# Comparison of Interior Point Filter Line Search Strategies for Constrained Optimization by Performance Profiles

M. Fernanda P. Costa and Edite M. G. P. Fernandes

***Abstract*—This paper presents a performance evaluation of three sets of modifications that can be incorporated into the primal-dual interior point filter line search method for nonlinear programming herein illustrated. In this framework, each entry in the filter relies on three components, the feasibility, the centrality and the optimality, that are present in the first-order optimality conditions. The modifications are concerned with an acceptance condition, a barrier parameter update formula and a set of initial approximations to the dual variables. Performance profiles are plotted to compare the obtained numerical results using the number of iterations and the number of the optimality measure evaluations.**

***Index Terms*—Nonlinear optimization, Interior point method, Filter line search method, Performance profiles.**

## I. Introduction

The filter technique of Fletcher and Leyffer [5] has been used to globalize primal-dual interior point methods for solving a nonlinear constrained optimization problem. This technique incorporates the concept of nondominance to build a filter that is able to reject poor trial iterates and enforce global convergence from arbitrary starting points [6]. The filter replaces the use of merit functions, avoiding therefore the update of penalty parameters that are associated with the penalization of the constraints in merit functions.

Ulbrich, Ulbrich and Vicente in [11] define two components for each entry in the filter and use a trust-region strategy. The two components combine the three criteria of the first-order optimality conditions: the first component is a measure of quasi-centrality and the second is an optimality measure combining complementarity and criticality. Global convergence to first-order critical points is also proved. The filter methods in [1], [13]-[15] rely on a line search strategy and define two components for each entry in the filter: the barrier objective function and the constraints violation. The global convergence is analyzed in [13].

The algorithm herein illustrated is a primal-dual interior point method with a line search approach but considers three components for each entry in the filter. Primal-dual interior point methods seem adequate to the filter implementation as the feasibility, centrality and optimality measures in the first-order optimality conditions are natural candidates to the components of the filter. The algorithm also incorporates a restoration phase that aims to improve either feasibility or centrality.

In this paper, a performance evaluation is carried out using a benchmarking tool, known as Dolan and Moré performance profiles [4], to assess the performance of three sets of modifications that we propose to incorporate into the original algorithm. The modifications rely:

i) on a condition that is used to decide if a trial point is acceptable;
ii) on the strategy to update the barrier parameter $\mu$, at each iteration;
iii) on the initial approximations for the dual variables.

The paper is organized as follows. Section II briefly describes the interior point method and Section III is devoted to explain the filter line search method. Section IV includes three sets of modifications that are proposed to accelerate convergence and improve robustness. Section V describes the numerical experiments that were carried out in order to compare the original algorithm with the proposed modifications using performance profiles, and the conclusions make Section VI.

## II. The interior point paradigm

For easy of presentation, we consider the formulation of a constrained nonlinear optimization problem as follows:

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & F(x) \\ \text{s.t.} & h(x) \geq 0 \end{array} \tag{1}$$

where $h_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \ldots, m$ and $F : \mathbb{R}^n \to \mathbb{R}$ are nonlinear and twice continuously differentiable functions.

The primal-dual interior point method for solving (1) uses nonnegative slack variables $w$, to transform (1) into

$$\begin{array}{c} \min_{x\in\mathbb{R}^n, w\in\mathbb{R}^m} \ \varphi_\mu(x,w) \equiv F(x) - \mu \sum_{i=1}^{m} \log(w_i) \\ \text{s.t. } h(x) - w = 0, \end{array} \tag{2}$$

where $\varphi_\mu(x,w)$ is the barrier function and $\mu$ is a positive barrier parameter. The first-order KKT conditions for a minimum of (2) define a nonlinear system of $n+2m$ equations in $n+2m$ unknowns

$$\left\{ \begin{array}{l} \nabla F(x) - A^T y = 0 \\ -\mu W^{-1} e + y = 0 \\ h(x) - w = 0 \end{array} \right. \tag{3}$$

M. Fernanda P. Costa is with the Department of Mathematics for Science and Technology, University of Minho, Campus de Azurém, 4800-058 Guimarães, Portugal (E-mail: mfc@mct.uminho.pt).

