A note on the use of vector barrier parameters for interior-point methods

Javier M. Moguerza [a,*], Alberto Olivares [a], Francisco J. Prieto [b]

[a] *School of Engineering, University Rey Juan Carlos, C/ Tulipan s/n, 28933 Mostoles, Madrid, Spain*
[b] *Department of Statistics, University Carlos III, C/ Madrid 126, 28903 Getafe, Madrid, Spain*

Abstract

A key feature to ensure desirable convergence properties in an interior point method is the appropriate choice of an updating rule for the barrier parameter. In this work we analyze and describe updating rules based on the use of a vector of barrier parameters. We show that these updating rules are well defined and satisfy sufficient conditions to ensure convergence to the correct limit points. We also present some numerical results that illustrate the improved performance of these strategies compared to the use of a scalar barrier parameter.

Keywords: Interior point methods; Primal dual methods; Non convex optimization; Central path

1. Introduction

Interior-point methods can be used to compute local solutions for nonlinear, and possibly non-convex, problems of the form

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & c(x) = 0, \\ & x \geqslant 0, \end{aligned} \tag{1}$$

where $f\colon \mathbb{R}^n \mapsto \mathbb{R}$ and $c\colon \mathbb{R}^n \mapsto \mathbb{R}^m$. These methods have proved to be very successful for the solution of linear and general convex problems. Recently, a significant amount of effort has been devoted to extending these procedures to non-convex problems, see for example El-Bakry et al. [2], Gajulapalli [4], Gay et al. [5],

[*] Corresponding author. Tel.: +34 91 488 7189; fax: +34 91 488 7338.

E mail addresses: javier.moguerza@urjc.es (J.M. Moguerza), alberto.olivares@urjc.es (A. Olivares), franciscojavier.prieto@uc3m.es (F.J. Prieto).

Vanderbei and Shanno [15], Yamashita [17], Tits et al. [14], Moguerza and Prieto [10], among others. These methods proceed by (approximately) solving a sequence of equality-constrained problems of the form

$$\begin{aligned} \min_x \quad & f(x) - \mu \sum_i \log x_i \\ \text{s.t.} \quad & c(x) = 0, \end{aligned} \tag{2}$$

for a sequence $\{\mu^{(k)}\}$ of values of the scalar barrier parameter μ such that $\mu^{(k)} \to 0$.

An appropriate choice of values for the parameter μ may have a significant impact on the practical performance of the algorithm, both on its convergence (a sequence that converges to zero at an excessively fast rate may imply numerical difficulties and lack of convergence), and its rate of convergence (a slowly convergent sequence will imply a slow algorithm). Several updating rules have been proposed in the literature, such as for example the widely used rule described in El-Bakry et al. [2].

In this paper we are interested in exploring the possibility of using a vector barrier parameter, $\mu \in \mathbb{R}^n$, to improve the practical convergence properties of the algorithm, and particularly its speed of convergence. A vector parameter should be better adapted to the possibly heterogeneous convergence behavior of the different variables in the problem. The use of vector parameters has been analyzed in the solution of linear problems by Nesterov and Nemirovskii [13] and Jansen et al. [8], among others. These approaches, although interesting from a theoretical point of view, did not result in major practical breakthroughs. A reason why these approaches have not been very successful in practice is that they increase the algorithmic complexity of the methods. Our goal is to design a robust and efficient updating strategy for a vector barrier parameter to be used within an interior-point method for the general nonlinear case.

As our interest focuses on this parameter, we will carry out the studies in this paper considering different definitions within the framework of the algorithm proposed by Moguerza and Prieto [10], an interior-point linesearch method using directions of negative curvature. The search directions are computed to approximate the solutions of the barrier problems, similar to (2),

$$\begin{aligned} \min_x \quad & f(x) - \sum_i \mu_i \log x_i \\ \text{s.t.} \quad & c(x) = 0. \end{aligned} \tag{3}$$

In the remainder of the paper we will use the bracketed superindex (k) to denote iterations and the subscript i to indicate each one of the components of the vector μ in (3).

The procedure to combine the directions is related to that in Moré and Sorensen [12]. A linesearch has been introduced in our algorithm as a mechanism to enforce good global convergence properties. We compute the iterates in such a manner that the value of an augmented Lagrangian merit function is decreased in each iteration. For problem (3) this merit function takes the form

$$L_A(x, \lambda; \rho, \mu) = f(x) - \sum_i \mu_i \log x_i - \lambda^T c(x) + \frac{1}{2} \sum_j \rho_j c_j(x)^2. \tag{4}$$

Under suitable assumptions the local minimizers for problem (3) are minimizers for this merit function, if all components of ρ are large enough and the vector λ corresponds to the optimal multipliers, see Bertsekas [1], for example.

The paper is organized as follows: In Section 2, we introduce the notation used in this work. Section 3 describes the general results that motivate our proposals. In Section 4, we present several updating strategies for the vector of barrier parameters μ and comment on some of their theoretical properties. Section 5 includes the results of a numerical comparison among the proposed strategies on a problem test set. Section 6 introduces some conclusions.

2. Notation and background

The first-order Karush Kuhn Tucker (KKT) conditions for problem (1) are:

$$\begin{aligned}&\nabla f(x)-\nabla c^T(x)\lambda-z=0,\\&c(x)=0,\\&Zx=0,\\&x,z\geqslant 0,\end{aligned}\tag{5}$$

where λ and z are the multipliers for the equality and bound constraints respectively, and Z denotes a diagonal matrix having as entries the elements of z, $Z \quad \mathrm{diag}(z)$.

In the algorithm proposed in Moguerza and Prieto [10], instead of considering directly the preceding conditions, a sequence of problems (3) such that $\mu_i \to 0$ for all i is solved. The first-order KKT conditions for (3) are:

$$\begin{aligned}&\nabla f(x)-\nabla c^T(x)\lambda-X^{-1}\mu=0,\\&c(x)=0,\end{aligned}\tag{6}$$

where $X \quad \mathrm{diag}(x)$. Replacing

$$z=X^{-1}\mu\tag{7}$$

in the first equation of (6), the first-order KKT conditions for the barrier problem can be rewritten as

$$\begin{aligned}&\nabla f(x)-\nabla c^T(x)\lambda-z=0,\\&c(x)=0,\\&Zx=\mu.\end{aligned}\tag{8}$$

