SIAM J. CONTROL AND OPTIMIZATION
Vol. 31, No. 6, pp. 1578–1598, November 1993

010

A PATH-FOLLOWING ALGORITHM FOR LINEAR PROGRAMMING USING QUADRATIC AND LOGARITHMIC PENALTY FUNCTIONS*

PAUL TSENG†

Abstract. Motivated by a recent work of Setiono, a path-following algorithm for linear programming using both logarithmic and quadratic penalty functions is proposed. In the algorithm, a logarithmic and a quadratic penalty is placed on, respectively, the nonnegativity constraints and an arbitrary subset of the equality constraints; Newton's method is applied to solve the penalized problem, and after each Newton step the penalty parameters are decreased. This algorithm maintains neither primal nor dual feasibility and does not require a Phase I. It is shown that if the initial iterate is chosen appropriately and the penalty parameters are decreased to zero in a particular way, then the algorithm is linearly convergent. Numerical results are also presented suggesting that the algorithm may be competitive with interior point algorithms in practice, requiring typically between 30–45 iterations to accurately solve each Netlib problem tested.

Key words. linear program, path-following, Newton step, penalty function

AMS subject classifications. 49, 90

1. Introduction.

Since the pioneering work of Karmarkar [Kar84], much interest has focused on solving linear programs using interior point algorithms. These interior point algorithms may be classified roughly as either (i) projective-scaling (or potential reduction), (ii) affine-scaling, or (iii) path-following. We will not attempt to review the literature on this subject, which is vast (see for example [Meg89], [Tod89] for surveys). Our interest is in algorithms of the path-following type, of the sort discussed in [GaZ81]. These interior point algorithms typically penalize the nonnegativity constraints by a logarithmic function and use Newton's method to solve the penalized problem, with the penalty parameters decreased after each Newton step (see, for example, [Gon89], [GMSTW86], [KMY89], [MoA87], [Ren88], [Tse89]).

One disadvantage of interior point algorithms is the need for an initial interior feasible solution. A common technique for handling this is to add an artifical column (see [AKRV89], [BDDW89], [GMSTW86], [Lus90], [MMS89], [MSSPB89], [MoM87]), but this itself has disadvantages. For example, the cost of the artificial column must be estimated, and some type of rank-1 updating is needed to solve each least square problem which can significantly increase the solution time and degrade the numerical accuracy of the solutions.

Recently, Setiono [Set89] proposed an interesting algorithm that combines features of a path-following algorithm with those of the method of multipliers [HaB70], [Hes69], [Pow69] (also see [Roc76], [Ber82]). This algorithm does not require a feasible solution to start and is comparable to interior point algorithms both in terms of work per iteration and, according to the numerical results reported in [Set89], in terms of the total number of iterations. To describe the basic idea in Setiono's algorithm, consider

* Received by the editors April 16, 1990; accepted for publication (in revised form) June 11, 1992. This research was supported by the U.S. Army Research Office, contract DAAL03-86-K-0171, and was conducted while the author was with the Laboratory for Information and Decision Systems and the Center for Intelligent Control Systems, Massachusetts Institute of Technology, Cambridge.

† Department of Mathematics, GN-50, University of Washington, Seattle, Washington 98195 (tseng@math.washington.edu).

a linear program in the standard dual form

$$\begin{array}{ll} \text{minimize} & -b^T p \\ \text{subject to} & t + A^T p = c, \quad t \geq 0, \end{array}$$

where A is some matrix and b and c are vectors of appropriate dimension. Let us attach a Lagrange multiplier vector x to the constraints $t + A^T p = c$ and apply the method of multipliers to the above linear program. This produces the following iterations

$$x^{k+1} = x^k + \frac{1}{\epsilon^k}(t^k + A^T p^k - c), \quad k = 1, 2, ...,$$

where $\{\epsilon^k\}$ is a sequence of monotonically decreasing positive scalars and (t^k, p^k) is some (inexact) solution of the augmented Lagrangian subproblem

$$\text{(1.1)} \qquad \begin{array}{l} \text{minimize} \quad -b^T p + (x^k)^T(t + A^T p - c) + \frac{1}{2\epsilon^k}||t + A^T p - c||^2 \\ \text{subject to } t \geq 0. \end{array}$$

(An advantage of the above multiplier iterations is that they do not need a feasible solution to start.) A key issue associated with the above multiplier iterations concerns the efficient generation of an inexact solution (t^k, p^k) of the convex quadratic program (1.1) for each k. (Note that as ϵ^k decreases, the objective function of (1.1) becomes progressively more ill-conditioned.) Setiono's algorithm may be viewed as the method of multipliers in which (t^k, p^k) is generated according to the following scheme, reminiscent of the path-following idea: Add a logarithmic penalty function $-\gamma^k \sum_{j=1}^m \ln(t_j)$ to the objective of (1.1), where γ^k is some positive scalar monotonically decreasing with k, and apply a single Newton step, starting from (t^{k-1}, p^{k-1}), to the resulting problem. (If the t^k thus obtained lies outside the positive orthant, it is moved back towards t^{k-1} until it becomes positive.[1])

In this paper, inspired by the preceding work of Setiono, we study an algorithm that also adds to the objective a quadratic penalty on the equality constraints and a logarithmic penalty on the nonnegativity constraints; and then solves the penalized problem using Newton's method, with the penalty parameters decreased after each Newton step. Unlike Setiono's algorithm, our algorithm does not use the multiplier vector x^k (so it may be viewed as a pure penalty method) and allows any subset of the equality constraints to be penalized. We show that if the problem is primal nondegenerate and the iterates start near the optimal solution of the initial penalized problem, then the penalty parameters can be decreased at the rate of a geometric progression and the iterates converge linearly. To the best of our knowledge, this is the first (global) linear convergence result for an noninterior point path-following algorithm. We also present numerical results indicating that the algorithm may potentially be competitive with interior point algorithms in practice. We remark that penalty methods that use either the quadratic or the logarithmic penalty function have been well studied (see, for example, [Ber82], [FiM68], [Fri57], [JiO78], [Man84],

[1] More precisely, t^k is given by the formula

$$t^k = t^{k-1} + .98\lambda^k \Delta t^k,$$

where Δt^k is the Newton direction (projected onto the space of t) and λ^k is the largest $\lambda \in (0, 1]$ for which $t^{k-1} + \lambda \Delta t^k$ is nonnegative. (The choice of .98 is arbitrary—any number between 0 and 1 would do.)

[WBD88]), but very little is known about penalty methods that use both types of penalty functions (called mixed interior point-exterior point algorithms in [FiM68]).

Upon completion of this paper, the author learned from Professor O. L. Mangasarian that the algorithm discussed here is essentially the IDLN (Interior Dual Least Norm) algorithm described in the recent Ph.D. thesis of Setiono [Set90] (also see the report [Set91]), with minor differences in the choice of the penalty parameters. The same reference includes an (asymptotic) linear rate convergence analysis and extensive numerical results showing that the IDLN algorithm outperforms the popular simplex code MINOS by a factor of 2 or more on the 63 Netlib problems tested. In short, Setiono's independent work provides further evidence of the practical efficiency of the mixed interior point-exterior point solution approach. Finally, we note that while this paper was under review, other noninterior point methods relating to that studied here have been proposed. One method, brought to our attention by one of the referees, is a certain augmented Lagrangian algorithm for stochastic programming (see [MuR90]); another method is a primal-dual exterior point algorithm for linear programming (see [KMM91]). However, neither of these methods has been shown to possess the nice theoretical/numerical properties enjoyed by the algorithm studied here. For example, no convergence result is given for the method in [MuR90] and no numerical or rate of convergence result is given for the method in [KMM91].

This paper proceeds as follows: In §2 we describe the basic algorithm; in §3 we analyze its convergence; and in §4 we recount our numerical experience with it. In §5 we discuss extensions of this work.

In our notation, every vector is a column vector in some k-dimensional real space $\Re^k$, and superscript T denotes transpose. For any vector x, we denote by x_j the jth coordinate of x, by $\mathrm{Diag}(x)$ the diagonal matrix whose jth diagonal entry is x_j, and by $\|x\|_1$, $\|x\|$, $\|x\|_\infty$ the L_1-norm, the L_2-norm, and the L_∞-norm of x, respectively. For any matrix A, we denote by A_j the jth column of A. We also denote by e the vector of 1's (whose dimension will be clear from the context) and denote by $\ln(\cdot)$ the natural logarithm function.

2. Algorithm description.

Let A be an $n \times m$ matrix, B be an $l \times m$ matrix, b be an n-vector, c be an m-vector, and d be an l-vector. Consider the following linear program associated with A, B, b, c, and d:

$$(\mathcal{D}) \qquad \begin{array}{c} \text{minimize } -b^T p \\ \text{subject to } t + A^T p = c,\ Bt = d,\ t \geq 0, \end{array}$$

which we call the dual problem. The dual problem may be viewed as a standard linear program in t, in which we arbitrarily partition the equality constraints into two subsets and express one subset in the generator form $t + A^T p = c$. (To see this, consider the problem of minimizing $a^T t$ subject to $t \in R \cap S$, $t \geq 0$, where a is an m-vector and R and S are affine sets in $\Re^m$. We can always express $R = \{\ t \mid t = c - A^T p \text{ for some } p\ \}$ and $S = \{\ t \mid Bt = d\ \}$ for some matrices A and B and some vectors c and d. Using $t = c - A^T p$ to eliminate t from the objective function, the problem is now in the form $(\mathcal{D})$.) The constraints $t + A^T p = c$ can be thought of as the complicating constraints which, if removed, would make $(\mathcal{D})$ much easier to solve. The advantages of splitting the constraints in this manner will be explained at the end of §2. Finally, we note that the form in which we write the equality constraints is unimportant, and the above form is adopted for notational convenience only.

By attaching Lagrange multiplier vectors x and y to the constraints $c - A^T p = t$

and $Bt = d$, respectively, we obtain the following dual of $(\mathcal{D})$:

$$(\mathcal{P}) \qquad \begin{array}{c} \text{minimize } c^T x + d^T y \\ \text{subject to } Ax = b, \; x + B^T y \geq 0, \end{array}$$

which we call the primal problem.

