Exactness Conditions for a Convex Differentiable Exterior Penalty for Linear Programming

O. L. Mangasarian[1,2], E. W. Wild[1]

[1] Computer Sciences Department, University of Wisconsin, Madison, WI 53706, {*olvi,wildt*}*@cs.wisc.edu*
[2] Department of Mathematics, University of California at San Diego, La Jolla, CA 92093.

Abstract Sufficient conditions are given for a classical dual exterior penalty function of a linear program to be independent of its penalty parameter. This ensures that an exact solution to the primal linear program can be obtained by minimizing the dual exterior penalty function. The sufficient conditions give a precise value to such a penalty parameter introduced in (Mangasarian, 2005), where no quantification of the parameter was given. Computational results on linear programs with up to one million variables or constraints compare favorably to CPLEX 9.0 (ILO, 2003) and validate the proposed approach.

Keywords: linear programming, exact penalty function, explicit penalty parameter magnitude

Introduction

In (Mangasarian, 2005), a classical exterior penalty formulation for the dual of a linear program was shown to yield an exact solution to the primal linear program provided that the penalty parameter was sufficiently large, but finite. However, no precise value for the penalty parameter was given that would guarantee an exact primal solution. In the present work we give sufficient conditions on the penalty parameter so that the corresponding computed primal variable is an exact solution. In Section 1 we derive our sufficient conditions for the penalty parameter to be large enough to generate an exact primal solution. Section 2 briefly details a generalized Newton algorithm as well as a new and fast iterative algorithm that solves a sequence of linear equations. Both of these methods will be used to obtain our numerical results. Section 3 gives sufficient conditions for dual penalty exactness for primal linear programs with nonnegative variables. In Section 4 we give numerical results for linear programs with up to one million variables or constraints. These results show that our penalty parameter magnitude condition is satisfied by the exact solutions obtained. Section 5 concludes the paper.

We now describe our notation and give some background material. All vectors will be column vectors unless transposed to a row vector by a prime $'$. For a vector x in the n-dimensional real space R^n, x_+ denotes the vector in R^n with all of its negative components set to zero. This corresponds to projecting x onto the nonnegative orthant. For a vector $x \in R^n$, $sign(x)$ denotes a vector in R^n with components of ± 1 for positive and negative components of x respectively and 0 for zero components of x. Thus, for a vector $x \in R^n$ the expression $diag\Big(sign\big((-x)_+\big)\Big)$ will denote a diagonal matrix of ones and zeros in $R^{n\times n}$ where the zeros correspond to nonnegative components of x and ones correspond to negative components of x. For $x \in R^n$, $\|x\|$ will denote the 2−norm of x. For simplicity we drop the 2 from $\|x\|_2$. The notation $A \in R^{m\times n}$ will signify a real $m \times n$ matrix. For such a matrix A' will denote the transpose of A, A_i will denote the i-th row of A and A_{ij} will denote the ij-th element of A. A vector of ones or zeroes in a real space of arbitrary dimension will be denoted by e or 0, respectively. For a piecewise-quadratic function, such as $f(x) = \frac{1}{2}||(Ax-b)_+||^2 + \frac{1}{2}x'Px$, where $A \in R^{m\times n}$, $P \in R^{n\times n}$, $P = P'$, P positive semidefinite and $b \in R^m$, the ordinary Hessian does not exist because its gradient, the $n \times 1$ vector $\nabla f(x) = A'(Ax-b)_+ + Px$, is not differentiable but is Lipschitz continuous with a Lipschitz constant of $\|A'\|\,\|A\| + \|P\|$. However, one can define its **generalized Hessian** (Hiriart-Urruty et al., 1984; Facchinei, 1995; Mangasarian, 2001) which is the $n \times n$ symmetric positive semidefinite matrix:

$$\partial^2 f(x) = A'diag\Big(sign\big((Ax-b)_+\big)\Big)A + P,$$

where $diag\Big(sign\big((Ax-b)_+\big)\Big)$ denotes an $m \times m$ diagonal matrix with diagonal elements $sign\big((A_ix-b_i)_+\big)$, $i = 1,\ldots,m$. The generalized Hessian has many of the properties of the regular Hessian (Hiriart-Urruty et al., 1984; Facchinei, 1995; Mangasarian, 2001) in relation to $f(x)$. If the smallest eigenvalue of $\partial^2 f(x)$ is greater than some positive constant for all $x \in R^n$, then $f(x)$ is a strongly convex piecewise-quadratic function on R^n. The abbreviation "*s.t.*" stands for "subject to."

