], [19], [24], [25], [26], [28] and [32].
- The algorithms of class (a) are known to have polynomial-time complexity requiring  $O(nL)$  iterations. However, these methods appear not to perform well in practice [30]. In contrast, the algorithms of group (b), while not known to have polynomial-time complexity, have exhibited good behavior on real world linear programs [1], [20], [23], [31]. Most path following algorithms of group (c) have been shown to require  $O(\sqrt{n}L)$  iterations. These algorithms use Newton's method to trace the path of minimizers for the logarithmic barrier family of problems, the so-called central path. The logarithmic barrier function approach is usually attributed to Frisch [12] and is formally studied in Fiacco and McCormick [11] in the context of nonlinear optimization. Continuous trajectories for interior point methods were proposed by Karmarkar [16] and are extensively studied in Bayer and Lagarias [6] [7], Megiddo [21] and Megiddo and Shub [22]. Megiddo [21] related the central path to the classical barrier path in the framework of the primal-dual complementarity relationship. Kojima, Mizuno and Yoshise [19] used this framework to describe a primal-dual interior point algorithm that traces the central trajectory and has a worst time complexity of  $O(nL)$  iterations.

\*Received April 11, 1988; revised October 20, 1988.

*AMS 1980 subject classification.* Primary: 90C05; Secondary 90C25.

*IAOR 1973 subject classification.* Main: Programming: Linear.

*OR/MS Index 1978 subject classification.* Primary: 644 Programming/Linear/Algorithms/Ellipsoidal. Secondary: 656 Programming/Nonlinear/convex.

*Key words.* Programming, affine scaling, power series extensions.

† This research was partially funded by the United States Navy Office of Naval Research, under contract N00014-87-K-0202 and by the Brazilian Postgraduate Education Agency—CAPES.

‡ AT & T Bell Laboratories.

§ University of California, Berkeley.

\*\* AT & T Bell Laboratories.

Monteiro and Adler [25] present a path following primal-dual algorithm that requires  $O(\sqrt{n}L)$  iterations.

This paper describes a modification of the algorithm of Monteiro and Adler [25] and shows that the resulting algorithm can be interpreted as an affine scaling algorithm in the primal-dual setting. We also show polynomial-time convergence for the primal-dual affine scaling algorithm by using a readily available starting primal-dual solution lying on the central path and a suitable fixed step size. Furthermore, we show finite global convergence (not necessarily polynomial) for any starting primal-dual solution. In [21] it is shown that there exists a path of minimizers for the weighted barrier family of problems, that passes through any given primal-dual interior point. The direction generated by our primal-dual affine scaling algorithm is precisely the tangent vector to the weighted barrier path at the current iterate. Hence, the infinitesimal trajectory determined by the current iterate is the weighted barrier path specified by this iterate.

We also present an algorithm based on power series approximations of the weighted barrier path that passes through the current iterate. We show that the complexity of the number of iterations is given by  $O(n^{\frac{1}{2}(1+1/r)}L^{(1+1/r)})$  and that the work per iteration is  $O(n^3 + n^2r)$  arithmetic operations, where  $r$  is the order of the power series approximation used and  $L$  is the size of the problem. Hence, as  $r \rightarrow \infty$ , the number of iterations required approaches  $O(\sqrt{n}L)$ . We develop this algorithm in the context of convex quadratic programming because it provides a more general setting and no additional complication arises in doing so. We should mention that the idea of using higher order approximation by truncating power series is suggested in [17] and also is present in [1], [7] and [21]. However, no convergence analysis is discussed there.

The importance of starting the algorithm at a point close to the central path is also analyzed. More specifically, the complexity of the number of iterations is given as a function of the “distance” of the starting point to the central path. It should be noted that Megiddo and Shub [22] have analyzed how the starting point affects the behavior of the continuous trajectory for the projective and affine scaling algorithms.

This paper is organized as follows. In §2 we motivate the first order approximation algorithm, by showing its relationship to the algorithm of Monteiro and Adler. We also interpret this first order approximation algorithm as an affine scaling algorithm in the primal-dual setup. In §3 we present polynomial-time complexity results for the primal-dual affine scaling algorithm (first order power series) in the context of linear programming and under the assumption that the starting point lies on the central path. In §4, we analyze the higher order approximation algorithm in the more general context of convex quadratic programming. We also analyze how the choice for the starting point affects the complexity of the number of iterations. Concluding remarks are made in §5.

**2. Motivation.** In this section we provide some motivation for the first order version of the algorithm that will be described in this paper. We concentrate our discussion on the relationship between this algorithm and the algorithm of Monteiro and Adler [25]. We also give an interpretation of the first order algorithm as an affine scaling algorithm in the primal-dual setup.

Throughout this paper we adopt the notation used in [19] and [25]. If the lower case  $x = (x_1, \dots, x_n)$  is an  $n$ -vector, then the corresponding upper case  $X$  denotes the diagonal matrix  $\text{diag}(x) = \text{diag}(x_1, \dots, x_n)$ . We denote the  $j$ th component of an  $n$ -vector  $x$  by  $x_j$ , for  $j = 1, \dots, n$ . A point  $(x, y, z) \in \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^n$  will be denoted by the lower case  $w$ . The logarithm of a real number  $a > 0$  on the natural base and on base 2 will be denoted by  $\ln a$  and  $\log a$  respectively. We denote the 2-norm and the  $\infty$ -norm in  $\mathcal{R}^n$  by  $\|\cdot\|$  and  $\|\cdot\|_\infty$  respectively. Finally, for  $w = (x, y, z) \in \mathcal{R}^n \times \mathcal{R}^m$

$\times \mathcal{R}^n$ , we denote by  $f(w) = (f_1(w), \dots, f_n(w))^T \in \mathcal{R}^n$ , the  $n$ -vector defined by

$$(1) \quad f_i(w) = x_i z_i, \quad i = 1, \dots, n.$$

Consider the pair of the standard form linear program

$$(2) \quad (\text{P}) \text{ minimize } c^T x$$

$$(3) \quad \text{subject to: } Ax = b,$$

$$(4) \quad x \geq 0,$$

and its dual

$$(5) \quad (\text{D}) \text{ maximize } b^T y$$

$$(6) \quad \text{subject to: } A^T y + z = c,$$

$$(7) \quad z \geq 0,$$

where  $A$  is an  $m \times n$  matrix,  $x$ ,  $c$  and  $z$  are  $n$ -vectors and  $b$  and  $y$  are  $m$ -vectors. We assume that the entries of  $A$ ,  $b$  and  $c$  are integer.

We define the sets of interior feasible solutions of problems (P) and (D) as

$$(8) \quad S = \{x \in \mathcal{R}^n; Ax = b, x > 0\},$$

$$(9) \quad T = \{(y, z) \in \mathcal{R}^m \times \mathcal{R}^n; A^T y + z = c, z > 0\}$$

respectively, and let

$$(10) \quad W = \{(x, y, z); x \in S, (y, z) \in T\}.$$

We define the duality gap at a point  $w = (x, y, z) \in W$  as  $c^T x - b^T y$ . One can easily verify that for any  $w \in W$ ,  $c^T x - b^T y = x^T z$ . In view of this relation, we refer to the duality gap as the quantity  $x^T z$  instead of the usual  $c^T x - b^T y$ . We make the following assumptions regarding (P) and (D):

*Assumption 2.1.* (a)  $S \neq \emptyset$ .

(b)  $T \neq \emptyset$ .

(c)  $\text{rank}(A) = m$ .

Before we describe the primal-dual affine scaling algorithm, we briefly review the concept of solution pathways for the weighted logarithmic barrier function family of problems associated with problem (P). For a comprehensive discussion of this subject, see [11] and [21].

The weighted barrier function method works on a parametrized family of problems penalized by the weighted barrier function as follows. The weighted barrier function problem with parameter  $\mu > 0$  and weights  $s_j > 0$ ,  $j = 1, \dots, n$  is:

$$(11) \quad (\text{P}_\mu) \text{ minimize } c^T x - \mu \sum_{j=1}^n s_j \ln x_j$$

$$(12) \quad \text{subject to: } Ax = b,$$

$$(13) \quad x > 0.$$

Conditions (a)–(b) of Assumption 2.1 imply that the set of optimal solutions of (P) is nonempty and bounded [25]. This fact implies that  $(P_\mu)$  has a *unique* global optimal solution  $x \equiv x^s(\mu)$  that is characterized by the following Karush-Kuhn-Tucker stationary condition (cf. [11], [21]):

$$(14) \quad ZXs - \mu s = 0,$$

$$(15) \quad Ax - b = 0, \quad x > 0,$$

$$(16) \quad A^T y + z - c = 0,$$

where  $s = (s_1, \dots, s_n)$  denotes the vector of weights,  $y \equiv y^s(\mu) \in \mathcal{R}^m$  and  $z \equiv z^s(\mu) \in \mathcal{R}^n$ . Furthermore, as  $\mu \rightarrow 0^+$ , the solution  $x^s(\mu)$  for (14)–(16) converges to an optimal solution of (P) and the corresponding pair  $(y^s(\mu), z^s(\mu)) \in T$  converges to an optimal solution of (D) [11], [21]. We refer to the path  $w^s: \mu \rightarrow w^s(\mu) \equiv (x^s(\mu), y^s(\mu), z^s(\mu))$  as the path of solutions of problem (P) with weight  $s = (s_1, \dots, s_n)$ .

We define the central path  $w(\mu)$  as the path of solutions  $w^s(\mu)$  of problem (P) with  $s = (1, \dots, 1)$  and let  $\Gamma$  denote the set of points traced by the central path, that is,

$$(17) \quad \Gamma = \{ w = (x, y, z) \in W; \text{ for some } \mu > 0, x_i z_i = \mu, i = 1, \dots, n \}.$$

For convenience, we also refer to the set  $\Gamma$  as the central path.

