Mathematical Programming 44 (1989) 27-41
North-Holland

INTERIOR PATH FOLLOWING PRIMAL-DUAL ALGORITHMS. PART I: LINEAR PROGRAMMING

Renato D.C. MONTEIRO and Ilan ADLER

Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, USA

Received 3 August 1987
Revised manuscript received 18 April 1988

We describe a primal-dual interior point algorithm for linear programming problems which requires a total of $O(\sqrt{n}L)$ number of iterations, where L is the input size. Each iteration updates a penalty parameter and finds the Newton direction associated with the Karush-Kuhn-Tucker system of equations which characterizes a solution of the logarithmic barrier function problem. The algorithm is based on the path following idea.

Key words: Interior-point methods, linear programming, Karmarkar's algorithm, polynomial-time algorithms, logarithmic barrier function, path following.

1. Introduction

Consider the linear programming problem

$$(\mathrm{P}) \qquad \begin{aligned} \min \quad & c^{\mathrm{T}}x \\ \text{s.t.} \quad & Ax = b, \\ & x \geqslant 0, \end{aligned}$$

where A is an $m \times n$-matrix. Assume that the data A, b and c are integer, and the input size is L. This paper presents an algorithm for linear programming (LP) problems based on the logarithmic barrier function method and on the idea of following 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 [3] and is formally studied in Fiacco and McCormick [2] in the context of nonlinear optimization. The introduction of the new interior point algorithm by Karmarkar, in his seminal paper [6], led researchers to reconsider the application of the logarithmic barrier function method to LP problems. Recently, this method was first considered by Gill et al. [4] to develop a projected Newton barrier method for solving LP problems. The presentation of a continuous trajectory of the iterative Karmarkar method was first described by Karmarkar [7] and extensively studied by Bayer and Lagarias [1]. Megiddo [9] related this path to the classical barrier path in the framework of complementarity relationship between the primal and dual linear programming problems. Kojima, Mizuno and Yoshise

[8], used this framework to present a primal-dual algorithm that traces the central path. Their algorithm is shown to converge in at most $O(nL)$ iterations with a computational effort of $O(n^3)$ arithmetic operations per iteration, resulting in a total of $O(n^4L)$ arithmetic operations.

In this paper, we build on the ideas in [8] and [9], and obtain a faster algorithm. The directions generated by our algorithm are the same as the directions generated by the algorithm presented in [8]. However, working closer to the central path, as defined and developed in [9], we are able to obtain convergence in at most $O(\sqrt{n}L)$ iterations. Each iteration involves the inversion of a $n \times n$ matrix which can be done in at most $O(n^3)$ arithmetic operations. Based on ideas presented in [6], one can approximate the matrix to be inverted so that each iteration can be executed, by way of rank-one updates, in an average of $O(n^{2.5})$ arithmetic operations, as will be described in Part II. Thus overall our algorithm requires $O(n^3L)$ arithmetic operations. It should be noted that the breakthrough in this line of research was obtained by Renegar [11], who was the first to achieve a speed of convergence of $O(\sqrt{n}L)$ iterations, where each iteration involves $O(n^3)$ arithmetic operations. His algorithm is based on the method of centers following the central trajectory. Subsequently, Vaidya [12] improved Renegar's complexity to a total of $O(n^3L)$ arithmetic operations using the same approach of the method of centers and the updating scheme described in [6]. Independently, an equivalent complexity was also obtained by Gonzaga [5], using the logarithmic barrier function approach. Both Vaidya's and Gonzaga's algorithms are primal algorithms. It should be noted that in order to simplify the complexity analysis presentation, we assume throughout the paper that $m = O(n)$.

In Part II, we extend our results to introduce a primal-dual algorithm that solves convex quadratic programming problems in $O(\sqrt{n}L)$ iterations with a total of $O(n^3L)$ arithmetic operations. In order to emphasize the simplicity of the basic ideas underlying our approach, we choose to defer some of the details of the proofs and the speedup resulting from the rank-one updates of the matrix to be inverted in each iteration to the second part of the paper.

Our paper is organized as follows. In Section 2, we present some theoretical background. In Section 3, we present the algorithm. In Section 4, we prove results related to the convergence properties of the algorithm. In Section 5, we discuss the initialization of the algorithm. Finally, in Section 6, we conclude the paper with some remarks.

2. Theoretical background

In order to facilitate the reading of this paper, we use a notation roughly similar to the one in [8]. A discussion of the main results necessary to motivate the development of our algorithm is presented in this section. A detailed discussion of these results can be found in [9].

We consider the pair of the standard form linear program and its dual

$$\text{(P)} \qquad \begin{aligned} \min \quad & c^{\mathrm{T}}x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned}$$

$$\text{(D)} \qquad \begin{aligned} \max \quad & b^{\mathrm{T}}y \\ \text{s.t.} \quad & A^{\mathrm{T}}y + z = c, \\ & z \geq 0, \end{aligned}$$

where A is an $m \times n$-matrix and b, c are vectors of length m and n respectively. We assume that the entries of the vectors b, c and the matrix A are integral. The algorithm we consider in this paper is motivated by the application of the logarithmic barrier function technique to problem (P). The logarithmic barrier function method consists of examining the family of problems

$$(\mathrm{P}_{\mu}) \qquad \begin{aligned} \min \quad & c^{\mathrm{T}}x - \mu \sum_{j=1}^{n} \ln x_j \\ \text{s.t.} \quad & Ax = b, \\ & x > 0, \end{aligned}$$

where $\mu > 0$ is the barrier penalty parameter. This technique is well-known in the context of general constrained optimization problems. One solves the problem penalized by the logarithmic barrier function term for several values of the parameter μ, with μ decreasing to zero, and the result is a sequence of feasible points converging to a solution of the original problem.