1 Sufficient Conditions for Dual Exterior Penalty Function Exactness

We shall consider the solvable linear program (LP):

$$\min_{y\in R^\ell} d'y \;\; s.t. \;\; By \ge b, \tag{1}$$

where $d \in R^{\ell}$, $B \in R^{m\times\ell}$, $b \in R^m$, and its dual:

$$\max_{u\in R^m} b'u \;\; s.t. \;\; B'u = d, \;\; u \geq 0. \tag{2}$$

The classical asymptotic exterior penalty problem for the dual linear program is:

$$\min_{u\in R^m} \epsilon(-b'u) + \frac{1}{2}(\|B'u - d\|^2 + \|(-u)_+\|^2). \tag{3}$$

Dividing Equation (3) by ϵ^2 and letting:

$$\frac{u}{\epsilon} \rightarrow u, \;\; \alpha = \frac{1}{\epsilon}, \tag{4}$$

Equation (3) becomes:

$$\min_{u\in R^m} f(u) = -b'u + \frac{1}{2}(\|B'u - \alpha d\|^2 + \|(-u)_+\|^2). \tag{5}$$

Applying Proposition 1 of (Mangasarian, 2005) to the dual exterior penalty problem (5) we get:

Proposition 1 Exact Primal Solution Computation *Let the primal LP (1) be solvable. Then the dual exterior penalty problem (5) is solvable for all $\alpha > 0$. For any $\alpha \geq \bar{\alpha}$ for some finite $\bar{\alpha} > 0$, any solution u of (5) generates an exact solution to primal LP (1) as follows:*

$$y = B'u - \alpha d. \tag{6}$$

In addition, this y minimizes:

$$\|y\|^2 + \|By - b\|^2, \tag{7}$$

over the solution set of the primal LP (1).

What we are after in the present work are sufficient conditions to ensure that $\alpha \geq \bar{\alpha}$, which we proceed to obtain now by first establishing the following lemma based on the convexity of the penalty function (5).

Lemma 1 Optimality Condition for Solving the Exterior Penalty (5) *A necessary and sufficient condition for u to be a solution of the exterior penalty function (5) is that:*

$$\nabla f(u) = -b + B(B'u - \alpha d) + Pu = 0, \tag{8}$$

where $P \in R^{m\times m}$ is a diagonal matrix of ones and zeros defined as follows:

$$P = diag\Big(sign\big((-u)_+\big)\Big). \tag{9}$$

Proof Setting the gradient of the convex differentiable exterior penalty function (5) equal to zero gives:

$$-b + B(B'u - \alpha d) - (-u)_+ = 0, \tag{10}$$

which, upon making use of the definition (9) of P, gives (8). $\square$

Note that the matrix P in the condition (8) depends on u and hence cannot be used directly to solve for u. However, P will be used to derive the conditions needed to ensure that $\alpha \geq \bar{\alpha}$, as we shall proceed to do now. We note first that by Proposition 1, Equation (8) is solvable for any $\alpha > 0$ whenever the linear program (1) is solvable. Hence, for such a case an explicit solution to (8) is given as follows:

$$u = (BB' + P)\backslash(\alpha Bd + b), \tag{11}$$

where the MATLAB (MATLAB, 1994-2006) backslash notation $u = H\backslash h$ denotes a solution to a solvable system of linear equations $Hu = h$. If we now substitute for u in (6) we obtain the following expression for the solution y of the primal linear program (1) when $\alpha \geq \bar{\alpha}$:

$$y = B'\Big((BB' + P)\backslash b\Big) + \alpha\Big(B'\big((BB' + P)\backslash(Bd)\big) - d\Big). \tag{12}$$

We note immediately that y depends on α through the explicit term α in the above Equation (12), as well as through the dependence of P on u in its definition (9), and the dependence of u on α through (11). Hence, in order for y to be independent of α, as would be the case when $\alpha \geq \bar{\alpha}$, we have the following result which follows directly from the expression (12) for y.

