Practical implementation of an interior point nonmonotone line search filter method

M. FERNANDA P. COSTA† and EDITE M.G.P. FERNANDES∗‡

†Mathematics for Science and Technology Department, University of Minho, 4800-058 Guimarães, Portugal
‡Production and Systems Department, University of Minho, 4710-057 Braga, Portugal

(*v3.2 released May 2006*)

Here we present a primal-dual interior point nonmonotone line search filter method for nonlinear programming. The filter relies on three measures, the feasibility, the centrality and the optimality presented in the optimality conditions, considers relaxed acceptability criteria for the step size and includes a feasibility restoration phase. The evaluation of the method is until now made on small problems and a comparison is provided with a merit function approach.

Keywords: Nonlinear programming; Interior point method; Filter method; Line search method; Nonmonotone strategy

AMS Subject Classifications: 90C51; 90C30

1. Introduction

In this paper we propose a primal-dual interior point method for nonlinear optimization that relies on a line search filter strategy to allow convergence from poor starting points. Some line search frameworks use penalty merit functions to enforce progress toward the solution. Fletcher and Leyffer [1] recently proposed a filter method, as an alternative to merit functions, as a tool to guarantee global convergence in algorithms for nonlinear optimization. This technique incorporates the concept of nondominance to build a filter that is able to accept trial points if they improve the objective function or the constraints violation instead of a combination of those two measures defined by a merit function. The filter replaces the use of merit functions, so avoiding the update of penalty parameters that are associated with the penalization of the constraints in a merit function. The filter technique has already been adapted to interior point methods. The authors in [2] define two entries for the filter and use a trust-region strategy. The two entries combine the three criteria of the first order optimality conditions: the first entry is a measure of quasi-centrality and the second is an optimality measure combining complementarity and criticality. Work done in [3–6] also considers two entries in the filter, one is the barrier objective function and the other is the constraints violation, but uses a line search strategy.

The algorithm herein presented considers three entries in the filter and uses a line search approach. Further, a nonmonotone strategy, similar to the one proposed in [7], is imposed on the trial point acceptability conditions of the algorithm. Primal-dual interior point methods seem adequate to the filter implementation as the feasibility, centrality and optimality measures in the optimality conditions are natural candidates to the components of the filter. The nonmonotone line search technique that has been successfully applied in an unconstrained optimization context [7,8] and in systems of nonlinear equations [9] seems quite interesting in the sense that the conditions that are used to define a new acceptable point are relaxed without affecting the global convergence.

∗Corresponding author. Email: emgpf@dps.uminho.pt

The paper is organized as follows. Section 2 briefly describes the interior point framework, Section 3 presents the main ideas of the nonmonotone line search filter method and Section 4 contains the numerical experiments. The conclusions and future developments make Section 5.

2.   The interior point framework

The herein adopted formulation of a general nonlinear constrained optimization problem with simple bounds is as follows:

$$
\begin{aligned}
\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & F(\boldsymbol{x}) \\
\text{s.t.} \quad & \boldsymbol{b} \leq \boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{b} + \boldsymbol{r} \\
& \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}
\end{aligned}
\tag{1}
$$

where $h_j : \mathbb{R}^n \to \mathbb{R}$ for $j = 1, \ldots, m$ and $F : \mathbb{R}^n \to \mathbb{R}$ are nonlinear and twice continuously differentiable functions. $\boldsymbol{r}$ is the vector of ranges on the constraints $\boldsymbol{h}$, $\boldsymbol{u}$ and $\boldsymbol{l}$ are the vectors of upper and lower bounds on the variables and $\boldsymbol{b}$ is assumed to be a finite real vector. Elements of the vectors $\boldsymbol{r}$, $\boldsymbol{l}$ and $\boldsymbol{u}$ are real numbers subject to the following limitations: $0 \leq r_j \leq \infty$, $-\infty \leq l_i, u_i \leq \infty$ for $j = 1, \ldots, m$, $i = 1, \ldots, n$. Constraints of the form $\boldsymbol{b} \leq \boldsymbol{h}(\boldsymbol{x}) \leq \boldsymbol{b} + \boldsymbol{r}$ are denoted by range constraints. When upper and lower bounds on the $\boldsymbol{x}$ variables do not exist, the vector $\boldsymbol{x}$ is considered free. Equality constraints are treated as range constraints with $\boldsymbol{r} = \boldsymbol{0}$.