Before we can apply the logarithmic barrier function method, some assumptions on the problems (P) and (D) are necessary. We impose the following assumptions:

Assumption 2.1. (a) The set $S \equiv \{x \in \mathbb{R}^n;\ Ax = b,\ x > 0\}$ is non-empty.

(b) The set $T \equiv \{(y, z) \in \mathbb{R}^m \times \mathbb{R}^n;\ A^{\mathrm{T}}y + z = c,\ z > 0\}$ is non-empty.

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

We say that points in the sets S and T are interior feasible solutions of problems (P) and (D) respectively. The need for (a) is evident since the logarithmic barrier function method is always applied in the interior of the set defined by the inequality constraints. Assumptions (b) and (c) are also necessary as will become clear from the discussion that follows.

Throughout this paper, we use the following notation. A point $(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ will be denoted by the lower case letter w, that is, $w \equiv (x, y, z)$. If x is a lower case letter that denotes a vector $x = (x_1, \ldots, x_n)^{\mathrm{T}}$, then a capital letter will denote the diagonal matrix with the components of the vector on the diagonal, i.e., $X = \operatorname{diag}(x_1, \ldots, x_n)$. Given a real number $a > 0$, we denote its logarithm to the natural base and to the base 2 by ln a and log a respectively. Also, W will denote the set defined as follows:

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

Observe that the objective function of problem (P_μ) is a strictly convex function. This implies that problem (P_μ) has at most one global minimum, and that this global minimum, if it exists, is completely characterized by the Karush–Kuhn–Tucker stationary condition:

$$c - \mu X^{-1} e - A^{\mathrm{T}} y = 0,$$
$$Ax = b, \quad x > 0,$$

where e denotes the n-vector of all ones and y is the vector of Lagrangian multiplier associated with the equality constraints of problem (P_μ). By introducing the n-vector z, this system can be rewritten in an equivalent way as

$$\begin{aligned} &ZXe - \mu e = 0, \\ &Ax - b = 0, \quad x > 0, \\ &A^{\mathrm{T}} y + z - c = 0. \end{aligned} \tag{2.1}$$

A necessary and sufficient condition for problem (P_μ) to have an optimal solution for all $\mu > 0$ is given by the following result.

Proposition 2.1. *Assume Assumption* 2.1(a) *holds and let $\mu > 0$ be given. Then problem* (P_μ) *has an optimal solution if and only if the set of optimal solutions of problem* (P) *is non-empty and bounded.*

A proof of Proposition 2.1 can be found in [9] (see also [2]). From this result, we immediately conclude that if (P_μ) has a solution for some $\mu > 0$ then it has a solution for all $\mu > 0$. The role played by Assumption 2.1(b) is now provided by the following result.

Proposition 2.2. *Assume that problem* (P) *is feasible. Then the set of optimal solutions of problem* (P) *is non-empty and bounded if and only if Assumption* 2.1(b) *holds, that is, the set of interior feasible solutions of the dual problem* (D) *is non-empty.*

The proof of Proposition 2.2 is an application of duality theory for linear programming. As a consequence of the two previous propositions, we have the following corollary.

Corollary 2.1. *Under Assumptions* 2.1(a) *and* (b), *problem* (P_μ) *(and consequently system* (2.1)) *has a unique solution $x(\mu)$, for all $\mu > 0$.*

The Karush–Kuhn–Tucker system (2.1) provides important information which we now point out. Assume that $\mu > 0$ is fixed in system (2.1). Since $x > 0$, the first equation in system (2.1) implies that $z>0$. The third equation in (2.1) then implies that the point (y, z) is an interior feasible solution for the dual problem (D). From Assumption 2.1(c), it follows that there is a unique y satisfying (2.1). We denote the unique point (x, y, z) that satisfies (2.1) by $w(\mu)=(x(\mu), y(\mu), z(\mu))$. Obviously $w(\mu)\in W$. The duality gap at point $w\in W$ is by definition given by

$$g(w)\equiv c^{\mathrm{T}}x-b^{\mathrm{T}}y.$$

Using the two last equations in (2.1), one can easily verify that

$$g(w)=x^{\mathrm{T}}z, \quad w\in W. \tag{2.2}$$

In view of the above relation, we will always refer to the duality gap as the quantity $x^{\mathrm{T}}z$ instead of the usual one $c^{\mathrm{T}}x-b^{\mathrm{T}}y$. In particular, using the first equation in (2.1), we obtain $g(w(\mu))=n\mu$, for all μ, and therefore $g(w(\mu))$ converges to zero as μ goes to zero. This implies that $c^{\mathrm{T}}x(\mu)$ and $b^{\mathrm{T}}y(\mu)$ converge to the common optimal value of problems (P) and (D) respectively. In fact, we have the following stronger result (cf. [9]).

Proposition 2.3. *Under Assumptions* 2.1(a), (b) *and* (c), *as* $\mu\to 0$, $x(\mu)$ *and* $(y(\mu), z(\mu))$ *converge to optimal solutions of problems* (P) *and* (D) *respectively.*

The following notation will be useful later. Let $w=(x, y, z)\in W$. We denote by $f(w)=(f_1(w),\ldots,f_n(w))^{\mathrm{T}}\in\mathbb{R}^n$ the n-vector defined by

$$f_i(w)=x_iz_i, \quad i=1,\ldots,n.$$

With this notation, the first equation of (2.1) written coordinate-wise becomes:

$$f_i(w(\mu))\equiv x_i(\mu)z_i(\mu)=\mu, \quad i=1,\ldots,n.$$

We denote by Γ the set (or path) of solutions $w(\mu)$, $\mu>0$ for system (2.1), i.e.,

$$\Gamma\equiv\{w(\mu)\equiv(x(\mu), y(\mu), z(\mu));\ \mu>0\}.$$

The path Γ is usually referred to as the central path associated with the LP problem (P). The algorithm which will be presented in the next section is based on the idea of following the central path Γ closely with the objective of approaching the desired solutions of problems (P) and (D). The path Γ will serve as a criterion to guide the points generated by the algorithm.