Edite M. G. P. Fernandes is with the Department of Production and Systems, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (corresponding author E-mail: emgpf@dps.uminho.pt).

where $\nabla F$ is the gradient vector of $F$, $A$ is the Jacobian matrix of the constraints $h$, $y$ is the vector of dual variables, $W = diag(w_i)$ is a diagonal matrix, and $e$ is a $m$ vector of all ones. Applying the Newton's method to solve (3), the following reduced KKT system

$$\begin{bmatrix} -H(x,y) & A^T \\ A & \mu^{-1}W^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma \\ \pi \end{bmatrix} \quad (4)$$

and

$$\Delta w = \mu^{-1}W^2 \left(\gamma_w - \Delta y\right), \quad (5)$$

are obtained to compute the search directions $\Delta x$, $\Delta w$, $\Delta y$, where

$$H(x,y) = \nabla^2 F(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x)$$

is the Hessian matrix of the Lagrangian function and

$$\begin{aligned} &\sigma = \nabla F(x) - A^T y, \\ &\pi = \rho + \mu^{-1}W^2\gamma_w, \\ &\gamma_w = \mu W^{-1}e - y, \\ &\rho = w - h(x). \end{aligned}$$

Given initial approximations to the variables $x_0$, $w_0 > 0$, $y_0 > 0$, this iterative process chooses, at each iteration $k$, a step length $\alpha_k$, and defines a new estimate to the optimal solution using

$$\begin{aligned} &x_{k+1} = x_k + \alpha_k \Delta x_k \\ &w_{k+1} = w_k + \alpha_k \Delta w_k \\ &y_{k+1} = y_k + \alpha_k \Delta y_k. \end{aligned}$$

The step length $\alpha_k$ is chosen to ensure the nonnegativity of the slack and dual variables. The procedure that decides which step length is accepted is a filter line search method.

After a new point has been computed, the barrier parameter $\mu$ is updated as a fraction of the average complementarity, i.e.,

$$\mu_{k+1} = \delta_\mu \frac{w_{k+1}^T y_{k+1}}{m} \quad (6)$$

where $\delta_\mu \in [0, 1)$.

Our algorithm is a quasi-Newton based method in the sense that a symmetric positive definite quasi-Newton BFGS approximation, $B_k$, is used to approximate the Hessian of the Lagrangian $H$, at each iteration $k$ [9].

## III. Filter line search method

In this section, we present the line search filter framework. To simplify the notation, we introduce the vectors:

$$\begin{aligned} &u = (x, w, y), \quad && \Delta = (\Delta x, \Delta w, \Delta y), \\ &u^1 = (x, w), && \Delta^1 = (\Delta x, \Delta w), \\ &u^2 = (w, y), && \Delta^2 = (\Delta w, \Delta y), \\ &u^3 = (x, y), && \Delta^3 = (\Delta x, \Delta y). \end{aligned}$$

The methodology of a filter as outline in [5] and [6] is adapted to this interior point method. In our case, three components for each entry in the filter are defined. The first component measures feasibility, the second measures centrality and the third optimality. Based on the optimality conditions (3) the following measures are used:

$$\theta_f(u^1) = \|\rho\|_2, \; \theta_c(u^2) = \|\gamma_w\|_2, \; \theta_{op}(u^3) = \frac{1}{2}\|\sigma\|_2^2.$$

After a search direction $\Delta_k$ has been computed, a backtracking line search procedure is implemented, where a decreasing sequence of step sizes

$$\alpha_{k,l} \in (0, \alpha_k^{\max}], \; l = 0, 1, ...,$$

with $lim_l\ \alpha_{k,l} = 0$, is tried until a set of acceptance conditions are satisfied. Here, we use $l$ to denote the iteration counter for the inner loop. The parameter $\alpha_k^{\max}$ represents the longest step size that can be taken along the direction before violating the nonnegativity conditions $u_k^2 \geq 0$. If we assume that the starting point $u_0$ satisfies $u_0^2 > 0$, the maximal step size $\alpha_k^{\max} \in (0, 1]$ is defined by

$$\alpha_k^{\max} = \max\{\alpha \in (0, 1] : u_k^2 + \alpha\Delta_k^2 \geq (1 - \varepsilon)u_k^2\} \quad (7)$$

for a fixed parameter $\varepsilon \in (0, 1)$.

### A. Acceptance conditions

In this algorithm, the trial point $u_k(\alpha_{k,l}) = u_k + \alpha_{k,l}\Delta_k$ is acceptable by the filter, if it leads to sufficient progress in one of the three measures compared to the current iterate,

$$\begin{aligned} &\theta_f(u_k^1(\alpha_{k,l})) \leq \left(1 - \gamma_{\theta_f}\right)\theta_f(u_k^1) \\ &\text{or} \;\; \theta_c(u_k^2(\alpha_{k,l})) \leq \left(1 - \gamma_{\theta_c}\right)\theta_c(u_k^2) \\ &\text{or} \;\; \theta_{op}(u_k^3(\alpha_{k,l})) \leq \theta_{op}(u_k^3) - \gamma_{\theta_o}\theta_f(u_k^1) \end{aligned} \quad (8)$$

where $\gamma_{\theta_f}, \gamma_{\theta_c}, \gamma_{\theta_o} \in (0, 1)$ are fixed constants.