This section briefly describes an infeasible primal-dual interior point method for solving (1). We refer to [10] for details. Adding nonnegative slack variables $\boldsymbol{w}$, $\boldsymbol{p}$, $\boldsymbol{g}$ and $\boldsymbol{t}$, problem (1) becomes

$$
\begin{aligned}
\min \quad & F(\boldsymbol{x}) \\
\text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{w} = \boldsymbol{b} \\
& \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{p} = \boldsymbol{b} + \boldsymbol{r} \\
& \boldsymbol{x} - \boldsymbol{g} = \boldsymbol{l} \\
& \boldsymbol{x} + \boldsymbol{t} = \boldsymbol{u} \\
& \boldsymbol{w}, \boldsymbol{p}, \boldsymbol{g}, \boldsymbol{t} \geq \boldsymbol{0}.
\end{aligned}
\tag{2}
$$

Incorporating the nonnegativity constraints in logarithmic barrier terms in the objective function, problem (2) is transformed into

$$
\min \Phi_\mu(\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{p}, \boldsymbol{g}, \boldsymbol{t}) \equiv F(\boldsymbol{x}) - \mu \sum_{j=1}^{m} \ln(w_j) - \mu \sum_{j=1}^{m} \ln(p_j) - \mu \sum_{i=1}^{n} \ln(g_i) - \mu \sum_{i=1}^{n} \ln(t_i)
$$

subject to the same set of equality constraints, where μ is a positive barrier parameter. The optimality conditions for this problem produce the standard primal-dual system

$$
\begin{aligned}
\boldsymbol{\nabla} F(\boldsymbol{x}) - \boldsymbol{\nabla} h(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{y} - \boldsymbol{z} + \boldsymbol{s} &= \boldsymbol{0} \\
\boldsymbol{y} + \boldsymbol{q} - \boldsymbol{v} &= \boldsymbol{0} \\
\mathbf{WV} \boldsymbol{e}_1 &= \mu \boldsymbol{e}_1 \\
\mathbf{PQ} \boldsymbol{e}_1 &= \mu \boldsymbol{e}_1 \\
\mathbf{GZ} \boldsymbol{e}_2 &= \mu \boldsymbol{e}_2 \\
\mathbf{TS} \boldsymbol{e}_2 &= \mu \boldsymbol{e}_2 \\
\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{w} - \boldsymbol{b} &= \boldsymbol{0} \\
\boldsymbol{w} + \boldsymbol{p} - \boldsymbol{r} &= \boldsymbol{0} \\
\boldsymbol{x} - \boldsymbol{g} - \boldsymbol{l} &= \boldsymbol{0} \\
\boldsymbol{x} + \boldsymbol{t} - \boldsymbol{u} &= \boldsymbol{0},
\end{aligned}
\tag{3}
$$

where we have introduced a new vector $\boldsymbol{y} = \boldsymbol{v} - \boldsymbol{q}$ and used an equivalent condition $\boldsymbol{w} + \boldsymbol{p} = \boldsymbol{r}$ instead of $\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{p} = \boldsymbol{b} + \boldsymbol{r}$ in order to put the system to be solved as efficient as possible. The vectors $\boldsymbol{v}$, $\boldsymbol{q}$, $\boldsymbol{z}$ and

s are the dual variables, $\boldsymbol{\nabla}F$ is the gradient vector of F, $\boldsymbol{\nabla}h$ is the Jacobian matrix of the constraints $\boldsymbol{h}$, $\mathbf{W} = \mathrm{diag}(w_j)$, $\mathbf{P} = \mathrm{diag}(p_j)$, $\mathbf{G} = \mathrm{diag}(g_i)$, $\mathbf{T} = \mathrm{diag}(t_i)$, $\mathbf{V} = \mathrm{diag}(v_j)$, $\mathbf{Q} = \mathrm{diag}(q_j)$, $\mathbf{Z} = \mathrm{diag}(z_i)$ and $\mathbf{S} = \mathrm{diag}(s_i)$ are diagonal matrices, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are m and n vectors of all ones.

