Performance Evaluation of an Interior Point Filter Line Search Method for Constrained Optimization

M. FERNANDA P. COSTA
University of Minho
Dep. of Math. for Science and Technology
Campus de Azurem, 4800-058 Guimarães
PORTUGAL

EDITE M. G. P. FERNANDES
University of Minho
Dep. of Production and Systems
Campus de Gualtar, 4710-057 Braga
PORTUGAL

Abstract: Here we present a performance evaluation of three versions of a primal-dual interior point filter line search method for nonlinear programming. Each entry in the filter relies on three components, the feasibility, centrality and optimality, that are present in the first-order optimality conditions. The versions differ in a set of acceptance conditions that are used to consider a trial iterate to be acceptable to the filter. Performance profiles are used to compare the obtained numerical results in terms of the number of iterations and the number of the optimality measure evaluations.

Key–Words: Nonlinear optimization, Interior point method, Filter line search method, Performance profiles

1 Introduction

The filter technique of Fletcher and Leyffer [4] is used to globalize the primal-dual interior point method 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. 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.

The filter technique has already been adapted to interior point methods. For example, Ulbrich, Ulbrich and Vicente in [9] 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, 11, 12, 13] 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 [11].

The algorithm herein presented 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 also carried out using a benchmarking tool, known as performance profiles [3], to compare three sets of concurrent trial iterate acceptance conditions.

The paper is organized as follows. Section 2 briefly describes the interior point method and Section 3 is devoted to introduce the filter line search method, including the three sets of acceptance conditions under study. Section 4 describes the numerical experiments that were carried out in order to define the performance profiles, and the conclusions make Section 5.

2 The interior point method

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

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

where $\varphi_\mu(x,w)$ is the barrier function and μ 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

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

where ∇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

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

and

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

are obtained to compute the search directions Δx, Δw, Δy, 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

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

Once the search directions have been determined, the algorithm proceeds iteratively from an initial point x_0, $w_0 > 0$, $y_0 > 0$ choosing a step length α_k, at each iteration, and defining a new estimate to the optimal solution by

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

The step length α_k is chosen to ensure the nonnegativity of slack and dual variables. In the algorithm, the procedure that decides which trial step size is accepted is a filter line search method.

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 [7].

To compute μ at each iteration, a fraction of the average complementarity

$$\mu = \delta_\mu \frac{w^T y}{m} \tag{6}$$

is used, where $\delta_\mu \in [0,1)$. We refer to [8, 10] for details.

3 Filter line search method

To simplify the notation, we introduce the vectors:

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

The methodology of a filter as outline in [4] is adapted to this interior point method. 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 Δ_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, l denotes the iteration counter for the inner loop. $\alpha_k^{\max}$ is the longest step size that can be taken along the direction before violating the nonnegativity conditions $u_k^2 \geq 0$. Assuming that the starting point u_0 satisfies $u_0^2 > 0$, the maximal step size $\alpha_k^{\max} \in (0,1]$ is defined by

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

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

In the initial version of the 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,

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

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

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

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

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

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 [11]), a trial step size $\alpha_{k,l}$ is called a θ_{op}-step if (10) holds. Similarly, if a θ_{op}-step is accepted as the final step size α_k in iteration k, then k is referred to as a θ_{op}-type iteration.

In order 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 θ_f, θ_c and θ_{op}, that are prohibited for a successful trial point in iteration k [9, 11, 12, 13]. Thus, a trial point $u_k(\alpha_{k,l})$ is acceptable, if

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

The filter is initialized to

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

for some positive constants $\theta_f^{\max}, \theta_c^{\max}$ and $\theta_{op}^{\max}$, and is updated using the formula

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

after every iteration in which the accepted trial step size satisfies (8). On the other hand, if (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 reverts to a restoration phase. Here, the algorithm tries to find a new iterate u_{k+1} that is acceptable to the current filter, i.e., condition (8) holds, by decreasing either the feasibility or the centrality.

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* ε_{tol}.
4. Compute search direction. *Compute the search direction* Δ_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 recalculate* μ *using (6).*
   - 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* θ_{op}*-step size ((9) holds) and the Armijo condition (10) for the* θ_{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 θ_{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., $\left(\theta_f(u^1_{k+1}), \theta_c(u^2_{k+1}), \theta_{op}(u^3_{k+1})\right) \notin \overline{F}_k$. Augment the filter using (12) and continue with the regular iteration in step 8.*

3.1 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. Thus, new functions are introduced

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

The restoration algorithm works with the steps Δ^1 and Δ^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, $\theta_{op}(u^3_k(\alpha_{k,l}))$, does not deviate too much from the current value, $\theta_{op}(u^3_k)$. The reader is referred to [2] for details.