However, to prevent convergence to a feasible but nonoptimal point, and whenever for the trial step size $\alpha_{k,l}$, the following switching conditions

$$\begin{aligned} &m_k(\alpha_{k,l}) < 0 \;\; \text{and} \\ &\left[-m_k(\alpha_{k,l})\right]^{s_o}\left[\alpha_{k,l}\right]^{1-s_o} > \delta\left[\theta_f(u_k^1)\right]^{s_f} \;\; \text{and} \\ &\left[-m_k(\alpha_{k,l})\right]^{s_o}\left[\alpha_{k,l}\right]^{1-s_o} > \delta\left[\theta_c(u_k^2)\right]^{s_c} \end{aligned} \quad (9)$$

hold, with fixed constants $\delta > 0$, $s_f > 1, s_c > 1$, $s_o \geq 1$, where

$$m_k(\alpha) = \alpha\nabla\theta_{op}(u_k^3)^T\Delta_k^3,$$

then the trial point must satisfy the Armijo condition with respect to the optimality measure

$$\theta_{op}(u_k^3(\alpha_{k,l})) \leq \theta_{op}(u_k^3) + \eta_o m_k(\alpha_{k,l}), \quad (10)$$

instead of (8) to be acceptable. Here, $\eta_o \in (0, 0.5)$ is a constant.

According to previous publications on filter methods (for example [13]), a trial step size $\alpha_{k,l}$ is called a $\theta_{op}$-step if (10) holds. Similarly, if a $\theta_{op}$-step is accepted as the final step size $\alpha_k$ in iteration $k$, then $k$ is referred to as a $\theta_{op}$-type iteration.

### B. The filter

To prevent cycling between iterates that improve either the feasibility, or the centrality, or the optimality, at each iteration $k$, the algorithm maintains a filter that is a set $\overline{F}_k$ that contains values of $\theta_f$, $\theta_c$ and $\theta_{op}$, that are prohibited for a successful trial point in iteration $k$ [11], [13]-[15]. Thus, a trial point $u_k(\alpha_{k,l})$ is acceptable, if

$$\left(\theta_f(u_k^1(\alpha_{k,l})), \theta_c(u_k^2(\alpha_{k,l})), \theta_{op}(u_k^3(\alpha_{k,l}))\right) \notin \overline{F}_k.$$

At the beginning of the iterative process the filter is initialized to

$$\overline{F}_0 \subseteq \left\{(\theta_f, \theta_c, \theta_{op}) \in \mathbb{R}^3 : \theta_f \geq \theta_f^{\max}, \theta_c \geq \theta_c^{\max}, \theta_{op} \geq \theta_{op}^{\max}\right\}, \tag{11}$$

for some positive constants $\theta_f^{\max}, \theta_c^{\max}$ and $\theta_{op}^{\max}$, and is updated after every iteration in which the accepted trial step size satisfies (8), using the formula

$$\overline{F}_{k+1} = \overline{F}_k \cup \left\{(\theta_f, \theta_c, \theta_{op}) \in \mathbb{R}^3 : \theta_f > \left(1 - \gamma_{\theta_f}\right) \theta_f(u_k^1) \text{ and } \theta_c > (1 - \gamma_{\theta_c})\, \theta_c(u_k^2) \text{ and } \theta_{op} > \theta_{op}(u_k^3) - \gamma_{\theta_o} \theta_{feas}(u_k^1)\right\}. \tag{12}$$

However, when (9) and (10) hold for the accepted step size the filter remains unchanged.

Whenever the backtracking line search finds a trial step size $\alpha_{k,l}$ that is smaller than a minimum desired step size $\alpha_k^{\min}$ (see [2] for details), the algorithm enters into a restoration phase that aims to find a new iterate $u_{k+1}$ that is acceptable to the current filter by decreasing either the feasibility or the centrality.