The first two equations define the conditions of dual feasibility, the next four equations are the centrality conditions and the last four define the primal feasibility. This is a nonlinear system of $5n + 5m$ equations in $5n + 5m$ unknowns. Applying the Newton's method to solve system (3), we obtain the linear system of equations

$$
\begin{aligned}
-\mathbf{H}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{\Delta}x + \boldsymbol{\nabla}h(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Delta}y + \boldsymbol{\Delta}z - \boldsymbol{\Delta}s &= \boldsymbol{\nabla}F(\boldsymbol{x}) - \boldsymbol{\nabla}h(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{z} + \boldsymbol{s} \equiv \boldsymbol{\sigma} \\
\boldsymbol{\Delta}y - \boldsymbol{\Delta}q + \boldsymbol{\Delta}v &= \boldsymbol{y} + \boldsymbol{q} - \boldsymbol{v} \equiv \boldsymbol{\beta} \\
\boldsymbol{\Delta}w + \mathbf{V}^{-1}\mathbf{W}\boldsymbol{\Delta}v &= \mu\mathbf{V}^{-1}\boldsymbol{e}_1 - \boldsymbol{w} \equiv \boldsymbol{\gamma}_w \\
\mathbf{P}^{-1}\mathbf{Q}\boldsymbol{\Delta}p + \boldsymbol{\Delta}q &= \mu\mathbf{P}^{-1}\boldsymbol{e}_1 - \boldsymbol{q} \equiv \boldsymbol{\gamma}_q \\
\mathbf{G}^{-1}\mathbf{Z}\boldsymbol{\Delta}g + \boldsymbol{\Delta}z &= \mu\mathbf{G}^{-1}\boldsymbol{e}_2 - \boldsymbol{z} \equiv \boldsymbol{\gamma}_z \\
\mathbf{T}^{-1}\mathbf{S}\boldsymbol{\Delta}t + \boldsymbol{\Delta}s &= \mu\mathbf{T}^{-1}\boldsymbol{e}_2 - \boldsymbol{s} \equiv \boldsymbol{\gamma}_s \\
\boldsymbol{\nabla}h(\boldsymbol{x})\boldsymbol{\Delta}x - \boldsymbol{\Delta}w &= \boldsymbol{w} + \boldsymbol{b} - \boldsymbol{h}(\boldsymbol{x}) \equiv \boldsymbol{\rho} \\
\boldsymbol{\Delta}p + \boldsymbol{\Delta}w &= \boldsymbol{r} - \boldsymbol{w} - \boldsymbol{p} \equiv \boldsymbol{\alpha} \\
\boldsymbol{\Delta}x - \boldsymbol{\Delta}g &= \boldsymbol{l} - \boldsymbol{x} + \boldsymbol{g} \equiv \boldsymbol{\upsilon} \\
\boldsymbol{\Delta}x + \boldsymbol{\Delta}t &= \boldsymbol{u} - \boldsymbol{x} - \boldsymbol{t} \equiv \boldsymbol{\tau}
\end{aligned}
\tag{4}
$$

for the search directions $\boldsymbol{\Delta}x, \boldsymbol{\Delta}w, \boldsymbol{\Delta}p, \boldsymbol{\Delta}g, \boldsymbol{\Delta}t, \boldsymbol{\Delta}y, \boldsymbol{\Delta}v, \boldsymbol{\Delta}q, \boldsymbol{\Delta}z$ and $\boldsymbol{\Delta}s$, where $\mathbf{H}(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\nabla}^2 F(\boldsymbol{x}) - \sum_{j=1}^{m} y_j \boldsymbol{\nabla}^2 h_j(\boldsymbol{x})$.

Based on the computed search directions, the algorithm proceeds iteratively, choosing a step size $\overline{\alpha} \in (0, \overline{\alpha}^{\mathrm{max}}]$ at each iteration and determining the new iterates by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \overline{\alpha}\boldsymbol{\Delta}x_k$, $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \overline{\alpha}\boldsymbol{\Delta}w_k$, ..., $\boldsymbol{s}_{k+1} = \boldsymbol{s}_k + \overline{\alpha}\boldsymbol{\Delta}s_k$. $\overline{\alpha}^{\mathrm{max}}$ is the longest step size that can be taken, with an upper bound of 1, along the search directions to ensure the nonnegativity of the slack and dual variables.

Implementation details to provide initial values for all the variables, as well as to solve system (4) and to compute $\overline{\alpha}^{\mathrm{max}}$ are described in [10]. The strategy to recover μ at each iteration considers a fraction of the average complementarity

$$
\mu = \delta_\mu \frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{p}^{\mathrm{T}}\boldsymbol{q} + \boldsymbol{g}^{\mathrm{T}}\boldsymbol{z} + \boldsymbol{t}^{\mathrm{T}}\boldsymbol{s}}{2m + 2n}
\tag{5}
$$

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

The procedure that decides which trial step size is accepted in this interior point paradigm is a nonmonotone line search filter method.