3.2 Acceptance conditions

The acceptance condition (8) is a natural extension of the condition in [4], 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 propose two other sets of acceptance conditions. They are overall more restrictive than the original (8) since sufficient progress is required in some cases in two components. Based on performance profiles, a comparative study is also carried out to evaluate the efficiency of these versions of the filter line search method. This is the main contribution of the paper.

The first set 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

$$\begin{array}{l}\left(\theta_f(u^1_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_f}\right)\theta_f(u^1_k) \text{ and}\right.\\ \left.\theta_c(u^2_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_c}\right)\theta_c(u^2_k)\right)\\ \text{ or } \theta_{op}(u^3_k(\alpha_{k,l})) \le \theta_{op}(u^3_k) - \gamma_{\theta_o}\theta_f(u^1_k)\end{array} \qquad (13)$$

holds. Thus, a new version of the Algorithm 1 is defined with condition (8) replaced by (13).

The other set of conditions is still more restrictive and accepts a trial iterate if sufficient progress is obtained in any two of the three proposed measures compared to the current iterate,

$$\begin{array}{l}\left(\theta_f(u^1_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_f}\right)\theta_f(u^1_k) \text{ and}\right.\\ \left.\theta_c(u^2_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_c}\right)\theta_c(u^2_k)\right)\\ \text{or}\\ \left(\theta_f(u^1_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_f}\right)\theta_f(u^1_k) \text{ and}\right.\\ \left.\theta_{op}(u^3_k(\alpha_{k,l})) \le \theta_{op}(u^3_k) - \gamma_{\theta_o}\theta_f(u^1_k)\right)\\ \text{or}\\ \left(\theta_c(u^2_k(\alpha_{k,l})) \le \left(1-\gamma_{\theta_c}\right)\theta_c(u^2_k) \text{ and}\right.\\ \left.\theta_{op}(u^3_k(\alpha_{k,l})) \le \theta_{op}(u^3_k) - \gamma_{\theta_o}\theta_f(u^1_k)\right).\end{array} \qquad (14)$$

In this version of the interior point filter line search method, condition (14) replaces (8). The original and these two new versions of the Algorithm 1 are tested with a well-known set of problems and compared using the performance profiles.

4 Numerical results

To analyze the performance of the three proposed versions of the interior point filter line search method we used 111 constrained problems from the Hock and Schittkowski test set [6]. 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 [5].

The chosen values for the constants are: $\theta_f^{\max} = 10^4 \max\left\{1, \theta_f(u^1_0)\right\}$, $\theta_c^{\max} = 10^4 \max\left\{1, \theta_c(u^2_0)\right\}$, $\theta_{op}^{\max} = 10^4 \max\left\{1, \theta_{op}(u^3_0)\right\}$, $\gamma_{\theta_f} = \gamma_{\theta_c} = \gamma_{\theta_o} = 10^{-5}$, $\delta = 1$, $s_f = 1.1$, $s_c = 1.1$, $s_o = 2.3$, $\eta_o = 10^{-4}$, $\varepsilon_{tol} = 10^{-4}$, $\delta_\mu = 0.1$ and $\varepsilon = 0.95$.

Four experiments were carried out with each proposed version. First, with the initial approximation x_0 given in [6], the algorithm recomputes a better approximation, say $\widetilde{x}_0$, as well as y_0, by solving a simplified reduced KKT (see (4)). Then, in the first experience, the initial matrix B_0 is a positive definite modification of $\nabla^2 F\left(\widetilde{x}_0\right)$ 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