The set of equations (8) is known as the primal-dual equations for problem (3), see Mehrotra [9] for example. The algorithm will compute search directions based on these primal-dual KKT equations. In addition to trying to satisfy these conditions, to ensure that the logarithmic terms in the objective function of (3) are well defined, the algorithm should force the variables x_i to remain strictly positive. From the comparison of these conditions and (5), it is also of interest to satisfy $z \geqslant 0$ in all iterations of the algorithm. A brief summary of the algorithm is given in the following Box.

```
Initialize variables ($x^{(0)}, \lambda^{(0)}, z^{(0)}$), barrier ($\mu^{(0)}$) and penalty ($\rho^{(0)}$) parameters
repeat
  From the Newton primal-dual equations:
    Compute a descent direction, $d_x^{(k)}$, for the primal variables $x$
    Compute search directions, $d_\lambda^{(k)}$ and $d_z^{(k)}$, for the multipliers $\lambda$ and $z$
  Compute, if it exists, $d_n^{(k)}$, a direction of negative curvature
  Adjust the penalty parameter $\rho^{(k)}$
  Compute $\alpha_p$ using a curvilinear search procedure
  Update the primal variables using $x^{(k+1)} = x^{(k)} + \alpha_p^2 d_x^{(k)} + \alpha_p d_n^{(k)}$
  Update the multipliers $\lambda^{(k)}$, $z^{(k)}$
  Decrease the barrier parameter vector $\mu^{(k)}$
until convergence
```

We refer the reader to Moguerza and Prieto [10] for a detailed description of the method.

3. The vector of barrier parameters

The vector of barrier parameters in (3) is updated to ensure convergence to the desired limit points. A first global convergence result is a direct adaptation of the basic results of Fiacco and McCormick [3]. We will assume in what follows that problem (1) satisfies the following assumptions:

- Functions f and c have continuous second derivatives.
- The linear independence constraint qualification holds at all second-order KKT points.
- Strict complementarity is satisfied at all second-order KKT points.
- The second-order sufficient conditions also hold at all second-order KKT points.

We will also use the notation $x^*(\mu)$ to denote a KKT point for problem (3), for a given value of the vector barrier parameter μ. From the preceding conditions it should follow that, sufficiently close to a second-order KKT point of (1), $x^*(\mu)$ will be unique when it exists.

In Moguerza and Prieto [10] it is shown that the algorithm is globally convergent under the preceding conditions. Therefore, this convergence is assumed throughout the rest of the paper. The updating rules considered below will also satisfy the conditions in Moguerza and Prieto [10].

Theorem 1. *Under the preceding assumptions, if the sequence $\{\mu^{(k)}\}$ converges to zero, all limit points of the sequence of values $\{x^*(\mu^{(k)})\}$ are KKT points of problem (1).*

Proof. It follows from a direct adaptation of the arguments in Fiacco and McCormick [3, Theorems 8 and 10]. □

To ensure convergence, it is enough to compute the values $\{x^*(\mu^{(k)})\}$ for any sequence $\{\mu^{(k)}\}$ converging to zero. In practice, obtaining values for $x^*(\mu^{(k)})$ with high precision is not computationally efficient. The most common alternative is to approximate these values with limited precision and then update the values of $\mu^{(k)}$ along an appropriate sequence. Let $x^{(k)}$ denote the current primal iterate at iteration k and x^* denote the second-order KKT point for (1) closest to $x^*(\mu^{(k\ \ 1)})$. As

$$\|x^{(k)} - x^*\| \leqslant \|x^{(k)} - x^*(\mu^{(k-1)})\| + \|x^*(\mu^{(k-1)}) - x^*\|, \tag{9}$$

inequality (9) and Theorem 1 imply that, for a sequence $\{\mu^{(k)}\}$ such that $\mu^{(k)} \to 0$, if $\|x^{(k)} - x^*(\mu^{(k\ \ 1)})\| \to 0$ then $\|\ x^{(k)} - x^*\| \to 0$. Thus, it is enough to solve each subproblem with increasing accuracy.

To define efficient ways to implement criteria that yield the desired accuracy, we observe that Theorem 1 implies $\|x^*(\mu^{(k\ \ 1)}) - x^*\| \to 0$. Based on (9), the values of $\mu^{(k)}$ are updated along a sequence that converges to zero, only when the values of $x^{(k)} - x^*(\mu^{(k\ \ 1)})$ are small enough.

We need to decide when $x^{(k)} - x^*(\mu^{(k\ \ 1)})$ is considered to be small enough. We will make this decision taking into account our concerns about both the convergence of the procedure and its rate of convergence. In this sense, it is reasonable to require that the sizes of $x^{(k)} - x^*(\mu^{(k\ \ 1)})$ and $x^*(\mu^{(k\ \ 1)}) - x^*$ are not too different whenever we update $\mu^{(k)}$, as from (9) the rate of convergence of the algorithm would be dominated by the slowest of both sequences. Thus, we will modify the value of $\mu^{(k)}$ whenever the value of $\|x^{(k)} - x^*(\mu^{(k\ \ 1)})\|$ is comparable to that of $\|x^*(\mu^{(k\ \ 1)}) - x^*\|$.

Unfortunately, these values are not directly available in the algorithm, but the following result provides alternative reference values. Let $F(x,\lambda,z)$ be a measure of the satisfaction of the first-order KKT conditions for problem (1) at the current iterate, that is

$$F(x,\lambda,z) = \begin{pmatrix} \nabla f(x) - \nabla c(x)^T\lambda - z \\ c(x) \\ Zx \end{pmatrix} = \begin{pmatrix} \nabla f(x) - \nabla c(x)^T\lambda - z \\ c(x) \\ Zx - \mu \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \mu \end{pmatrix} = \overline{F}(x,\lambda,z;\mu) + \begin{pmatrix} 0 \\ 0 \\ \mu \end{pmatrix}, \tag{10}$$

where $\overline{F}$ denotes the corresponding satisfaction measure for problem (3). Also let

$$\theta = \begin{cases} \|F(x,\lambda,z)\|^2 & \text{if } \|F(x,\lambda,z)\| < 1, \\ \|F(x,\lambda,z)\| & \text{otherwise,} \end{cases} \qquad \theta = \begin{cases} \|\overline{F}(x,\lambda,z;\mu)\|^2 & \text{if } \|\overline{F}(x,\lambda,z;\mu)\| < 1, \\ \|\overline{F}(x,\lambda,z;\mu)\| & \text{otherwise.} \end{cases} \tag{11}$$

Far from a solution the norm is not squared in the preceding definitions to avoid unnecessarily large values for θ. Under our assumptions, these quantities are related to the distances to the respective solutions, as we show in the following lemma.

Lemma 1. *Let* $w = \begin{pmatrix} x^{\mathrm{T}} & \lambda^{\mathrm{T}} & z^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}}$. *Under the preceding assumptions, asymptotically it holds that*

$$\|w^{(k)} - w^*\|^2 = \mathrm{O}(\theta^{(k)}), \quad \|w^{(k)} - w^*(\mu^{(k)})\|^2 = \mathrm{O}(\theta^{(k)}) \quad \textit{and} \quad \|w^*(\mu^{(k)}) - w^*\| = \mathrm{O}(\|\mu^{(k)}\|),$$

where w^* *is the vector having components corresponding to a second-order KKT point of problem* (1) *closest to* w^* $(\mu^{(k)})$.