3. Nonmonotone line search filter method

In this section we present the main ideas of a triple entry filter method based on a nonmonotone line search approach. To abbreviate the notation we set

$$
\begin{aligned}
\boldsymbol{u} &= (\boldsymbol{x},\boldsymbol{w},\boldsymbol{p},\boldsymbol{g},\boldsymbol{t},\boldsymbol{y},\boldsymbol{v},\boldsymbol{q},\boldsymbol{z},\boldsymbol{s}), & \boldsymbol{\Delta} &= (\boldsymbol{\Delta}x,\boldsymbol{\Delta}w,\boldsymbol{\Delta}p,\boldsymbol{\Delta}g,\boldsymbol{\Delta}t,\boldsymbol{\Delta}y,\boldsymbol{\Delta}v,\boldsymbol{\Delta}q,\boldsymbol{\Delta}z,\boldsymbol{\Delta}s), \\
\boldsymbol{u}^1 &= (\boldsymbol{x},\boldsymbol{w},\boldsymbol{p},\boldsymbol{g},\boldsymbol{t}), & \boldsymbol{\Delta}^1 &= (\boldsymbol{\Delta}x,\boldsymbol{\Delta}w,\boldsymbol{\Delta}p,\boldsymbol{\Delta}g,\boldsymbol{\Delta}t), \\
\boldsymbol{u}^2 &= (\boldsymbol{w},\boldsymbol{p},\boldsymbol{g},\boldsymbol{t},\boldsymbol{v},\boldsymbol{q},\boldsymbol{z},\boldsymbol{s}), & \boldsymbol{\Delta}^2 &= (\boldsymbol{\Delta}w,\boldsymbol{\Delta}p,\boldsymbol{\Delta}g,\boldsymbol{\Delta}t,\boldsymbol{\Delta}v,\boldsymbol{\Delta}q,\boldsymbol{\Delta}z,\boldsymbol{\Delta}s), \\
\boldsymbol{u}^3 &= (\boldsymbol{x},\boldsymbol{y},\boldsymbol{v},\boldsymbol{q},\boldsymbol{z},\boldsymbol{s}), & \boldsymbol{\Delta}^3 &= (\boldsymbol{\Delta}x,\boldsymbol{\Delta}y,\boldsymbol{\Delta}v,\boldsymbol{\Delta}q,\boldsymbol{\Delta}z,\boldsymbol{\Delta}s).
\end{aligned}
$$

To adapt the methodology of a filter as outline in [1] to this interior point method, we propose three quantities for the filter entries. The first component corresponds to feasibility, the second component corresponds to centrality and the third to optimality. With this choice of filter components, it seems natural to use the optimality conditions (3) to define these components. Therefore, in terms of our filter

approach, we choose the following measures of *feasibility, centrality* and *optimality*:

$$\theta_{feas}(\boldsymbol{u}^1) = \|\boldsymbol{\rho}\|_2 + \|\boldsymbol{\alpha}\|_2 + \|\boldsymbol{\upsilon}\|_2 + \|\boldsymbol{\tau}\|_2,$$
$$\theta_c(\boldsymbol{u}^2) = \|\boldsymbol{\gamma}_w\|_2 + \|\boldsymbol{\gamma}_q\|_2 + \|\boldsymbol{\gamma}_z\|_2 + \|\boldsymbol{\gamma}_s\|_2,$$
$$\theta_{op}(\boldsymbol{u}^3) = \tfrac{1}{2}\left(\|\boldsymbol{\sigma}\|_2^2 + \|\boldsymbol{\beta}\|_2^2\right).$$

After a search direction $\boldsymbol{\Delta}_k$ has been computed, we borrow the ideas in [7] and consider a backtracking nonmonotone line search procedure, where a decreasing sequence of step sizes $\overline{\alpha}_{k,l} \in (0, \overline{\alpha}_k^{\max}]$ $(l = 0, 1, \ldots)$, with $lim_l \, \overline{\alpha}_{k,l} = 0$, is tried until an acceptance criterion is satisfied. Here, we use l to denote the iteration counter for the inner loop.

3.1.   *Nonmonotone sufficient reduction*

Nonmonotone line search methods that use a merit function ensure sufficient progress toward the solution by imposing that the merit function value at each new iterate satisfies an Armijo condition with respect to the maximum merit function value of a prefixed number of previous iterates. We refer to [7] for details. Thus, the normal condition which implies a monotonic decrease of the merit function is relaxed without affecting the global convergence.