### C. The algorithm

Our interior point filter line search algorithm for solving constrained optimization problems is as follows:

*Algorithm 1:* (interior point filter line search algorithm)

1) *Given*: Starting point $x_0$, $u_0^2 > 0$, constants $\theta_f^{\max} \in (\theta_f(u_0^1), \infty]$; $\theta_c^{\max} \in (\theta_c(u_0^2), \infty]$; $\theta_{op}^{\max} \in (\theta_{op}(u_0^3), \infty]$; $\gamma_{\theta_f}, \gamma_{\theta_c}, \gamma_{\theta_o} \in (0, 1)$; $\delta > 0$; $s_f > 1$; $s_c > 1$; $s_o \geq 1$; $\eta_o \in (0, 0.5]$; $\varepsilon_{tol} \ll 1$; $\varepsilon \in (0, 1)$; $\delta_\mu \in [0, 1)$.
2) *Initialize*. Initialize the filter (using (11)) and the iteration counter $k \leftarrow 0$.
3) *Check convergence*. Stop if the relative measures of primal and dual infeasibility are less or equal to $\varepsilon_{tol}$.
4) *Compute search direction*. Compute the search direction $\Delta_k$ from the linear system (4), and (5).
5) *Backtracking line search.*
   - 5.1) *Initialize line search.* Compute the longest step length $\alpha_k^{\max}$ using (7) to ensure positivity of slack and dual variables. Set $\alpha_{k,l} = \alpha_k^{\max}$, $l \leftarrow 0$.
   - 5.2) *Compute new trial point.* If the trial step size becomes too small, i.e., $\alpha_{k,l} < \alpha_k^{\min}$, go to restoration phase in step 9. Otherwise, compute the trial point $u_k(\alpha_{k,l})$ and $\mu_{k+1}$.
   - 5.3) *Check acceptability to the filter*. If $\left(\theta_f(u_k^1(\alpha_{k,l})), \theta_c(u_k^2(\alpha_{k,l})), \theta_{op}(u_k^3(\alpha_{k,l}))\right) \in \overline{F}_k$, reject the trial step size and go to step 5.6.
   - 5.4) *Check sufficient decrease with respect to current iterate.* If $\alpha_{k,l}$ is an $\theta_{op}$-step size ((9) holds) and the Armijo condition (10) for the $\theta_{op}$ function holds, accept the trial step and go to step 6.
   - 5.5) *Check sufficient decrease with respect to current iterate.* If (8) holds, accept the trial step and go to step 6. Otherwise go to step 5.6.
   - 5.6) *Choose new trial step size.* Set $\alpha_{k,l+1} = \alpha_{k,l}/2$, $l \leftarrow l + 1$, and go back to step 5.2.
6) *Accept trial point.* Set $\alpha_k \leftarrow \alpha_{k,l}$ and $u_{k+1} \leftarrow u_k(\alpha_k)$.
7) *Augment the filter if necessary.* If $k$ is not an $\theta_{op}$-type iteration, augment the filter using (12). Otherwise, leave the filter unchanged.
8) *Continue with next iteration*. Increase the iteration counter $k \leftarrow k + 1$ and go back to step 3.
9) *Restoration phase.* Use a restoration algorithm to produce a point $u_{k+1}$ that is acceptable to the filter, i.e., $\left(\theta_f(u_{k+1}^1), \theta_c(u_{k+1}^2), \theta_{op}(u_{k+1}^3)\right) \notin \overline{F}_k$. Augment the filter using (12) and continue with the regular iteration in step 8.

### D. Restoration phase

The task of the restoration phase is to compute a new iterate acceptable to the filter by decreasing either the feasibility or the centrality, whenever the regular backtracking line search procedure cannot make sufficient progress and the step size becomes too small. The new functions for measuring feasibility and centrality are

$$\theta_{2,f}(u^1) = \frac{1}{2}\left\|\rho\right\|_2^2 \text{ and } \theta_{2,c}(u^2) = \frac{1}{2}\left\|\gamma_w\right\|_2^2$$

respectively. The restoration algorithm works with the steps $\Delta^1$ and $\Delta^2$, computed from (4) and (5), that are descent directions for $\theta_{2,f}(u^1)$ and $\theta_{2,c}(u^2)$, respectively.

A sufficient reduction in one of the measures $\theta_{2,f}$ and $\theta_{2,c}$ is required for a trial step size to be acceptable. Additionally, we also ensure that the value of the optimality measure at the new trial point, $\theta_{op}(u_k^3(\alpha_{k,l}))$, does not deviate too much from the current value, $\theta_{op}(u_k^3)$. The reader is referred to [2] for details.

## IV. Algorithm modifications

In the sequence of our previous work [3], which carried out a comparison of three types of acceptance conditions, we propose now other modifications to the original algorithm, in an attempt to accelerate convergence and improve robustness. They are focused:

i) on the acceptance conditions that are used to decide if a trial point is acceptable;
ii) on the dynamic update of the barrier parameter $\mu$, at each iteration;
iii) on the initial settings to the dual variables.

To assess the performance of the proposed modifications, numerical experiments with a set of well-known problems are carried out and a benchmarking tool, known as performance profiles [4] is used. Section V summarizes the numerical results.