We make the following blanket assumptions, which are standard for interior point algorithms, regarding $(\mathcal{D})$ and $(\mathcal{P})$.

Assumption A.

(a) $\{ (x, y) \mid Ax = b, x + B^T y > 0 \}$ is nonempty.

(b) $\{ (t, p) \mid Bt = d, t > 0, t + A^T p = c \}$ is nonempty.

(c) A has full row rank.

It is well known that, under parts (a) and (b) of Assumption A, both $(\mathcal{D})$ and $(\mathcal{P})$ have nonempty bounded optimal solution sets.

Consider the dual problem $(\mathcal{D})$. Suppose that we place a quadratic penalty on the constraints $t + A^T p = c$ with a penalty parameter $1/\epsilon$ $(\epsilon > 0)$ and we place a logarithmic penalty on the constraints $t \geq 0$ with a penalty parameter $\gamma > 0$. This gives the following approximation to $(\mathcal{D})$:

$$(\mathcal{D}_{\epsilon,\gamma}) \qquad \begin{array}{c} \text{minimize } f_{\epsilon,\gamma}(t, p) \\ \text{subject to } Bt = d, \quad t > 0, \end{array}$$

where $f_{\epsilon,\gamma} : (0, \infty)^m \times \Re^n$ is the penalized objective function given by

$$(2.1) \qquad f_{\epsilon,\gamma}(t, p) = \frac{1}{2}||c - t - A^T p||^2 - \epsilon\gamma \sum_{j=1}^{m} \ln(t_j) - \epsilon b^T p \quad \forall t > 0, \; \forall p.$$

The penalized problem $(\mathcal{D}_{\epsilon,\gamma})$ has the advantage that its objective function $f_{\epsilon,\gamma}$ is twice differentiable and the Hessian $\nabla^2 f_{\epsilon,\gamma}$ is positive definite everywhere (via Assumption A(c)). Since the feasible set of $(\mathcal{D}_{\epsilon,\gamma})$ is nonempty (by Assumption A(b)) and its intersection with any level set of $f_{\epsilon,\gamma}$ is bounded (by Assumption A(a)), it is readily seen that $(\mathcal{D}_{\epsilon,\gamma})$ has an optimal solution which, by the strict convexity of $f_{\epsilon,\gamma}$, is unique.

Note 1. We can use penalty functions other than the quadratic and the logarithmic. For example, we can use a cubic in place of the quadratic and $t_j \ln(t_j)$ in place of $-\ln(t_j)$. The quadratic and the logarithmic function, however, have nice properties (such as the second derivative of the logarithmic function equals minus the square of its first derivative) which make global convergence analysis possible.

It is well known (see [Roc70]) that (t, p) is the optimal solution of $(\mathcal{D}_{\epsilon,\gamma})$ if and only if it satisfies, together with some $u \in \Re^l$, the Kuhn–Tucker conditions

$$(2.2) \qquad t > 0, \qquad Bt = d, \qquad \nabla f_{\epsilon,\gamma}(t, p) + \begin{pmatrix} B^T u \\ 0 \end{pmatrix} = 0.$$

Straightforward calculation using (2.1) finds that

$$(2.3) \qquad \nabla f_{\epsilon,\gamma}(t, p) = \begin{pmatrix} t + A^T p - c - \epsilon\gamma(T)^{-1} e \\ A(t + A^T p - c) - \epsilon b \end{pmatrix},$$

$$(2.4) \qquad \nabla^2 f_{\epsilon,\gamma}(t, p) = \begin{pmatrix} I + \epsilon\gamma(T)^{-2} & A^T \\ A & AA^T \end{pmatrix},$$

where $T = \mathrm{Diag}(t)$. The above formulas will be used extensively in the subsequent analysis. Note that $\nabla^2 f_{\epsilon,\gamma}$ is ill-conditioned at the boundary of its domain.

It is not difficult to show that, as ϵ and γ tend to zero, the optimal solution of $(\mathcal{D}_{\epsilon,\gamma})$ approaches the optimal solution set of $(\mathcal{D})$ (see Lemma 3.1). This suggests the following algorithm for solving $(\mathcal{D}_{\epsilon,\gamma})$. At each iteration, we are given ϵ, γ and a (t,p) which is an approximate solution of $(\mathcal{D}_{\epsilon,\gamma})$; we apply a Newton step to $(\mathcal{D}_{\epsilon,\gamma})$ at (t,p) to generate a new (t,p) and then we decrease ϵ and γ. In other words, we consider a sequence of penalized problems $(\mathcal{D}_{\epsilon^k,\gamma^k})$, $k = 1,2,...$, with $\epsilon^k \downarrow 0$ and $\gamma^k \downarrow 0$, and we use a Newton step to follow the optimal solution of one penalized problem to that of the next. We now formally state this algorithm, which we call the QLPPF (short for Quadratic-Logarithmic Penalty Path-Following) algorithm.

QLPPF Algorithm

Iteration 0. Choose $\epsilon^1 > 0$ and $\gamma^1 > 0$. Choose $(t^1, p^1) \in (0,\infty)^m \times \Re^n$ with $Bt^1 = d$.
Iteration k. Given $(t^k, p^k) \in (0,\infty)^m \times \Re^n$ with $Bt^k = d$, compute $(\Delta t^k, \Delta p^k, u^k)$ to be a solution of

$$(2.5)\quad \nabla^2 f_{\epsilon^k,\gamma^k}(t^k,p^k)\begin{pmatrix}\Delta t^k\\ \Delta p^k\end{pmatrix} + \nabla f_{\epsilon^k,\gamma^k}(t^k,p^k) + \begin{pmatrix}B^T u^k\\ 0\end{pmatrix} = 0, \quad B\Delta t^k = 0,$$

and set

$$(2.6)\qquad t^{k+1} = t^k + \Delta t^k, \qquad p^{k+1} = p^k + \Delta p^k,$$

$$(2.7)\qquad \gamma^{k+1} = \alpha^k \gamma^k, \qquad \epsilon^{k+1} = \alpha^k \epsilon^k,$$

where α^k is some scalar in (0,1).

Note 2. As we noted earlier, the QLPPF Algorithm is closely linked to the algorithms proposed by Setiono. In particular, it can be seen from (2.3), (2.4) that, in the special case where B is the zero matrix, the direction finding problem (2.5) is identical to that in the IDLN algorithm of Setiono (see [Set91, Eq. (16)]) and differs from that in the IDPP algorithm of Setiono (see [Set89, Eq. (6)]) by only an order ϵ^k term on the right-hand side (which tends to zero as ϵ^k tends to zero).

Note 3. A unique feature of the QLPPF Algorithm lies in its handling of the two sets of constraints $t + A^T p = c$ and $Bt = d$, whereby quadratic penalties are placed only on the first set while the second set is maintained to be satisfied at all iterations. This feature has the advantage that it enables the QLPPF Algorithm to provide a unified framework for interior point methods and exterior point methods. As an example, if we put all the equality constraints into $Bt = d$ (and correspondingly set $A = I$ and $c = 0$), then it can be seen that the QLPPF Algorithm with $\gamma^k = 1$ for all k reduces to the well-known primal path-following algorithm for maximizing $b^T t$ subject to the constraints $Bt = d$, $t \geq 0$ (see [Gon89], [Tse89], [Ye87]). If we put all the equality constraints into $t + A^T p = c$ (and correspondingly set $B = 0$ and $d = 0$), then, as was noted above, the QLPPF Algorithm reduces to the IDLN algorithm for maximizing $b^T p$ subject to the constraints $t + A^T p = c$, $t \geq 0$. For other choices of constraint splitting, we obtain algorithms somewhere in between. We can then envision choosing a constraint splitting so the corresponding QLPPF Algorithm is in some sense most efficient (e.g., fastest convergence) for the given problem. Alternatively, we may choose a constraint splitting so to ensure that certain "critical" equality constraints are satisfied exactly at all iterations (by putting these constraints into $Bt = d$).

3. Global convergence. In this section, we show that if $(\mathcal{D})$ is in some sense primal nondegenerate and if (t^1, p^1) is "close" to the optimal solution of $(\mathcal{D}_{\epsilon^1,\gamma^1})$ in the QLPPF Algorithm, then, by decreasing the α^k's at an appropriate rate, the iterates $\{(t^k, p^k)\}$ generated by the QLPPF Algorithm approach the optimal solution set of $(\mathcal{D})$ (see Theorem 3.4). Because the Hessian $\nabla^2 f_{\epsilon,\gamma}$ is ill-conditioned at the boundary of its domain, the proof of this result is quite involved and relies critically on finding a suitable Lyapunov (i.e., merit) function to monitor the progress of the algorithm.

For any $\epsilon > 0$ and $\gamma > 0$, let $\rho_{\epsilon,\gamma} : (0,\infty)^m \times \Re^n \times \Re^l$ be the function given by

$$\begin{aligned} \rho_{\epsilon,\gamma}(t,p,u) = \max\{ & \frac{1}{\epsilon\gamma}||T(c-t-A^Tp-B^Tu)+\epsilon\gamma e||, \\ & \frac{1}{\sqrt{\epsilon\gamma}}||A(c-t-A^Tp)+\epsilon b|| \ \} \qquad \forall t>0,\ \forall p,\ \forall u, \end{aligned} \tag{3.1}$$

where $T = \mathrm{Diag}(t)$. From (2.2) and (2.3) we see that (t,p) is a solution of $(\mathcal{D}_{\epsilon,\gamma})$ if and only if (t,p) satisfies $Bt = d, t > 0$ and $\rho_{\epsilon,\gamma}(t,p,u) = 0$ for some u. Hence $\rho_{\epsilon,\gamma}$ acts as a Lyapunov function which measures how far (t,p) is from solving $(\mathcal{D}_{\epsilon,\gamma})$. This notion is made precise in the following lemma.

For any $\epsilon > 0$, let

$$\mathcal{Y}_\epsilon = \{ \ (t,p) \mid A(c-t-A^Tp) = -\epsilon b,\ Bt = d,\ t > 0 \ \}. \tag{3.2}$$

The following lemma shows that any $(t,p) \in \mathcal{Y}_\epsilon$ that satisfies $\rho_{\epsilon,\gamma}(t,p,u) \le \beta$, for some u, is within an order of $\epsilon + (1+\beta)\gamma$ of being an optimal solution of $(\mathcal{D})$.