Monteiro and Adler [25] present an interior path following primal-dual algorithm which requires at most  $O(\sqrt{n}L)$  iterations. This primal-dual algorithm assumes given constants  $\theta$  and  $\delta$  satisfying

$$(18) \quad 0 \leq \theta < \frac{1}{2},$$

$$(19) \quad 0 < \delta < \sqrt{n},$$

$$(20) \quad \frac{\theta^2 + \delta^2}{2(1 - \theta)} \leq \theta \left( 1 - \frac{\delta}{\sqrt{n}} \right)$$

(e.g.  $\theta = \delta = 0.35$ ) and an initial feasible interior solution  $w^0 \in W$  satisfying the following criterion of closeness to the central path  $\Gamma$ :

$$(21) \quad \|f(w^0) - \mu^0 e\| \leq \theta \mu^0,$$

where  $\mu^0 = (x^0)^T z^0 / n$ . Also assumed given is a positive tolerance  $\epsilon$  for the duality gap. The algorithm iterates until the duality gap  $(x^k)^T z^k$  falls below the tolerance  $\epsilon$ .

For  $w \in W$  and  $\mu > 0$ , we denote the feasible direction  $\Delta w \equiv (\Delta x, \Delta y, \Delta z)$  obtained by solving the system of linear equations

$$(22) \quad Z \Delta x + Z \Delta z = X Z e - \mu e,$$

$$(23) \quad A \Delta x = 0,$$

$$(24) \quad A^T \Delta y + \Delta z = 0,$$

by  $\Delta w(w, \mu)$ . The direction  $\Delta w(w, \mu)$  is the Newton direction associated with system (14)–(16) for the parameter  $\mu$  fixed and the weights  $s_j = 1, j = 1, \dots, n$  [19], [25].

System (22)–(24) has the following solution:

$$\begin{aligned} \Delta x &= \left[ Z^{-1} - Z^{-1}XA^T(AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe - \mu e) \\ \Delta y &= - \left[ (AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe - \mu e) \\ \Delta z &= \left[ A^T(AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe - \mu e) \end{aligned}$$

The algorithm is given next.

*Algorithm 2.1.* The Algorithm of Monteiro and Adler.

---

**procedure** PrimalDual ( $A, b, c, \epsilon, w^0, \delta$ )  
 1. Set  $k := 0$  and  $\mu^0 = (x^0)^T z^0 / n$ ;  
 2. **do**  $(x^k)^T z^k > \epsilon \rightarrow$   
 3.      $\mu^{k+1} := \mu^k (1 - \delta / \sqrt{n})$ ;  
 4.      $\Delta w^k := \Delta w(w^k, \mu^{k+1})$ ;  
 5.      $w^{k+1} := w^k - \Delta w^k$ ;  
 6.     Set  $k := k + 1$ ;  
 7. **od**;  
**end** PrimalDual;

---

Note that in line 5, no step size is used to define the next iterate. Instead, we can view  $\mu$  as playing the role of step size. The following theorem, which is proved in [25], leads to polynomial-time complexity of the above algorithm.

**THEOREM 2.1.** *Let  $\theta$  and  $\delta$  be constants satisfying (18)–(20). Assume that  $w = (x, y, z) \in W$  satisfies*

$$(25) \quad \|f(w) - \mu e\| \leq \theta \mu$$

where  $\mu = x^T z / n$ . Let  $\hat{\mu} > 0$  be defined as  $\hat{\mu} = \mu(1 - \delta / \sqrt{n})$ . Consider the point  $\hat{w} = (\hat{x}, \hat{y}, \hat{z}) \in \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^n$  given by  $\hat{w} = w - \Delta w$ , where  $\Delta w \equiv \Delta w(w, \hat{\mu})$  satisfies (22)–(24). Then we have

- (a)  $\|f(\hat{w}) - \hat{\mu} e\| \leq \theta \hat{\mu}$ ,
- (b)  $\hat{w} \in W$ ,
- (c)  $\hat{x}^T \hat{z} = n \hat{\mu}$ .

The approach of the algorithm of this paper is to compute the search direction  $\Delta w$  by solving system (22)–(24) with  $\mu = 0$ , and introduce a step size  $\alpha$  so that the new iterate  $\hat{w}$  is found from the current iterate  $w$  as follows:

$$(26) \quad \hat{w} = w - \alpha \Delta w.$$

More specifically, the direction  $\Delta w = (\Delta x, \Delta y, \Delta z)$  is determined by the following system of linear equations

$$(27) \quad Z \Delta x + X \Delta z = XZe,$$

$$(28) \quad A \Delta x = 0,$$

$$(29) \quad A^T \Delta y + \Delta z = 0,$$

which results in the following direction:

$$(30) \quad \begin{aligned} \Delta x &= \left[ Z^{-1} - Z^{-1}XA^T(AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe) \\ &= \left[ Z^{-1} - Z^{-1}XA^T(AZ^{-1}XA^T)^{-1}AZ^{-1} \right] Xc, \end{aligned}$$

$$(31) \quad \begin{aligned} \Delta y &= - \left[ (AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe) \\ &= - (AZ^{-1}XA^T)^{-1}b, \end{aligned}$$

$$(32) \quad \begin{aligned} \Delta z &= \left[ A^T(AZ^{-1}XA^T)^{-1}AZ^{-1} \right] (XZe) \\ &= A^T(AZ^{-1}XA^T)^{-1}b, \end{aligned}$$

where the second equalities in (30)–(32) follow from the fact that  $z = c - A^T y$  and  $Ax = b$ . Note that the computation of  $\Delta y$  and  $\Delta z$  is a byproduct of the computation of  $\Delta x$ . We denote the solution  $\Delta w$  of system (27)–(29) by  $\Delta w(w)$ .

We show that, by appropriately choosing the step size  $\alpha > 0$  and an initial starting point  $w^0 \in W$  (via artificial variables), the algorithm outlined above has polynomial-time complexity. A detailed description of the algorithm is presented in §3 together with a proof of polynomial-time complexity.

We now give an interpretation of this algorithm as an affine scaling algorithm in the primal-dual setting. Before, we need to describe a general framework for affine scaling algorithms. An affine scaling algorithm assumes a feasible interior point  $x^0 \in S$  is given as a starting point. Given the  $k$ th iterate  $x \equiv x^k \in S$ , the algorithm computes a search direction  $\Delta x \equiv \Delta x^k$  as follows. Let  $D \equiv D^k$  be a diagonal matrix with strictly positive diagonal entries. Consider the linear scaling transformation  $\Psi_D: \mathcal{R}^n \rightarrow \mathcal{R}^n$ , where  $\Psi_D(x) = D^{-1}x$ . In the transformed space problem (P) becomes

$$(33) \quad (P_D) \text{ minimize } (Dc)^T v$$

$$(34) \quad \text{subject to: } ADv = b,$$

$$(35) \quad v \geq 0.$$

The search direction  $d$  in the transformed space is obtained by projecting the gradient vector  $Dc$  orthogonally onto the linear subspace  $\{v: ADv = 0\}$  to obtain a feasible direction that yields the maximum rate of variation in the transformed objective function. Specifically, this direction is given by

$$(36) \quad d = \left[ I - DA^T(AD^2A^T)^{-1}AD \right] Dc.$$

Hence, in the original space the direction  $\Delta x$  is given by

$$(37) \quad \Delta x = \Psi_D^{-1}(d)$$

$$(38) \quad = D \left[ I - DA^T(AD^2A^T)^{-1}AD \right] Dc.$$

Since (P) is posed in minimization form the next iterate  $\hat{x} \equiv x^{k+1}$  is given by

$$(39) \quad \hat{x} = x - \alpha \Delta x,$$

where  $\alpha > 0$  is selected so as to guarantee that the iterate  $\hat{x} > 0$ .

When the scaling matrix  $D \equiv X$ , (38) is the direction generated by the primal affine scaling algorithm [5], [10], [33]. Note that in this case, the primal affine transformation  $\Psi_X$  maps the current iterate  $x$  in the original space into the vector of all ones in the transformed space. Commonly, for the primal affine scaling algorithm, the step size  $\alpha$  is computed by performing a ratio test and multiplying the step size resulting from the ratio test by a fixed positive constant less than 1 (see for example [5], [10], [30] and [33] for details).

The primal-dual algorithm can also be viewed as a special case of this general framework if we assume that besides the current primal iterate  $x \in S$ , we also have a current dual iterate  $(y, z) \in T$  in the background. In this case, if we let the scaling matrix  $D \equiv (Z^{-1}X)^{1/2}$ , then (38) is exactly the direction given by (30). Note that now the current iterate  $x$  in the original space is mapped, under the affine transformation  $\Psi_D$ , into the following vector in the transformed space

$$(40) \quad (XZ)^{1/2}e = \left(\sqrt{x_1z_1}, \dots, \sqrt{x_nz_n}\right)^T.$$

The above framework was described for problems posed in standard form. A similar description can be done for problems posed in format of the dual problem (D). In this case, the affine transformation  $\Psi_D$  is used to scale the slack vector  $z$ . When the scaling matrix  $D \equiv Z^{-1}$ , we obtain the dual affine algorithm [1]. More specifically, if  $(y, z) \in T$  is the current iterate, the direction computed by the dual affine scaling algorithm is given by

$$(41) \quad \Delta y = -(AD^2A^T)^{-1}b,$$

$$(42) \quad \Delta z = A^T(AD^2A^T)^{-1}b,$$

where  $D \equiv Z^{-1}$  and the next iterate  $(\hat{y}, \hat{z}) \in T$  is found by setting  $\hat{y} = y - \alpha \Delta y$  and  $\hat{z} = z - \alpha \Delta z$ . The step size  $\alpha$  is computed in a way similar to the one in the primal affine scaling algorithm and guarantees that  $\hat{z} > 0$ . The dual affine scaling algorithm has been shown to perform well in practice [1], [2], [20], [23]. In this dual framework, if the scaling matrix  $D \equiv (Z^{-1}X)^{1/2}$ , then (41) and (42) are identical to (31) and (32) respectively. Thus, in this case, we again obtain the primal-dual affine scaling algorithm.