### A. Condition for a trial iterate to be acceptable

The acceptance condition (8) is a natural extension of the condition in [5], in the sense that a sufficient reduction in just one component of the filter is imposed for a trial iterate to be acceptable.

Here, we try another acceptance condition. It is more restrictive than the original (8) since sufficient progress may be required in two components of the filter - the feasibility and the centrality components. Thus, the condition considers the trial point $u_k(\alpha_{k,l})$ to be acceptable if it leads to sufficient progress either in both the feasibility and centrality measures or in the optimality measure, i.e., if

$$\begin{array}{l} \left(\theta_f(u_k^1(\alpha_{k,l})) \leq \left(1-\gamma_{\theta_f}\right)\theta_f(u_k^1) \right. \\ \text{and } \; \theta_c(u_k^2(\alpha_{k,l})) \leq \left(1-\gamma_{\theta_c}\right)\theta_c(u_k^2)\big) \\ \text{or } \; \theta_{op}(u_k^3(\alpha_{k,l})) \leq \theta_{op}(u_k^3) - \gamma_{\theta_o}\theta_f(u_k^1) \end{array} \tag{13}$$

holds.

### B. Barrier parameter update

In order to guarantee a positive decreasing sequence of $\mu$ values, an alternative formula to (6) may be used to update the barrier parameter. The proposed formula couples the theoretical requirement defined on the first-order KKT conditions (3) with a simple heuristic. Thus, we update $\mu$ using

$$\mu_{k+1} = \max\{\epsilon, \min\{\kappa_\mu \mu_k, 10^{-j}\delta_\mu \frac{w_{k+1}^T y_{k+1}}{m}\}\} \tag{14}$$

where $j$ is the first element of the set $\{0,1,2,3\}$ for which

$$10^{-j}\delta_\mu \frac{w_{k+1}^T y_{k+1}}{m} < \mu_k$$

holds. If this condition does not hold, then $j$ is set to 3 in (14). The constant $\kappa_\mu \in (0,1)$. The tolerance $\epsilon$ is used to prevent $\mu$ from becoming too small so avoiding numerical difficulties at the end of the iterative process.

We further remark that each time the barrier parameter is updated, the $\theta_c$ component of the filter should be recalculated. In practice, only $\theta_c^{\max}$ is reevaluated.

### C. Initial values for the slack and dual variables

This interior point method requires that the slack and the dual variables are nonnegative at the beginning of the process and are maintained as so throughout the entire iterative process. Further, for a given initial point $x_0$, we should guarantee that the initial slack variables are sufficiently away from the boundary, using

$$w_0 = \max\{h(x_0), \epsilon_w\}$$

for a fixed positive constant $\epsilon_w$.

Our formulation of the constraints in (1), $h(x) \geq 0$, includes the pure inequality constraints, as well as the simple bound constraints on the variables.

For the initial dual variables, the new proposed strategy sets the dual variables, associated with the pure inequality constraints, to one, and the dual variables associated with the bound constraints to the absolute value of $x_0$ (componentwise), as long as this is far away from zero, i.e.,

$$y_0 = \begin{cases} 1, & \text{for pure inequality constraints} \\ \max\{|x_0|, \epsilon_w\}, & \text{for bound constraints.} \end{cases} \tag{15}$$

## V. Numerical results

To analyze the performance of the proposed modifications to the interior point filter line search method we use 111 constrained problems from the Hock and Schittkowski test set [8]. The tests were done in double precision arithmetic with a Pentium 4. The algorithm is coded in the C programming language and includes an interface to AMPL to read the problems that are coded in the AMPL modeling language [7].

The chosen values for the constants are: $\theta_f^{\max} = 10^4 \max\left\{1, \theta_f(u_0^1)\right\}$, $\theta_c^{\max} = 10^4 \max\left\{1, \theta_c(u_0^2)\right\}$, $\theta_{op}^{\max} = 10^4 \max\left\{1, \theta_{op}(u_0^3)\right\}$, $\gamma_{\theta_f} = \gamma_{\theta_c} = \gamma_{\theta_o} = 10^{-5}$, $\delta = 1$, $s_f = 1.1$, $s_c = 1.1$, $s_o = 2.3$, $\eta_o = 10^{-4}$, $\varepsilon_{tol} = 10^{-4}$, $\delta_\mu = 0.1$, $\varepsilon = 0.95$, $\epsilon_w = 0.1$, $\kappa_\mu = 0.1$ and $\epsilon = 10^{-9}$.

In our comparative studies we plot the performance profiles as outline in [4]. A brief explanation of this performance assessment follows.