LEMMA 3.1. *Fix any $\epsilon > 0$, $\gamma > 0$ and $\beta \in (0,1]$. For any $(t,p) \in \mathcal{Y}_\epsilon$ that satisfies $\rho_{\epsilon,\gamma}(t,p,u) \le \beta$ for some u, the vector (x,y) given by*

$$x = (t + A^Tp - c)/\epsilon, \qquad y = u/\epsilon \tag{3.3}$$

is feasible for $(\mathcal{P})$ and, together with (t,p), satisfies

$$\begin{aligned} c^Tx + d^Ty &\le v^* + \eta^*\epsilon + m(1+\beta)\gamma, \\ b^Tp &\ge v^* - m(1+\beta)\gamma, \\ ||t + A^Tp - c||^2 &\le 2(\eta^*(\epsilon)^2 + m(1+\beta)\epsilon\gamma), \end{aligned}$$

where v^* denotes the optimal cost of $(\mathcal{P})$ and $\eta^* = \min\{ \ ||x^*||^2/2 \mid (x^*, y^*)$ is an optimal solution of $(\mathcal{P})$ for some $y^* \ \}$.

Proof. Since (t,p) is in $\mathcal{Y}_\epsilon$, it follows from the definition of $\mathcal{Y}_\epsilon$ (see (3.2)) and (3.3) that $Ax = b$. Since $\rho_{\epsilon,\gamma}(t,p,u) \le \beta$, we have from the definition of $\rho_{\epsilon,\gamma}$ (see (3.1)) and (3.3) that $||T(-x - B^Ty) + \gamma e|| \le \gamma\beta$, where $T = \mathrm{Diag}(t)$. Thus

$$\gamma(1-\beta)e \le T(x + B^Ty) \le \gamma(1+\beta)e. \tag{3.4}$$

Since $t > 0$ and $\beta \le 1$, the first inequality in (3.4) yields $x + B^Ty \ge \gamma(1-\beta)T^{-1}e \ge 0$. Hence, (x,y) is feasible for $(\mathcal{P})$.

Let

$$\begin{aligned} d_\epsilon(\tau,\pi) &= b^T\pi - \frac{1}{2\epsilon}||\tau + A^T\pi - c||^2 \quad \forall(\tau,\pi), \\ p_\epsilon(\xi,\psi) &= \frac{\epsilon}{2}||\xi||^2 + c^T\xi + d^T\psi \quad \forall(\xi,\psi). \end{aligned}$$

Straightforward algebra using (3.3) yields

$$p_\epsilon(x,y) = d_\epsilon(t,p) + t^T(x + B^Ty).$$

Also, it follows from strong duality for a convex quadratic program and its dual that

$$\min_{(\xi,\psi)}\{\ p_\epsilon(\xi,\psi) \mid A\xi = b, \xi + B^T\psi \geq 0\ \} = \max_{(\tau,\pi)}\{\ d_\epsilon(\tau,\pi) \mid B\tau = d, \tau \geq 0\ \}.$$

Thus, by letting (x^*, y^*) be any optimal solution of $(\mathcal{P})$ with $||x^*||^2/2 = \eta^*$, we obtain from the above two relations and $c^Tx^* + d^Ty^* = v^*$ that

$$\begin{aligned}\frac{\epsilon}{2}||x||^2 + c^Tx + d^Ty &= p_\epsilon(x,y)\\ &= d_\epsilon(t,p) + t^T(x + B^Ty)\\ &\leq p_\epsilon(x^*,y^*) + t^T(x + B^Ty)\\ &= \epsilon\eta^* + v^* + t^T(x + B^Ty),\end{aligned}$$

Similarly, by letting (t^*, p^*) be any optimal solution of $(\mathcal{D})$, we obtain from the same two relations and the facts $b^Tp^* = v^*$ and $t^* + A^Tp^* = c$ that

$$\begin{aligned}b^Tp - \frac{1}{2\epsilon}||t + A^Tp - c||^2 &= d_\epsilon(t,p)\\ &= p_\epsilon(x,y) - t^T(x + B^Ty)\\ &\geq d_\epsilon(t^*,p^*) - t^T(x + B^Ty)\\ &= v^* - t^T(x + B^Ty).\end{aligned}$$

Since $0 \leq t^T(x + B^Ty) \leq m(1+\beta)\gamma$ (cf. (3.4)), the above two relations yield

$$\begin{aligned}c^Tx + d^Ty &\leq v^* + \epsilon\eta^* + m(1+\beta)\gamma\\ b^Tp &\geq v^* - m(1+\beta)\gamma.\end{aligned}$$

Finally, since (x, y) is feasible for $(\mathcal{P})$ so that $c^Tx + d^Ty \geq v^*$, the first of these two relations also yields

$$\frac{\epsilon}{2}||x||^2 + v^* \leq \epsilon\eta^* + v^* + m(1+\beta)\gamma.$$

Canceling v^* from both sides and using (3.3) completes the proof. □

Since we are dealing with linear programs, Lemma 3.1 implies that, as $\epsilon \downarrow 0$ and $\gamma \downarrow 0$, the (x, y) given by (3.3) approaches the optimal solution set of $(\mathcal{P})$ and (t, p) approaches the optimal solution set of $(\mathcal{D})$. In fact, it suffices to decrease ϵ and γ as far as $2^{\kappa L}$, where κ is some scalar constant and L is the size of the problem encoding in binary (defined as, say, in [Kar84]), at which time an optimal solution of $(\mathcal{P})$ and of $(\mathcal{D})$ can be recovered by using the techniques described in, for example, [Kar84] and [PaS82].

For each $\lambda > 0$, let $\theta_\lambda : (0,\infty)^m \to [0,\infty)$ be the function given by

$$\theta_\lambda(t) = (||E + ED^{1/2}A^TFAD^{1/2}E|| + ||ED^{1/2}A^TF||)^2 \quad \forall t > 0, \tag{3.5}$$

where

$$D = (I + \lambda T^{-2})^{-1}, \tag{3.6}$$

$$E = I - D^{1/2}B^T[BDB^T]^{-1}BD^{1/2}, \tag{3.7}$$

$$F = [A(I - D^{1/2}ED^{1/2})A^T]^{-1}, \tag{3.8}$$

and $T = \mathrm{Diag}(t)$. (F is well defined because $||E|| \le 1$ (E is a projection matrix) and $||D|| < 1$, so that $I - D^{1/2}ED^{1/2}$ is positive definite. We also use the assumption that A has full row rank.) The quantity $\theta_\lambda(t)$ estimates the norm squared of certain projection-like operator depending on λ and t, and it will be used extensively in our analysis. In general, $\theta_\lambda(t)$ is rather cumbersome to evaluate, but, as we shall see, it suffices for our analysis to bound $\theta_\lambda(t)$ from above (see Lemma 3.3(b)).

3.1. Analyzing a Newton step. In this subsection we prove a key lemma that states that if $(\bar{t}, \bar{p})$ is "close" to the optimal solution of $(\mathcal{D}_{\bar{\epsilon},\bar{\gamma}})$, then (t, p) generated by applying one Newton step to $(\mathcal{D})$ at $(\bar{t}, \bar{p})$ is close to the optimal solution of $(\mathcal{D}_{\epsilon,\gamma})$ for some $\epsilon < \bar{\epsilon}$ and some $\gamma < \bar{\gamma}$. The notion of "closeness" is measured by the Lyapunov function $\rho_{\epsilon,\gamma}$ and the proof of the lemma is based on the ideas used in [Tse89, §2].

LEMMA 3.2. *For any $\bar{\epsilon} > 0$, any $\bar{\gamma} > 0$ and any $(\bar{t}, \bar{p}, \bar{u}) \in (0, \infty)^m \times \Re^n \times \Re^l$ with $B\bar{t} = d$, let (t, p, u) be given by*

$$t = \bar{t} + \Delta t, \tag{3.9}$$

$$p = \bar{p} + \Delta p, \tag{3.10}$$

where u and $(\Delta t, \Delta p)$ together solve the following system of linear equations

$$\nabla^2 f_{\bar{\epsilon},\bar{\gamma}}(\bar{t}, \bar{p}) \begin{pmatrix} \Delta t \\ \Delta p \end{pmatrix} + \nabla f_{\bar{\epsilon},\bar{\gamma}}(\bar{t}, \bar{p}) + \begin{pmatrix} B^T u \\ 0 \end{pmatrix} = 0, \qquad B\Delta t = 0. \tag{3.11}$$

Suppose that $\rho_{\bar{\epsilon},\bar{\gamma}}(\bar{t}, \bar{p}, \bar{u}) \le \beta$ for some $\beta < \min\{1, 1/\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})\}$. Then the following hold:

(a) $(t, p) \in \mathcal{Y}_{\bar{\epsilon}}$.

(b) *For any α satisfying*

$$\max\left\{ \sqrt{\frac{\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})(\beta)^2 + \sqrt{m}}{\beta + \sqrt{m}}}, \frac{1}{1 + \beta\sqrt{\bar{\gamma}/\bar{\epsilon}}/||b||} \right\} \le \alpha \le 1, \tag{3.12}$$

we have $\rho_{a\bar{\epsilon},a\bar{\gamma}}(t, p, u) \le \beta$.