Global, though not polynomial, convergence proofs exist for the affine scaling algorithms under the assumption of nondegeneracy [5], [10], [33]. It is conjectured, however, that both the primal and dual affine algorithms have worst case time complexity that are not polynomial. By appropriately choosing a starting primal-dual solution and a suitable fixed step size, we show in this paper that in the primal-dual setting, the affine scaling algorithm has polynomial-time complexity.

**3. The algorithm and convergence result.** In this section, we complete the description of the primal-dual affine scaling algorithm that was briefly outlined in §2. Polynomial-time complexity for this algorithm is established by selecting a suitable starting point and an appropriate step size. We make one further assumption regarding problems (P) and (D).

*Assumption 3.1.* An initial point  $w^0 = (x^0, y^0, z^0) \in W$  is given such that the following condition holds:

$$(43) \quad x_i^0 z_i^0 = \mu^0, \quad i = 1, 2, \dots, n,$$

where  $0 < \mu^0 = 2^{O(L)}$ .

Relation (43) is equivalent to requiring that  $w^0 \equiv w(\mu^0)$  where  $w(\mu)$  is the central path. Observe that Assumption 3.1 implies (a) and (b) of Assumption 2.1. Given a linear program in standard form, an associated augmented linear program in standard form can be constructed satisfying Assumptions 2.1 and 3.1 and whose solution yields a solution for the original problem, if such exists. Indeed, in [25], it is shown that the augmented problem can be constructed in such a way that a initial point  $w^0$  lying in the central path is readily available and that the size of the original problem and that of the augmented problem are of the same order. The point  $w^0$  is used as the algorithm's initial iterate.

The algorithm generates a sequence of points  $w^k \in W$ , ( $k = 1, 2, \dots$ ) starting from  $w^0$  as follows. Given  $w^k \in W$ , the search direction  $\Delta w(w^k)$  is computed according to (30)–(32) and  $w^{k+1}$  is found by setting

$$(44) \quad w^{k+1} = w^k - \alpha^k \Delta w(w^k)$$

where  $\alpha^k$  is the step size at the  $k$ th iteration. For the purpose of this paper, which is limited to a theoretical analysis, we choose a constant step size  $\alpha^k = \alpha$  (for  $k = 0, 1, 2, \dots$ ), to be described next. Let  $\epsilon$  be a given tolerance for the duality gap, *i.e.* the algorithm terminates when the duality gap  $(x^k)^T z^k$  is no longer greater than  $\epsilon$ . The step size is chosen to depend on the parameter  $\mu^0$ , the dimension  $n$  and the tolerance  $\epsilon$  as follows:

$$(45) \quad \alpha = \frac{1}{n \lceil \ln(n\epsilon^{-1}\mu^0) \rceil}$$

where  $\lceil x \rceil$  denotes the smallest integer greater than or equal to  $x$ . We also assume that  $\alpha < 1/2$ , which can be insured by the choice of the tolerance  $\epsilon$ . Note that the larger  $\epsilon^{-1}$ ,  $\mu^0$  and  $n$  are, the smaller the step size  $\alpha$  is. We are now ready to describe the algorithm, which is presented below.

*Algorithm 3.1.* The Primal-Dual Affine Scaling Algorithm.

---

```

procedure PrimalDualAffine ( $A, b, c, \epsilon, w^0$ )
1. Set  $k := 0$ ;
2. do  $(x^k)^T z^k > \epsilon \rightarrow$ 
3.   Compute  $\Delta w(w^k)$  according to (30)–(32);
4.   Set  $w^{k+1} := w^k - \alpha \Delta w(w^k)$  where  $\alpha$  is a constant given by (45);
5.   Set  $k := k + 1$ ;
6. od;
end PrimalDualAffine;
    
```

---

Algorithm 3.1 is given as input the data  $A, b, c$ , a tolerance  $\epsilon > 0$  for the duality gap stopping criterion and the initial iterate  $w^0$  as the one specified in Assumption 3.1.

The following theorem, whose proof we defer to later in this section, describes the behavior of one iteration of Algorithm 3.1 given that a general step size  $\alpha$  is taken.

**THEOREM 3.2.** *Let  $w = (x, y, z) \in W$  be given such that*

$$(46) \quad \|f(w) - \mu e\|_\infty \equiv \max_{1 \leq i \leq n} |x_i z_i - \mu| \leq \theta \mu$$



where  $\mu \equiv x^T z/n > 0$  and  $0 \leq \theta < 1$ . Consider the point  $\hat{w} = (\hat{x}, \hat{y}, \hat{z})$  defined as  $\hat{w} = w - \alpha \Delta w$ , where  $\Delta w \equiv \Delta w(w)$  and  $\alpha \in (0, 1)$ . Let  $\hat{\mu} \equiv (1 - \alpha)\mu$  and

$$(47) \quad \hat{\theta} \equiv \theta + \frac{n\alpha^2}{2(1 - \alpha)}.$$

Then we have:

- (a)  $\|f(\hat{w}) - \hat{\mu}e\|_\infty \leq \hat{\theta}\hat{\mu}$ ,
- (b) If  $\hat{\theta} < 1$  then  $\hat{w} \in W$ ,
- (c)  $\hat{\mu} = \hat{x}^T \hat{z}/n$ .

Theorem 3.2 parallels Theorem 2.1 closely. In spite of the fact that Theorem 3.2 was formulated in terms of the  $\infty$ -term, as compared to the 2-norm formulation of Theorem 2.1, we should point out that Theorem 3.2 also holds for the 2-norm as will become clear from its proof. The reason we state Theorem 3.2 in terms of the  $\infty$ -norm is discussed in the next section where we prove convergence (not necessarily polynomial) of algorithm 3.1 for any given starting point  $w^0 \in W$ . Polynomial convergence will only be guaranteed in the case that the initial starting point is in some sense *close* to the central path. In that context, the  $\infty$ -norm will play an important role.

We can view  $f(w)$  as a map from  $\mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^n$  into  $\mathcal{R}^n$ , mapping  $w = (x, y, z)$  into the complementarity vector  $XZe$ . Under this map, the set  $W$  is mapped onto the positive orthant, the central path  $\Gamma$  is mapped onto the *diagonal* line  $f(\Gamma) = \{\mu e; \mu > 0\}$  and an optimal solution  $w^* = (x^*, y^*, z^*)$  for the pair of problems (P) and (D) is mapped into the zero vector [25]. The image under  $f$  of the set of points  $w \in W$  such that  $\|f(w) - \mu e\| \leq \theta\mu$  with  $\mu = x^T z/n$  is a cone in the positive orthant of  $\mathcal{R}^n$  having the *diagonal* line  $f(\Gamma)$  as a central axis and the zero vector as an extreme point. The central axis forms a common angle with all the extreme rays of the cone and this angle is an increasing function of  $\theta$ . For this reason, we refer to  $\theta$  as the opening of the cone. Theorem 2.1 states that if we start at a point inside this cone, then all iterates will remain within the same cone and will approach the optimal solution  $f(w^*)$  at a rate given by  $(1 - \delta/\sqrt{n})$ . This is to be contrasted with Theorem 3.2, where the iterates are guaranteed to be in cones with openings that gradually increase from one iteration to the other.

Note that by (c) of Theorem 3.2, we have

$$(48) \quad \hat{x}^T \hat{z}^T = n\hat{\mu} = (1 - \alpha)n\mu = (1 - \alpha)x^T z$$

that is, the duality gap is reduced by a factor of  $(1 - \alpha)$  at each iteration. Therefore, it is desirable to choose  $\alpha$  as large as possible in order to obtain as large as possible a decrease in the duality gap. Once  $\alpha$  is specified, the number of iterations necessary to reduce the duality gap to a value  $\leq \epsilon$  is not greater than

$$(49) \quad K = \alpha^{-1} \lceil \ln(n\mu^0 \epsilon^{-1}) \rceil$$

which is immediately implied by the fact that

$$(50) \quad (x^K)^T z^K = (1 - \alpha)^K (x^0)^T z^0 = (1 - \alpha)^K n\mu^0 \leq \epsilon$$

where the second equality is due to (43) and the inequality follows by the choice of  $K$ . The choice of  $\alpha$  should now be made to guarantee feasibility of all iterates  $w^k$ , ( $k = 0, 1, \dots, K$ ) and toward this objective, (b) of Theorem 3.2 will play an important

role. The choice of  $\alpha$  given by relation (45) becomes clear in the proof of the following result.

**COROLLARY 3.3.** *Let  $K$  be as in (49) and consider the first  $K$  iterates generated by Algorithm 3.1, i.e. the sequence  $\{w^k\}_{k=0}^K$ . Let  $\mu^k \equiv (1 - \alpha)^k \mu^0$  and  $\theta^k \equiv kn\alpha^2$ , for all  $k = 0, 1, 2, \dots, K$ . Then, for all  $k = 0, 1, 2, \dots, K$  we have:*

- (a)  $\|f(w^k) - \mu^k e\|_\infty \leq \theta^k \mu^k$ ,
- (b)  $w^k \in W$ ,
- (c)  $(x^k)^T z^k / n = \mu^k$ .

**PROOF.** From (45), (49) and the definition of  $\theta^k$ , it follows that

$$(51) \quad \theta^k \leq Kn\alpha^2 = 1$$

for all  $k = 0, 1, \dots, K$ . The proof of (a), (b) and (c) is by induction on  $k$ . Obviously (a), (b) and (c) hold for  $k = 0$ , due to Assumption 3.1. Assume (a), (b) and (c) hold for  $k$ , where  $0 \leq k < K$ . Since  $\alpha < 1/2$ , it follows that

$$(52) \quad \theta^k + \frac{n\alpha^2}{2(1 - \alpha)} < \theta^k + n\alpha^2 = \theta^{k+1} \leq 1.$$

In view of the last relation, we can apply Theorem 3.2 with  $w \equiv w^k$ ,  $\mu \equiv \mu^k$  and  $\theta \equiv \theta^k$  to conclude that (a), (b) and (c) hold for  $k + 1$ . This completes the proof of the corollary. ■

We now discuss some consequences of the above corollary. Let  $L$  denote the size of linear programming problem (P). If we set  $\epsilon = 2^{-O(L)}$ , then by (50), the iterate  $w^K$  generated by Algorithm 3.1, where  $K$  is given by (49), satisfies  $(x^K)^T z^K \leq \epsilon = 2^{-O(L)}$ . Then, from  $w^K$ , one can find exact solutions of problems (P) and (D) by solving a system of linear equations which involves at most  $O(n^3)$  arithmetic operations [27]. Using this observation, we obtain the main result of this section.