### A. Performance profiles

To evaluate and compare the performance of the herein proposed modifications to the interior point filter line search method, presented in Sections II and III, we use the performance profiles as outline in [4]. These profiles represent the cumulative distribution functions for the performance ratios based on a chosen metric.

Let $\mathcal{P}$ be the set of problems and $\mathcal{C}$ the set of codes (implementation of the algorithms) used in the comparative study. Let $t_{p,c}$ be the performance metric (for example, the number of iterations) required to solve problem $p$ by code $c$. Then, the comparison is based on the performance ratios

$$r_{p,c} = \frac{t_{p,c}}{\min\{t_{p,c}, c \in \mathcal{C}\}}, p \in \mathcal{P}, c \in \mathcal{C}$$

and the overall assessment of the performance of a particular code $c$ is given by

$$\rho_c(\tau) = \frac{1}{n_P}\text{size}\{p \in \mathcal{P} : \log_2(r_{p,c}) \leq \tau\}$$

where $n_P$ is the number of problems in the set $\mathcal{P}$. Here, we use a $\log_2$ scaled of the performance profiles. "size" is the number of problems in the set such that the $\log_2$ of the performance ratio $r_{p,c}$ is less than or equal to $\tau$ for code $c$. Thus, $\rho_c(\tau)$ gives the probability (for code $c \in \mathcal{C}$) that the $\log_2$ of the performance ratio $r_{p,c}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. The function $\rho_c$ is the cumulative distribution function for the performance ratio.

The value of $\rho_c$ for $\tau = 0$ gives the probability that the code $c$ will win over the others in the set. However, for large values of $\tau$, the value of $\rho_c$ measures the code robustness.

### B. Comparison of acceptance conditions

To assess the performance of the new acceptance condition (13), when compared with the original (8), within the illustrated interior point filter line search method, we plot the performance profiles for the number of iterations and for the number of $\theta_{op}$ evaluations required to solve a set of problems, according to the convergence criteria defined in the Algorithm 1.

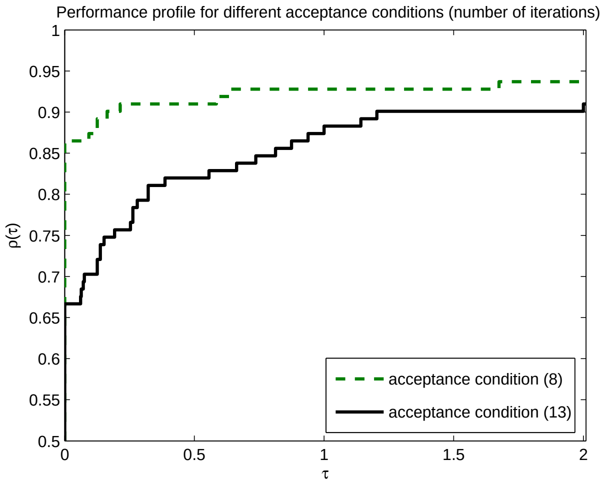


Fig. 1. Performance profiles in a $log_2$ scale: number of iterations

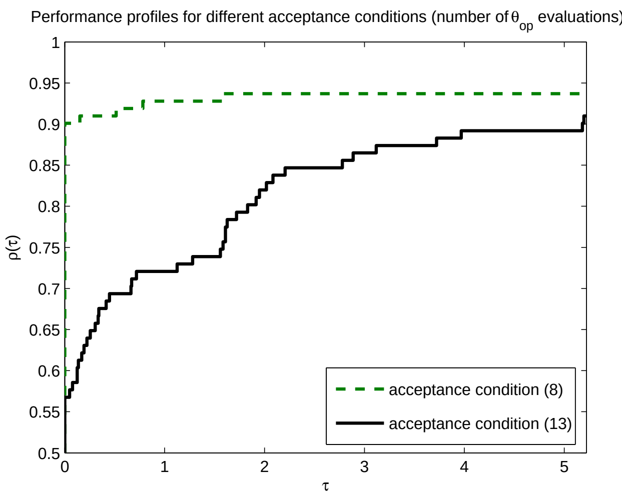


Fig. 2. Performance profiles in a $log_2$ scale: number of optimality measure evaluations

Four experiments were carried out with each proposed version. First, with the initial approximation $x_0$ given in [8], the algorithm recomputes a better approximation, say $\widetilde{x}_0$, as well as $y_0$, by solving the simplified reduced KKT:

$$\left[\begin{array}{cc} -(B_0+I) & A(x_0)^T \\ A(x_0) & I \end{array}\right]\left[\begin{array}{c} \widetilde{x}_0 \\ y_0 \end{array}\right]=\left[\begin{array}{c} \nabla F(0) \\ 0 \end{array}\right].$$

Then, in the first experience, the initial matrix $B_0$ is a positive definite modification of $\nabla^2 F(x_0)$ and, in the second experience, $B_0$ is set to the identity matrix.