Proof. Let

$$\bar{r} = \bar{T}(c - \bar{t} - A^T\bar{p} - B^T\bar{u}) + \bar{\epsilon}\bar{\gamma}e, \tag{3.13}$$

$$\bar{s} = A(c - \bar{t} - A^T\bar{p}) + \bar{\epsilon}b, \tag{3.14}$$

where $\bar{T} = \mathrm{Diag}(\bar{t})$. Then, by (3.1),

$$\max\{||\bar{r}||/(\bar{\epsilon}\bar{\gamma}), ||\bar{s}||/\sqrt{\bar{\epsilon}\bar{\gamma}}\} = \rho_{\bar{\epsilon},\bar{\gamma}}(\bar{t}, \bar{p}, \bar{u}) \le \beta. \tag{3.15}$$

By using (2.3), (2.4) and (3.13), (3.14), we write (3.11) equivalently as

$$\begin{aligned} D^{-1}\Delta t + A^T\Delta p + B^T(u - \bar{u}) &= \bar{T}^{-1}\bar{r}, \\ A\Delta t + AA^T\Delta p &= \bar{s}, \\ B\Delta t &= 0, \end{aligned}$$

where for convenience we let $D = (I + \bar{\epsilon}\bar{\gamma}\bar{T}^{-2})^{-1}$. Solving for Δt gives

$$\Delta t = D^{1/2}(E + ED^{1/2}A^TFAD^{1/2}E)D^{1/2}\bar{T}^{-1}\bar{r} - D^{1/2}ED^{1/2}A^TF\bar{s},$$

where we let $E = I - D^{1/2}B^T[BDB^T]^{-1}BD^{1/2}$ and $F = [A(I - D^{1/2}ED^{1/2})A^T]^{-1}$. ($F$ is well defined by the same reasoning that the matrix given by (3.8) is well defined.) Then, we can bound $\bar{T}^{-1}\Delta t$ as follows:

$$||\bar{T}^{-1}\Delta t|| \le ||\bar{T}^{-1}D^{1/2}||^2||E+ED^{1/2}A^TFAD^{1/2}E||||\bar{r}||+||\bar{T}^{-1}D^{1/2}||||ED^{1/2}A^TF||||\bar{s}||.$$

Since $\bar{T}^{-2}D$ is diagonal and each of its diagonal entry is less than $1/(\bar{\epsilon}\bar{\gamma})$, we obtain that $\|\bar{T}^{-2}D\| < 1/(\bar{\epsilon}\bar{\gamma})$ and hence

$$\begin{aligned} \|\bar{T}^{-1}\Delta t\| &\leq \|E + ED^{1/2}A^T FAD^{1/2}E\|\|\bar{r}\|/(\bar{\epsilon}\bar{\gamma}) + \|ED^{1/2}A^T F\|\|\bar{s}\|/\sqrt{\bar{\epsilon}\bar{\gamma}} \\ &\leq \sqrt{\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})}\beta, \end{aligned} \tag{3.16}$$

where the last inequality follows from (3.15) and the definition of $\bar{T}$, D, E, F, and $\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})$ (see (3.5)–(3.8)).

Now, by using (2.3) and (2.4), we can write (3.11) equivalently as $B\Delta t = 0$ and

$$\bar{\epsilon}\bar{\gamma}\bar{T}^{-1}\Delta t = \bar{T}(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u) + \bar{\epsilon}\bar{\gamma}e, \tag{3.17}$$

$$0 = A(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p) + \bar{\epsilon}b, \tag{3.18}$$

so from (3.9) and (3.10) we obtain

$$\begin{aligned} T(c - t - A^T p - B^T u) + \bar{\epsilon}\bar{\gamma}e &= (\bar{T} + \Delta T)(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u) + \bar{\epsilon}\bar{\gamma}e \\ &= \bar{T}(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u) \\ &\quad + \bar{\epsilon}\bar{\gamma}e + \Delta T(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u) \\ &= \bar{\epsilon}\bar{\gamma}\bar{T}^{-1}\Delta t + \Delta T(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u) \\ &= \bar{T}^{-1}\Delta T(\bar{\epsilon}\bar{\gamma}e + \bar{T}(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p - B^T u)) \\ &= \bar{\epsilon}\bar{\gamma}\bar{T}^{-2}\Delta T\Delta t, \end{aligned}$$

where $T = \text{Diag}(t)$, $\Delta T = \text{Diag}(\Delta t)$, and the third and the last equality follow from (3.17). This implies

$$\begin{aligned} \|T(c - t - A^T p - B^T u) + \bar{\epsilon}\bar{\gamma}e\| &\leq \bar{\epsilon}\bar{\gamma}\|\bar{T}^{-2}\Delta T\Delta t\| \\ &\leq \bar{\epsilon}\bar{\gamma}\|\bar{T}^{-2}\Delta T\Delta t\|_1 \\ &= \bar{\epsilon}\bar{\gamma}\|\bar{T}^{-1}\Delta t\|^2 \\ &\leq \bar{\epsilon}\bar{\gamma}\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})(\beta)^2, \end{aligned} \tag{3.19}$$

where the third inequality follows from (3.16).

(a) We have from (3.16) and the hypothesis $\beta < 1/\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})$ that $\|\bar{T}^{-1}\Delta t\|^2 \leq \theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})(\beta)^2 < \beta < 1$, so (3.9) and $\bar{t} > 0$ yields $t = \bar{t} + \Delta t > 0$. Also, $B\bar{t} = d$ together with $B\Delta t = 0$ (see (3.11)) and (3.9) yields $Bt = d$, and (3.18) together with (3.9), (3.10) yields $0 = A(c - t - A^T p) + \bar{\epsilon}b$. Hence $(t, p) \in \mathcal{Y}_{\bar{\epsilon}}$.

(b) Fix any α satisfying (3.12) and let $\gamma = \alpha\bar{\gamma}$, $\epsilon = \alpha\bar{\epsilon}$. (Note that because $\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})\beta < 1$, the left-hand quantity in (3.12) is strictly less than 1, so such an α exists.) Let

$$r = T(c - t - A^T p - B^T u) + \epsilon\gamma e.$$

Then the triangle inequality and (3.19) imply

$$\begin{aligned} \|r\|/(\epsilon\gamma) &\leq \|T(c - t - A^T p - B^T u) + \bar{\epsilon}\bar{\gamma}e\|/(\epsilon\gamma) + (1 - (\alpha)^2)\sqrt{m}/(\alpha)^2 \\ &\leq \theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})(\beta)^2/(\alpha)^2 + (1/(\alpha)^2 - 1)\sqrt{m}, \end{aligned}$$

which together with the fact (see (3.12)) $(\theta_{\bar{\epsilon}\bar{\gamma}}(\bar{t})(\beta)^2 + \sqrt{m})/(\beta + \sqrt{m}) \leq (\alpha)^2$, yields

$$\|r\|/(\epsilon\gamma) \leq \beta. \tag{3.20}$$

Let $s = A(c - t - A^T p) + \epsilon b$. By using (3.9), (3.10), and (3.18), we have

$$s = A(c - \bar{t} - \Delta t - A^T\bar{p} - A^T\Delta p) + \alpha\bar{\epsilon}b = (\alpha - 1)\bar{\epsilon}b,$$

which together with the fact (see (3.12))

$$1/(1 + \beta\sqrt{\bar{\gamma}/\bar{\epsilon}}/||b||) \leq \alpha \leq 1,$$

yields

$$||s||/\sqrt{\epsilon\gamma} = (1/\alpha - 1)||b||\sqrt{\bar{\epsilon}/\bar{\gamma}} \leq \beta.$$

This together with (3.20) and the definition of $\rho_{\epsilon,\gamma}(t,p,u)$ (see (3.1)) proves our claim. □

3.2. Bounds. Lemma 3.2 shows that if the rate α at which the penalty parameters ϵ and γ are decreased is not too small (see (3.12)), then a single Newton step suffices to keep the current iterate close to the optimal solution of the penalized problem $(\mathcal{D}_{\epsilon,\gamma})$. Thus, in order to establish the (linear) convergence of the QLPPF Algorithm, it suffices to bound α away from 1 which, according to (3.12), amounts to bounding $\theta_{\epsilon\gamma}(t)$ by some quantity independent of t and ϵ, γ. It is not difficult to see that such a bound does not exist for arbitrary t. Fortunately, we need to consider only those t that, together with some p, are close to the optimal solution of $(\mathcal{D}_{\epsilon,\gamma})$, in which case, as we show below, such a bound does exist (provided that a certain primal nondegeneracy assumption also holds). The proof of this is somewhat intricate: For ϵ and γ large, we argue by showing that t cannot be too large, i.e., of the order $\epsilon+\gamma+1$ (see Lemma 3.3(a)) and, for ϵ and γ small, we argue by showing that, under the primal nondegeneracy assumption, the columns of A corresponding to those components of t which are small (i.e., of the order γ) are of rank n.

LEMMA 3.3. (a) *For all $\epsilon > 0$, all $\gamma > 0$, and all $(t,p) \in \mathcal{Y}_\epsilon$ satisfying $\rho_{\epsilon,\gamma}(t,p,u) \leq 1$ for some u, we have*

$$||t|| \leq M_1(\epsilon + \gamma + 1),$$

where $M_1 > 0$ is some scalar depending on A, B, b, c, and d only.

(b) *Suppose that $(\mathcal{P})$ is primal nondegenerate in the sense that, for every optimal solution (x^*, y^*) of $(\mathcal{P})$, those columns of A corresponding to the positive components of $x^* + B^T y^*$ have rank n. Then, for all $\epsilon > 0$, all $\gamma > 0$, all $\lambda \geq \epsilon\gamma/2$, and all $(t,p) \in \mathcal{Y}_\epsilon$ satisfying $\rho_{\epsilon,\gamma}(t,p,u) \leq 1$ for some u, we have*

$$\theta_\lambda(t) \leq \psi(\epsilon/\gamma),$$

where we define

$$\psi(\omega) = M_2(\sqrt{\omega} + 1/\sqrt{\omega})^4 \quad \forall\ \omega > 0, \tag{3.21}$$

with $M_2 \geq 1$ some scalar depending on A, B, b, c, and d only.

Proof. (a) The proof is by contradiction. Suppose the contrary, so that there exists a sequence $\{(t^k, p^k, u^k, \epsilon^k, \gamma^k)\}$ such that $\epsilon^k > 0$ and $\gamma^k > 0$ and

$$(t^k, p^k) \in \mathcal{Y}_{\epsilon^k}, \quad \rho_{\epsilon^k,\gamma^k}(t^k, p^k, u^k) \leq 1 \qquad \forall k, \tag{3.22}$$

$$||t^k||/(\epsilon^k + \gamma^k) \to \infty, \qquad ||t^k|| \to \infty. \tag{3.23}$$

By passing into a subsequence if necessary we will assume that $(t^k, p^k)/||(t^k, p^k)||$ converges to some limit point, say (t^∞, p^∞) (so $(t^\infty, p^\infty) \neq 0$). We have from (3.22) (also using (3.1) and (3.2)) that

$$Bt^k = d, \quad t^k > 0 \qquad \forall k,$$

and from Lemma 3.1 that

$$b^T p^k \geq v^* - 2m\gamma^k, \quad ||t^k + A^T p^k - c|| \leq \sqrt{2(\eta^*(\epsilon^k)^2 + 2m\epsilon^k\gamma^k)} \qquad \forall k.$$

Upon dividing both sides of the above four relations by $||(t^k, p^k)||$ and letting $k \to \infty$, we obtain from (3.23) and $(t^k, p^k)/||(t^k, p^k)|| \to (t^\infty, p^\infty)$ that

$$Bt^\infty = 0, \quad t^\infty \geq 0, \quad b^T p^\infty \geq 0, \quad ||t^\infty + A^T p^\infty|| = 0.$$

Then, any optimal solution to $(\mathcal{D})$ would remain optimal when moved along the direction (t^∞, p^∞), contradicting the boundedness of the optimal solution set of $(\mathcal{D})$ (via Assumption A(a)).