Proof. From the necessary conditions at w^* and the Taylor series expansions for $F(w)$ around w^* we have

$$F(w^{(k)}) = \begin{pmatrix} \nabla f(x^{(k)}) - \nabla c(x^{(k)})^{\mathrm{T}}\lambda^{(k)} - z^{(k)} \\ c(x^{(k)}) \\ Z^{(k)}x^{(k)} \end{pmatrix} = F(w^*) + \begin{pmatrix} \nabla_{xx}L(x^*,\lambda^*) & -\nabla c(x^*)^{\mathrm{T}} & -I \\ \nabla c(x^*) & 0 & 0 \\ Z^* & 0 & X^* \end{pmatrix}(w^{(k)} - w^*)$$
$$+ o(\|w^{(k)} - w^*\|) = F(w^*) + J(w^*)(w^{(k)} - w^*) + o(\|w^{(k)} - w^*\|).$$

The desired result for $w^{(k)} - w^*$ follows from the fact that $F(w^*) \quad 0$ and, under the assumptions, the matrix of coefficients for the linear term in the preceding expansion, $J(w^*)$, is invertible.

The equivalent result for $w^{(k)} - w^*(\mu^{(k)})$ is obtained by applying the same argument to $\overline{F}(w;\mu)$, as $\overline{F}(w^*(\mu^{(k)});\mu^{(k)}) = 0$ and the same invertible matrix $J(w^*)$ appears in the corresponding Taylor series expansion.

The third result follows from:

$$\overline{F}^{(k)}(w^*(\mu^{(k)});\mu^{(k)}) = 0 = \overline{F}(w^*;\mu^{(k)}) + J(w^*)(w^*(\mu^{(k)}) - w^*) + o(\|w^*(\mu^{(k)}) - w^*\|),$$

the invertibility of $J(w^*)$ and also from (10), implying

$$\overline{F}(w^*;\mu^{(k)}) = F(w^*) + \begin{pmatrix} 0 \\ 0 \\ \mu^{(k)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu^{(k)} \end{pmatrix}. \qquad \square$$

Our general strategy will use a modified Newton's method to solve problem (3) for fixed μ and to ensure $w^{(k)} - w^*\ (\mu^{(k)}) \to 0$. Assuming global convergence for this method, see Moguerza and Prieto [10], we can use the result in Lemma 1 jointly with (9) to define an update rule that will ensure global convergence for problem (1).

Let $\omega^{(k)} \equiv (x^{(k)})^{\mathrm{T}}z^{(k)}/n$. A reasonable update rule for $\mu^{(k)}$ has the following form:

$$\mu_i^{(k)} = \begin{cases} \mu_i^{(k-1)} & \text{if } \mu_i^{(k-1)} < \max(\phi(\omega^{(k)}), \zeta(\theta^{(k)})v_i^{(k)}), \\ \psi(\mu_i^{(k-1)}, \theta^{(k)}) & \text{otherwise,} \end{cases} \tag{12}$$

for continuous functions ϕ, ζ and ψ such that ϕ and ζ are monotonically non-decreasing functions and $\psi(x,y) \leqslant \delta x$, for some $\delta < 1$. The vector v must be chosen to satisfy the bound $\max_i v_i^{(k)} \leqslant \max(1, 1/\|y^{(k)}\|)$, where $y^{(k)} \equiv X^{(k)}z^{(k)}$. Also, we impose the conditions that $v_i^{(k)} \geqslant \beta_n \max_j v_j^{(k)}$ for all iterations k, all components i and some positive constant β_n, and $\zeta\ (x) \leqslant \beta_e x$ for all positive x and some positive constant $\beta_e < 1/n$.

We must decide how rapidly should we decrease $\mu^{(k)}$, once the condition to modify μ is satisfied, that is, how to define the function ψ in (12). This modification rule has an impact on the convergence rate for the algorithm. Again from (9) the rate of convergence of the algorithm will be defined as the slowest rate of convergence of the sequences $\{w^{(k)} - w^*(\mu^{(k\ \ 1)})\}$ and $\{w^*(\mu^{(k)}) - w^*\}$. Assuming that Newton's method is used to generate values of $w^{(k)}$ for fixed values of μ, the sequence $\{w^{(k)} - w^*(\mu^{(k\ \ 1)})\}$ will have quadratic convergence. Thus, it would seem of interest to define ψ so that both $\{\mu^{(k)}\}$ and, as a consequence of Lemma 1, also $\{w^*(\mu^{(k)}) - \ w^*\}$, converge quadratically.

Unfortunately, the quadratic convergence of this sequence may not ensure the quadratic convergence of the algorithm, as the barrier parameter is not updated in all iterations. In particular, the number of iterations where the barrier parameter is updated may be very small if the reduction in its value after each update is large. A detailed theoretical study of this issue depends on specific details of the implementation of the algorithm and lies beyond the scope of this work, see Gould et al. [6], Moguerza and Prieto [11] and Yamashita and Yabe [18] for example.

In practice, the usual approach is to reduce the value of $\mu^{(k)}$ to ensure superlinear convergence of the parameter, and thus of the sequence $\{w^*(\mu^{(k)}) - w^*\}$, at all iterates where the parameter is modified. If this decrease is slower than that of $\|w^{(k)} - w^*(\mu^{(k\;\;1)})\|$, that is, slower than quadratic, it should be possible to prove that the update will eventually take place in all iterations, and the whole sequence will converge superlinearly.

We will impose a superlinear reduction in the norm of the barrier parameter, that is, the function ψ in (12) should be such that $\psi(\mu^{(k\;\;1)},\theta^{(k)})\quad o(\|\mu^{(k\;\;1)}\|)$.

The preceding comment only provides information on the change in the norm of $\mu^{(k)}$. As we are interested in using a vector barrier parameter, the update rule needs to be completed with a specification of the modification in each component of $\mu^{(k)}$. This situation implies a major difference between our proposal and previous ones.

The basic motivation for the updating rules of scalar barrier parameters is based in the decomposition presented in (9).

Definition 1. For problem (2), the central path is defined as the set

$$\overline{C}_\mu \equiv \{(x,\lambda,z) | \exists \mu > 0, \overline{F}(x,\lambda,z;\mu e) = 0\},$$

where $e = (\,1 \quad \ldots \quad 1\,)^{\mathrm{T}} \in \mathbb{R}^n$.

In (9) the progress to the solution is decomposed into the approximation of the iterates to a point in the central path, and the movement of the reference point along the central path closer to the solution x^*.