3. The algorithm

As in the previous section, we denote a point $(x, y, z)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^n$ by the lower case letter w. The algorithm generates a sequence of points $w^k\in W$, $k=0,1,2,\ldots,$ where the initial point w^0 is provided as input to the algorithm. We require that the initial point $w^0\in W$ be a point satisfying some criterion of closeness with respect to the central path Γ. Given an LP problem in standard form, in Section 5 we show how to construct an augmented LP problem so that Assumption 2.1 is satisfied. As a consequence of this construction, we also show how to obtain an initial point $w^0 \in W$ satisfying the criterion of closeness.

Given a current iterate $(x, y, z) \in W$, a vector of directions $\Delta w \equiv (\Delta x, \Delta y, \Delta z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ needs to be generated for the determination of the next iterate. Let $(\hat{x}, \hat{y}, \hat{z})$ denote the next iterate. We obtain $(\hat{x}, \hat{y}, \hat{z})$ by setting $\hat{x} \equiv x - \Delta x$, $\hat{y} \equiv y - \Delta y$ and $\hat{z} \equiv z - \Delta z$ or in more compact notation, $\hat{w} \equiv w - \Delta w$.

According to [8], the direction Δw chosen to generate the next iterate $\hat{w}$ is defined as the Newton direction associated with the Karush–Kuhn–Tucker system of equations (2.1). If we denote the left hand side of the system of equations (2.1) by $H(w) \equiv H(x, y, z)$, the Newton direction Δw at the point $w \in W$ is determined by the following system of linear equations:

$$D_w H(w) \Delta w = H(w)$$

where $D_w H(w)$ denotes the Jacobian of H at $w \equiv (x, y, z)$. More specifically, the Jacobian of H at w is given by

$$D_w H(w) = \begin{bmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^{\mathrm{T}} & I \end{bmatrix}$$

and the Newton direction $\Delta w = (\Delta x, \Delta y, \Delta z)$ is given by the following system of linear equations:

$$Z\Delta x + X\Delta z = XZe - \hat{\mu} e, \tag{3.1a}$$

$$A\Delta x = 0, \tag{3.1b}$$

$$A^{\mathrm{T}}\Delta y + \Delta z = 0, \tag{3.1c}$$

where $\hat{\mu} > 0$ is some prespecified penalty parameter. Note that the solution $\Delta w = (\Delta x, \Delta y, \Delta z)$ of the system of equations (3.1) clearly depends on the current iterate $w = (x, y, z)$ and on the penalty parameter $\hat{\mu} > 0$. In order to indicate this dependence, we denote the solution of system (3.1) by $\Delta w(w, \hat{\mu})$.

By simple calculation, we obtain the following expressions for Δx, Δy, Δz:

$$\Delta x = [Z^{-1} - Z^{-1}XA^{\mathrm{T}}(AZ^{-1}XA^{\mathrm{T}})^{-1}AZ^{-1}](XZe - \hat{\mu} e),$$

$$\Delta y = -[(AZ^{-1}XA^{\mathrm{T}})^{-1}AZ^{-1}](XZe - \hat{\mu} e),$$

$$\Delta z = [A^{\mathrm{T}}(AZ^{-1}XA^{\mathrm{T}})^{-1}AZ^{-1}](XZe - \hat{\mu} e).$$

Therefore, to calculate the direction $\Delta w \equiv (\Delta x, \Delta y, \Delta z)$, the inverse of the matrix $(AZ^{-1}XA^{\mathrm{T}})$ needs to be calculated. Observe that all the other operations involved in the computation of $\Delta w \equiv \Delta w(w, \hat{\mu})$ are of the order of $\mathrm{O}(n^2)$ arithmetic operations.

We are now ready to describe the algorithm. Let θ and δ be constants satisfying

$$0 \leq \theta < \tfrac{1}{2}, \quad 0 < \delta < \sqrt{n}, \tag{3.2a}$$

$$\frac{\theta^2 + \delta^2}{2(1-\theta)} \leq \theta(1 - \delta/\sqrt{n}), \tag{3.2b}$$

where n is the number of columns of the constraint matrix A. One possible set of values for the constants satisfying (3.2) is $\theta = \delta = 0.35$. At the beginning of the algorithm, we assume that an initial point $w^0 \equiv (x^0, y^0, z^0) \in W$ is available such that the following criterion of closeness with respect to the central path Γ is satisfied:

$$\|\mathrm{f}(w^0) - \mu_0 e\| \leq \theta\mu_0 \tag{3.3}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\mu_0 = (x^0)^{\mathrm{T}} z^0 / n$.

We now state the algorithm.

Algorithm 3.1.

Step 0: Assume θ and δ are fixed constants satisfying (3.2). Let $w^0 \in W$ satisfy (3.3). Let $\varepsilon > 0$ be a tolerance for the duality gap.
Set $k := 0$.

Step 1: If $(x^k)^{\mathrm{T}} z^k \leq \varepsilon$, stop.

Step 2: Set $\mu_{k+1} := \mu_k(1 - \delta/\sqrt{n})$.
Calculate $\Delta w^k \equiv \Delta w(w^k, \mu_{k+1})$.

Step 3: Set $w^{k+1} := w^k - \Delta w^k$.
Set $k := k+1$ and go to Step 1.