**THEOREM 3.4.** *Algorithm 3.1 solves the pair of problems (P) and (D) in at most  $O(nL^2)$  iterations, where each iteration involves  $O(n^3)$  arithmetic operations.*

**PROOF.** From (45), (50) and the fact that  $\epsilon = 2^{-O(L)}$  and  $\mu = 2^{O(L)}$ , it follows that the algorithm takes at most

$$(53) \quad K = n \left[ \ln(n\epsilon^{-1}\mu^0) \right]^2 = O(nL^2)$$

iterations to find a point  $w^k \in W$  satisfying  $(x^k)^T z^k \leq \epsilon = 2^{-O(L)}$ . The work in each iteration is dominated by the effort required to compute and invert the matrix  $A(Z^k)^{-1} X^k A^T$ , namely,  $O(n^3)$  arithmetic operations. This proves the theorem. ■

We now turn our attention towards proving Theorem 3.2. The proof requires some technical lemmas.

**LEMMA 3.5.** *Let  $w = (x, y, z) \in W$  be given. Consider the point  $\hat{w} = (\hat{x}, \hat{y}, \hat{z})$  given by  $\hat{w} = w - \alpha \Delta w$ , where  $\Delta w \equiv \Delta w(w) = (\Delta x, \Delta y, \Delta z)$  and  $\alpha > 0$ . Then we have:*

$$(54) \quad \hat{x}_i \hat{z}_i = (1 - \alpha)x_i z_i + \alpha^2 \Delta x_i \Delta z_i \quad \text{and}$$

$$(55) \quad (\Delta x)^T \Delta z = 0.$$

PROOF. First, we show (54). We have:

$$\begin{aligned}
 \hat{x}_i \hat{z}_i &= (x_i - \alpha \Delta x_i)(z_i - \alpha \Delta z_i) \\
 &= x_i z_i - \alpha(x_i \Delta z_i + z_i \Delta x_i) + \alpha^2 \Delta x_i \Delta z_i \\
 &= x_i z_i - \alpha x_i z_i + \alpha^2 \Delta x_i \Delta z_i \\
 &= (1 - \alpha)x_i z_i + \alpha^2 \Delta x_i \Delta z_i
 \end{aligned}$$

where the third equality is implied by (27). This completes the proof of (54). To show (55) multiply (28) and (29) on the left by  $(\Delta y)^T$  and  $(\Delta x)^T$ , respectively, and combine the two resulting expressions. This shows (55) and completes the proof of the lemma. ■

The next lemma appears as Lemma 4.7 in [25], where it is proved.

LEMMA 3.6. *Let  $r$ ,  $s$  and  $t$  be real  $n$ -vectors satisfying  $r + s = t$  and  $r^T s \geq 0$ . Then we have:*

$$(56) \quad \max(\|r\|, \|s\|) \leq \|t\|,$$

$$(57) \quad \|RSe\| \leq \frac{\|t\|^2}{2}$$

where  $R$  and  $S$  denote the diagonal matrices corresponding to the vectors  $r$  and  $s$ , respectively.

As a consequence of the previous lemma, we have the following result.

LEMMA 3.7. *Let  $w = (x, y, z) \in W$  be given. Consider the direction  $\Delta w \equiv \Delta w(w) = (\Delta x, \Delta y, \Delta z)$ . Define  $\Delta f \in \mathcal{R}^n$  as  $\Delta f = (\Delta X)(\Delta Z)e$ , where  $(\Delta X)$  and  $(\Delta Z)$  are the diagonal matrices corresponding to  $\Delta x$  and  $\Delta z$ , respectively. Then we have*

$$(58) \quad \|\Delta f\| \leq \frac{x^T z}{2}.$$

PROOF. Let  $D = (Z^{-1}X)^{1/2}$ . Multiplying both sides of (27) by  $(XZ)^{-1/2}$ , we have

$$(59) \quad D^{-1} \Delta x + D \Delta z = (XZ)^{1/2} e.$$

By (55) we have that  $(D^{-1} \Delta x)^T (D \Delta z) = 0$ . Hence, we can apply Lemma 3.6 with  $r = D^{-1} \Delta x$ ,  $s = D \Delta z$  and  $t = (XZ)^{1/2} e$  resulting in

$$(60) \quad \|(D^{-1} \Delta X)(D \Delta Z)e\| \leq \frac{\|(XZ)^{1/2} e\|^2}{2}$$

which is equivalent to (58). This completes the proof of the lemma. ■

We are now ready to prove Theorem 3.2.

PROOF (THEOREM 3.2). From (54) and the fact that  $\hat{\mu} = (1 - \alpha)\mu$ , it follows that

$$(61) \quad \hat{x}_i \hat{z}_i - \hat{\mu} = (1 - \alpha)(x_i z_i - \mu) + \alpha^2 \Delta x_i \Delta z_i.$$

Since  $\mu = x^T z/n$ , it follows from Lemma 3.7 that

$$(62) \quad |\Delta x_i, \Delta z_i| \leq \|\Delta f\| \leq \frac{x^T z}{2} = \frac{n\mu}{2}.$$

Using relations (46), (61), (62) and the fact that  $\hat{\mu}(1 - \alpha)\mu$ , we obtain

$$(63) \quad \begin{aligned} |\hat{x}_i \hat{z}_i - \hat{\mu}| &\leq (1 - \alpha)|x_i z_i - \mu| + \alpha^2 |\Delta x_i \Delta z_i| \\ &\leq (1 - \alpha)\theta\mu + \frac{n\alpha^2\mu}{2} \\ &= \left[ \theta + \frac{n\alpha^2}{2(1 - \alpha)} \right] \hat{\mu}. \end{aligned}$$

Since the last relation holds for all  $i = 1, \dots, n$ , (a) follows.

We now show that  $\hat{w} \in W$  under the assumption that

$$(64) \quad \theta + \frac{n\alpha^2}{2(1 - \alpha)} < 1.$$

To show that  $\hat{w} \in W$ , it suffices to show that  $\hat{x} > 0$  and  $\hat{z} > 0$ . Assume by contradiction that  $\hat{x}_i \leq 0$  or  $\hat{z}_i \leq 0$ , for some  $i$ . Using relations (63) and (64), it follows that  $\hat{x}_i \hat{z}_i > 0$ . Hence, it must be the case that  $\hat{x}_i < 0$  and  $\hat{z}_i < 0$ . This requires that  $\alpha \Delta x_i > x_i$  and  $\alpha \Delta z_i > z_i$ , which implies that

$$(65) \quad \alpha^2 \Delta x_i \Delta z_i > x_i z_i \geq x_i z_i \geq (1 - \theta)\mu.$$

This last inequality and (62) imply that

$$(66) \quad \frac{n\alpha^2\mu}{2} > (1 - \theta)\mu$$

which contradicts the fact that

$$(67) \quad \theta + \frac{n\alpha^2}{2} \leq \theta + \frac{n\alpha^2}{2(1 - \alpha)} < 1.$$

This shows (b). Summing (54) over all  $i = 1, 2, \dots, n$  and noting (55), we obtain (c). This completes the proof of Theorem 3.2. ■

**4. Primal-dual power series algorithm.** The algorithm of §3 can be viewed as generating points based on a first order approximation of the weighted logarithmic barrier path of solutions determined by the current iterate. This observation will be examined later in more detail. In this section, we present an algorithm based on power series approximation of the path of solutions that passes through the current iterate. As one should expect, faster convergence is obtained. More interestingly, we show that the complexity of the number of iterations depends on the order of approximation, say  $r$ , and moreover, as  $r \rightarrow \infty$ , the number of iterations asymptotically approaches the complexity of the number of iterations of the primal-dual path following algorithm [25], namely,  $O(\sqrt{n}L)$  iterations. We develop the algorithm in this section in the

context of convex quadratic programming problems because it provides a more general setting for the algorithm without additional complications.

We start by briefly extending the concepts introduced in §2 to convex quadratic programming problems. Consider the convex quadratic programming problem as follows. Let

$$(68) \quad \text{(P) minimize } c^T x + \frac{1}{2} x^T Q x$$

$$(69) \quad \text{subject to: } Ax = b,$$

$$(70) \quad x \geq 0,$$

where  $A$ ,  $b$ ,  $c$  and  $x$  are as in §2 and  $Q$  is an  $n \times n$  symmetric positive semidefinite matrix. Its associated Lagrangian dual problem is given by

$$(71) \quad \text{(D) maximize } -\frac{1}{2} x^T Q x + b^T y$$

$$(72) \quad \text{subject to: } -Qx + A^T y + z = c,$$

$$(73) \quad x \geq 0,$$

where  $y$  is an  $m$ -vector and  $z$  is an  $n$ -vector. We define the sets of interior feasible solutions of problems (P) and (D) to be

$$(74) \quad S = \{x \in \mathcal{R}^n; Ax = b, x > 0\},$$

$$(75) \quad T = \{(x, y, z) \in \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^n; -Qx + A^T y + z = c, z > 0\}$$

respectively and  $W$  is now defined to be

$$(76) \quad W = \{(x, y, z); x \in S, (x, y, z) \in T\}.$$

The duality gap at a point  $w \in W$ , which is defined as  $c^T x + x^T Q x - b^T y$ , can be easily shown to be given by  $x^T z$ . We make the following assumptions regarding problems (P) and (D):

*Assumption 4.1.* (a) A point  $w^0 = (x^0, y^0, z^0) \in W$  is given.