The adjustment in the value of μ controls this last movement along the central path. The introduction of a vector barrier parameter modifies this strategy as there is no longer any reference central path, but rather a central surface.

Definition 2. For problem (3), the central surface is defined as the set

$$C_\mu \equiv \{(x,\lambda,z) | \exists \mu > 0, F(x,\lambda,z;\mu) = 0\}.$$

The following lemma and theorem show that this central surface exists under the assumptions on problem (3).

Lemma 2. *For any small enough μ it holds that problem (3) has a solution and the system of equations $\overline{F}(x,\lambda,z;\mu) = 0$ also has a solution, that is, the central surface is not empty.*

Proof. The result follows from our assumptions and an adaptation of the arguments for the linear case given in Theorem 2.8 of Wright [16]. □

Theorem 2. *Let $x^*(\mu)$ denote a solution of problem (2) for a small enough value of $\mu \in \mathbb{R}$. There exists a positive constant $\epsilon(\mu)$ and positive vectors $\mu \in \mathbb{R}^n$ such that all values x satisfying $c(x) = 0$ and $\|x - x^*(\mu)\| < \epsilon(\mu)$ are solutions of problem (3) for some appropriate vector μ.*

Proof. From the optimality conditions (5) for problems (2) and (3), we have that $x_i^*(\mu)z_i^*(\mu) = \mu > 0$, implying both $x_i^*(\mu) > 0$ and $z_i^*(\mu) > 0$. Also, $\nabla f(x^*(\mu)) - \nabla c(x^*(\mu))^{\mathrm{T}}\lambda^*(\mu) = z(\mu) > 0$.

Consider a value x such that $c(x)\quad 0$ and $\|x - x^*(\mu)\|$ is small enough that both $x_i > 0$ and $(\nabla f(x) - \nabla c(x)^{\mathrm{T}}\lambda^*(\mu))_i \equiv z_i > 0$ for all i. Then, this value x is a KKT point for problem (3) for a value of the barrier parameter $\mu\quad Xz$. □

4. Updating rules

As a consequence of the results in the preceding section, there is no simple reference to follow on the way to the solution, and a reasonable question is how to select such a reference among all the alternatives, for a given value of $\mu^{(k)}$.

Practical efficiency considerations imply that one of the main goals in the process of generating iterates on the way to the solution is to avoid getting too close to any of the non-negativity constraints. This closeness may lead to numerical instabilities and an increase in the iteration count for the algorithm. Thus, it would seem reasonable to look for updates that keep the iterates sufficiently separated from these constraints.

The equation in (8) that links the values of x and μ is the complementarity condition $Xz \quad \mu$. Our proposals will make use of this condition, since this is the usual approach in the literature, see El-Bakry et al. [2] or Yamashita and Yabe [18].

Going back to the definition of ψ, we introduce first the rule for the norm of the update. As we mentioned above, we wish to impose a condition that ensures a superlinear reduction on the size of μ. On the other hand, by Lemma 1 relating the size of θ to the distance to the solution, it would be reasonable to make use of θ in the update rule. The following lemma relates $\theta^{(k)}$ to $\|\mu^{(k)}\|$.

Lemma 3. *For fixed* μ *with* $\|\mu\| \leqslant 1$, $\theta^{(k)} \to \|\mu\|^2$.

Proof. From (10) and (11),

$$\theta^{(k)} = \|F(w^{(k)})\|^2 = \|\overline{F}(w^{(k)};\mu)\|^2 + \|\mu\|^2 + 2(Z^{(k)}x^{(k)} - \mu)^{\mathrm{T}}\mu.$$

The global convergence of the algorithm implies $\|w^{(k)} - w^*(\mu)\| \to 0$; thus, it holds that $\|\overline{F}(w^{(k)};\mu)\| \to 0$ and $Z^{(k)}x^{(k)} \to \mu$, implying the desired result. □

Due to these considerations, we have chosen to implement condition (12) and the definition of ψ in terms of θ instead of using μ. In our studies, we implement the conditions by ensuring that

$$\|\psi(\mu,\theta)\| = \mathrm{O}(\delta(\theta)\theta), \tag{13}$$

where δ is a continuous increasing function such that $0 < \delta(\theta^{(k)}) < 1$ for all $\theta^{(k)} > 0$ and $\delta\ (\theta^{(k)}) \quad o(1)$ for $\theta^{(k)} \to 0$. We also assume that the condition to modify μ is satisfied at the current iteration.

The following four steps summarize the general strategy we have considered to define the update of the barrier vector μ, that is, the form of the function ψ in (12):

1. let $\mu_l^{(k)} = \delta(\theta^{(k)})((x^{(k)})^{\mathrm{T}}z^{(k)})/n$,
2. let $\mu_u^{(k)} = \mu^{(k-1)}$,
3. let $\mu_n^{(k)} = \delta(\theta^{(k)})\theta^{(k)}v^{(k)}$ for some vector $v^{(k)}$,
4. define $\mu_i^{(k)} = \min((\mu_u^{(k)})_i, \max(\mu_l^{(k)}, (\mu_n^{(k)})_i))$.

In our implementations we have always carried out steps (1) (4), although some of them may be omitted for alternative implementations of the algorithm.

Step (1) defines a lower bound for each one of the components in $\mu^{(k)}$. It has been introduced as a consequence of the observation that, if this step is omitted, some of the components of μ may become very small with respect to others along the algorithm, causing numerical problems. To avoid this situation, we introduce a lower bound on the values of the components. This lower bound could be defined in terms of the value of $\theta^{(k)}$, but we have chosen to define it only in terms of the complementarity gap $(x^{(k)})^{\mathrm{T}}z^{(k)}$.

The definition in step (2), together with step (4), enforces the condition that all components of μ must decrease monotonically in each iteration. We have found that this monotonic decrease is helpful both from a theoretical point of view (see [10]), and also with regard to the practical behavior of the algorithm.

The most relevant step in the update of the barrier parameter is the choice of the vector $v^{(k)}$ in step (3) of the procedure. To satisfy the bound conditions in (12) we define $v^{(k)}$ from another vector $v^{(k)}$ as $v_i^{(k)} = \max(v_i^{(k)}, \beta_n\|v^{(k)}\|_\infty)$. To define v we have tried different approaches, that we describe in the following proposals.

Proposal 1. A first alternative is to select the values of the components in $v^{(k)}$ and $\mu^{(k)}$ assuming that the new values of x and z will be such that the condition $Xz \quad \mu$ is approximately satisfied. Our goal is to obtain values in the next iteration that are similar for all components of x that are converging to zero; this property would guarantee that we do not get too close to the bounds for these variables, avoiding numerical difficulties.