In the next section, we prove that all points generated by the algorithm above lie in the set W and that they remain close to the central path Γ in a sense to be described later. We also show that Algorithm 3.1 terminates in at most $\mathrm{O}(\sqrt{n} \max(\log \varepsilon^{-1}, \log n, \log \mu_0))$ iterations. This fact will enable us to show that Algorithm 3.1 performs no more than $\mathrm{O}(n^{3.5} \max(\log \varepsilon^{-1}, \log n, \log \mu_0))$ arithmetic operations until its termination.

4. Convergence results

We begin this section by stating the main result and its consequences. The main result is stated as follows.

Theorem 4.1. *Let θ and δ be constants satisfying relations* (3.2). *Assume that $w = (x, y, z) \in W$ satisfies*

$$\|f(w) - \mu e\| \leq \theta\mu \tag{4.1}$$

where $\mu = x^{\mathrm{T}} z / n$ and $\|\cdot\|$ denotes the Euclidean norm. Let $\hat{\mu} > 0$ be defined as

$$\hat{\mu} = \mu(1 - \delta/\sqrt{n}). \tag{4.2}$$

Consider the point $\hat{w} \equiv (\hat{x}, \hat{y}, \hat{z}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ *given by* $\hat{w} \equiv w - \Delta w$, *where* $\Delta w \equiv \Delta w(w, \hat{\mu})$. *Then, we have*

(a) $\|f(\hat{w}) - \hat{\mu} e\| \leq \theta \hat{\mu}$,

(b) $\hat{w} \in W$,

(c) $g(\hat{w}) \equiv \hat{x}^{\mathrm{T}} \hat{z} = n\hat{\mu}$.

The proof of Theorem 4.1 will be given at the end of this section. We now give an intuitive interpretation of the measure of closeness (4.1). We can view $f(w)$ not only as denoting the vector XZe but also as a map from $\mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ into $\mathbb{R}^n$ mapping $w = (x, y, z)$ into the "complementary" vector XZe. Under this map, the central path Γ is mapped into the "diagonal" line $f(\Gamma) = \{\mu e;\ \mu > 0\}$. Also, if we let $w^* = (x^*, y^*, z^*)$ denote the optimal solution which $w(\mu)$ converges to, as μ approaches zero, then w^* is mapped by f into the origin of $\mathbb{R}^n$ (cf. Proposition 2.3). The measure of closeness has its natural interpretation when viewed in the "complementarity" space $\mathbb{R}^n$. The image under f of the set of points w for which the measure of closeness is satisfied for some $\mu > 0$, is a "spherical" cone having as central axis the "diagonal" line $f(\Gamma)$ and having as extreme point the origin of $\mathbb{R}^n$. Also, the angle formed by the central axis with all the extreme rays of the cone is constant and depends on θ.

Note that $(x^{\mathrm{T}} z / n) e$ is the closest point to $f(w)$, $w \in W$, lying in the "diagonal" line $f(\Gamma)$. Hence, $\|f(w) - \mu e\|$ measures the Euclidean distance between $f(w)$ and the line $f(\Gamma)$. So, (4.1) can be interpreted as keeping the distance between the images of w and Γ, under f, relatively small.

As a consequence of Theorem 4.1, we have the following result:

Corollary 4.1. *The sequence of points* (w^k) *generated by Algorithm* 3.1 *satisfies*

(a) $\|f(w^k) - \mu_k e\| \leq \theta \mu_k$, *for all* $k = 1, 2, \ldots$,

(b) $w^k \in W$, *for all* $k = 1, 2, \ldots$,

(c) $g(w^k) = (x^k)^{\mathrm{T}} z^k = n\mu_k$, *for all* $k = 1, 2, \ldots$,

where $\mu_k \equiv \mu_0 (1 - \delta/\sqrt{n})^k$ *for* $k = 1, 2, \ldots$.

Proof. This result follows trivially by arguing inductively and using Theorem 4.1. □

Viewed with respect to the "complementarity space", Corollary 4.1 says that, all iterates generated by Algorithm 3.1 will lie within the cone and will get closer to the extreme point of the cone, i.e. the origin of $\mathbb{R}^n$, by a factor of $(1 - \delta/\sqrt{n})$ at each iteration. Therefore we can view Algorithm 3.1 as a path-following procedure, i.e., the sequence of iterates (w^k) attempts to trace the central path Γ so that it eventually converges to an optimal solution $w^* = (x^*, y^*, z^*)$ for the pair of LP problems (P) and (D).

We now derive an upper bound on the total number of iterations performed by Algorithm 3.1.

Proposition 4.1. *The total number of iterations performed by Algorithm* 3.1 *is no greater than* $\hat{k} \equiv \lceil \ln(n\varepsilon^{-1}\mu_0)\sqrt{n}/\delta \rceil$ *where* $\varepsilon > 0$ *denotes the tolerance for the duality gap and* $\mu_0 \equiv (x^0)^{\mathrm{T}} z^0/n$ *is the initial penalty parameter.*

Proof. From (c) of Corollary 4.1, the algorithm terminates whenever $n\mu_k \leqslant \varepsilon$. Thus, it is enough to show that $\hat{k}$ satisfies this inequality. By the definition of $\hat{k}$, we have

$$\begin{aligned}
\ln \varepsilon &\geqslant -\frac{\hat{k}\delta}{\sqrt{n}} + \ln(n\mu_0) \\
&\geqslant \hat{k} \ln\left(1 - \frac{\delta}{\sqrt{n}}\right) + \ln(n\mu_0) \\
&= \ln\left[n\mu_0 \left(1 - \frac{\delta}{\sqrt{n}}\right)^{\hat{k}} \right] \\
&= \ln n\mu_{\hat{k}}.
\end{aligned}$$