(b)  $\text{rank}(A) = m$ .

The point  $w^0$  will serve as the initial iterate for the algorithm described below. Observe that (a) of Assumption 4.1 is weaker than Assumption 3.1 since we do not require  $w^0$  to lie in the central path. As a result, the upper bound on the number of iterations for the algorithm described in this section will be given in terms of some measure of *distance* of  $w^0$  with respect to the central path and also in terms of the duality gap at  $w^0$ .

In the context of convex quadratic programming problems, the path of solutions for the weighted barrier function family of problems associated with problem (P), where the weights are  $s = (s_1, \dots, s_n)$ , is determined implicitly by the following parametrized system of equations:

$$(77) \quad x_i^s(\mu) z_i^s(\mu) = s_i \mu, \quad i = 1, \dots, n,$$

$$(78) \quad Ax^s(\mu) = b,$$

$$(79) \quad -Qx^s(\mu) + A^T y^s(\mu) + z^s(\mu) = c.$$

Under Assumption 4.1 and for  $\mu > 0$  fixed, this system is ensured to have a unique solution  $w^s(\mu) \equiv (x^s(\mu), y^s(\mu), z^s(\mu))$ . Furthermore, as  $\mu \rightarrow 0^+$ , the solution  $x^s(\mu) \in S$  for (77)–(79) converges to an optimal solution of (P) and  $w^s(\mu) \equiv (x^s(\mu), y^s(\mu), z^s(\mu)) \in W$  converges to an optimal solution of (D) [11], [21]. With these definitions and notations, the central path associated with problems (P) and (D) is defined as in §2.

Given a point  $w = (x, y, z) \in W$  and letting  $s_i = x_i z_i, i = 1, \dots, n$ , it follows that  $w^s(1) = w$ . Therefore, for this particular set of weights, the path of solutions contains the point  $w$ . The idea of the  $r$ th degree truncated power series approach can be motivated as follows. In order to obtain an approximation to the point  $w^s(1 - \alpha)$  for  $\alpha > 0$ , we consider the  $r$ th order Taylor polynomial,  $r \geq 1$ , of the function  $h: \alpha \rightarrow w^s(1 - \alpha)$  at  $\alpha = 0$  as follows:

$$(80) \quad \begin{aligned} w^r(w, \alpha) &\equiv \sum_{k=0}^r \frac{(\alpha)^k}{k!} \frac{d^k h}{d\alpha^k}(0) \\ &= w + \sum_{k=1}^r \frac{(-\alpha)^k}{k!} \frac{d^k w^s}{d\mu^k}(1) \end{aligned}$$

where for  $k \geq 1, d^k/d\mu^k$  is the  $k$ th derivative operator and for  $k = 0, d^0/d\mu^0$  is defined as the identity operator, that is  $d^0 w^s/d\mu^0(\mu) \equiv w^s(\mu)$ , for all  $\mu$ . If the  $k$ th derivative  $d^k w^s/d\mu^k(1), (k = 1, \dots, r)$ , is known, then one can use  $w^r(w, \alpha)$  to estimate the point  $w^s(1 - \alpha)$ , for  $\alpha$  sufficiently small.

We next show how the  $k$ th derivative

$$(81) \quad \frac{d^k w^s}{d\mu^k}(1) \equiv \left( \frac{d^k x^s}{d\mu^k}(1), \frac{d^k y^s}{d\mu^k}(1), \frac{d^k z^s}{d\mu^k}(1) \right) \text{ for } k = 1, \dots, r$$

can be computed. Taking the derivative of (77)–(79)  $k$  times, and setting  $\mu = 1$ , we obtain

$$(82) \quad \sum_{l=0}^k \binom{k}{l} \frac{d^l x_i}{d\mu^l}(1) \frac{d^{(k-l)} z_i}{d\mu^{(k-l)}}(1) = \begin{cases} x_i z_i & \text{if } k = 1, \\ 0 & \text{if } k \geq 2, \end{cases}$$

$$(83) \quad A \frac{d^k x}{d\mu^k}(1) = 0,$$

$$(84) \quad -Q \frac{d^k x}{d\mu^k}(1) + A^T \frac{d^k y}{d\mu^k}(1) + \frac{d^k z}{d\mu^k}(1) = 0.$$

To eliminate the binomial coefficients above and simplify the expressions below, let  $\Delta^{(k)} w \equiv [d^k w/d\mu^k(1)]/k!$ , for  $k = 0, 1, \dots, r$ . With this notation relations (82)–(84) become

$$(85) \quad \sum_{l=0}^k (\Delta^{(l)} x_i)(\Delta^{(k-l)} z_i) = \begin{cases} x_i z_i & \text{if } k = 1, \\ 0 & \text{if } k \geq 2, \end{cases}$$

$$(86) \quad A \Delta^{(k)} x = 0,$$

$$(87) \quad -Q \Delta^{(k)} x + A^T \Delta^{(k)} y + \Delta^{(k)} z = 0.$$

In terms of the direction  $\Delta^{(k)}w$ ,  $1 \leq k \leq r$ , the right-hand side of (80) becomes

$$(88) \quad w^r(w, \alpha) \equiv \sum_{k=0}^r (-\alpha)^k \Delta^{(k)}w = w + \sum_{k=1}^r (-\alpha)^k \Delta^{(k)}w.$$

Let  $\Delta^{(k)}X$  and  $\Delta^{(k)}Z$  be the diagonal matrices corresponding to the vectors  $\Delta^{(k)}x$  and  $\Delta^{(k)}z$ , respectively. Assume that  $\Delta^{(l)}w \equiv (\Delta^{(l)}x, \Delta^{(l)}y, \Delta^{(l)}z)$ ,  $0 \leq l < k$  have already been computed. Then we compute  $\Delta^{(k)}w = (\Delta^{(k)}x, \Delta^{(k)}y, \Delta^{(k)}z)$  by solving the following system of linear equations, which is exactly system (85)–(87) written in a different format.

$$(89) \quad Z \Delta^{(k)}x + X \Delta^{(k)}z = \begin{cases} XZe & \text{if } k = 1, \\ - \sum_{l=1}^{k-1} (\Delta^{(l)}X)(\Delta^{(k-l)}Z)e & \text{if } k \geq 2, \end{cases}$$

$$(90) \quad A \Delta^{(k)}x = 0,$$

$$(91) \quad -Q \Delta^{(k)}x + A^T \Delta^{(k)}y + \Delta^{(k)}z = 0.$$

Sometimes, we denote the directions  $\Delta^{(k)}w = (\Delta^{(k)}x, \Delta^{(k)}y, \Delta^{(k)}z)$  by  $\Delta^{(k)}w(w)$  to indicate their dependence on the point  $w$ . Note that the coefficients of the system above are the same for the computation of all the directions  $\Delta^{(k)}w$ ,  $1 \leq k \leq r$ . Once the computation of  $\Delta^{(1)}w$  is performed, which takes  $O(n^3)$  arithmetic operations, the directions  $\Delta^{(k)}w$ ,  $2 \leq k \leq r$ , can each be computed in  $O(n^2)$  arithmetic operations. Thus, the overall computation of  $\Delta^{(j)}w$ ,  $1 \leq j \leq r$ , takes  $O(n^3 + rn^2)$  arithmetic operations.

In fact, explicit expressions for  $\Delta^{(k)}w \equiv (\Delta^{(k)}x, \Delta^{(k)}y, \Delta^{(k)}z)$  in terms of the previous directions  $\Delta^{(l)}w$ ,  $l = 1, 2, \dots, k - 1$  are given as follows:

$$\Delta^{(k)}x = (Z + XQ)^{-1} \left[ I - XA^T(A(Z + XQ)^{-1}XA^T)^{-1}A(Z + XQ)^{-1} \right] u,$$

$$\Delta^{(k)}y = - \left[ (A(Z + XQ)^{-1}XA^T)^{-1}A(Z + XQ)^{-1} \right] u,$$

$$\Delta^{(k)}z = Q \Delta^{(k)}x - A^T \Delta^{(k)}y, \quad \text{where}$$

$$(92) \quad u = \begin{cases} XZe & \text{if } k = 1, \\ - \sum_{l=1}^{k-1} (\Delta^{(l)}X)(\Delta^{(k-l)}Z)e & \text{if } k \geq 2. \end{cases}$$

Note that when the matrix  $Q = 0$ , that is, problem (P) is a linear program, then the direction  $\Delta^{(1)}w$  is exactly the direction  $\Delta w \equiv \Delta w(w)$  as defined in §3. Thus, one can easily see that the algorithm to be described next, when  $r = 1$ , generalizes the one presented in the previous section for linear programming. When we consider the infinitesimal version of the algorithm described in the previous section, or more generally, the one presented in this section when  $r = 1$ , we are led to consider the solution of the following differential equation in the set  $W$  of primal-dual interior feasible solutions:

$$(93) \quad \frac{dw}{d\mu}(\mu) = \Delta^{(1)}w(w(\mu)),$$

$$(94) \quad w(\mu^0) = w,$$

where  $\mu^0$  and  $w = (x, y, z) \in W$  are assumed given and (94) determines the initial condition for (93). The trajectories of the differential equation (93) are said to be induced by the vector field  $w \in W \rightarrow \Delta^{(1)}(w) \in \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^n$ . It turns out, by the way we motivate our algorithm, that the trajectory induced by this vector field and passing through the point  $w = (x, y, z)$  is exactly the locus of points traced by the path of solutions  $w^s(\mu)$  of system (77)–(79) when the weights  $s = (s_1, \dots, s_n)$  are given by  $s_i = x_i z_i$ .