Proposition 2 Sufficient Conditions for Independence on α *The optimal solution y given by (12) for a solvable primal linear program (1) is independent of α if:*

$$\textit{The diagonal matrix } P \text{ is independent of } \alpha \tag{13}$$

and

$$B'\big((BB' + P)\backslash(Bd)\big) - d = 0. \tag{14}$$

We note immediately that condition (14) is easy to verify once the exterior penalty problem (3) is solved. However, although condition (13) is difficult to verify without varying α, it turns out that in all our computational results presented in this work, condition (14) is satisfied whenever an exact solution to the linear program (1) is obtained. Hence, for all intents and purposes, (13) can be ignored. Similarly, even though we have not established the necessity of conditions (13) and (14), the necessity of (14) appears to hold in our computational results.

We now describe our computational algorithms.

2 Computational Algorithms

In this section we present two algorithms for solving the linear program (1) by solving the dual exterior penalty problem (5). Both algorithms utilize the sufficient condition (14) for independence on the penalty parameter α as a stopping criterion.

2.1 Generalized Newton Algorithm

Our first algorithm will make use of the generalized Newton algorithm, utilized in (Mangasarian, 2005) for 1-norm support vector machine classification problems, to solve our unconstrained minimization problem (5) with an appropriate value of the penalty parameter α by utilizing the new sufficient condition (14) for exactness. For that purpose we define $f(u)$ as the objective function of (5), that is:

$$f(u) = -b'u + \frac{1}{2}(\|B'u - \alpha d\|^2 + \|(-u)_+\|^2). \tag{15}$$

The gradient and generalized Hessian as defined in the Introduction are given as follows.

$$\nabla f(u) = -b + B(B'u - \alpha d) - (-u)_+. \tag{16}$$

$$\partial^2 f(u) = BB' + diag\Big(sign\big((-u)_+\big)\Big). \tag{17}$$

We now incorporate our new sufficient condition (14) into the generalized Newton algorithm (Mangasarian, 2005, Algorithm 3) for solving the unconstrained minimization problem (5). In particular the sufficient condition (14) will be used as a stopping criterion in Step (II) of the following Algorithm 1, as well as in Algorithm 2 further on.

Algorithm 1 Generalized Newton Algorithm for (5) *Let $f(u)$, $\nabla f(u)$ and $\partial^2 f(u)$ be defined by (15),(16) and (17). Set the parameter values α, δ, tol and imax (typically $\alpha = 100$, $\delta = 1e{-}6$, tol=1e−3 and imax=5000). Start with a random $u^0 \in R^m$. For $i = 0, 1, \ldots$:*

(I) $u^{i+1} = u^i - \lambda_i(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i) = u^i + \lambda_i t^i$,
where the Armijo stepsize $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ *is such that:*

$$f(u^i) - f(u^i + \lambda_i t^i) \geq -\frac{\lambda_i}{4}\nabla f(u^i)' t^i, \tag{18}$$

and t^i is the modified Newton direction:

$$t^i = -(\partial^2 f(u^i) + \delta I)^{-1}\nabla f(u^i). \tag{19}$$

In other words, start with $\lambda_i = 1$ and keep multiplying λ_i by $\frac{1}{2}$ until (18) is satisfied.

(II) Stop if $\|\nabla f(u^i)\| \leq tol$ *& norm*$\Big(B'\big((BB'+P^i)\backslash(Bd)\big)-d\Big) \leq tol$ *where* $P^i = diag\Big(sign\big((-u^i)_+\big)\Big)$.
(III) If $i = imax$ *then* $\alpha \to 10\alpha$, $imax \to 2 \cdot imax$
(IV) $i \to i+1$ *and go to (I)*

The iterates u^i of the above algorithm either terminate or converge as follows (Mangasarian, 2005, Proposition 4).

Proposition 3 Generalized Newton Algorithm 1 Convergence *Let* $tol = 0$ *and assume that (14) implies that* $\alpha \geq \bar{\alpha}$*. Then either Algorithm 1 terminates at an* i *such that* u^i *solves the exterior penalty problem (5) and consequently* $y^i = B'u^i - \alpha d$ *solves the primal linear program (1), or any accumulation point* $\bar{u}$ *of the sequence* $\{u^i\}$ *generated by Algorithm 1 solves the exterior penalty problem (5) and the corresponding* $\bar{y} = B'\bar{u} - \alpha d$ *solves the primal linear program (1).*

Proof If Algorithm 2 terminates at some i, then, by the stopping criterion (II), $\nabla f(u^i) = 0$, and $B'\big((BB'+P^i)\backslash(Bd)\big)-d = 0$. Hence, $P^{i+1} = P^i$, condition (14) is satisfied for P^i, and u^i solves (8). Consequently, u^i solves the dual penalty minimization problem (5), and since, by assumption, condition (14) implies that $\alpha \geq \bar{\alpha}$, it follows that $y^i = B'u^i - \alpha d$ solves the primal linear program (1).