The remaining two experiments consider different initial primal and dual variables. They use the given $x_0$, and $y_0$ is obtained by formula (15). The third experience uses $B_0 \approx \nabla^2 F(x_0)$, with guaranteed positive definiteness, and the fourth uses $B_0 = I$.

We then combine the results of the four experiments and select the best result for each problem.

We use two performance metrics: the number of iterations and the number of $\theta_{op}$ evaluations. The corresponding performance profile plots are illustrated in Fig. 1 and 2.

Fig. 1 shows the performance profiles for the number of iterations required to solve the problems, by the two different

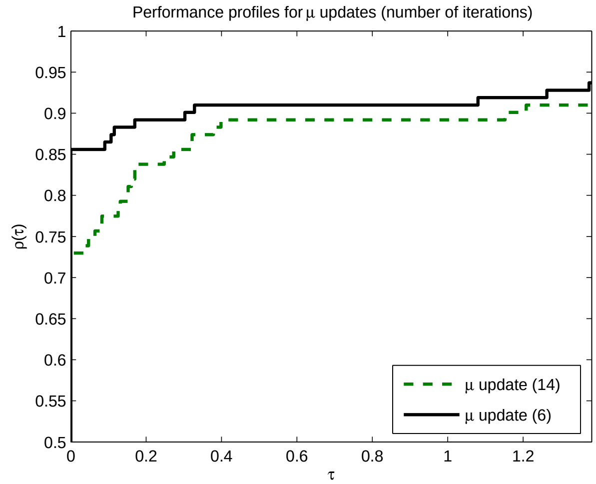


Fig. 3. Performance profiles for different $\mu$ updates: number of iterations

conditions (8) and (13). The figure gives a clear indication that the condition (8) is the most efficient, in terms of number of iterations, on almost 86% of the problems. From Fig. 2 we may conclude that (8) is the most efficient on almost 90% of the problems when the number of $\theta_{op}$ evaluations is under comparison. Observing the right side of the plots in both figures, we conclude that condition (8) solves most problems to optimality (approximately 93%).

### *C. Comparison of barrier parameter updates*

We aim to compare the two dynamic update strategies for the barrier parameter, illustrated in (6) and (14). We also combine the results of the four previously described experiments and select the best result for each problem. For the performance metric, we select the number of iterations required by each code to solve each problem. The performance profile plots are illustrated in Fig 3. Update (6) is the most efficient, on almost 86% of the problems, and it wins also on robustness.

One of the four experiments, that were carried out in order to obtain the best possible result for each problem, solves more problems in the set than the others. The configuration of this experiment is defined by the following initial approximations: $x_0$ given by [8], $y_0$ obtained by formula (15) and $B_0 = I$. Based only on the results of this experiment, we plot in Fig. 4 the performance profiles for the number of iterations taken by both formulae (6) and (14). Although update (6) is slightly better than update (14) for small values of $\tau$ ($\tau < 0.2$), at the end, the other formula turns out to be more robust.

### *D. Different initial values for the dual variables*

Finally, in the sequence of the last experience where, for a particular set of initial values for the primal, slack and dual variables, the update formula for the barrier parameter (14) wins over the other, we decided to further analyze the dependency of the new update formula on the initial approximations. The alternatives for the initial values for the dual variables $y$ are:

i) using (15);

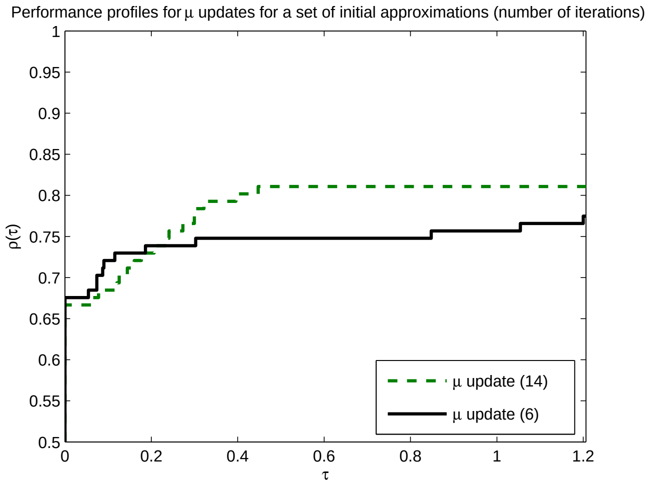


Fig. 4. Performance profiles for different $\mu$ updates with a particular set of initial approximations: number of iterations