Consider those components of x converging to zero and define the corresponding components of μ as $\mu_i^{(k)} = \gamma z_i^{(k)}$ for some constant γ; under strict complementarity the corresponding components of z will be converging to strictly positive values, and their relative changes will be small; under these assumptions, the components of x in the next iteration should be close to $\mu_i^{(k)}/z_i^{(k)} = \gamma$, that is, these components would all be similar in value, as desired. For the other components we could apply the analogous approach to z. In general, we could define the vector $\mu_n^{(k)}$ in (3) to be proportional to the vector $y = \max(x, z)$, by selecting

$$u^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|}.$$

Proposal 2. If some components of x in the current iteration have become small (but not unreasonably so), imposing the preceding rule may force these components to increase (although in later iterations these components will have to decrease again). A better measure of scale would be needed in this case to avoid this behavior. This measure can be defined in terms of the current value of $X^{(k)}z^{(k)}$, that is, the value that should be close to $\mu^{(k\quad 1)}$, appropriately scaled. In this way, we would have a value of μ that is better adapted to those components of x that are small. Also, a less demanding adjustment is introduced for those components for which it is more difficult to satisfy the complementarity condition.

Let $y^{(k)} \quad X^{(k)}z^{(k)}$; we define the vector $\mu_n^{(k)}$ in this case from

$$u^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|}.$$

There exists an interesting alternative interpretation for this update. The value of μ_n associated to this definition is the solution of the quadratic problem

$$\begin{aligned} \min \quad & \mu^{\mathrm{T}}\mu \\ \text{s.t.} \quad & \mu^{\mathrm{T}}y^{(k)}/\|y^{(k)}\| = \delta(\theta^{(k)})\theta^{(k)}, \\ & \mu \geqslant 0. \end{aligned} \tag{14}$$

As a consequence, the preceding definition for μ_n provides the smallest possible vector such that its inner product with the vector $X^{(k)}z^{(k)}$ has the desired value for the norm of the new parameter, except when an adjustment to its smallest components is needed in (12). We allocate the value $\delta(\theta^{(k)})\ \theta^{(k)}$ (the norm of the update) to the different components of μ in a manner related to the relative size of the components of $X^{(k)}z^{(k)}$.

Proposal 3. A slight modification of the preceding rule is to assign the values of $\mu_n^{(k)}$ in a manner that is proportional to the absolute size of the components of $y^{(k)}$, instead of their relative size. We could define μ_n in terms of the modified problem

$$\begin{aligned} \min \quad & \mu^{\mathrm{T}}\mu \\ \text{s.t.} \quad & \mu^{\mathrm{T}}y^{(k)} = \delta(\theta^{(k)})\theta^{(k)}, \\ & \mu \geqslant 0. \end{aligned} \tag{15}$$

This is equivalent to defining $v^{(k)}$ in step 3 as

$$u^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|^2}.$$

Proposal 4. For comparison purposes, we have also considered a more "classical" definition, based on a scalar update. Using the update proposed in El-Bakry et al. [2], for example, we have also implemented the method corresponding to

$$u^{(k)} = \gamma^{(k)}((x^{(k)})^{\mathrm{T}}z^{(k)}/n)e, \tag{16}$$

where $\gamma^{(k)} = \min(\gamma, n/((x^{(k)})^{\mathrm{T}}z^{(k)}))$.

Note that all the preceding definitions satisfy the bound $v_i^{(k)} \leqslant \max(1, 1/\|y^{(k)}\|)$.

We define the function ϕ in (12) as

$$\phi(\theta) = \delta(\theta)\theta. \tag{17}$$

Finally, we show that these updating rules imply $\mu \to 0$, assuming that problem (3) has a solution. First we show that, under the hypothesis that for a fixed value of μ the algorithm converges to a KKT point of problem (3), the updating strategy for vector μ guarantees that at least one of its components will decrease. Then, based on this result, we show that $\mu \to 0$.

Lemma 4. *If the algorithm converges to a KKT point of problem* (3) *for fixed* μ, *then any of the preceding updating strategies for vector* μ *guarantees that its norm will be decreased after a finite number of iterations.*

Proof. Assume that there exists an iteration k such that μ is not modified in any iteration beyond k, that is, $\mu^{(k)} \quad \mu$ for all $k \geqslant k$. From our assumption that the algorithm converges to a KKT point of (3) and (10), then $X^{(k)}z^{(k)} \to \mu$ and $\omega^{(k)} \equiv (x^{(k)})^{\mathrm{T}} z^{(k)}/n \to \mu^{\mathrm{T}}e/n$. In this case, from the continuity of ϕ, $\phi(\omega^{(k)}) \to \phi(\mu^{\mathrm{T}}e/n)$, satisfying from (17)

$$\phi(\mu^{\mathrm{T}}e/n) = \delta(\mu^{\mathrm{T}}e/n)\mu^{\mathrm{T}}e/n < \mu^{\mathrm{T}}e/n \leqslant \max_i \mu_i.$$

But $\omega^{(k)} \to \mu^{\mathrm{T}}e/n$ implies the existence of an iteration $\tilde{k} \geqslant k$ such that for all $k \geqslant \tilde{k}$ it holds $\phi(\omega^{(k)}) \leqslant (1 + \delta(\mu^{\mathrm{T}}e/n))/2\mu^{\mathrm{T}}\ e/n < \max_l \mu_l$.

Consider now the behavior of $\zeta(\theta^{(k)})v^{(k)}$. From the convergence assumptions and Lemma 3 it holds that $y^{(k)} \to \mu$ and

$$\theta^{(k)} \to \begin{cases} \|\mu\|^2 & \text{if } \|\mu\| \leqslant 1, \\ \|\mu\| & \text{otherwise.} \end{cases}$$

As a consequence,

$$\theta^{(k)} \max(1, 1/\|y^{(k)}\|) \to \|\mu\|,$$

and as $\beta_e < 1/n$, using the properties of ζ in both cases there exists an iteration $\hat{k} \geqslant k$ such that

$$\zeta(\theta^{(k)})v_i^{(k)} \leqslant \beta_e \theta^{(k)} \max(1, 1/\|y^{(k)}\|) < \|\mu\|/n \leqslant \max_l \mu_l,$$

for all i and $k \geqslant \hat{k}$. Therefore, for all $k \geqslant \max(\tilde{k}, \hat{k})$ it holds that $\max(\phi(\omega^{(k)}), \zeta(\theta^{(k)})v_i^{(k)}) < \max_l \mu_l$ and this component of μ must be updated in the algorithm, contradicting our assumption that μ remains constant for all large enough iterations. □

Theorem 3. *If the algorithm converges to a KKT point of problem* (3) *for fixed values of* μ, *then the updating strategy for vector* μ *guarantees that* $\mu^{(k)} \to 0$.

Proof. By Lemma 4, $\|\mu^{(k)}\|$ will decrease after a finite number of iterations. Let $\mathscr{K}$ denote the (infinite) sequence of iterations where $\mu^{(k)}$ is modified, and let $J^{(k)}$ denote the set of components that decrease at iteration $k \in \mathscr{K}$.