For the case when $\{i\}$ does not terminate, since the corresponding sequence of diagonal matrices of ones and zeros $\{P^i\}$ has a finite number of possible configurations, at least one such configuration must occur infinitely often and, by (Mangasarian, 2005, Proposition 4), for any accumulation point $\bar{u}$ of $\{u^i\}$ there exists a subsequence $\{u^{i_j}, P^{i_j}\}$ with constant P^{i_j} that converges to $(\bar{u}, \bar{P})$ such that $\nabla f(\bar{u}) = 0$ and $B'\big((BB'+\bar{P})\backslash(Bd)\big)-d = 0$. Consequently, $\bar{u}$ solves the minimization problem (15) and since, by assumption, (14) implies that $\alpha \geq \bar{\alpha}$, it follows that $\bar{y} = B'\bar{u} - \alpha d$ solves the primal linear program (1). □

We now give a considerably simpler algorithm based on the successive direct solution of a linear equation obtained from the sufficient condition (8).

2.2 Direct Linear Equation (DLE) Algorithm

Our direct linear equation algorithm consists of successively solving the necessary and sufficient optimality condition (8) for updated values of the diagonal matrix P. The sufficient condition (14) for independence on α is satisfied in our computational examples.

Algorithm 2 DLE: Direct Linear Equation Algorithm for (8) *Set the parameter values* α, δ, *tol*, *tol1 and imax (typically* $\alpha = 100$, $\delta = 1e{-}8$, *tol*$=1e{-}16$, *tol1*$=1e{-}3$ *and imax=500). Start with a random* $u^0 \in R^m$*. For* $i = 0, 1, \ldots$*:*

(I) Set P^i to:

$$P^i = diag\Big(sign\big((-u^i)_+\big)\Big). \tag{20}$$

(II) Solve for u^{i+1} as follows:

$$u^{i+1} = (BB' + P^i)\backslash(b + \alpha Bd). \tag{21}$$

(III) $u^{i+1} \rightarrow \big(u^i + \lambda_i(u^{i+1} - u^i)\big)$, where the Armijo stepsize $\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \ldots\}$ is such that:

$$f(u^i) - f\big(u^i + \lambda_i(u^{i+1} - u^i)\big) \geq -\frac{\lambda_i}{4}\nabla f(u^i)'(u^{i+1} - u^i). \tag{22}$$

(IV) Stop if $\|u^{i+1} - u^i\| \leq tol$ & $norm\Big(B'\big((BB' + P^i)\backslash(Bd)\big) - d\Big) \leq tol1$.
(V) If $i = imax$ then $\alpha \rightarrow 10\alpha$, $imax \rightarrow 2 \cdot imax$.
(VI) $i \rightarrow i + 1$ and go to (I).

We state now the following convergence result for the DLE Algorithm 2. The result is based on the fact that the direction $(u^{i+1} - u^i)$ is a descent direction for the dual penalty function (15), and satisfies the convergence conditions required by (Mangasarian, 1995, Theorem 2.1) .

Proposition 4 DLE Algorithm 2 Convergence *Let $tol = tol1 = 0$ and assume that (14) implies $\alpha \geq \bar{\alpha}$. Then either Algorithm 2 terminates at an i such that u^i solves (5) and consequently $y^i = B'u^i - \alpha d$ solves the primal linear program (1), or any accumulation point $\bar{u}$ of the sequence $\{u^i\}$ generated by Algorithm 2 solves the exterior penalty problem (5) and the corresponding $\bar{y} = B'\bar{u} - \alpha d$ solves the primal linear program (1), provided the matrices of the sequence $\{(BB' + P^i)\}$ are nonsingular.*

Proof If Algorithm 2 terminates at some i, then $u^{i+1} = u^i$ and $B'\big((BB' + P^i)\backslash(Bd)\big) - d = 0$. Hence $P^{i+1} = P^i$, $u^i = (BB' + P^i)\backslash(b + \alpha Bd)$, and (14) holds for $u = u^i$ and $P = P^i$. Thus the pair (u^i, P^i) solves (8). Consequently, u^i solves the dual penalty minimization problem (5), and since, by assumption, condition (14) implies that $\alpha \geq \bar{\alpha}$, it follows that $y^i = B'u^i - \alpha d$ solves the primal linear program (1).