(b) Fix any $\epsilon > 0$, $\gamma > 0$, $\lambda \geq \epsilon\gamma/2$, and any $(t, p) \in \mathcal{Y}_\epsilon$ satisfying $\rho_{\epsilon,\gamma}(t, p, u) \leq 1$ for some u. Let $T = \mathrm{Diag}(t)$ and let D, E, F be given by, respectively, (3.6), (3.7), and (3.8). Then, $F^{-1} = A(I - D^{1/2}ED^{1/2})A^T$, $||E|| \leq 1$ and $D = (I + \lambda T^{-2})^{-1}$. From the definition of $\theta_\lambda(t)$ (see (3.5)) we then obtain

$$\begin{aligned} \theta_\lambda(t) &= (||E + ED^{1/2}A^T FAD^{1/2}E|| + ||ED^{1/2}A^T F||)^2 \\ &\leq (||E|| + ||E||^2||D^{1/2}||^2||A||^2||F|| + ||E||||D^{1/2}||||A||||F||)^2 \\ &< (1 + ||A||^2||F|| + ||A||||F||)^2, \end{aligned} \tag{3.24}$$

where the strict inequality follows from the facts $||D|| < 1$, $||E|| \leq 1$. Now we bound $||F||$. We have

$$\begin{aligned} z^T(F^{-1})z &= z^T A(I - D^{1/2}ED^{1/2})A^T z \\ &\geq z^T A(I - D)A^T z \\ &= \sum_{j=1}^m (A_j^T z)^2/((t_j)^2/\lambda + 1) \\ &\geq \lambda \sum_{j=1}^m (A_j^T z)^2/(t_j)^2 \\ &\geq \lambda \sum_{j=1}^m (A_j^T z)^2/||t||^2 \\ &\geq \lambda\sigma||z||^2/||t||^2 \quad \forall z, \end{aligned} \tag{3.25}$$

where the first inequality follows from $||E|| \leq 1$ and $\sigma > 0$ denotes the smallest eigenvalue of AA^T. ($\sigma > 0$ because A has full row rank.) Hence, $\lambda \geq \epsilon\gamma/2$ yields

$$||F|| \leq 2||t||^2/(\sigma\epsilon\gamma). \tag{3.26}$$

For ϵ and γ near zero, we give below a different bound on $||F||$. By the primal nondegeneracy assumption, there exists a constant $\delta > 0$ depending on A, B, b, c, and d only such that if (x, y) is any feasible solution of $(\mathcal{P})$ with $c^T x + d^T y \leq v^* + \delta$, then the columns $\{ A_j \mid j \in \{1, \ldots, m\}$ with $x_j + B_j^T y \geq \delta \}$ have rank n. Consider the case in which $\eta^* \epsilon + 2m\gamma < \delta$, where η^* is defined as in Lemma 3.1. Since $(t, p) \in \mathcal{Y}_\epsilon$ and $\rho_{\epsilon,\gamma}(t, p, u) \leq 1$, we have from Lemma 3.1 that the columns $\{ A_j \mid j \in \{1, \ldots, m\}$ with $(t_j + A_j^T p + B_j^T u - c_j)/\epsilon \geq \delta \}$ have rank n. Since $(t, p) \in \mathcal{Y}_\epsilon$ and $\rho_{\epsilon,\gamma}(t, p, u) \leq 1$, we have $T(c - t - A^T p - B^T u) \geq -2\epsilon\gamma e$ (cf. (3.1) and (3.2)) so that $(t_j + A_j^T p + B_j^T u - c_j)/\epsilon \geq \delta$ implies $t_j \leq 2\gamma/\delta$. Hence, we obtain from (3.25) that

$$\begin{aligned} z^T(F^{-1})z &\geq \lambda \sum_{j=1}^{m} (A_j^T z)^2/(t_j)^2 \\ &\geq \lambda(\delta)^2 \sum_{t_j \leq 2\gamma/\delta} (A_j^T z)^2/(2\gamma)^2 \\ &\geq \lambda(\delta)^2 \sigma' ||z||^2/(2\gamma)^2 \quad \forall z, \end{aligned}$$

where the last inequality follows from the fact that those A_j for which $t_j \leq 2\gamma/\delta$ have rank n, and σ' is some positive scalar depending on A only. Since $\lambda \geq \epsilon\gamma/2$, we then have

$$||F|| \leq (2\gamma)^2/(\lambda(\delta)^2\sigma') \leq 8\gamma/(\epsilon(\delta)^2\sigma') = 8/(\omega(\delta)^2\sigma'),$$

where for convenience we let $\omega = \epsilon/\gamma$. Now, consider the remaining case where $\eta^*\epsilon + 2m\gamma \geq \delta$. Then, $\gamma \geq \delta/(2m)$, so part (a) and (3.26) together yield

$$\begin{aligned} ||F|| &\leq \frac{2(M_1)^2}{\sigma\epsilon\gamma}(\epsilon + \gamma + 1)^2 \\ &= \frac{2(M_1)^2}{\sigma}\left(\sqrt{\omega} + \frac{1}{\sqrt{\omega}} + \frac{1}{\gamma\sqrt{\omega}}\right)^2 \\ &\leq \frac{2(M_1)^2}{\sigma}\left(\sqrt{\omega} + \frac{1}{\sqrt{\omega}} + \frac{2m}{\delta\sqrt{\omega}}\right)^2. \end{aligned}$$

Combining the above two cases and we conclude that

$$||F|| \leq \begin{cases} K/\omega & \text{if } \eta^*\epsilon + 2m\gamma < \delta; \\ K\left(\sqrt{\omega} + 1/\sqrt{\omega}\right)^2 & \text{otherwise,} \end{cases}$$

for some positive scalar K depending on A, B, b, c, and d only. Combining the above bound with (3.24) and we conclude that $\theta_\lambda(t) \leq M_2(\sqrt{\omega} + 1/\sqrt{\omega})^4$ for some scalar $M_2 \geq 1$ depending on A, B, b, c, and d only. □

3.3. Main convergence result. By combining Lemmas 3.1 to 3.3, we obtain the following global convergence result for the QLPPF Algorithm.

THEOREM 3.4. *Suppose that* $(\mathcal{P})$ *is primal nondegenerate in the sense of Lemma* 3.3(b) *and let* $\psi(\cdot)$ *be given by* (3.21). *If in the* QLPPF *Algorithm* (t^1, p^1) *together with some* u^1 *satisfies*

$$t^1 > 0, \quad Bt^1 = d, \tag{3.27}$$

$$\theta_{\epsilon^1\gamma^1}(t^1) \leq \psi(\epsilon^1/\gamma^1), \tag{3.28}$$

$$\rho_{\epsilon^1,\gamma^1}(t^1, p^1, u^1) \leq \beta, \tag{3.29}$$

for some scalar

$$0 < \beta < \frac{1}{\psi(\epsilon^1/\gamma^1)}, \tag{3.30}$$

and if we choose

$$\alpha^k = \max\left\{ \sqrt{\frac{\theta_{\epsilon^k\gamma^k}(t^k)(\beta)^2+\sqrt{m}}{\beta+\sqrt{m}}}, \frac{1}{1+\beta\sqrt{\gamma^1/\epsilon^1}/\|b\|} \right\} \quad \forall k, \tag{3.31}$$

then $\epsilon^k \downarrow 0$, $\gamma^k \downarrow 0$ *linearly, and* $\{((t^k + A^T p^k - c)/\epsilon^k, u^k/\epsilon^k)\}$, $\{(t^k, p^k)\}$ *approach the optimal solution set of, respectively,* $(\mathcal{P})$ *and* $(\mathcal{D})$.

Proof. First note from $\psi(\omega) > 1$ for all $\omega > 0$ (see (3.21) and $M_2 \geq 1$) and (3.30) that $\beta < 1$. Also note from (2.7) that

$$\epsilon^k/\gamma^k = \epsilon^1/\gamma^1 \quad \forall k. \tag{3.32}$$

We claim that

$$(t^k, p^k) \in \mathcal{Y}_{\epsilon^{k-1}}, \quad \rho_{\epsilon^{k-1},\gamma^{k-1}}(t^k, p^k, u^k) \leq \beta, \quad \rho_{\epsilon^k,\gamma^k}(t^k, p^k, u^k) \leq \beta, \tag{3.33}$$

for all $k \geq 2$. It is easily seen by using (3.27)–(3.31) and (2.5)–(2.7) and Lemma 3.2 that (3.33) holds for $k = 2$. Suppose that (3.33) holds for all $k \leq h$, for some $h \geq 2$. Then, $(t^h, p^h) \in \mathcal{Y}_{\epsilon^{h-1}}$ and $\rho_{\epsilon^{h-1},\gamma^{h-1}}(t^h, p^h, u^h) \leq \beta < 1$. Since we also have from (2.7) that $\epsilon^h = \alpha^{h-1}\epsilon^{h-1}$ and $\gamma^h = \alpha^{h-1}\gamma^{h-1}$ and from (3.31) that $(\alpha^{h-1})^2 \geq \sqrt{m}/(1+\sqrt{m}) \geq 1/2$, we can apply Lemma 3.3(b) to conclude that

$$\theta_{\epsilon^h\gamma^h}(t^h) \leq \psi(\epsilon^{h-1}/\gamma^{h-1}) = \psi\left(\epsilon^1/\gamma^1\right), \tag{3.34}$$

where the equality follows from (3.32). Then, by (3.30), $\beta < 1/\theta_{\epsilon^h\gamma^h}(t^h)$. Since (3.33) holds for $k = h$, we also have $\rho_{\epsilon^h,\gamma^h}(t^h, p^h, u^h) \leq \beta$, so Lemma 3.2 together with (2.5)–(2.7) and (3.31), (3.32) yields

$$(t^{h+1}, p^{h+1}) \in \mathcal{Y}_{\epsilon^h}, \quad \rho_{\epsilon^h,\gamma^h}(t^{h+1}, p^{h+1}, u^{h+1}) \leq \beta, \quad \rho_{\epsilon^{h+1},\gamma^{h+1}}(t^{h+1}, p^{h+1}, u^{h+1}) \leq \beta.$$

Hence, (3.33) holds for $k = h + 1$.