Define I as the set of components of μ such that $i \in I$ if there exists an iteration index k satisfying $i \notin J^{(k)}$ for all $k \geqslant k$. Let $\bar{I} = \{1, \ldots, n\} \setminus I$, that is, the set of components in μ that change an infinite number of times in the algorithm; Lemma 4 implies that $\bar{I}$ is non-empty. Finally, define $l \equiv \max\{k \in \mathscr{K} | I \cap J^{(k)} \neq \emptyset\}$, $k \equiv \min\{k \in \mathscr{K} | k > l\}$ and $\mu \equiv \max_{j \in I} \mu_j^{(k)}$.

For all $j \in \bar{I}$, the update formula (12) and $\delta < 1$ imply $\mu_j^{(k)} \to 0$. As a consequence, there exists an iteration $\tilde{k} > k$ such that $\mu \geqslant \beta_r \max_{j \in \bar{I}} \mu_j^{(k)}$ for all $k \geqslant \tilde{k}$, where $\beta_r \quad \max(1, 1/\beta_n)$ and β_n has been defined in (12). Thus, for any $k \in \mathscr{K}$ with $k \geqslant \tilde{k}$ it holds that $\mu_i^{(k)} \geqslant \beta_r \mu_j^{(k)}$ for any $i \in I$ and $j \in \bar{I}$, and from (12) $\mu_j^{(k)} \geqslant \max(\phi(\theta^{(k)}), \zeta(\theta^{(k)})v_j^{(k)})$ for $j \in J^{(k)} \subseteq \bar{I}$.

Using $\beta_r \geqslant 1$ and $v_j^{(k)} \geqslant \beta_n v_i^{(k)}$ we then have

$$\mu_i^{(k)} \geqslant \beta_r \mu_j^{(k)} \geqslant \beta_r \max(\phi(\theta^{(k)}), \zeta(\theta^{(k)}) v_j^{(k)}) \geqslant \max(\phi(\theta^{(k)}), \zeta(\theta^{(k)}) \beta_r v_j^{(k)}) \geqslant \max(\phi(\theta^{(k)}), \zeta(\theta^{(k)}) \beta_r \beta_n v_i^{(k)})$$
$$\geqslant \max(\phi(\theta^{(k)}), \zeta(\theta^{(k)}) v_i^{(k)}),$$

and from (12) $\mu_i^{(k)}$ must be modified, contradicting our assumption that $i \notin J^{(k)}$. It must follow that $I \quad \emptyset$ and $\mu^{(k)} \to 0$. □

5. Numerical results

In this section we compare numerically the strategies described above. We have conducted a set of numerical experiments on a collection of test problems. The algorithm in Moguerza and Prieto [10] has been implemented using MATLAB 6.5 and the four alternative strategies. The test set we have considered is composed of 145 small problems included in the CUTEr collection [7], selected from those nonlinearly constrained problems having less than one hundred variables and continuous derivatives (note that exact first and second derivatives have been used). The initial points specified in CUTEr were used within the algorithm. The parameter values used in the implementation have been selected as follows:

1. The function $\delta(\theta)$ in Step (3) is defined as
   $$\delta(\theta) = \min(0.15, \exp(-1/\theta)).$$
2. The scalar γ in (16) has been defined as
   $$\gamma^{(k)} = \delta(\theta^{(k)}).$$
3. The scalars β_n and β_e, both implicit in (12), have been set equal to $10 \quad ^{8}$.
4. The termination criterion has been defined as
   $$\frac{\|F(x, \lambda, z)\|}{1 + |f(x)|} < 10^{-8}.$$
   The same parameter values were used for all problems.

Table 1 shows the results obtained using the four strategies described in the preceding section, respectively. The notation used in the Table is:

- **Prob.:** problem name.
- **Iter.:** iteration count (number of factorizations of the primal-dual system).
- **Eval.:** number of evaluations of the objective function and the constraints.

The percentage of successfully solved problems is significantly higher for all the vectorial strategies (Proposals 1 3) compared to the extensively used scalar strategy (Proposal 4). In particular, Proposal 3 solves all problems but two. On the other hand, the scalar updating rule (Proposal 4) fails to solve nine problems. In summary, all proposals based on a vector barrier parameter solve over 95% of the problems (Proposal 3 solves 98.6%), while the scalar proposal solves 93.7% of the problems (see the first row in Table 4).

Tables 2 and 3 show a comparison of iteration and function evaluation counts among the four strategies for the whole set. The average numbers of iterations and function evaluations are shown at the bottom of the tables. Proposal 3 seems to be the most efficient one, while Proposal 4 is again the least efficient one, with an average increase of 12.01 iterations and 16.94 function evaluations, an improvement of 39% and 40% respectively. These tables also show the total numbers of iterations and function evaluations.

Table 4 contains a summary of the success rates for all strategies. The rows in the table present the percentage of problems in which each proposal requires the smallest/largest number of iterations and function evaluations respectively (excluding ties). Proposal 1 is the best one regarding the number of function evaluations. Proposal 2 is the most efficient according to the smallest number of iterations. Proposal 3 is the best when considering both the largest number of iterations and function evaluations required.