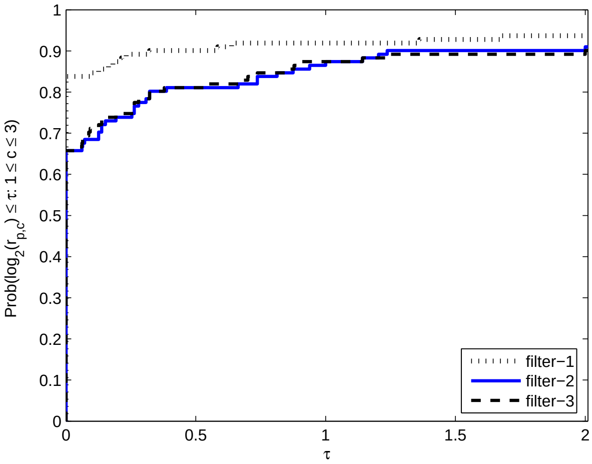

Figure 1: Performance profiles in a log_2 scale: number of iterations

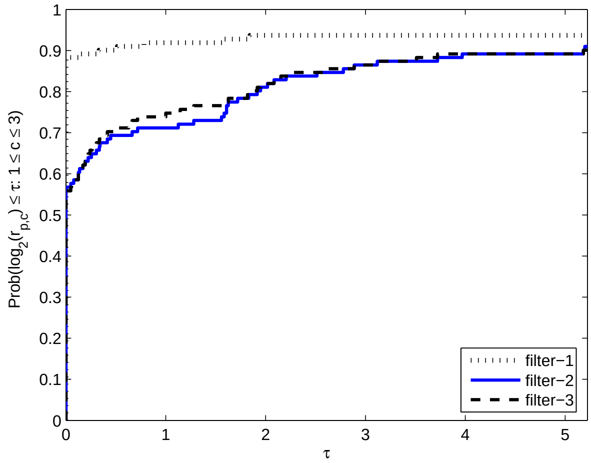

Figure 2: Performance profiles in a log_2 scale: number of optimality measure evaluations

given x_0 and set $y_0 = 1$. The third experience sets $B_0 \approx \nabla^2 F(x_0)$, with guaranteed positive definiteness, and the fourth uses $B_0 = I$.

For each version, we combine the results of the four experiments and select the best result for each problem. Then, we record the following performance metrics: number of iterations and number of θ_{op} evaluations.

To evaluate and compare the performance of the three proposed versions for the interior point filter line search method we use the performance profiles as outline in [3]. These profiles represent the cumulative distribution function for the performance ratio based on a chosen metric. A brief explanation follows.

Let $\mathcal{P}$ be the set of problems and $\mathcal{C}$ the set of codes 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

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

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 τ 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 ρ_c is the cumulative distribution function for the performance ratio.

To simplify the notation we denote the three tested versions by:

`filter-1` (filter method based on original condition (8);

`filter-2` (filter method based on the acceptance condition (13));

`filter-3` (filter based on the acceptance condition (14)).

Figure 1 shows the performance profiles for the number of iterations required to solve the problems, considering the convergence criteria of Algorithm 1, of the three versions. The figure gives a clear indication of the relative performance of each code/version.

The value of $\rho(\tau)$ for $\tau \approx 0$ gives the probability that the code will win over the others in the set. However, for large values of τ, the $\rho(\tau)$ measures the code robustness. The code with largest $\rho(\tau)$ is the one that solves more problems in the set $\mathcal{P}$.

We observe from Figure 1 that on this test set the performance profile for code `filter-1` (original version) lies above the other two. By examining ρ at the left side of the plot, one may conclude that `filter-1` is the most efficient, in terms of number of iterations, on almost 83% of the problems. Observing the other end of the plot, we conclude that `filter-1` solves most problems to optimality (approximately 93%). Figure 1 also shows that each of the codes fails on at least 7% of the problems.

Figure 2 shows the performance profiles for the number of θ_{op} evaluations. Similar conclusions can be drawn from these profiles.

5 Conclusions

A primal-dual interior point method based on a filter line search approach is presented. The new approach defines three components for each entry in the filter: the feasibility, centrality and optimality. We propose three different versions for some of the acceptance conditions for a trial iterate to be acceptable to the filter.

The versions are tested with a set of well-known problems and compared using a benchmarking tool with performance profiles. The used metrics were the number of iterations and the number of optimality measure evaluations. The numerical results show that the original filter line search algorithm, based on condition (8), is superior in terms of efficiency to the other versions.

Thus, using more restrictive conditions to consider a trial iterate to be acceptable to the filter than condition (8), does not seem to be effective as far as iteration and θ_{op} counts are considered. We note 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 tested.

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* 23, 2002, pp. 257–272.

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

[3] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance porfiles, *Mathematical Programming A* 91, 2002, pp. 201–213.

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

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

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

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

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

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

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

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

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

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