The second inequality is due to the fact that $\ln(1-x) \leqslant -x$ for all $x < 1$ and the last equality follows from the definition of μ_k. Therefore $\hat{k}$ satisfies $n\mu_{\hat{k}} \leqslant \varepsilon$ and this completes the proof of the proposition. $\square$

Define the size $L(A, b, c)$ of a linear programming problem in the standard form (P) as

$$\begin{aligned}
L(A, b, c) = {} & \left\lceil \log \left(\begin{matrix} \text{largest absolute value of the determinant} \\ \text{of any square submatrix of } A \end{matrix} + 1 \right) \right\rceil \\
& + \lceil \log(1 + \max_j |c_j|) \rceil + \lceil \log(1 + \max_i |b_i|) \rceil + \lceil \log(m+n) \rceil.
\end{aligned}$$

It is a well-known fact that if, at some iteration k, we have a point $w^k \in W$ such that the duality gap satisfies $(x^k)^{\mathrm{T}} z^k \leqslant 2^{-\mathrm{O}(L)}$, then from w^k, we may obtain optimal solutions for problems (P) and (D) in $\mathrm{O}(m^2 n)$ arithmetic operations (cf. [10]). Using this observation, we obtain

Corollary 4.2. *If the initial penalty parameter* μ_0 *satisfies* $\mu_0 = 2^{\mathrm{O}(L)}$ *then Algorithm* 3.1 *solves the pair of* LP *problems* (P) *and* (D) *in at most* $\mathrm{O}(\sqrt{n}L)$ *iterations.*

Proof. Follows directly from the previous proposition. $\square$

In Section 5, we will see that the initial point w^0 can be chosen in such a way that $\mu_0 \equiv (x^0)^{\mathrm{T}} z^0/n$ satisfies $\mu_0 = 2^{\mathrm{O}(L)}$. As a consequence of the previous result, we have:

Corollary 4.3. *Algorithm* 3.1 *solves the pair of* LP *problems* (P) *and* (D) *in no more than* $\mathrm{O}(n^{3.5}L)$ *arithmetic operations.*

Proof. At every iteration, the computational effort is dominated by the calculation of the inverse of the matrix $A(Z^k)^{-1}X^kA^{\mathrm{T}}$ which requires $\mathrm{O}(n^3)$ arithmetic operations. By Corollary 4.2, Algorithm 3.1 terminates in at most $O(\sqrt{n}L)$ iterations. These two observations immediately conclude the proof of the corollary. $\square$

In Part II of this paper, we will see that Algorithm 3.1 can be modified to yield an algorithm which does not only solve linear programming problems but also convex quadratic programming problems in at most $O(n^3L)$ arithmetic operations. This reduction in the complexity will be achieved by introducing a suitable approximation of the inverse of the matrix involved in the computation of the directions generated by Algorithm 3.1.

We now concentrate our effort towards proving Theorem 4.1. Let $w = (x, y, z) \in W$ and $\hat{\mu} > 0$. Let $\Delta w = (\Delta x, \Delta y, \Delta z)$ be the direction $\Delta w(w, \hat{\mu})$. Let $\hat{w} = w - \Delta w$. The next result provides expressions for the product of complementary variables $f_i(\hat{w})$, $i = 1, \ldots, n$ and the duality gap $g(\hat{w}) \equiv \hat{x}^{\mathrm{T}}\hat{z}$.

Proposition 4.2. *Let w, $\hat{\mu}$, Δw and $\hat{w}$ be as above. Then, we have*

$$f_i(\hat{w}) = \hat{\mu} + \Delta x_i \Delta z_i, \tag{4.3}$$

$$(\Delta x)^{\mathrm{T}}(\Delta z) = 0, \tag{4.4}$$

$$g(\hat{w}) = n\hat{\mu}. \tag{4.5}$$

Proof. By definition, we have for all $i = 1, \ldots, n$,

$$\begin{aligned} f_i(\hat{w}) &\equiv \hat{x}_i \hat{z}_i \\ &= (x_i - \Delta x_i)(z_i - \Delta z_i) \\ &= x_i z_i - (x_i \Delta z_i + z_i \Delta x_i) + \Delta x_i \Delta z_i \\ &= \hat{\mu} + \Delta x_i \Delta z_i \end{aligned}$$

where the last equality follows from (3.1a). This shows (4.3). Multiplying (3.1b) and (3.1c) on the left by $(\Delta y)^{\mathrm{T}}$ and $(\Delta x)^{\mathrm{T}}$ respectively, and combining, we obtain (4.4). Recall from (2.2) that $g(\hat{w}) = \sum_{i=1}^{n} f_i(\hat{w})$. Summing expression (4.3) over all indices $i = 1, \ldots, n$ and noting (4.4), we obtain (4.5). This completes the proof of the proposition. $\square$

We now state and prove some preliminary results that will be useful in the proof of Theorem 4.1.

Lemma 4.1. *Let r, s and t be real n-vectors satisfying $r + s = t$ and $r^{\mathrm{T}}s \geq 0$. Then, we have*

$$\|RSe\| \leq \frac{\|t\|^2}{2}$$

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

Proof. Using the assumptions, we obtain

$$\|r\|^2+\|s\|^2\leq\|r\|^2+\|s\|^2+2r^{\mathrm{T}}s=\|r+s\|^2=\|t\|^2$$

which implies that $\|t\|^2/2\geq\|r\|\,\|s\|\geq\|RSe\|$, where the last inequality follows by using the definition of the Euclidean norm. This completes the proof of the lemma. □

Let $w=(x,y,z)\in W$ and $\hat{\mu}>0$. Let $\Delta w=(\Delta x,\Delta y,\Delta z)$ be the direction $\Delta w(w,\hat{\mu})$. Let ΔX and ΔZ denote the diagonal matrices corresponding to the vectors Δx and Δz. Consider the vector $\Delta f\in\mathbb{R}^n$ defined as $\Delta f\equiv(\Delta X)(\Delta Z)e$. The next result provides an upper bound on the Euclidean norm of the vector Δf.