For the case when $\{i\}$ does not terminate, for each accumulation point $\bar{u}$ of $\{u^i\}$, there exists a subsequence $\{u^{i_j}, P^{i_j}\}$ of $\{u^i, P^i\}$ for which $\{P^{i_j}\}$ is constant and equal to $\bar{P}$, and $\{u^{i_j}, P^{i_j}\}$ converges to $(\bar{u}, \bar{P})$. This is because $\{P^i\}$ is a sequence of diagonal matrices of ones and zeros with a finite number of values. We will now show that $\nabla f(\bar{u}) = 0$ and consequently $\bar{u}$ minimizes the dual exterior penalty function (15).

From (8) we have that:

$$\nabla f(u^i) = (BB' + P^i)u^i - (\alpha Bd + b). \tag{23}$$

By the assumption that $(BB' + P^i)$ is nonsingular, it follows from (21), before the Armijo stepsize is taken, that:

$$u^{i+1} = (BB' + P^i)^{-1}(\alpha Bd + b). \tag{24}$$

Define now the direction t^i as:

$$t^i := u^{i+1} - u^i = (BB' + P^i)^{-1}(\alpha Bd + b) - u^i. \tag{25}$$

Consequently, after some algebra we have that:

$$-\nabla f(u^i)'t^i = u^{i'}(BB'+P^i)u^i - 2u^{i'}(\alpha Bd+b) + (\alpha Bd+b)'(BB'+P^i)^{-1}(\alpha Bd+b). \tag{26}$$

We will now show that the right hand side of (26) is a forcing function of $\nabla f(u^i)$, that is, it is nonnegative and if it approaches zero then $\|\nabla f(u^i)\|$ approaches zero. Define the quantities a and M as:

$$a = (\alpha Bd + b),\ M = (BB' + P^i). \tag{27}$$

Then, Equation (26) reduces to:

$$-\nabla f(u^i)'t^i = u^{i'}Mu^i - 2u^{i'}a + a'M^{-1}a. \tag{28}$$

Define now

$$h = M^{-\frac{1}{2}}a,\ g = M^{\frac{1}{2}}u^i. \tag{29}$$

Hence

$$\begin{aligned} -\nabla f(u^i)'t^i &= g'g - 2g'M^{-\frac{1}{2}}M^{\frac{1}{2}}h + h'h = \|g-h\|^2 \\ &= \|M^{\frac{1}{2}}u^i - M^{-\frac{1}{2}}a\|^2 \\ &= \|M^{-\frac{1}{2}}(Mu^i - a)\|^2 \\ &= \|M^{-\frac{1}{2}}\big((BB'+P^i)u^i - (\alpha Bd+b)\big)\|^2 \\ &= \|(BB'+P^i)^{-\frac{1}{2}}\nabla f(u^i)\|^2. \end{aligned} \tag{30}$$

The last term of (30) is a forcing function of $\nabla f(u^i)$. Hence, by (Mangasarian, 1995, Theorem 2.1) it follows that for each accumulation point $\bar{u}$, $\nabla f(\bar{u}) = 0$, and for a subsequence $\{u^{i_j}, P^{i_j}\}$ with constant P^{i_j} that converges to $(\bar{u}, \bar{P})$, the pair $(\bar{u}, \bar{P})$ solves (8) and $B'\big((BB'+\bar{P})\backslash(Bd)\big) - d = 0$. Consequently, $\bar{u}$ solves the minimization problem (5) and since (14) implies that $\alpha \geq \bar{\alpha}$, it follows that $\bar{y} = B'\bar{u} - \alpha d$ solves the primal linear program (1). $\square$

Before presenting our computational results we state results corresponding to those of Section 1 for linear programs with explicit nonnegative variable constraints. We do this because it will allow us to find an exact solution for the linear program (2), that is the dual of (1), which will allow us to efficiently obtain solutions of (1) when $m >> \ell$.