Table 1
Results

Prob.	Proposal 1		Proposal 2		Proposal 3		Proposal 4	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
AIRPORT	15	15	14	14	15	15	15	15
ALSOTAME	8	8	8	8	8	8	8	8
BIGGSC4	18	19	20	20	19	19	19	19
BT13	23	23	21	25	21	25	23	23
CANTILVR	27	44	19	26	15	18	35	51
CB2	7	7	12	15	11	12	9	9
CB3	8	8	13	17	10	11	10	12
CHACONN1	8	12	7	7	9	11	9	12
CHACONN2	9	10	12	16	10	12	10	12
CONGIGMZ	23	26	32	34	33	37	28	31
CSFI1	65	97	29	40	33	46	60	98
CSFI2	56	100	>250	>350	58	62	>250	>350
DEMYMALO	33	42	12	15	13	14	26	28
DIPIGRI	8	12	9	13	9	13	9	13
DISC2	93	112	19	21	45	53	42	64
DUAL1	17	17	20	20	18	18	18	18
DUAL2	14	14	16	16	14	14	14	14
DUAL4	14	14	16	16	14	14	14	14
EXPFITA	19	19	22	22	33	33	33	33
FCCU	7	7	7	7	8	8	7	7
GIGOMEZ1	80	103	12	16	17	19	35	43
HATFLDH	12	12	13	13	12	12	12	12
HIMMELBI	27	27	23	23	26	26	26	26
HIMMELBK	56	70	23	26	25	28	25	28
HIMMELP2	16	34	19	69	16	34	16	34
HIMMELP3	15	33	15	33	15	33	15	33
HIMMELP4	22	45	25	52	24	50	24	50
HIMMELP5	61	89	55	78	68	112	39	51
HIMMELP6	30	41	39	52	41	63	19	20
HONG	7	7	8	9	7	7	8	8
HS10	11	14	13	21	15	22	11	14
HS11	7	7	7	8	7	8	7	7
HS12	12	31	10	19	10	19	12	31
HS13	34	37	>250	>350	>250	>350	>250	>350
HS14	7	8	9	15	7	10	7	8
HS15	24	28	23	26	17	17	16	16
HS16	15	16	14	15	15	17	15	17
HS17	17	19	15	15	16	16	19	23
HS18	12	12	13	15	13	15	12	12
HS19	18	18	17	17	17	17	18	18
HS20	7	8	9	11	8	9	8	10
HS21	5	5	6	6	5	5	5	5
HS21MOD	11	11	12	12	12	12	12	12
HS22	6	6	6	6	6	6	6	7
HS23	9	9	10	14	9	10	9	10
HS24	8	11	7	7	8	10	8	12
HS29	8	13	10	17	10	17	8	13
HS30	6	7	6	7	6	7	6	7
HS31	5	5	5	5	5	5	5	5
HS32	16	16	16	16	15	15	15	15
HS33	8	9	8	9	9	10	9	10
HS34	9	11	9	11	9	9	11	13
HS35	7	7	7	7	7	7	7	7
HS36	7	7	9	9	8	8	7	7
HS37	6	6	6	6	7	8	7	7

(*continued on next page*)

Table 1 (*continued*)

Prob.	Proposal 1		Proposal 2		Proposal 3		Proposal 4	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
HS41	7	7	7	7	7	7	7	7
HS43	11	14	13	19	14	23	11	14
HS44	9	9	9	9	9	9	9	9
HS44NEW	9	9	9	9	9	9	9	9
HS53	4	4	4	4	4	4	4	4
HS57	8	8	15	22	22	49	20	25
HS59	47	93	>250	>350	26	31	>250	>350
HS60	7	7	6	7	7	7	7	7
HS63	7	8	6	9	6	9	6	7
HS64	30	34	29	30	25	29	>250	>350
HS65	14	25	16	25	15	26	15	26
HS66	7	7	9	11	8	8	8	8
HS67	11	19	11	19	11	19	11	19
HS68	24	38	24	37	24	35	24	35
HS69	10	10	12	12	9	9	11	11
HS70	21	22	20	26	24	27	25	29
HS71	7	7	8	8	8	8	6	6
HS72	29	29	22	23	23	23	>250	>350
HS73	14	14	14	14	16	16	15	15
HS74	7	7	9	9	8	8	7	7
HS75	7	7	10	11	9	9	8	8
HS76	7	7	7	7	7	7	7	7
HS80	10	10	9	9	9	9	10	10
HS81	9	9	9	9	9	9	9	9
HS83	20	20	16	16	18	18	24	24
HS84	37	44	25	43	36	43	37	44
HS86	11	11	14	14	14	14	11	11
HS88	18	75	19	38	25	33	15	16
HS91	21	25	17	20	14	18	21	36
HS92	25	40	33	67	21	27	36	45
HS93	6	6	10	11	9	9	6	6
HS95	8	8	18	24	11	11	7	7
HS96	7	7	19	24	11	11	7	7
HS97	27	35	20	26	13	13	34	38
HS98	21	23	22	37	16	19	21	29
HS99	4	4	10	41	7	36	4	4
HS100	8	12	9	13	9	13	8	12
HS104	11	11	11	11	12	12	11	11
HS105	11	11	17	21	17	21	19	21
HS106	42	46	17	19	12	13	40	44
HS107	12	15	12	17	9	13	11	14
HS108	18	19	16	17	19	24	31	41
HS109	>250	>350	>250	>350	>250	>350	>250	>350
HS110	4	4	5	5	5	5	4	4
HS111	12	18	11	11	12	18	12	18
112	11	11	11	11	11	11	10	10
HS113	49	57	15	21	18	25	40	47
HS114	16	17	17	19	16	17	16	17
HS116	123	220	>250	>350	32	36	>250	>350
HS117	27	30	25	31	23	27	28	31
HS118	13	13	13	13	14	14	13	13
HS119	16	16	11	11	12	12	16	16
HS268	15	15	21	23	18	19	15	15
HUBFIT	7	7	7	7	7	7	7	7
KIWCRESC	8	10	15	28	10	15	9	11
LAUNCH	32	54	38	75	36	70	36	70
LIN	20	20	8	8	8	9	20	20

Table 1 (*continued*)

Prob.	Proposal 1		Proposal 2		Proposal 3		Proposal 4	
	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.	Iter.	Eval.
LOADBAL	13	13	13	13	13	13	13	13
MADSEN	11	14	10	12	11	12	14	24
MAKELA1	15	16	13	13	12	12	12	13
MAKELA2	6	6	8	8	7	7	6	6
MAKELA3	25	30	12	15	16	21	15	23
MATRIX2	33	33	33	33	33	33	33	33
MIFFLIN1	6	6	6	6	6	6	6	6
MIFFLIN2	>250	>350	26	34	23	27	115	174
MINMAXBD	26	49	26	49	27	50	27	50
MINMAXRB	11	15	7	9	8	10	8	11
MISTAKE	9	9	11	11	11	11	11	11
ODFITS	7	7	11	21	7	7	7	7
POLAK1	9	10	8	8	7	7	9	10
POLAK2	>250	>350	>250	>350	14	30	>250	>350
POLAK3	25	28	>250	>350	22	29	43	47
POLAK4	6	6	18	25	66	89	19	31
POLAK6	22	41	22	29	20	34	21	40
PRODPL0	14	14	13	13	13	13	13	13
PRODPL1	15	15	15	16	16	16	16	16
RK23	7	7	7	7	7	7	7	7
ROSENMMX	228	321	97	143	33	53	33	53
S268	15	15	21	23	18	19	15	15
TAME	3	3	3	3	3	3	3	3
TENBARS4	>250	>350	28	39	34	43	>250	>350
TRUSPYR1	8	8	16	19	9	9	10	10
TRUSPYR2	9	9	17	20	10	11	10	10
TRY B	13	13	9	9	10	10	13	15
TWOBARS	6	6	6	6	6	6	6	6
WOMFLET	15	187	17	138	15	159	15	188
ZECEVIC2	8	9	8	8	8	8	8	8
ZECEVIC3	15	16	16	25	9	12	20	24
ZECEVIC4	13	14	9	9	10	11	13	14
ZY2	6	6	6	6	6	6	6	6