Lemma 4.2. *Let Δf be defined as above. Then, we have*

$$\|\Delta f\|\leq\frac{\|f(w)-\hat{\mu}e\|^2}{f_{\min}}$$

where $f_{\min}\equiv\min\{x_iz_i\,;\ i=1,\ldots,n\}$.

Proof. Let $D\equiv(Z^{-1}X)^{1/2}$. Multiplying both sides of equation (3.1a) by $(XZ)^{-1/2}$, we obtain

$$D^{-1}\Delta x+D\Delta z=(XZ)^{-1/2}(f(w)-\hat{\mu}e).$$

Note also that from (4.4), we have $(D^{-1}\Delta x)^{\mathrm{T}}(D\Delta z)=0$. In view of these two relations, we can apply Lemma 4.1 with $r=D^{-1}\Delta x$, $s=D\Delta z$ and $t=(XZ)^{-1/2}(f(w)-\hat{\mu}e)$ to obtain

$$\begin{aligned}\|\Delta f\|&\equiv\|(\Delta X)(\Delta Z)e\|\\&=\|(D^{-1}\Delta X)(D\Delta Z)e\|\\&\leq\tfrac{1}{2}\|(XZ)^{-1/2}(f(w)-\hat{\mu}e)\|^2\\&=\frac{1}{2}\sum_{i=1}^{n}\frac{(f_i(w)-\hat{\mu})^2}{x_iz_i}\\&\leq\frac{\|f(w)-\hat{\mu}e\|^2}{2f_{\min}}\end{aligned}$$

and this completes the proof of the lemma. □

We are finally ready to prove Theorem 4.1.

Proof of Theorem 4.1. We will first prove (a). In view of relation (4.3), we need to show that $\|\Delta f\|\leq\theta\hat{\mu}$. To show this inequality, note that by Lemma 4.2, it is enough to show that

$$\frac{\|f(w)-\hat{\mu}e\|^2}{2f_{\min}}\leq\theta\hat{\mu}.$$

Since $\mu = x^{\mathrm{T}}z/n$, it follows that $(f(w)-\mu e)^{\mathrm{T}}e = 0$. Using this equality, relations (4.1) and (4.2) and the fact that $\|e\| = \sqrt{n}$, we obtain

$$\begin{aligned}\|f(w)-\hat{\mu}e\|^2 &= \|f(w)-\mu e\|^2 + \|\mu e - \hat{\mu}e\|^2 \\ &\leq (\theta\mu)^2 + (|\mu - \hat{\mu}|\,\|e\|)^2 \\ &= (\theta\mu)^2 + (\delta\mu)^2 \\ &= (\theta^2+\delta^2)\mu^2.\end{aligned}$$

Note also that by (4.1), we have $f_{\min} \geq (1-\theta)\mu$. Thus

$$\begin{aligned}\frac{\|f(w)-\hat{\mu}e\|}{2f_{\min}} &\leq \frac{\theta^2+\delta^2}{2(1-\theta)}\mu \\ &\leq \theta(1-\delta/\sqrt{n})\mu = \theta\hat{\mu}\end{aligned}$$

where the last inequality follows from (3.2b) and the equality follows from (4.2). This proves (a).

We will now prove (b). From (3.1b), (3.1c) and the fact that $w \in W$, it follows that $\hat{w} \equiv (\hat{x}, \hat{y}, \hat{z})$ satisfies $A\hat{x} = b$ and $A^{\mathrm{T}}\hat{y} + \hat{z} = c$. We have just to show that $\hat{x}$ and $\hat{z}$ are strictly positive vectors to conclude that $\hat{w} \in W$. The proof is by contradiction. Assume that $\hat{x}_i < 0$ or $\hat{z}_i < 0$ for some index i. From (a), we have $\hat{x}_i\hat{z}_i \geq (1-\theta)\hat{\mu} > 0$. Then, it follows that $\hat{x}_i < 0$ and $\hat{z}_i < 0$, which in turn implies that $\Delta x_i \Delta z_i > x_i z_i$. On the other hand, we have $\Delta x_i \Delta z_i \leq \|\Delta f\| \leq \theta\hat{\mu} \leq \theta\mu$. Hence, we have $\theta\mu > x_i z_i \geq (1-\theta)\mu$ which contradicts the fact that $\theta < \frac{1}{2}$. This proves (b).

Statement (c) follows immediately from (4.5). This completes the proof of the theorem. □

Note that by achieving the measure of closeness in (a) of Theorem 4.1, we automatically maintain feasibility. This raises the question of how large δ (which measures the decrease rate in the duality gap) can be in order to guarantee that $\hat{w} \in W$ (and disregarding the measure of closeness in (a) of Theorem 4.1). Specifically, suppose we are given a point $w \in W$ satisfying (4.1) and our aim is only to guarantee that $\hat{w} = w - \Delta w \in W$ where $\Delta w = \Delta(w, \hat{\mu})$ with $\hat{\mu}$ given by (4.2). It is a consequence of the proof of Theorem 4.1 that if we choose δ satisfying

$$\frac{\theta^2+\delta^2}{2(1-\theta)(1-\delta/\sqrt{n})} \leq 1, \qquad \frac{\theta^2+\delta^2}{2(1-\theta)^2} < 1,$$

then $\hat{w} \in W$. In particular, if $w \in \Gamma$, and therefore we can set $\theta = 0$, then $\delta \leq [2(1-\delta/\sqrt{n})]^{1/2}$ is sufficient to guarantee that $\hat{w} \in W$. Thus, in this case, as $n \to \infty$, the largest possible δ, which provably guarantees that $\hat{w} \in W$, approaches $\sqrt{2}$.