Before we describe the algorithm based on the  $r$ th degree truncated power series, we need to introduce some further notation. For  $w \in W$ , let  $f_{\min}(w) \equiv \min_{1 \leq i \leq n} f_i(w)$  and  $f_{\max}(w) = \max_{1 \leq i \leq n} f_i(w)$ . Consider now the point  $w^0 \in W$  mentioned in Assumption 4.1 and let

$$(95) \quad \mu^0 \equiv (f_{\max}(w^0) + f_{\min}(w^0))/2,$$

$$(96) \quad \theta^0 \equiv \frac{f_{\max}(w^0) - f_{\min}(w^0)}{f_{\max}(w^0) + f_{\min}(w^0)}.$$

Note that  $\theta^0 < 1$  and that  $\mu^0$  and  $\theta^0$  satisfy

$$(97) \quad \|f(w^0) - \mu^0 e\|_{\infty} \leq \theta^0 \mu^0.$$

The fact that we are using the  $\infty$ -norm is crucial here in order to guarantee that, given  $w^0 \in W$ , there exist constants  $\mu^0$  and  $\theta^0$  such that  $\theta^0 < 1$  and such that relation (97) holds. In general, given any  $w^0 \in W$ , the above property does not hold if we use the 2-norm. This is the main reason for using the  $\infty$ -norm instead of the 2-norm.

We now have all the ingredients to describe the truncated power series algorithm of degree  $r$ . The truncated power series algorithm of degree  $r$  studied in this section generates a sequence of points  $w^k \in W$  ( $k = 1, 2, \dots$ ), starting from the point  $w^0 \in W$  (cf. Assumption 4.1) as follows. Given  $w^k \in W$ ,  $w^{k+1}$  is found by setting  $w^{k+1} \equiv w^r(w^k, \alpha)$  (cf. (88)), where  $\alpha > 0$  is the step size. As in §3, we assume that the same step size is used for all iterations. The step size  $\alpha > 0$  is determined as follows. Let  $\epsilon > 0$  be a tolerance for the duality gap  $(x^k)^T(z^k)$ , so that, like in §3, we terminate the algorithm as soon as  $(x^k)^T(z^k) \leq \epsilon$ . The step size is determined as a function of the degree of approximation  $r$ , the dimension  $n$ , the parameter  $\mu^0$ , the constant  $\theta^0$  and the tolerance  $\epsilon$  as follows:

$$(98) \quad \alpha = \left( \left[ 2^{(1/2+3/2r)} \gamma^{(1/2+1/2r)} q(r)^{1/r} n^{(1/2+1/2r)} \left[ \ln(2n\epsilon^{-1}\mu^0) \right]^{1/r} \right] \right)^{-1}$$

where  $\gamma = 2/(1 - \theta^0)$  and  $q(r) = \sum_{k=r+1}^{2r} p(k)$  with the sequence  $p(k)$  defined recursively as follows:

$$(99) \quad p(1) = 1,$$

$$(100) \quad p(k) = \sum_{j=1}^{k-1} p(j)p(k-j), \quad k \geq 2.$$

We also assume that the tolerance  $\epsilon$  is given small enough to ensure that  $\alpha \leq 1/2$ . The solution of the recurrence relation (99)–(100) is well known and is given by

$$(101) \quad p(k) = \frac{1}{k} \binom{2k-1}{k-1}.$$

The following estimate of  $q(r)^{1/r}$  will be useful later.



LEMMA 4.2.  $\sup_r q(r)^{1/r} \leq 16$ .

PROOF. Using the formula for  $p(k)$  above and the fact that  $\binom{n}{k} \leq 2^n$  for all  $n$  and  $k \leq n$ , we obtain

$$(102) \quad q(r) \equiv \sum_{k=r+1}^{2r} p(k) \leq rp(2r) \leq \binom{4r}{2r} \leq 2^{4r}$$

and this completes the proof of the lemma. ■

We are now ready to describe the algorithm, which is presented below.

Algorithm 4.1. The Truncated Power Series Algorithm of Degree  $r$ .

---

**procedure** TruncatedPowerSeries ( $a, b, c, \epsilon, w^0$ )

1. Set  $k := 0$ ;
2. **do**  $(x^k)^T z^k > \epsilon \rightarrow$
3.     Compute  $\Delta^{(k)} w(w^k)$ , for  $k = 1, 2, \dots, r$ , as described above;
4.     Set  $w^{k+1} := w^r(w^k, \alpha)$  where  $\alpha$  is the constant given by (98);
5.     Set  $k := k + 1$ ;
6. **od**;

**end** TruncatedPowerSeries;

---

The next theorem is a generalization of Theorem 3.2.

THEOREM 4.3. Let  $w = (x, y, z) \in W$  and  $\mu > 0$  be given such that

$$(103) \quad \|f(w) - \mu e\|_\infty \equiv \max_{1 \leq i \leq n} |x_i z_i - \mu| \leq \theta \mu$$

for some  $0 \leq \theta < 1$ . Consider the point  $\hat{w} \equiv (\hat{x}, \hat{y}, \hat{z})$  given by  $\hat{w} \equiv w^r(w, \alpha)$ , where  $\alpha \in (0, 1)$ . Let  $\hat{\mu} \equiv (1 - \alpha)\mu$  and

$$(104) \quad \hat{\theta} \equiv \theta + \frac{1 - \theta}{1 - \alpha} \sum_{l=r+1}^{2r} \left[ \left( \frac{1 + \theta}{1 - \theta} \right)^{1/2} n^{1/2} \alpha \right]^l p(l).$$

Then we have:

- (a)  $\|f(\hat{w}) - \hat{\mu} e\|_\infty \leq \hat{\theta} \hat{\mu}$ .
- (b) If  $\hat{\theta} < 1$  then  $\hat{w} \in W$ .

Note that the opening of the cone described in the discussion following Theorem 3.2 gradually increases by a term that depends on the  $k$ -powers of the step size,  $r + 1 \leq k \leq 2r$ . The consequences of Theorem 4.3 are as follows.

COROLLARY 4.4. Let  $K = \alpha^{-1} \lceil \ln(2n\epsilon^{-1}\mu^0) \rceil$ . Consider the first  $K$  iterates generated by Algorithm 4.1, that is, the sequence  $\{w^k\}_{k=1}^K$ . Let  $\mu^k = (1 - \alpha)^k \mu^0$  and

$$(105) \quad \theta^k = \theta^0 + k \left[ 2^{(r+3)/2} \gamma^{(r-1)/2} n^{(r+1)/2} \right] \alpha^{r+1} q(r).$$

Then, for all  $k = 0, 1, \dots, K$ , we have

- (a)  $\|f(w^k) - \mu^k e\|_\infty \leq \theta^k \mu^k$ .
- (b)  $w^k \in W$ .
- (c)  $(x^k)^T (z^k) \leq (1 + \theta^k) n \mu^k \leq 2n \mu^k$ .

PROOF. From the definition of  $\theta^k$ ,  $K$ , relation (98) and the fact that  $\gamma = 2/(1 - \theta^0)$ , we have that for all  $k = 1, 2, \dots, K$

$$\begin{aligned}
 (106) \quad \theta^k &\leq \theta^0 + K2^{(r+3)/2}\gamma^{(r-1)/2}n^{(r+1)/2}\alpha^{r+1}q(r) \\
 &= \theta^0 + 2^{(r+3)/2}\gamma^{(r-1)/2}n^{(r+1)/2}\alpha^r q(r) \left[ \ln(2n\epsilon^{-1}\mu^0) \right] \\
 &\leq \theta^0 + \gamma^{-1} \\
 &= \theta^0 + \frac{1 - \theta^0}{2} \\
 &= \frac{1 + \theta^0}{2}.
 \end{aligned}$$

Since  $\theta^0 < 1$ , it follows that  $\theta^k < 1$  for  $k = 1, 2, \dots, K$ . Note that (a) and the fact that  $\theta^k < 1$  immediately imply (c). The proof of (a) and (b) is by induction on  $k$ . Obviously (a) and (b) hold for  $k = 0$  due to (97) and Assumption 4.1. Assume (a) and (b) hold for  $k$ , where  $0 \leq k < K$ . We will show that (a) and (b) hold for  $k + 1$ . If

$$(107) \quad \theta^k + \frac{1 - \theta^k}{1 - \alpha} \sum_{l=r+1}^{2r} \left[ \left( \frac{1 + \theta^k}{1 - \theta^k} \right)^{1/2} n^{1/2}\alpha \right]^l p(l) \leq \theta^{k+1}$$

then, by applying Theorem 4.3 with  $w \equiv w^k$ ,  $\hat{w} \equiv w^{k+1}$ ,  $\mu \equiv \mu^k$  and  $\theta \equiv \theta^k$ , it follows that (a) holds for  $k + 1$  and that (b) also holds for  $k + 1$  since  $\theta^{k+1} < 1$ . Therefore, we only have to show (107) to complete the proof of the corollary. Note that (106) implies that

$$(108) \quad 1 - \theta^k \geq \frac{1 - \theta^0}{2} = \gamma^{-1}.$$

Using relation (98), (108) and the fact that  $(1 + \theta^k) \leq 2$ , one can easily verify that

$$(109) \quad \left[ \left( \frac{1 + \theta^k}{1 - \theta^k} \right)^{1/2} n^{1/2}\alpha \right] < 1.$$

Hence, from the definition of  $q(r)$ ,  $\theta^k$  and the fact that  $\alpha \leq 1/2$ , it follows that

$$\begin{aligned}
 &\theta^k + \frac{1 - \theta^k}{1 - \alpha} \sum_{l=r+1}^{2r} \left[ \left( \frac{1 + \theta^k}{1 - \theta^k} \right)^{1/2} n^{1/2}\alpha \right]^l p(l) \\
 &\leq \theta^k + \frac{1 - \theta^k}{1 - \alpha} \left[ \left( \frac{1 + \theta^k}{1 - \theta^k} \right)^{1/2} n^{1/2}\alpha \right]^{r+1} \sum_{l=r+1}^{2r} p(l) \\
 &\leq \theta^k + \frac{1 + \theta^k}{1 - \alpha} \left( \frac{1 + \theta^k}{1 - \theta^k} \right)^{(r-1)/2} n^{(r+1)/2}\alpha^{r+1}q(r) \\
 &\leq \theta^k + 4(2\gamma)^{(r-1)/2}n^{(r+1)/2}\alpha^{r+1}q(r) \\
 &= \theta^k + 2^{(r+3)/2}\gamma^{(r-1)/2}n^{(r+1)/2}\alpha^{r+1}q(r) \\
 &= \theta^{k+1}
 \end{aligned}$$

where in the third inequality we use the fact that  $(1 + \theta^k) \leq 2$  and  $(1 - \theta^k)^{-1} \leq \gamma$ . This shows (107) and concludes the proof of the corollary. ■

As an immediate consequence of the above theorem, we have the following result.