Table 2
Comparison of iteration counts on the whole set

Prob.	Number of iterations			
	Proposal 1	Proposal 2	Proposal 3	Proposal 4
Total	3648	3829	2702	4443
Average	25.16	26.41	18.63	30.64

Table 3
Comparison of function evaluations on the whole set

Prob.	Number of function evaluations			
	Proposal 1	Proposal 2	Proposal 3	Proposal 4
Total	5022	5295	3613	6070
Average	34.63	36.52	24.92	41.86

Table 5 compares the results from the four proposed updating strategies to those of other interior-point codes reported in the literature. In particular, those from Vanderbei and Shanno [15], Gay et al. [5] and

Table 4
Summary statistics

	Percentage of successes and best performances			
	Proposal 1	Proposal 2	Proposal 3	Proposal 4
% Successes	97.24	95.17	98.62	93.79
% Worst Iter.	13.79	29.66	9.66	13.79
% Best Eval.	17.24	15.86	13.79	5.52
% Worst Eval.	13.79	29.66	8.97	14.48

Table 5
Iterations counts for different nonlinear interior point codes

Prob.	Number of iterations						
	P1	P2	P3	P4	VS	Y	GOW
HS64	30	29	25		28	29	
HS65	14	16	15	15	14	15	10
HS71	7	8	8	6	12	8	15
HS72	29	22	23		21	43	
HS73	14	14	16	15	20	12	11
HS83	20	16	18	24	15	16	16
HS84	37	25	36	37	18	21	25
HS93	6	10	9	6	10	29	17
HS95	8	18	11	7	18	13	26
HS96	7	19	11	7	22	12	27
HS97	27	20	13	34	18	22	31
HS98	21	22	16	21	19	20	27
HS100	8	9	9	8	11	16	10
HS104	11	11	12	11	14	19	12
HS106	42	17	12	40	33	39	45
HS108	18	16	19	31	23	62	13
HS109					49	21	32
HS113	49	15	18	40	16	25	13
HS114	16	17	16	16	31	47	15
HS116	123		32		33	82	
HS117	27	25	23	28	22	36	33
HS118	13	13	14	13	17	34	17
Average	25.10	17.10	16.95	19.94	21.09	28.23	20.79

Yamashita [17], on a set of 22 HS problems (all the problems that were reported in all of the references). The columns in the table correspond to the number of iterations (matrix factorizations) required by:

- **Px**: the results for proposal x, for x 1,2,3,4.
- **VS**: iteration counts for LOQO, as reported in Vanderbei and Shanno [15].
- **Y**: iteration counts reported in Yamashita [17].
- **GOW**: iteration counts reported in Gay et al. [5].

For these problems Proposals 2 and 3 work better on the average than any of the other codes. All initial points for the algorithms are those indicated in Gould et al. [7]. For the GOW algorithm, only those results corresponding to these starting points are shown. The termination conditions for our codes are similar to those used in Gay et al. [5], and comparable or slightly more restrictive than those implemented in the other two codes.

6. Conclusions

In this work we have compared four updating rules for the barrier parameter within an interior-point method. The first three proposals use a vector of barrier parameters, while the last one is based on a widely used scalar updating strategy. We have shown that the new vector strategies are well defined, and all the components of the vector converge to zero under reasonable assumptions.

The computational experiments show that, for small problems, the vectorial strategies improve in average the iteration count and the number of function evaluations, and they are also successful on a larger number of problems than the scalar approach.

Acknowledgements

We wish to thank an anonymous referee for his/her careful reading and useful suggestions. This work has been partially supported by Spanish grants MCYT TIC2003-05892-C05-05, MEC MTM2006-14961-C05-05 and MEC MTM2004-02334.

References

[1] D.P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Academic Press, New York, 1982.

[2] A.S. El Bakry, R.A. Tapia, T. Tsuchiya, Y. Zhang, On the formulation and theory of the Newton interior point method for nonlinear programming, Journal of Optimization Theory and Applications 89 (1996) 507 541.

[3] A.V. Fiacco, G.P. McCormick, Nonlinear programming: Sequential unconstrained minimization techniques, Society for Industrial And Applied Mathematics, Philadelphia, 1990 (Originally published by Research Analysis Corporation, McLean, Virginia).

[4] R.S. Gajulapalli, An interior point algorithm for large scale nonlinear optimization, Ph.D. Thesis. The University of Texas at Austin, 1995.

[5] D.M. Gay, M.L. Overton, M.H. Wright, A primal dual interior method for nonconvex nonlinear programming, in: Y. Yuan (Ed.), Advances in Nonlinear Programming, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 31 56.

[6] N.I.M. Gould, D. Orban, A. Sartenaer, Ph.L. Toint, Superlinear convergence of primal dual interior point algorithms for nonlinear programming, SIAM Journal on Optimization 11 (4) (2001) 974 1002.

[7] N.I.M. Gould, D. Orban, Ph.L. Toint, CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited, ACM Transactions on Mathematical Software 29 (4) (2003) 373 394.

[8] B. Jansen, C. Roos, T. Terlaky, J. Ph. Vial, Primal dual target following algorithms for linear programming, Annals of Operations Research 62 (1996) 197 231.

[9] S. Mehrotra, On the implementation of a primal dual interior point method, SIAM Journal on Optimization 2 (1992) 575 601.

[10] J.M. Moguerza, F.J. Prieto, An augmented Lagrangian interior point method using directions of negative curvature, Mathematical Programming A 95 (2003) 573 616.

[11] J.M. Moguerza, F.J. Prieto, Combining search directions using gradient flows, Mathematical Programming A 96 (2003) 529 559.

[12] J.J. Moré, D.C. Sorensen, On the use of directions of negative curvature in a modified Newton method, Mathematical Programming 16 (1979) 1 20.

[13] Y.E. Nesterov, A.S. Nemirovski, Multi parameter surfaces of analytic centers and long step surface following interior point methods, Mathematics of Operations Research 23 (1) (1998) 1 38.

[14] A.L. Tits, A. Wachter, S. Bakhtiari, T.J. Urban, C.T. Lawrence, A primal dual interior point method for nonlinear programming with strong global and local convergence properties, SIAM Journal on Optimization 14 (1) (2003) 173 199.

[15] R.J. Vanderbei, D.F. Shanno, An interior point algorithm for nonconvex nonlinear programming, Computational Optimization and Applications 13 (1999) 231 252.

[16] S.J. Wright, Primal Dual Interior Point Methods, SIAM, Philadelphia, 1997.

[17] H. Yamashita, A globally convergent primal dual interior point method for constrained optimization, Optimization Methods and Software 10 (1998) 443 469.

[18] H. Yamashita, H. Yabe, Superlinear and quadratic convergence of some primal dual interior point methods for constrained optimization, Mathematical Programming 75 (1996) 377 397.