3 Linear Programs with Nonnegative Variables

In solving the linear program (1) for the case when $m >> \ell$, the algorithms of Section 2 are quite inefficient because these algorithms involve the inversion of very large m-by-m matrices that contain BB' in (19) or (21). To avoid this difficulty we shall instead first solve the dual linear program (2) exactly by solving an exterior penalty problem for the primal linear program

(1). As shown below, this involves the inversion of much smaller ℓ-by-ℓ matrices. Once an exact dual solution u is obtained an exact primal solution is obtained by solving the following unconstrained minimization problem:

$$\min_{y\in R^\ell} \frac{1}{2}\|B^1y-b^1\|^2+\frac{1}{2}\|(-B^2y+b^2)_+\|^2+\frac{1}{2}(d'y-b'u)^2, \tag{31}$$

where the superscripts 1 and 2 denote components of the primal constraints $By \geq b$ corresponding to optimal dual variable components $u^1 > 0$ and $u^2 = 0$ respectively. It is easy to check that solving (31) involves the inversion of much smaller ℓ-by-ℓ matrices, as will be the case for the computations below.

In order to develop the above approach, we begin with the solvable dual linear program (2) with nonnegative variables (hence the title of this section) and consider the classical exterior penalty function for the corresponding primal problem (1):

$$\min_{y\in R^\ell} \epsilon d'y+\frac{1}{2}\|(-By+b)_+\|^2. \tag{32}$$

Dividing by ϵ^2 and letting:

$$\frac{y}{\epsilon}\rightarrow y \textit{ and } \frac{1}{\epsilon}\rightarrow \alpha, \tag{33}$$

the penalty problem (32) becomes:

$$\min_{y\in R^\ell} d'y+\frac{1}{2}\|(-By+\alpha b)_+\|^2. \tag{34}$$

Applying Proposition 1 of (Mangasarian, 2005) to the primal exterior penalty problem (34), we get:

Proposition 5 Exact Dual Solution Computation *Let the dual LP (2) be solvable. Then the primal exterior penalty problem (34) is solvable for all* $\alpha > 0$. *For any* $\alpha \geq \bar{\alpha}$ *for some finite* $\bar{\alpha} > 0$, *any solution* y *of (34) generates an exact solution to dual LP (2) as follows:*

$$u = (-By+\alpha b)_+. \tag{35}$$

In addition, this u *minimizes:*

$$\|u\|^2, \tag{36}$$

over the solution set of the dual LP (2).

Without giving proofs of results similar to those of Section 1, we now state results corresponding to Lemma 1 and Proposition 2.

Lemma 2 Optimality Condition for Solving the Primal Exterior Penalty (34) *A necessary and sufficient condition for u to be a solution of the exterior penalty function (34) is that:*

$$d - B'Q(-By + \alpha b) = 0, \tag{37}$$

where $Q \in R^{\ell \times \ell}$ is a diagonal matrix of ones and zeros defined as follows:

$$Q = diag\Big(sign\big((-By + \alpha b)_+\big)\Big). \tag{38}$$

We note that Equation (37) is solvable for any $\alpha > 0$ whenever the linear program (1) is solvable. Hence, for such a case an explicit solution to (37) is given as follows:

$$y = (B'QB)\backslash(\alpha B'Qb - d). \tag{39}$$

If we now substitute for y in (35) we obtain the following expression for an exact solution u of the dual linear program (2) when $\alpha \geq \bar{\alpha}$:

$$u = \Bigg(B\Big((B'QB)\backslash d\Big) - \alpha\Big(B\big((B'QB)\backslash(B'Qb)\big) - b\Big)\Bigg)_+ \tag{40}$$

We note that u depends on α through the explicit term α in the above Equation (40), as well as through the dependence of Q on y in the definition (38), and the dependence of y on α in (39). Hence, in order for u to be independent of α, as would be the case when $\alpha \geq \bar{\alpha}$, we have the following result which follows directly from the expression (40) for u being a constant function of α: for a constant Q, the subgradient (Rockafellar, 1970) of the expression for u (40) with respect to α vanishes as stated in (42) below.

Proposition 6 Sufficient Conditions for Independence on α *The optimal solution u given by (40) for a solvable dual linear program (2) is independent of α if:*

$$\textit{The diagonal matrix } Q \text{ is independent of } \alpha \tag{41}$$

and

$$diag\Big(sign(u)\Big)\ \Big(B\big((B'QB)\backslash(B'Qb)\big) - b\Big) = 0, \tag{42}$$

where u is defined in (40).

We turn now to our computational results based on Algorithms 1 and 2.

4 Computational Results

To illustrate the effectiveness of our approach, we report results on randomly generated linear programs. In these linear programs we selected the number of variables, constraints, nonzero coefficients, and generated optimal values for the primal and dual variables. We compare our Newton LP approach (Algorithm 1) and direct linear equation (DLE) approach (Algorithm 2) with CPLEX 9.0 (ILO, 2003), a state-of-the-art linear programming package.