Since (3.33) holds for all $k \geq 2$, we see that (3.34) holds for all $h \geq 2$. Then, by (3.30), $\theta_{\epsilon^h\gamma^h}(t^h)\beta$ is less than 1 and bounded away from 1 for all $h \geq 2$, so that, by (3.31), α^k is less than 1 and bounded away from 1 for all k. Hence $\epsilon^k \downarrow 0$, $\gamma^k \downarrow 0$ at the rate of a geometric progression. The remainder of the proof follows from (3.33) and Lemma 3.1. □

Note that instead of $\theta_{\epsilon^k\gamma^k}(t^k)$ we can use, for example, the upper bound $1/\psi(\epsilon^1/\gamma^1)$ in the formula (3.31), and linear convergence would be preserved. However, this bound is typically loose and difficult to compute. There is also the issue of finding ϵ^1, γ^1, β, (t^1, p^1) and u^1 satisfying (3.27)-(3.30), which we will address in §3.4.

3.4. Algorithm initialization. By Theorem 3.4, if the primal nondegeneracy assumption therein holds and if we can find ϵ, γ, (t, p) and u satisfying

$$t > 0, \quad Bt = d, \tag{3.35}$$

$$\theta_{\epsilon\gamma}(t) \leq \psi(\epsilon/\gamma), \tag{3.36}$$

$$\rho_{\epsilon,\gamma}(t, p, u) < 1/\psi(\epsilon/\gamma), \tag{3.37}$$

then we can set $\beta = \rho_{\epsilon,\gamma}(t, p, u)$ (assuming $\rho_{\epsilon,\gamma}(t, p, u) \neq 0$) and start the QLPPF Algorithm with ϵ, γ, (t, p), and we would obtain linear convergence. But how do we find such ϵ, γ, (t, p), and u?

One obvious way is to fix any $\epsilon > 0$, any $\gamma > 0$, and then solve the penalized problem $(\mathcal{D}_{\epsilon,\gamma})$. The solution (t,p) obtained satisfies

$$t > 0, \qquad Bt = d, \quad \nabla f_{\epsilon,\gamma}(t,p) + \begin{pmatrix} B^T u \\ 0 \end{pmatrix} = 0$$

for some u (see (2.2)) so, by (2.3) and (3.1), (3.2), we have $\rho_{\epsilon,\gamma}(t,p,u) = 0$ and $(t,p) \in \mathcal{Y}_\epsilon$. Hence (3.35) and (3.37) hold and, by Lemma 3.3(b), $\theta_{\epsilon\gamma}(t) \leq \psi(\epsilon/\gamma)$, so (3.36) also holds. (Of course $(\mathcal{D}_{\epsilon,\gamma})$ need not be solved exactly.) To solve the problem $(\mathcal{D}_{\epsilon,\gamma})$, we can use any method for convex differentiable minimization (e.g., gradient descent, coordinate descent), and we would typically want ϵ small and γ large so that $(\mathcal{D}_{\epsilon,\gamma})$ is well conditioned.

Suppose that there holds $Be = 0$ and $Ae = b$. (This holds, for example, when B is the zero matrix (which corresponds to the case when all equality constraints are penalized), and a change of variable $x' = (\bar{X})^{-1}x$, where $\bar{x}$ is any interior feasible solution of $(\mathcal{P})$ (i.e., $A\bar{x} = b$, $\bar{x} > 0$) and $\bar{X} = Diag(\bar{x})$, has been made in $(\mathcal{P})$.) Then we can find a usable ϵ, γ, (t,p), u immediately: Fix any $\epsilon > 2\psi(1)(||c|| + ||B^T(BB^T)^{-1}d||)$ and let $\gamma = \epsilon$. Also let $w = B^T(BB^T)^{-1}d$ and

$$\begin{aligned} p &= (AA^T)^{-1}A(c - w), \\ t &= \epsilon e + w, \\ u &= -(BB^T)^{-1}d. \end{aligned}$$

Then $Bw = d$, $A(c - A^T p) = Aw$, $At = \epsilon b + Aw$ and $B^T u = \epsilon e - t$, so that

$$\begin{aligned} Bt = Bw \ &= \ d, \\ A(c - t - A^T p) &= -\epsilon b, \\ T(c - t - A^T p - B^T u) + \epsilon\gamma e &= (\epsilon I + W)(c - \epsilon e - A^T p) + (\epsilon)^2 e \\ &= (\epsilon I + W)(c - A^T p) - \epsilon w, \end{aligned}$$

where $T = \mathrm{Diag}(t)$ and $W = \mathrm{Diag}(w)$. Also from $\psi(1) > 1$ and our choice of ϵ we have $\epsilon > ||w||$, so $t > 0$. Hence, by (3.2), we have $(t,p) \in \mathcal{Y}_\epsilon$ and, by (3.1) and $\epsilon = \gamma$, we have

$$\begin{aligned} \rho_{\epsilon,\gamma}(t,p,u) &= ||(\epsilon I + W)(c - A^T p) - \epsilon w||/(\epsilon)^2 \\ &\leq ||c - w - A^T p||/\epsilon + ||W||||c - A^T p)||/(\epsilon)^2 \\ &= ||(I - A^T(AA^T)^{-1}A)(c - w)||/\epsilon \\ &+ ||W||||(I - A^T(AA^T)^{-1}A)c + A^T(AA^T)^{-1}Aw||/(\epsilon)^2 \\ &\leq ||c - w||/\epsilon + ||w||(||c|| + ||w||)/(\epsilon)^2, \end{aligned} \tag{3.38}$$

where the last inequality follows from the triangle inequality and the nonexpansive property of projection matrices. From our choice of ϵ we see that $(||c|| + ||w||)/\epsilon < .5/\psi(1)$, so the right-hand side of (3.38) is bounded above by $.5/\psi(1) + (.5/\psi(1))^2 \leq 1/\psi(1) = 1/\psi(\epsilon/\gamma)$, where the inequality follows from $\psi(1) \geq 1$ and the equality follows from $\gamma = \epsilon$. Hence (3.35) and (3.37) hold. Also, since $(t,p) \in \mathcal{Y}_\epsilon$, then (3.37) together with Lemma 3.3(b) shows that (3.36) holds.

4. Numerical results. To study the performance of the QLPPF Algorithm in practice, we have implemented the algorithm to solve the special case of $(\mathcal{D})$ and $(\mathcal{P})$

in which B is the zero matrix, i.e., $(\mathcal{P})$ is of the form

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } Ax = b,\ x \geq 0. \end{aligned} \tag{4.1}$$

(This corresponds to penalizing all equality constraints in the corresponding dual problem.) Below we describe our implementation and present our very preliminary numerical experience.

1. Initialization. In our implementation, we set for all problems

$$\epsilon^1 = 10^{-7}||c||_1/m, \qquad \gamma^1 = 10^4||c||_1/m,$$

and set (somewhat arbitrarily)

$$p^1 = 0, \qquad t^1 = \max\{e, c/2\},$$

where "max" is taken componentwise. (Note that since B is the zero matrix, t^1 can be set to any positive vector.) We have chosen t^1 so to minimize $||T^1(c - t^1 - A^T p^1)||$ subject to $t^1 \geq e$. Also, care must be exercised in choosing ϵ^1 and γ^1: if their values are set too low, then the QLPPF Algorithm may fail to converge; if their values are set too high, then the QLPPF Algorithm may require many iterations to converge. (Notice that we set ϵ^1 and γ^1 directly proportional to the average cost $||c||_1/m$ so their values scale with c.)

2. Steplength selection. To ensure that the t^k's remain inside the positive orthant, we employ a backtracking scheme similar to that used by Setiono: whenever $t^k + \Delta t^k$ is outside the positive orthant, we replace the formula for t^{k+1} in (2.6) by

$$t^{k+1} = t^k + .98\lambda^k \Delta t^k, \tag{4.2}$$

where

$$\lambda^k = \min_{\Delta t_j^k < 0} -\frac{t_j^k}{\Delta t_j^k}. \tag{4.3}$$

However, this raises a difficulty, namely, for λ^k much smaller than 1, the vector $(.98\lambda^k\Delta t^k, \Delta p^k)$ may be far from the Newton direction $(\Delta t^k, \Delta p^k)$ and, as a consequence, the iterates may fail to converge. To remedy this, we replace (analogous to (4.2)) the formula for p^{k+1} in (2.6) by

$$p^{k+1} = p^k + .98\lambda^k \Delta p^k,$$

whenever nonconvergence is detected. (The parameter value .98 is chosen somewhat arbitrarily, but it works well in our tests.)

The proper choice of the α^k's is very important for the QLPPF Algorthm: if the α^k's are too near 1 (so the penalty parameters decrease slowly), then the algorithm would converge slowly; if the α^k's are too near 0 (so the penalty parameters decrease rapidly), then the algorithm may fail to converge. In our implementation we adjust the α^k's dynamically according to the following rule:

$$\alpha^k = \begin{cases} \max\{.3, .95\alpha^{k-1}\} & \text{if } \lambda^k = 1; \\ .6 & \text{if } \lambda^k \leq .2; \\ \alpha^{k-1} & \text{otherwise} \end{cases} \quad \forall k \geq 2,$$

with α^1 set to .5. The rational for this adjustment rule is that, if $\lambda^k = 1$, then the current iterate is closely following the solution trajectory (so we can decrease the penalty parameters at a faster rate and still retain convergence) and, if $\lambda^k \leq .2$, then the current iterate is unable to follow the solution trajectory (so we must decrease the penalty parameters at a slower rate).