Following this idea, the trial point $\boldsymbol{u}_k(\overline{\alpha}_{k,l}) = \boldsymbol{u}_k + \overline{\alpha}_{k,l}\boldsymbol{\Delta}_k$ in the backtracking line search technique is considered acceptable, if it leads to sufficient progress in one of the three measures compared to a previous iterate that yields the maximum value of the corresponding measure within the last $m(k)$ iterates, where $m(0) = 0$ and $0 \le m(k) \le \min\{m(k-1) + 1, M\}, k \ge 1$ (M is a nonnegative integer), i.e., if

$$\theta_{feas}(\boldsymbol{u}_k^1(\overline{\alpha}_{k,l})) \le \left(1 - \gamma_{\theta_f}\right)\theta_{feas}^{k-\overline{j}} \text{ or } \theta_c(\boldsymbol{u}_k^2(\overline{\alpha}_{k,l})) \le \left(1 - \gamma_{\theta_c}\right)\theta_c^{k-\overline{j}} \text{ or } \theta_{op}(\boldsymbol{u}_k^3(\overline{\alpha}_{k,l})) \le \theta_{op}^{k-\overline{j}} - \gamma_{\theta_o}\theta_{feas}^{k-\overline{j}} \quad (6)$$

holds for fixed constants $\gamma_{\theta_f}, \gamma_{\theta_c}, \gamma_{\theta_o} \in (0, 1)$, where we define

$$\theta_{feas}^{k-\overline{j}} = \max_{0 \le \overline{j} \le m(k)} \theta_{feas}(\boldsymbol{u}_{k-\overline{j}}^1), \ \theta_c^{k-\overline{j}} = \max_{0 \le \overline{j} \le m(k)} \theta_c(\boldsymbol{u}_{k-\overline{j}}^2) \ \text{and} \ \theta_{op}^{k-\overline{j}} = \max_{0 \le \overline{j} \le m(k)} \theta_{op}(\boldsymbol{u}_{k-\overline{j}}^3).$$

However, to prevent convergence to a feasible but nonoptimal point, and whenever, for the current iterate $\theta_{feas}(\boldsymbol{u}_k^1) \le \theta_{feas}^{\min}$, for some $\theta_{feas}^{\min} \in (0, \infty]$, and the following switching conditions

$$\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3 < 0 \ \text{ and } \ \overline{\alpha}_{k,l}\left[-\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3\right]^{s_{op}} > \delta\left[\theta_{feas}^{k-\overline{j}}\right]^{s_{feas}} \quad (7)$$

hold with fixed constants $\delta > 0, s_{feas} > 1, s_{op} > 2s_{feas}$, a different nonmonotone reduction criterion is imposed on the trial point $\boldsymbol{u}_k(\overline{\alpha}_{k,l})$

$$\theta_{op}(\boldsymbol{u}_k^3(\overline{\alpha}_{k,l})) \le \theta_{op}^{k-\overline{j}} + \eta_{op}\overline{\alpha}_{k,l}\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3, \quad (8)$$

instead of (6) [4–6]. Here $\eta_{op} \in (0, 0.5)$ is a constant.

According to a previous publication on filter methods [11], a trial step size $\overline{\alpha}_{k,l}$ for which (7) holds, is called a "θ_{op} -step". Similarly, if a "θ_{op}-step" is accepted as the final step size $\overline{\alpha}_k$ in iteration k, then k is referred as a "θ_{op}-type iteration".

In order to prevent cycling between iterates that improve either the feasibility, the centrality or the optimality, at each iteration k, the algorithm maintains a filter, here denoted by

$$\overline{F}_k \subseteq \left\{(\theta_{feas}, \theta_c, \theta_{op}) \in \mathbb{R}^3 : \theta_{feas} \ge 0, \theta_c \ge 0, \theta_{op} \ge 0\right\}.$$

Following the ideas in [2, 4–6], the filter here is a set $\overline{F}_k$ that contains those combinations of values of θ_{feas}, θ_c and θ_{op} that are prohibited for a successful trial point in iteration k. So, during the line search, a trial point $\boldsymbol{u}_k(\overline{\alpha}_{k,l})$ is rejected, if

$$\big(\theta_{feas}(\boldsymbol{u}_k^1(\overline{\alpha}_{k,l})), \theta_c(\boldsymbol{u}_k^2(\overline{\alpha}_{k,l})), \theta_{op}(\boldsymbol{u}_k^3(\overline{\alpha}_{k,l}))\big) \in \overline{F}_k.$$