All methods were run in a random order on each linear program. The reported results were obtained on an Intel Pentium 4 processor with 1 gigabyte of RAM running CentOS Linux 4.5 and MATLAB 7.3. For the DLE method, we set $\delta = 1e{-}8$, $tol = 1e{-}16$, $tol1 = 1e{-}3$, $imax = 500$, and $\alpha = 100$. For the Newton LP method, we set $\delta = 1e{-}6$, $tol = 1e{-}3$, $imax = 5000$, and $\alpha = 100$. For problems with more constraints than variables, we used $imax = 500$ for the Newton LP method. If the sufficient condition was not less than tol after the Newton method terminated, we tried again with 10α. This case occurred only a few times over all the experiments we report. We used only stepless methods. Although these parameters effect the solution time and accuracy, we did not experiment much with different settings and chose these particular values because they gave acceptable results on our experiments. Default parameters were used for CPLEX.

The sufficient conditions (14) and (42) play an important role in our algorithms in choosing the size of the penalty parameter α and as a stopping criterion. In fact, in all our tests the penalty parameter α was chosen sufficiently large so that it satisfied the appropriate sufficient condition (14) or (42). In every experiment we report, the relative difference between the returned and the true objective value, maximum constraint violation, and sufficient condition was less then $1e{-}3$, and in most experiments these values were of the order of $1e{-}7$ or smaller.

4.1 Square linear programs

Figure 1 shows results for our proposed Newton LP Algorithm 1 and DLE Algorithm 2 formulations, as well as CPLEX 9.0. Each point on the graph represents the average elapsed seconds over 10 square linear programs randomly generated with density 0.1. The standard deviations are too small to show on the plot, and our proposed approaches are both clearly faster than CPLEX for problems with more than 2000 variables and constraints. In general, CPLEX returned more accurate solutions than our approaches. We used the barrier method for CPLEX, which was faster than both the primal and dual simplex algorithms for these problems. Note that both of our approaches are more than twice as fast as CPLEX for problems with 5000 variables and 5000 constraints.

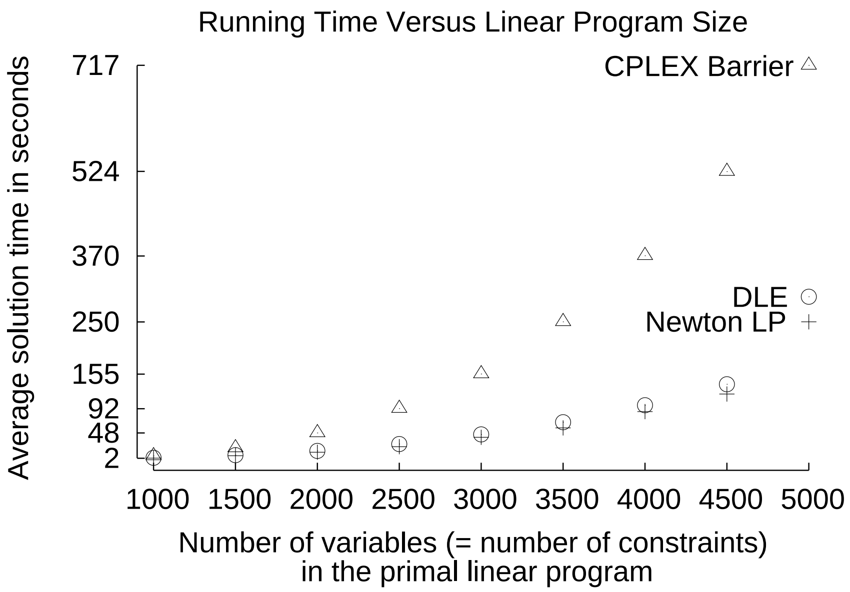

Fig. 1 **Average running times of our proposed approaches and the CPLEX 9.0 barrier method. Our Newton LP Algorithm 1 method is represented by '+', our DLE Algorithm 2 method is represented by '○', and CPLEX is represented by '△'. Each point is the average of 10 randomly generated square linear programs.**

4.2 Rectangular linear programs

Table 1 shows results for our proposed Newton LP Algorithm 1 and DLE Algorithm 2 methods, as well as CPLEX 9.0. We applied Algorithms 1 and 2 directly for problems with more variables than constraints, and used the technique described in Section 3 for problems with more constraints than variables. We found that the primal simplex method of CPLEX performed best for problems with more variables than constraints, while the dual simplex method of CPLEX performed best for problems with more constraints than variables. The results in Table 1 show that our approach can solve problems with up to a million variables or constraints, although not as quickly as the commercially available CPLEX solvers in all cases.