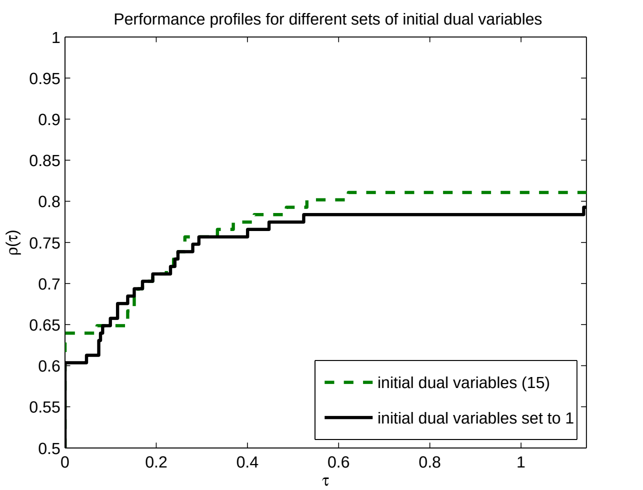


Fig. 5. Performance profiles in a $log_2$ scale: number of iterations

ii) setting $y_0 = 1$, for all the components.

Fig. 5 shows the performance profile plots of both cases. Initial dual variables defined by (15) give the most efficient run, on almost $64\%$ of the problems. They also win on robustness.

## VI. Conclusions

A primal-dual interior point method based on a filter line search approach is presented. This approach defines three components for each entry in the filter: the feasibility, the centrality and the optimality. We have presented a detailed comparative study of three types of modifications that can be introduced in the original algorithm.

The first modification is concerned with one of the conditions that are used to consider a point to be acceptable to the filter. The proposal is more restrictive than the original acceptance condition. The performance profiles show that neither efficiency nor robustness has improved with the modification.

The second modification focuses on the dynamic update of the barrier parameter. The new update formula, which guarantees a decreasing sequence of $\mu$ values over the iterative process, does not seem to improve the algorithm efficiency, except for a particular set of initial values for the primal, slack and dual variables.

The third set of comparative experiments aims to analyze the dependency of the $\mu$ update formula on the initial settings for the dual variables. The performance plots are not significantly different, although it is possible to point out the best set of initial values.

We remark that the performance profiles reflect only the performance of the tested codes on the data being used, so other test sets with larger and/or more difficult problems should be carried out.

## References

[1] H. Y. Benson, R. J. Vanderbei, and D.F. Shanno, "Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions," *Computational Optimization and Applications*, vol. 23, pp. 257–272, 2002.

[2] M. F. P. Costa and E. M. G. P. Fernandes, "A globally convergent interior point filter line search method for nonlinear programming," (submitted for publication to Journal of Numerical Analysis, Industrial and Applied Mathematics), 2007.

[3] M. F. P. Costa and E. M. G. P. Fernandes, "Performance evaluation of an interior point filter line search method for constrained optimization," in *Proc. 6th. WSEAS International Conference on System Science and Simulation in Engineering*, ISBN: 978-960-6766-14-4, 2007, pp. 18-23.

[4] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2002.

[5] R. Fletcher and S. Leyffer S. "Nonlinear programming without a penalty function," *Mathematical Programming*, vol. 91, pp. 239–269, 2002.

[6] R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wachter, "Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming," *SIAM Journal on Optimization*, vol. 13, pp. 635-659, 2002.

[7] R. Fourer, D. M. Gay, and B. Kernighan, "A modeling language for mathematical programming," *Management Science*, vol. 36, pp. 519–554, 1990.

[8] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming*. Springer-Verlag, 1981.

[9] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag, 1999.

[10] D. F. Shanno and R. J. Vanderbei, "Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods," *Mathematical Programming B*, vol. 87, pp. 303–316, 2000.

[11] M. Ulbrich, S. Ulbrich, and L. N. Vicente, "A globally convergent primal-dual interior-point filter method for nonlinear programming," *Mathematical Programming*, vol. 100, pp. 379–410, 2004.

[12] R. J. Vanderbei and D. F. Shanno, "An interior-point algorithm for nonconvex nonlinear programming," *Computational Optimization and Applications*, vol. 13, pp. 231–252, 1999.

[13] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: motivation and global convergence," *SIAM Journal on Optimization*, vol. 16, pp. 1–31, 2005.

[14] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: local convergence," *SIAM Journal on Optimization*, vol. 16, pp. 32–48, 2005.

[15] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2007.