3. Termination. To avoid numerical problems, we stop decreasing the penalty parameters ϵ and γ when they reach some prespecified tolerances $\epsilon_{\min}$ and $\gamma_{\min}$, respectively. In our tests we set

$$\epsilon_{\min} = 10^{-12}||c||_1/m, \qquad \gamma_{\min} = 10^{-9}||c||_1/m.$$

We terminate the QLPPF Algorithm when the relative duality gap and the violation of primal and dual feasibility are small. More specifically, we terminate whenever the current iterate, denoted by (t, p), satisfies

$$\frac{||X(A^Tp - c)||_1}{|c^Tx|} \leq 10^{-7}, \tag{4.4}$$

$$\max\{||Ax - b||_\infty, ||[-x]_+||_\infty\} \leq 10^{-7}, \tag{4.5}$$

$$||[A^Tp - c]_+||_\infty \leq 10^{-7}, \tag{4.6}$$

where $x = (t + A^Tp - c)/\epsilon$ (see Lemma 3.1), $X = \text{Diag}(x)$, and $[\cdot]_+$ denotes the orthogonal projection onto the nonnegative orthant. Only for three of our test problems could the above termination criterion not be met (owing to violation of (4.4) and (4.6)) in which case the algorithm is terminated whenever primal feasibility (4.5) is met and $|c^Tx - v^*|/|v^*|$ is less than $5 \cdot 10^{-7}$, where v^* denotes the optimal cost of (4.1).

4. Solving for the direction. The most expensive computation at each iteration of the QLPPF Algorithm lies in solving the system of linear equations (2.5). This can be seen to entail solving a single linear system of the form

$$AQA^Tw = z, \tag{4.7}$$

for w, where z is some n-vector and Q is some $m \times m$ diagonal matrix whose jth diagonal entry is

$$\frac{\epsilon\gamma}{\epsilon\gamma + (t_j)^2}, \tag{4.8}$$

with $\epsilon > 0$, $\gamma > 0$, and t some positive m-vector. (Linear system of the form (4.7) also arise in interior point algorithms, but (4.7) has the nice property that the condition number of Q can be controlled by adjusting the penalty parameters ϵ and γ.) In our implementation, (4.7) is solved using YSMP, a sparse matrix package for symmetric positive semidefinite systems developed at Yale University (see [EGSS79], [EGSS82]) and a precursor to the commercial package SMPAK (Scientific Computing Associates, 1985). YSMP comprises a set of Fortran routines implementing the Cholesky decomposition scheme and, as a preprocessor, the minimum-degree ordering algorithm (see, e.g., [GeL81]). In our implementation, the minimum-degree ordering routine ORDER is called first to obtain a permutation of the rows and columns of the matrix AA^T so that fill-in is reduced during factorization. Then, AA^T is symbolically factored using the routine SSF. (SSF is called only once since the nonzero pattern of AQA^T does not change with Q.) At each iteration, the matrix AQA^T is numerically factored by the routine SNF (taking advantage of information generated by SSF concerning the location of the nonzeros in the factorization), and the two triangular systems thus generated are solved by the routine SNS to obtain a solution of (4.7). (We also experimented with the public domain version of the sparse matrix package SPARSPAK [GeL81], presently avaliable from Netlib. We found SPARSPAK to be comparable to YSMP in solution time but somewhat inferior in solution accuracy.)

5. Data structure. The data structure used in our implementation is similar to that described in [AKRV89] and [MoM87]. Each matrix is stored in sparse format by row. To compute the nonzero entries of the matrix AQA^T efficiently for any Q, we also store the nonzero entries of the outer products $A_j(A_j)^T$, where A_j denotes the jth column of A. AQA^T is then computed using the formula

$$AQA^T = \sum_{j=1}^{m} q_j A_j (A_j)^T,$$

where q_j denotes the jth diagonal entry of Q and the product of q_j is taken with each nonzero entry of $A_j(A_j)^T$.

6. Test problems. Our test problems comprise the first 25 of the Netlib linear programming problems (see [Gay86]) used in the test of Monma and Morton [MoM87].[2] These problems range in size from 27 rows and 51 columns up to 1042 rows and 2869 columns and, for some of them, slack columns must be added and null rows must be removed to transform them into the form (4.1). (We also wrote a routine to convert these problems from their original MPS format to that used by our implementation.) The statistics for the test problems (after problem transformation) are summarized in Table 1.

7. Computing environment. Our implementation was written in Fortran and was compiled and ran on a Decstation 5000 under the Ultrix 4.2 operating system (similar to Berkeley 4.2 with some 4.3 enhancements). The default optimization setting for the compiler was used.

Table 2 summarizes the computational results obtained with our implementation of the QLPPF algorithm. Columns 2 and 3 show, respectively, the total number of iterations and the CPU time. Columns 4 and 6 show, respectively, the cost and the accuracy of the final primal solution (the latter is measured by the left-hand quantity in (4.5)). Analogously, columns 5 and 7 show, respectively, the cost and the accuracy of the final dual solution (the latter is measured by the left-hand quantity in (4.6)). For most of the problems, the primal cost agrees with the optimal cost in the first 7 digits and the accuracy of the primal solution is between 10^{-7} and 10^{-14}. Thus the quality of the computed solutions compares favorably with that of solutions generated by interior point algorithms. The number of iterations varies between 29 and 48 (except for Scagr25 which required 62 iterations) and the CPU time varies between 0.3 and 22 seconds, depending on the problem size and the sparsity of the constraint matrix. For most of the problems, over half of the CPU time is devoted to solving the linear system (4.7) at every iteration. (We also performed tests on a μVAX-2000 Workstation under the operating system VMS 4.1. The resulting number of iterations is roughly the same; the accuracy of the final solutions improves slightly; and the CPU times are from 10 to 15 times that on the Decstation 5000.)

The number of iterations for the QLPPF Algorithm is comparable to that for the projected Newton barrier method of Gill et al. [GMSTW86], but is typically more

[2] Possibly due to conversion error, our version of Scorpion has an optimal cost very different from that reported in [Gay86] and hence the problem is excluded.

TABLE 1
Test problem characteristics

Problem Name	Number of Rows	Number of Cols	Constraint Nonzeros[1]	Hessian Nonzeros[2]	Optimal Cost[3]
Afiro	27	51	102	90	-4.6475314E+2
Adlittle	56	138	424	384	2.2549496E+5
Scagr7	129	185	465	629	-2.3313892E+6
Sc205	205	317	665	656	-5.2202061E+1
Share2B	96	162	777	871	-4.1573224E+2
Share1B	117	253	1179	1001	-7.6589319E+4
Scagr25	471	671	1725	2393	-1.4753433E+4
ScTap1	300	660	1872	1686	1.4122500E+3
BrandY	193	303	2202	2734	1.5185099E+3
Scsd1	77	760	2388	1133	8.6666666E+0
Israel	174	316	2443	11227	-8.9664482E+5
BandM	305	472	2494	3724	-1.5862801E-2
Scfxm1	330	600	2732	3233	1.8416759E+4
E226	223	472	2768	2823	-1.8751929E+1
Scrs8	490	1275	3288	2198	9.0429695E+2
Beaconfd	173	295	3408	2842	3.3592486E+4
Scsd6	147	1350	4316	2099	5.0500000E+1
Ship04s	360	1506	4400	3272	1.7987147E+6
Scfxm2	660	1200	5469	6486	3.6660261E+4
Ship04l	360	2166	6380	4588	1.7933245E+6
Ship08s	712	2467	7194	5440	1.9200982E+6
ScTap2	1090	2500	7334	6595	1.7248071E+3
Scfxm3	990	1800	8206	9739	5.4901254E+4
Ship12s	1042	2869	8284	6387	1.4892361E+6
Scsd8	397	2750	8584	4280	9.0499999E+2

[1] The number of nonzero entries in A.

[2] The number of nonzero entries in AA^T.

[3] Cited from [Gay86].

than that for the affine-scaling algorithm or for Setiono's algorithm. Specifically, by comparing column 3 of [MoM87, Table 5] (also see [BDDW89], [MSSPB89]) with column 2 of Table 2, we see that the number of iterations for the QLPPF Algorithm can be up to $\frac{3}{2}$ times that for the affine-scaling algorithm. Similarly, the number of iterations for the QLPPF Algorithm can be up to $\frac{5}{3}$ times that for Setiono's algorithm (compare column 3 of [Set89, Table 3] with column 2 of Table 2). On the other hand, there are some problems on which the number of iterations is less for the QLPPF Algorithm than for the other algorithms.

In conclusion, our computational results indicate that, for linear programming, a mixed interior point-exterior point penalty method, as exemplified by the QLPPF Algorithm, can perform near the level of interior point algorithms. On the other hand, we caution that these results are very preliminary and thus should be viewed only as encouraging. In particular, the performance of the QLPPF Algorithm can be very sensitive to the choice of the initial parameters and the initial iterate. Of course, this also gives us hope that the performance of the QLPPF Algorithm can be improved with further fine tuning.

Finally, we remark it is typically beneficial to operate the QLPPF Algorithm with a large ratio of γ^k/ϵ^k. An intuitive explanation for this is that, if γ^k/ϵ^k is small, then γ^k is not a sufficiently large penalty (relative to ϵ^k) to maintain $t^k + \Delta t^k$ within the positive orthant. This results in small stepsizes λ^k (cf. (4.3)) and hence slow convergence.