We note that our proposed DLE Algorithm 2 method, which is implemented in 54 lines of MATLAB code, gives times which are faster than CPLEX for problems with more variables than constraints. In general, our proposed Newton method is slower on these problems, but is still able to give solutions in about half a minute. For problems with more constraints than variables, CPLEX is noticeably faster. Nevertheless, our proposed approaches are able to give accurate solutions to problems with as many as one million constraints and one hundred variables within a few minutes.

Constraints	Variables	CPLEX	Newton LP	DLE
100	1,000	0.0246 (0.0006)	0.0438 (0.0003)	0.0251 (0.0001)
100	10,000	0.0522 (0.0001)	0.2024 (0.0004)	0.0682 (0.0001)
100	100,000	0.837 (0.003)	3.219 (0.012)	0.905 (0.004)
100	1,000,000	17.9 (0.6)	29.1 (0.2)	9.3 (0.2)
1,000	100	0.0969 (0.0025)	0.1391 (0.0027)	0.1032 (0.0004)
10,000	100	0.267 (0.001)	0.970 (0.012)	0.469 (0.003)
100,000	100	2.96 (0.05)	13.02 (0.12)	5.50 (0.04)
1,000,000	100	44.8 (0.3)	173.5 (2.2)	70.9 (0.4)

Table 1 **Average running times of our proposed approaches and the CPLEX 9.0 simplex method. Ten linear programs were randomly generated for each number of variables and constraints, and the average solution time in seconds is given with the standard deviation in parentheses for each algorithm. Primal methods were used for problems with more variables than constraints, and dual methods were used for problems with more constraints than variables.**

5 Conclusion and Outlook

We have presented sufficient conditions for a classical dual exterior penalty function of a linear program to provide an exact solution to a primal linear program. These conditions allow us to give a precise termination condition to a Newton algorithm for linear programming introduced in (Mangasarian, 2005), and also to a new direct method based on solving the necessary and sufficient optimality condition (8). Experimental results indicate that both approaches are able to efficiently obtain accurate solutions on randomly generated linear programs with different numbers of constraints and variables. For some linear programs, our approaches implemented in MATLAB were as much as twice as fast as the commercial linear programming package CPLEX 9.0.

It is possible that our approaches could be extended to handle problems which are too big to fit in memory, so long as either the number of constraints or the number of variables is not too big, and the matrix multiplications are done externally. Other opportunities for future work include applying both approaches to real world linear programs, exploring the use of direct methods for other optimization problems, and further improving the performance of our approaches.

Acknowledgements The research described in this Data Mining Institute Report 07-01, July 2007, was supported by National Science Foundation Grants CCR-0138308 and IIS-0511905, the Microsoft Corporation, and ExxonMobil.

References

F. Facchinei. Minimization of SC^1 functions and the Maratos effect. *Operations Research Letters*, 17:131–137, 1995.

J.-B. Hiriart-Urruty, J. J. Strodiot, and V. H. Nguyen. Generalized Hessian matrix and second-order optimality conditions for problems with C^{L1} data. *Applied Mathematics and Optimization*, 11:43–56, 1984.

ILOG CPLEX 9.0 User's Manual. ILOG, Incline Village, Nevada, 2003. http://www.ilog.com/products/cplex/.

O. L. Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM Journal on Control and Optimization*, 33(6):1916–1925, 1995. ftp://ftp.cs.wisc.edu/tech-reports/reports/1993/tr1145.ps.

O. L. Mangasarian. A finite Newton method for classification problems. Technical Report 01-11, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, December 2001. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-11.ps.*Optimization Methods and Software* 17, 2002, 913-929.

O. L. Mangasarian. Exact 1-Norm support vector machines via unconstrained convex differentiable minimization. Technical Report 05-03, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, August 2005. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/05-03.ps. Journal of Machine Learning Research 7, 2006, 1517-1530.

MATLAB. *User's Guide*. The MathWorks, Inc., Natick, MA 01760, 1994-2006. http://www.mathworks.com.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.