**COROLLARY 4.5.** *The total number of iterations performed by Algorithm 4.1 is on the order of  $O(\varphi n^{(r+1)/2} [\max(\log n, \log \epsilon^{-1}, \log \mu^0)]^{(1+1/r)})$ , where  $\varphi \equiv f_{\max}(w^0)/f_{\min}(w^0)$ .*

**PROOF.** Let

$$(110) \quad K = \alpha^{-1} \lceil \ln(2n\epsilon^{-1}\mu^0) \rceil.$$

By (c) of Corollary 4.4, it follows that

$$(111) \quad (x^K)^T z^K \leq 2n\mu^K = 2n(1 - \alpha)^K \mu^0 \leq \epsilon,$$

where the last inequality follows from the definition of  $K$ . Hence, Algorithm 4.1 performs no more than  $K$  iterations. Using (96) and the fact that  $\gamma = 2/(1 - \theta^0)$ , it follows that

$$(112) \quad \gamma \leq 2f_{\max}(w^0)/f_{\min}(w^0).$$

By using the last relation, expressions (110), (98) and Lemma 4.2, the corollary follows. ■

Let  $L$  denote the size of the convex quadratic programming problem (P). Then if we set  $\epsilon = 2^{-O(L)}$ , then the observation preceding Theorem 3.4 still holds in the context of convex quadratic programming problems. Using this observation, we can now state the main result of this section, which is a direct consequence of the previous corollary.

**THEOREM 4.6.** *If the initial iterate is such that  $f_{\max}(w^0) = 2^{O(L)}$  and the ratio*

$$(113) \quad f_{\max}(w^0)/f_{\min}(w^0) = O(1)$$

*then Algorithm 4.1 solves the pair of problems (P) and (D) in at most*

$$(114) \quad O(n^{\frac{1}{2}(1+1/r)} L^{(1+1/r)})$$

*iterations, where each iteration involves  $O(n^3 + n^2r)$  arithmetic operations.*

We now turn our effort towards proving Theorem 4.3. The next result generalizes Lemma 3.5.

**LEMMA 4.7.** *Let  $w \equiv (x, y, z) \in W$  and  $\alpha > 0$  be given. Consider the point  $\hat{w} \equiv (\hat{x}, \hat{y}, \hat{z})$  defined as  $\hat{w} \equiv w'(w, \alpha)$ . Then we have:*

$$(115) \quad \hat{x}_i \hat{z}_i = (1 - \alpha)x_i z_i + \sum_{l=r+1}^{2r} (-1)^l \alpha^l \sum_{j=l-r}^r (\Delta^{(j)} x_i) (\Delta^{(l-j)} z_i),$$

$$(116) \quad (\Delta^{(j)} x)^T (\Delta^{(j)} z) \geq 0, \quad j = 1, \dots, r.$$

PROOF. Using (88), we obtain for all  $i = 1, \dots, n$

$$\begin{aligned} \hat{x}_i \hat{z}_i &= \left( x_i + \sum_{j=1}^r (-\alpha)^j \Delta^{(j)} x_i \right) \left( z_i + \sum_{j=1}^r (-\alpha)^j \Delta^{(j)} z_i \right) \\ &= x_i z_i + \sum_{l=1}^r (-\alpha)^l \sum_{j=0}^l \Delta^{(j)} x_i \Delta^{(l-j)} z_i \\ &\quad + \sum_{l=r+1}^{2r} (-\alpha)^l \sum_{j=l-r}^r \Delta^{(j)} x_i \Delta^{(l-j)} z_i \end{aligned}$$

From (85), it follows that

$$(117) \quad \sum_{l=1}^r (-\alpha)^l \sum_{j=0}^l \Delta^{(j)} x_i \Delta^{(l-j)} z_i = -\alpha x_i z_i.$$

Combining the two last expressions, we obtain (115). The proof of (116) is similar to the one given for (55) of Lemma 3.5 and follows from (90)–(91) and the fact that the matrix  $Q$  is positive semidefinite. This completes the proof of the lemma. ■

The next lemma provides some bounds on the scaled directions  $D^{-1} \Delta^{(k)} x$  and  $D \Delta^{(k)} z$ , where  $D = (Z^{-1} X)^{1/2}$ . It is a generalization of Lemma 3.7 and its proof is an application of Lemma 3.6.

LEMMA 4.8. *Let  $w = (x, y, z) \in W$  be given. Consider the directions*

$$(118) \quad \Delta^{(k)} w(w) \equiv (\Delta^{(k)} x, \Delta^{(k)} y, \Delta^{(k)} z)$$

for  $k \geq 1$ . Then we have

$$(119) \quad \max\{\|D^{-1} \Delta^{(k)} x\|, \|D \Delta^{(k)} z\|\} \leq \frac{p(k)(x^T z)^{k/2}}{f_{\min}^{(k-1)/2}}$$

where  $D \equiv (Z^{-1} X)^{1/2}$  and  $p(k)$  is defined as in (99) and (100).

PROOF. The proof is by induction on  $k$ . For  $k = 1$ , it follows that (119) holds by using relations (116), (89), (56) and an argument similar to the proof of Lemma 3.7. Assume (119) holds for all  $j$  with  $1 \leq j < k$ . We will show that (119) holds for  $k$ . By relation (89) and relation (116) of Lemma 4.7, we have

$$(120) \quad D^{-1} \Delta^{(k)} x + D \Delta^{(k)} z = -(XZ)^{-1/2} \sum_{j=1}^{k-1} (\Delta^{(j)} X)(\Delta^{(k-j)} Z) e,$$

$$(\Delta^{(k)} x)^T \Delta^{(k)} z \geq 0.$$

Letting  $r = D^{-1}\Delta^{(k)}x$ ,  $s = D\Delta^{(k)}z$  and  $t = -(XZ)^{-1/2}\sum_{j=1}^{k-1}(\Delta^{(j)}X)(\Delta^{(k-j)}Z)e$ , then, by Lemma 3.6, it follows that

$$(121) \quad \max\{\|r\|, \|s\|\} \leq \|t\|.$$

On the other hand, using the induction hypothesis, we obtain

$$\begin{aligned} \|t\| &= \left\| (XZ)^{-1/2} \sum_{j=1}^{k-1} (\Delta^{(j)}X)(\Delta^{(k-j)}Z)e \right\| \\ &\leq \frac{1}{f_{\min}^{1/2}} \left\| \sum_{j=1}^{k-1} (\Delta^{(j)}X)(\Delta^{(k-j)}Z)e \right\| \\ &= \frac{1}{f_{\min}^{1/2}} \left\| \sum_{j=1}^{k-1} (D^{-1}\Delta^{(j)}X)(D\Delta^{(k-j)}Z)e \right\| \\ &\leq \frac{1}{f_{\min}^{1/2}} \sum_{j=1}^{k-1} \|D^{-1}(\Delta^{(j)}x)\| \|D(\Delta^{(k-j)}z)\| \\ &\leq \frac{1}{f_{\min}^{1/2}} \sum_{j=1}^{k-1} \left( \frac{p(j)(x^Tz)^{j/2}}{f_{\min}^{(j-1)/2}} \right) \left( \frac{p(k-j)(x^Tz)^{(k-j)/2}}{f_{\min}^{(k-j-1)/2}} \right) \\ &= \frac{(x^Tz)^{k/2}}{f_{\min}^{(k-1)/2}} \sum_{j=1}^{k-1} p(j)(k-j) \\ &= \frac{p(k)(x^Tz)^{k/2}}{f_{\min}^{(k-1)/2}}, \end{aligned}$$

where the last equality follows by the definition of  $p(k)$ . The last relation and relation (121) show that (119) holds for  $k$  and this completes the proof of the lemma. ■

We are now ready to give the proof of Theorem 4.3.