TABLE 2
Computational results for the QLPPF *Algorithm.*

Problem Name	**Iters.**	**CPU (sec.)**[1]	**Primal Cost**[2]	**Dual Cost**[2]	**Primal Feas.**[3]	**Dual Feas.**[3]
Afiro	29	0.30	-4.6475313E+2	-4.6475316E+2	2E-08	0
Adlittle	37	0.64	2.2549499E+5	2.2549497E+5	3E-11	8E-09
Scagr7	43	0.98	-2.3313889E+6	-2.3313912E+6	3E-10	0
Sc205	36	1.07	-5.2202055E+1	-5.2202058E+1	1E-10	6E-12
Share2B	33	0.99	-4.1573226E+2	-4.1573228E+2	1E-10	0
Share1B	38	1.48	-7.6589327E+4	-7.6585942E+4	8E-08	2E-06
Scagr25	62	6.39	-1.4753429E+7	-1.4753437E+7	3E-09	9E-07
ScTap1	40	2.67	1.4122500E+3	1.4122499E+3	3E-12	0
BrandY	35	3.62	1.5185099E+3	1.5185098E+3	6E-08	1E-11
Scsd1	37	2.65	8.6666685E+0	8.6666668E+0	2E-15	0
Israel	44	21.10	-8.9664482E+5	-8.9659623E+5	4E-08	5E-05
BandM	37	4.31	-1.5862801E+2	-1.5862802E+2	4E-10	9E-10
Scfxm1	36	4.78	1.8416759E+4	1.8416758E+4	2E-08	3E-09
E226	43	4.71	-1.8751926E+1	-1.8751927E+1	7E-12	8E-11
Scrs8	48	7.43	9.0429699E+2	9.0429694E+2	6E-12	1E-08
Beaconfd	29	3.78	3.3592487E+4	3.3592485E+4	1E-09	3E-09
Scsd6	39	3.87	5.0500003E+1	5.0500000E+1	3E-13	0
Ship04s	35	4.43	1.7987148E+6	1.7987147E+6	8E-09	9E-09
Scfxm2	36	10.15	3.6660263E+4	3.6660260E+4	6E-08	3E-09
Ship04l	35	6.32	1.7933246E+6	1.7933245E+6	1E-10	0
Ship08s	40	9.68	1.9200982E+6	1.9200981E+6	2E-12	0
ScTap2	45	22.52	1.7248071E+3	1.7248071E+3	6E-13	3E-10
Scfxm3	36	17.93	5.4901256E+4	5.4901252E+4	2E-08	3E-09
Ship12s	38	12.44	1.4892362E+6	1.4892361E+6	1E-11	0
Scsd8	38	8.80	9.0500002E+2	9.0499999E+2	2E-13	0

[1] Obtained using the intrinsic function SECNDS on the Decstation 5000; does not include time to read the problem.

[2] Shown first 8 digits only.

[3] Shown first digit only.

5. Some extensions. We have thus far assumed that the parameters ϵ and γ are decreased at the same rate in the QLPPF Algorithm. Alternatively we can decrease them at different rates. For example, Setiono's algorithm employs the strategy whereby ϵ is first decreased with γ held fixed and, once ϵ reaches a prescribed tolerance, then γ is decreased with ϵ held fixed. (In [Set89], the product $\epsilon\gamma$ is what is referred to as γ.) We can also use different penalty parameters for different coordinates.

Our convergence results very possibly also extend to linear complementarity problems with positive semi-definite matrices—in the same manner that the results in [Tse89] can be extended to these problems (see [Tse92]). This is a topic for further study.

Acknowledgment. I thank Dr. R. Setiono for providing me with a copy of the YSMP code and for helpful discussions regarding implementation. I also thank Professor D. P. Bertsekas for his support of this project and the referees for their helpful comments.

REFERENCES

[AKRV89] I. ADLER, N. KARMARKAR, M. G. C. RESENDE, AND G. VEIGA, *Data structures and programming techniques for the implementation of Karmarkar's algorithm*, ORSA J. Comput., 1 (1989), pp. 84–106.

[Ber82] D. P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, NY, 1982.

[BDDW89] P. T. BOGGS, P. D. DOMICH, J. R. DONALDSON, AND C. WITZGALL, *Algorithmic enhancements to the method of centers for linear programming problems*, ORSA J. Comput., 1 (1989), pp. 159–171.

[EGSS79] S. C. EISENSTAT, M. C. GURSKY, M. H. SCHULTZ, AND A. H. SHERMAN, *Yale Sparse Matrix Package I. The Symmetric Codes*, Research Report #112, Department of Computer Science, Yale University, New Haven, CT (1979).

[EGSS82] ———, *Yale Sparse Matrix Package I. The symmetric codes*, Internat. J. Numer. Methods Engrg., 18 (1982), pp. 1145–1151.

[FiM68] A. V. FIACCO, AND G. P. MCCORMICK, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley, New York, NY, 1968.

[Fri57] K. R. FRISCH, *The Logarithmic Potential Method of Convex Programming*, Memorandom, University Institute of Economics, Oslo, Norway, May, 1955.

[GaZ81] C. B. GARCIA, AND W. I. ZANGWILL, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[Gay86] D. M. GAY, *Electronic Mail Distribution of Linear Programming Test Problems*, Manuscript 86-0, Department of Numerical Analysis, AT&T Bell Laboratories, Murray Hill, NJ, August, 1986.

[GeL81] J. A. GEORGE AND J. W.-H. LIU, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[GMSTW86] 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*, Math. Prog., 36 (1986), pp. 183–209.

[Gon89] C. C. GONZAGA, *An algorithm for solving linear programming problems in $O(n^3L)$ operations*, in Progress in Mathematical Programming: Interior-Point and Related Methods, N. Megiddo, Ed., Springer-Verlag, New York, 1989, pp. 1–28.

[HaB70] P. C. HAARHOFF AND J. D. BUYS, *A new method for the optimization of a nonlinear function subject to nonlinear constraints*, Computer J., 13 (1970), pp. 178–184.

[Hes69] M. R. HESTENES, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.

[Hua67] P. HUARD, *Resolution of mathematical programming with nonlinear constraints by the method of centers*, in Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1967, pp. 207–219.

[JiO78] K. JITTORNTRUM, AND M. R. OSBORN, *Trajectory analysis and extrapolation in barrier function methods*, J. Austral. Math. Soc. Series B, 20 (1978), pp. 352–369.

[Kar84] N. KARMARKAR, *A new polynomial-time algorithm for linear programming*, Combinatorica, 4 (1984), pp. 373–395.

[KMY89] M. KOJIMA, S. MIZUNO, AND A. YOSHISE, *A polynomial-time algorithm for a class of linear complementarity problems*, Math. Prog., 44 (1989), pp. 1–26.

[KMM91] M. KOJIMA, N. MEGIDDO, AND S. MIZUNO, *A Primal-Dual Exterior Point Algorithm for Linear Programming*, Technical Report, Mathematics and Computer Science Department, IBM Almaden Research Center, Almaden, CA, November, 1991.

[Lus89] H. LUSS, ED., *Special Issue on Optimization*, AT&T Technical Journal, 68 (1989).

[Lus90] I. J. LUSTIG, *Feasibility issues in a primal-dual interior-point method for linear programming*, Math. Prog., 49 (1990), pp. 145–162.

[Man84] O. L. MANGASARIAN, *Normal solutions of linear programs*, Math. Prog. Study, 22 (1984), pp. 206–216.

[MSSPB89] R. E. MARSTEN, M. J. SALTZMAN, D. F. SHANNO, G. S. PIERCE, AND J. F. BALLINTIJN, *Implementation of a dual affine interior point algorithm for linear programming*, ORSA J. Comput., 1 (1989), pp. 287–297.

[MMS89] K. A. MCSHANE, C. L. MONMA, AND D. SHANNO, *An implementation of a primal-dual interior point method for linear programming*, ORSA J. Comput., 1 (1989), pp. 70–83.

[MeS90] S. MEHROTRA, AND J. SUN, *An algorithm for convex quadratic programming that requires $O(n^{3.5}L)$ arithmetic operations*, Math. Oper. Res., 15 (1990), pp. 342–363.

[Meg89] N. MEGIDDO, ED., *Progress in Mathematical Programming: Interior-Point and Related Methods*, Springer-Verlag, New York, NY, 1989.

[MoM87] C. L. MONMA AND A. J. MORTON, *Computational experience with a dual affine variant of Karmarkar's method for linear programming*, Oper. Res. Letters, 6 (1987), pp. 261–267.

[MoA87] R. C. MONTEIRO AND I. ADLER, *Interior path following primal-dual algorithms - part* I: *Linear programming*, Math. Prog., 44 (1989), pp. 27–41.

[MuR90] J. M. Mulvey, and A. Ruszczyński, *A Diagonal Quadratic Approximation Method for Large Scale Linear Programs*, Technical Report, Department of Civil Engineering and Operations Research, Princeton University, NJ, September, 1990.

[PaS82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, NJ, 1982.

[Pow69] M. J. D. Powell, *A Method for Nonlinear Constraints in Minimization Problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 283–298.

[Ren88] J. Renegar, *A polynomial-time algorithm, based on Newton's method, for linear programming*, Math. Prog., 40 (1988), pp. 59–93.

[Roc70] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.

[Roc76] ———, *Augmented Lagrangians and applications for the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[Set89] R. Setiono, *An Interior Dual Proximal Point Algorithm for Linear Programs*, Technical report, Computer Sciences Department, University of Wisconsin, Madison, WI, September, 1989.

[Set90] ———, *Least Norm and Proximal Point Interior Algorithms for Linear Programming*, Ph.D. thesis, Computer Sciences Department, University of Wisconsin, Madison, WI, August, 1990.

[Set91] ———, *Interior Dual Least 2-Norm Algorithm for Linear Programs*, Technical Report, Computer Sciences Department, University of Wisconsin, Madison, WI (July 1991); SIAM J. Control Optim., 31 (1993), pp. 875–899.

[Tod89] M. J. Todd, *Recent Developments and New Directions in Linear Programming*, in Mathematical Programming: Recent Developments and Applications, M. Iri and K. Tanabe, eds., KTK Scientific Publishers, Tokyo, 1989, pp. 109–154.

[Tse89] P. Tseng, *A simple complexity proof for a polynomial-time linear programming algorithm*, Oper. Res. Letters, 8 (1989), pp. 155–159.

[Tse92] ———, *Complexity analysis of a linear complementarity algorithm based on a Lyapunov function*, Math. Prog., 53 (1992), pp. 297–306.

[Ye87] Y. Ye, *Further Development on the Interior Algorithm for Convex Quadratic Programming*, Technical Report, Integrated Systems Inc., Santa Clara, CA, November, 1987.

[WBD88] C. Witzgall, P. T. Boggs, and P. D. Domich, *On the Convergence Behavior of Trajectories for Linear Programming*, Technical Report, National Bureau of Standards, Boulder, CO, August, 1988.