5. Initialization of the algorithm

Given an LP in standard form and its dual, we have assumed in Section 2 that conditions (a) and (b) of Assumption 2.1 are valid. In general, this is not the case.

In order to circumvent this difficulty, we introduce another LP problem in standard form, which we call the augmented problem, whose solution will yield a solution to the original problem. Moreover, the size of the original problem and that of the augmented problem are of the same order. This fact will enable us to show that the complexity results of Section 4 also hold for LP problems which do not satisfy Assumption 2.1 or for which a proper initial point is not known a priori. For this augmented problem, we will see that an initial point w^0 lying on the central path Γ, and consequently satisfying the criterion of closeness (3.3), is readily available. The discussion in this section is intended to be brief. The reader is referred to Section 6 of Part II for a more detailed description which deals with initialization for convex quadratic programming problems.

Let the original problem be stated as follows.

$$(\tilde{\mathrm{P}}) \qquad \begin{aligned} \min \quad & \tilde{c}^{\mathrm{T}}\tilde{x} \\ \text{s.t.} \quad & \tilde{A}\tilde{x} = \tilde{b}, \\ & \tilde{x} \geq 0, \end{aligned}$$

where $\tilde{A}$ is an $\tilde{m} \times \tilde{n}$ matrix which has full row rank and $\tilde{b}$, $\tilde{c}$ are vectors of length $\tilde{m}$ and $\tilde{n}$ respectively. We assume that the entries of the vectors $\tilde{b}$, $\tilde{c}$ and the matrix $\tilde{A}$ are integral.

In order to state the augmented problem, we need to define some quantities. Let $n = \tilde{n}+2$ and $m = \tilde{m}+1$. Let $\tilde{L} = L(\tilde{A}, \tilde{b}, \tilde{c})$ denote the size of $(\tilde{\mathrm{P}})$. Let $\alpha = 2^{4\tilde{L}}$ and $\lambda = 2^{2\tilde{L}}$. Let K_b and K_c be constants defined as follows:

$$K_b = \alpha\lambda(\tilde{n}+1) - \lambda\tilde{c}^{\mathrm{T}}e, \qquad K_c = \alpha\lambda. \tag{5.1}$$

The augmented problem can be stated as follows.

$$(\mathrm{P}) \qquad \begin{aligned} \min \quad & \tilde{c}^{\mathrm{T}}\tilde{x} + K_c\tilde{x}_n \\ \text{s.t.} \quad & \tilde{A}\tilde{x} + (\tilde{b} - \lambda\tilde{A}e)\tilde{x}_n = \tilde{b}, \\ & (\alpha e - \tilde{c})^{\mathrm{T}}\tilde{x} + \alpha\tilde{x}_{n-1} = K_b, \\ & \tilde{x} \geq 0, \quad \tilde{x}_{n-1} \geq 0, \quad \tilde{x}_n \geq 0, \end{aligned}$$

where $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_{n-2})^{\mathrm{T}}$ is an $(n-2)$-vector and $\tilde{x}_{n-1}$ and $\tilde{x}_n$ are scalars. The dual of problem (P) is then given as follows.

$$(\mathrm{D}) \qquad \begin{aligned} \max \quad & \tilde{b}^{\mathrm{T}}\tilde{y} + K_b\tilde{y}_m \\ \text{s.t.} \quad & \tilde{A}^{\mathrm{T}}\tilde{y} + (\alpha e - \tilde{c})\tilde{y}_m + \tilde{z} = \tilde{c}, \\ & \alpha\tilde{y}_m + \tilde{z}_{n-1} = 0, \\ & (\tilde{b} - \lambda\tilde{A}e)^{\mathrm{T}}\tilde{y} + \tilde{z}_n = K_c, \\ & \tilde{z} \geq 0, \quad \tilde{z}_{n-1} \geq 0, \quad \tilde{z}_n \geq 0, \end{aligned}$$

where $\tilde{y}$ is an $(m-1)$-vector, $\tilde{z}$ is an $(n-2)$-vector and $\tilde{y}_m$, $\tilde{z}_{n-1}$ and $\tilde{z}_n$ are scalars. These problems can be recast in the notation of problems (P) and (D) of Section 2 as follows. Let $x=(\tilde{x}^{\mathrm{T}}, \tilde{x}_{n-1}, \tilde{x}_n)^{\mathrm{T}} \in \mathbb{R}^n$, $y=(\tilde{y}^{\mathrm{T}}, \tilde{y}_m)^{\mathrm{T}} \in \mathbb{R}^m$ and $z=(\tilde{z}^{\mathrm{T}}, \tilde{z}_{n-1}, \tilde{z}_n)^{\mathrm{T}} \in \mathbb{R}^n$. Define $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and $A \in \mathbb{R}^{m \times n}$ as follows:

$$b=\begin{pmatrix} \tilde{b} \\ K_b \end{pmatrix}, \quad c=\begin{pmatrix} \tilde{c} \\ 0 \\ K_c \end{pmatrix}, \quad A=\begin{bmatrix} \tilde{A} & 0 & \tilde{b}-\lambda\tilde{A}e \\ (\alpha e-\tilde{c})^{\mathrm{T}} & \alpha & 0 \end{bmatrix}. \tag{5.2}$$

With this notation, we can then rewrite problems (P) and (D) as in Section 2. We refer to these two formats interchangeably.