**PROOF (THEOREM 4.3).** We first show (a). From relation (115) and the definition of  $\hat{\mu}$ , it follows that

$$(122) \quad \hat{x}_i \hat{z}_i - \hat{\mu} = (1 - \alpha)(x_i z_i - \mu) + \sum_{l=r+1}^{2r} (-\alpha)^l \sum_{j=l-r}^r \Delta^{(j)}x_i \Delta^{(l-j)}z_i.$$

The absolute value of the summation on the right-hand side of the above expression can be bounded with the aid of Lemma 4.8 as follows. Let  $D \equiv (Z^{-1}X)^{1/2}$  and let  $D_{ii}$

denote the  $i$ th diagonal element of the matrix  $D$ . Then

$$(123) \quad \left| \sum_{l=r+1}^{2r} (-\alpha)^l \sum_{j=l-r}^r \Delta^{(j)} x_i \Delta^{(l-j)} z_i \right|$$

$$(124) \quad \leq \sum_{l=r+1}^{2r} \alpha^l \sum_{j=l-r}^r |\Delta^{(j)} x_i| |\Delta^{(l-j)} z_i|$$

$$(125) \quad = \sum_{l=r+1}^{2r} \alpha^l \sum_{j=l-r}^r |D_{ii}^{-1} \Delta^{(j)} x_i| |D_{ii} \Delta^{(l-j)} z_i|$$

$$(126) \quad \leq \sum_{l=r+1}^{2r} \alpha^l \sum_{j=l-r}^r \|D^{-1} \Delta^{(j)} x\| \|D \Delta^{(l-j)} z\|$$

$$(127) \quad \leq \sum_{l=r+1}^{2r} \alpha^l \sum_{j=l-r}^r \left( \frac{p(j)(x^T z)^{j/2}}{f_{\min}^{(j-1)/2}} \right) \left( \frac{p(l-j)(x^T z)^{(l-j)/2}}{f_{\min}^{(l-j-1)/2}} \right)$$

$$(128) \quad = \sum_{l=r+1}^{2r} \frac{\alpha^l (x^T z)^{l/2}}{f_{\min}^{l/2-1}} \sum_{j=l-r}^r p(j) p(l-j)$$

$$(129) \quad \leq \sum_{l=r+1}^{2r} \frac{\alpha^l [(1+\theta)n\mu]^{l/2} l^{-1}}{[(1-\theta)\mu]^{l/2-1}} \sum_{j=1}^{l-1} p(j) p(l-j)$$

$$(130) \quad = (1-\theta) \left[ \sum_{l=r+1}^{2r} \left[ \left( \frac{1+\theta}{1-\theta} \right)^{1/2} n^{1/2} \alpha \right]^l p(l) \right] \mu$$

where the third inequality follows from (119) and the last inequality follows from (103). The last relation, expressions (103) and (122) and the definition of  $\hat{\mu}$  imply (a). The proof of (b) is similar to the proof of (b) of Theorem 3.2. This completes the proof of the theorem. ■

**5. Concluding remarks.** It should be emphasized that the computed upper bound on the number of iterations required by the power series algorithm of §4 decreases with  $r$ , the order of the approximation, according to  $O(n^{\frac{1}{2}(1+1/r)} L^{(1+1/r)})$ . Therefore, as  $r \rightarrow \infty$ , the upper bound converges to  $O(\sqrt{n} L)$ , the upper bound on the number of iterations required by the path following algorithms of group (c) of §1. On the other hand, the work per iteration, namely  $O(n^3 + n^2 r)$  arithmetic operations, increases with  $r$ . When  $r = O(n)$  we still obtain  $O(n^3)$  arithmetic operations per iteration, which is the work per iteration required by all interior point based algorithms if no rank-one update trick is used [15].

The main purpose of this paper was to present a theoretical result. However, based on the good performance of both the primal affine [31] and dual affine scaling algorithms [1], [20], [23], we feel that the primal-dual affine scaling algorithm has the potential of becoming a competitive algorithm. For a practical implementation some modifications are required, such as: (1) introducing a larger step size computed by means of a ratio test in the first order approximation or by means of a binary search in the higher order approximation algorithms; (2) determining an appropriate starting

artificial problem that gives a good initial starting point; and (3) making a good choice of  $r$ .

Note that when  $r = 1$ , the primal-dual affine scaling algorithm described in §3 can be viewed as a simultaneous application of an affine scaling algorithm to the primal and dual problems, which implies that both the primal and dual objective functions monotonically approach the optimal value. For a practical implementation, this suggests that two ratio tests performed independently in the primal and the dual spaces respectively, might outperform one ratio test done simultaneously in the primal-dual space, since a larger decrease in the duality gap would be obtained. On the other hand, the last strategy would be more conservative in the sense that it would keep the iterates from coming too close to the boundary of the primal-dual feasible region.

### References

- [1] Adler, I., Karmarkar, N., Resende, M. G. C. and Veiga, G. (1989). An Implementation of Karmarkar's Algorithm for Linear Programming. *Math. Programming* **44**.
- [2] \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ (1989). Data Structures and Programming Techniques for the Implementation of Karmarkar's Algorithm. *ORSA Journal on Computing* **1** 84–106.
- [3] Anstreicher, K. M. (1986). A Monotonic Projective Algorithm for Fractional Linear Programming. *Algorithmica* **1** 483–498.
- [4] \_\_\_\_\_ (1989). A Combined 'Phase I-Phase II' Projective Algorithm for Linear Programming. *Math. Programming* **43** 209–223.
- [5] Barnes, E. R. (1986). A Variation on Karmarkar's Algorithm for Solving Linear Programming Problems. *Math. Programming* **36** 174–182.
- [6] Bayer, D. and Lagarias, J. C. (1989). The Nonlinear Geometry of Linear Programming. I. Affine and Projective Scaling Trajectories. *Trans. Amer. Math. Soc.* **314** 499–526.
- [7] \_\_\_\_\_ and \_\_\_\_\_ (1989). The Nonlinear Geometry of Linear Programming. II. Legendre Transform Coordinates and Central Trajectories. *Trans. Amer. Math. Soc.* **314** 527–581.
- [8] de Ghellinck, G. and Vial, J.-F. (1986). A Polynomial Newton Method for Linear Programming. *Algorithmica* **1** 425–454.
- [9] Dikin, I. I. (1967). Iterative Solution of Problems of Linear and Quadratic Programming. *Soviet Math. Dokl.* **8** 674–675.
- [10] \_\_\_\_\_ (1974). On the Speed of an Iterative Process. *Upravlyaemye Sistemi* **12** 54–60. In Russian.
- [11] Fiacco, A. V. and McCormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, New York.
- [12] Frisch, K. R. (1955). The Logarithmic Potential Method of Convex Programming. Technical report, University Institute of Economics, Oslo, Norway.
- [13] Gonzaga, C. C. (1989). An Algorithm for Solving Linear Programming in  $O(n^3L)$  Operations. In Nimrod Megiddo, editor, *Progress in mathematical programming—Interior-point and related methods*. Springer-Verlag, 1–28.
- [14] \_\_\_\_\_ (1989). An Conical Projection Algorithm for Linear Programming. *Math. Programming* **43** 151–173.
- [15] Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica* **4** 373–395.
- [16] \_\_\_\_\_ (1984). Talk given at the University of California, Berkeley.
- [17] \_\_\_\_\_ and Lagarias, J., Slutsman, L., and Wang, P. (1989). Power Series Variants of Karmarkar-type Algorithms. *AT & T Technical Journal* **68**(3) 20–36.
- [18] Kojima, M., Mizuno, S. and Yoshise, A. (1989). A Polynomial-Time Algorithm for a Class of Linear Complementarity Problems. *Mathematical Programming* **44** 1–26.
- [19] \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_ (1989). A Primal-Dual Interior Point Algorithm for Linear Programming. In Nimrod Megiddo, editor, *Progress in mathematical programming—Interior-point and related methods*. Springer-Verlag, 29–48.
- [20] Marsten, R. (1987). The IMP System on Personal Computers. Presentation given at the ORSA/TIMS Joint National Meeting, St. Louis, MO.
- [21] Megiddo, N. (1989). Pathways to the Optimal Set in Linear Programming. In Nimrod Megiddo, editor, *Progress in mathematical programming—Interior-point and related methods*. Springer-Verlag, 131–158.
- [22] \_\_\_\_\_ and Shub, M. (1989). Boundary Behaviour of Interior Point Algorithms in Linear Programming. *Mathematics of Operations Research* **14** 97–146.
- [23] Monma, C. L. and Morton, A. J. (1987). Computational Experimental with a Dual Affine Variant of Karmarkar's Method for Linear Programming. *Operations Research Letters* **6** 261–267.

- [24] Monteiro, R. D. C. and Adler, I. (1987). An Extension of Karmarkar Type Algorithm to a Class of Convex Separable Programming Problems with Global Linear Rate of Convergence. Technical Report ESRC 87-4, Engineering Systems Research Center, University of California, Berkeley, CA 94720, 1987. To appear in *Mathematics of Operations Research*.
- [25] \_\_\_\_\_ and \_\_\_\_\_ (1989). Interior Path Following Primal-Dual Algorithms—Part I. Linear Programming. *Math. Programming* **44** 27–42.
- [26] \_\_\_\_\_ and \_\_\_\_\_ (1989). Interior Path Following Primal-Dual Algorithms—Part II: Convex Quadratic Programming. *Math. Programming* **44** 43–66.
- [27] Papadimitriou, C. R. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs, NJ.
- [28] Renegar, J. (1986). A Polynomial-Time Algorithm Based on Newton's Method for Linear Programming. Technical report MSRI 07 118–86, Mathematical Sciences Research Institute, Berkeley, CA. To appear in *Math. Programming* **40**.
- [29] Todd, M. J. and Burrell, B. P. (1986). An Extension to Karmarkar's Algorithm for Linear Programming Using Dual Variables. *Algorithmica* **1** 409–424.
- [30] Tomlin, J. A. (1985). An Experimental Approach to Karmarkar's Projective Method for Linear Programming. Technical report, Ketron, Inc., Mountain View, CA.
- [31] \_\_\_\_\_ and Welch, J. S. (1986). Implementing an Interior Point Method in a Mathematical Programming System. Technical report, Ketron, Inc., Mountain View, CA.
- [32] Vaidya, P. M. (1987). An Algorithm for Linear Programming Which Requires  $O(((m+n)n^2 + (m+n)^{1.5})L)$  Arithmetic Operations. Technical report, AT & T Bell Laboratories, Murray Hill, NJ, 1987. To appear in *Mathematical Programming*.
- [33] Vanderbei, R. J., Meketon, M. S. and Freedman, B. A. (1986). A Modification of Karmarkar's Linear Programming Algorithm. *Algorithmica* **1** 395–407.
- [34] Ye, Y. and Kojima, M. (1987). Recovering Optimal Dual Solutions in Karmark's Polynomial Algorithm for Linear Programming. *Math. Programming* **39** 305–317.

MONTEIRO: AT & T BELL LABORATORIES, HOLMDEL, NEW JERSEY 07733.

ADLER: DEPARTMENT OF INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH,  
UNIVERSITY OF CALIFORNIA, BERKELEY, CALIFORNIA 94720

RESENDE: AT & T BELL LABORATORIES, MURRAY HILL, NEW JERSEY 07974