Consider the point $w^0=(x^0, y^0, z^0)$ defined as follows:

$$x^0 \equiv (\lambda, \ldots, \lambda, 1)^{\mathrm{T}} \in \mathbb{R}^n, \tag{5.3a}$$

$$y^0=(0, \ldots, 0, -1)^{\mathrm{T}} \in \mathbb{R}^m, \tag{5.3b}$$

$$z^0 \equiv (\alpha, \ldots, \alpha, \alpha\lambda)^{\mathrm{T}} \in \mathbb{R}^n. \tag{5.3c}$$

Using (5.1), one can easily verify that $Ax^0=b$ and $A^{\mathrm{T}}y^0+z^0=c$. Hence, $w^0 \in W$. Moreover, $f(w^0)=\alpha\lambda e$, which implies that w^0 lies on the central path Γ. If we let $\mu_0 \equiv \alpha\lambda$ then we obtain the criterion of closeness (3.3).

Note that, by construction, problems (P) and (D) have feasible solutions and therefore optimal solutions. In Section 6 of Part II, we show the following:
Let $x=(x_1, \ldots, x_n)^{\mathrm{T}}$ and $(y, z)=((y_1, \ldots, y_m)^{\mathrm{T}}, (z_1, \ldots, z_n)^{\mathrm{T}})$ be optimal solutions for problems (P) and (D) respectively. Then

(i) If $x_n z_{n-1}=0$ then, we have the following possible cases.

   (a) If $x_n=0$ and $z_{n-1}=0$ then $\tilde{x} \equiv (x_1, \ldots, x_{n-2})^{\mathrm{T}}$ and $(\tilde{y}, \tilde{z}) \equiv ((y_1, \ldots, y_{m-1})^{\mathrm{T}}, (z_1, \ldots, z_{n-2})^{\mathrm{T}})$ are optimal solutions for $(\tilde{\mathrm{P}})$ and $(\tilde{\mathrm{D}})$ respectively.

   (b) If $x_n \neq 0$ then $(\tilde{\mathrm{P}})$ is infeasible.

   (c) If $z_{n-1} \neq 0$ then $(\tilde{\mathrm{P}})$ is unbounded.

(ii) If $x_n z_{n-1} \neq 0$ then $(\tilde{\mathrm{P}})$ is either unbounded or infeasible. In this case, we solve the LP problem obtained by replacing the objective function of problem (P) by the linear function $K_c \tilde{x}_n$. If the resulting optimal solution $\bar{x}$, for this problem satisfies $\bar{x}_n=0$ then $(\tilde{\mathrm{P}})$ is unbounded. Otherwise, $(\tilde{\mathrm{P}})$ is infeasible.

(iii) The sizes of problems (P) and $(\tilde{\mathrm{P}})$ are of the same order.

It is clear from the above that one can solve problem $(\tilde{\mathrm{P}})$ via (P) in $\mathrm{O}(\sqrt{\tilde{n}}\tilde{L})$ iterations and with a total of $\mathrm{O}(\tilde{n}^{3.5}\tilde{L})$ arithmetic operations.

6. Remarks

The purpose of this paper is to present a theoretical result. Thus in order to simplify the presentation, we constructed $\hat{\mu}=\mu(1-\delta/\sqrt{n})$. Obviously, one can use $\hat{\mu}$ which is less than or equal to the above value, but which still guarantees (a) of Theorem 4.1. In this way, one can accelerate the convergence of the algorithm.

Additional improvements in actual implementation, such as more judicious selection of θ and δ, introduction of a step size for the direction combined with a search procedure for the parameter μ and a more practical choice of the initial point resulting in a smaller initial duality gap are possible and remain to be tested.

As mentioned in the introduction, we assumed throughout the paper that $m = O(n)$. However, if we drop this assumption then one can easily verify that the complexity achieved in this paper (Part I) expressed in terms of m and n is $O(n^{1.5}m^2L)$. Somewhat better bounds in terms of m and n can be achieved by using an approximation scheme as presented in Part II.

Acknowledgement

The authors wish to thank Michael Todd and two anonymous referees for many useful comments and in particular, for suggesting some simplifications in the proof of Theorem 4.1. This research was partially funded by the Brazilian Post-Graduate Education Federal Agency (CAPES) and by the United States Navy Office of Naval Research under contract N00014-87-K-0202. Their financial support is gratefully acknowledged.

References

[1] D.A. Bayer and J.C. Lagarias, "The nonlinear geometry of linear programming," manuscripts, AT&T Bell Laboratories (Murray Hill, NJ, 1986), to appear in *Transactions of the American Mathematical Society.*

[2] A. Fiacco and G. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (John Wiley and Sons, New York, 1968).

[3] K.R. Frisch, "The logarithmic potential method of convex programming," unpublished manuscript, University Institute of Economics (Oslo, Norway, 1955).

[4] P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method," *Mathematical Programming* 36 (1986) 183–209.

[5] C.C. Gonzaga, "An algorithm for solving linear programming problems in $O(n^3L)$ operations," Memorandum Number UCB/ERL M87/10, Electronics Research Laboratory, Universtiy of California (Berkeley, CA, March, 1987).

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

[7] N. Karmarkar, Talk at the University of California at Berkeley (Berkeley, CA, 1984).

[8] M. Kojima, S. Mizuno and A. Yoshise, "A primal-dual interior point algorithm for linear programming," Report No. B-188, Department of Information Sciences, Tokyo Institute of Technology (Tokyo, Japan, February, 1987).

[9] N. Megiddo, "Pathways to the optimal set in linear programming," Research Report, IBM Almaden Research Center (San Jose, CA, 1986).

[10] C.R. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, Englewood Cliffs, New Jersey, 1982).

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

[12] P.M. Vaidya, "An algorithm for linear programming which requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ arithmetic operations," preprint, AT&T Bell Laboratories (Murray Hill, NJ, 1987).