At the beginning of the optimization, the filter is initialized to

$$\overline{F}_0 \subseteq \big\{(\theta_{feas}, \theta_c, \theta_{op}) \in \mathbb{R}^3 : \theta_{feas} \geq \theta_{feas}^{\max}, \theta_c \geq \theta_c^{\max}, \theta_{op} \geq \theta_{op}^{\max}\big\} \tag{9}$$

for some $\theta_{feas}^{\max}, \theta_c^{\max}$ and $\theta_{op}^{\max}$, so that the algorithm will never allow trial points to be accepted that have values of θ_{feas}, θ_c and θ_{op} larger than $\theta_{feas}^{\max}$, $\theta_c^{\max}$ and $\theta_{op}^{\max}$, respectively.

The filter is augmented, using the update formula

$$\overline{F}_{k+1} = \overline{F}_k \cup \Big\{(\theta_{feas}, \theta_c, \theta_{op}) \in \mathbb{R}^3 : \theta_{feas} > \big(1 - \gamma_{\theta_f}\big)\theta_{feas}^{k-\overline{j}} \ \text{ and}$$
$$\theta_c > (1 - \gamma_{\theta_c})\theta_c^{k-\overline{j}} \ \text{ and } \ \theta_{op} > \theta_{op}^{k-\overline{j}} - \gamma_{\theta_o}\theta_{feas}^{k-\overline{j}}\Big\} \tag{10}$$

after every iteration in which the accepted trial step size satisfies (6). On the other hand, if (7) and (8) hold for the accepted step size, the filter remains unchanged.

Finally, in some cases it is not possible to find a trial step size $\overline{\alpha}_{k,l}$ that satisfies the above criteria. A minimum desired step size is defined using linear models of the involved functions

$$\overline{\alpha}_k^{\min} = \xi \begin{cases} \min\left\{\gamma_{\theta_f}, \dfrac{\theta_{op}(\boldsymbol{u}_k^3) - \theta_{op}^{k-\overline{j}} + \gamma_{\theta_o}\theta_{feas}^{k-\overline{j}}}{-\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3}, \dfrac{\delta\left[\theta_{feas}^{k-\overline{j}}\right]^{s_{feas}}}{\left[-\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3\right]^{s_{op}}}\right\}, \\[1em] \qquad \text{if } \boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3 < 0 \text{ and } \theta_{feas}(\boldsymbol{u}_k^1) \leq \theta_{feas}^{\min} \\[1em] \min\left\{\gamma_{\theta_f}, \dfrac{\theta_{op}(\boldsymbol{u}_k^3) - \theta_{op}^{k-\overline{j}} + \gamma_{\theta_o}\theta_{feas}^{k-\overline{j}}}{-\boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3}\right\}, \\[1em] \qquad \text{if } \boldsymbol{\nabla}\theta_{op}(\boldsymbol{u}_k^3)^{\mathrm{T}}\boldsymbol{\Delta}_k^3 < 0 \text{ and } \theta_{feas}(\boldsymbol{u}_k^1) > \theta_{feas}^{\min} \\[1em] \gamma_{\theta_f}, \qquad \text{otherwise} \end{cases} \tag{11}$$

with a safety factor $\xi \in (0, 1]$. If the backtracking nonmonotone line search technique finds a trial step size $\overline{\alpha}_{k,l} < \overline{\alpha}_k^{\min}$, the algorithm reverts to a feasibility restoration phase. Here, the algorithm tries to find a new iterate $\boldsymbol{u}_{k+1}$ which is acceptable to the current filter for which (6) holds, by decreasing either the infeasibility or the centrality.

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

Algorithm 1 (nonmonotone line search filter algorithm)

Given: Starting point $\boldsymbol{u}_0$, constants $\theta_{feas}^{\max} \in (\theta_{feas}(\boldsymbol{u}_0^1), \infty]$; $\theta_c^{\max} \in (\theta_c(\boldsymbol{u}_0^2), \infty]$; $\theta_{op}^{\max} \in (\theta_{op}(\boldsymbol{u}_0^3), \infty]$; $\theta_{feas}^{\min} > 0$; $\gamma_{\theta_f}, \gamma_{\theta_c}, \gamma_{\theta_o} \in (0, 1)$; $\delta > 0$; $\xi \in (0, 1]$; $s_{feas} > 1$; $s_{op} > 2s_{feas}$; $\eta_{op}, \eta_{\theta_{2,f}}, \eta_{\theta_{2,c}}, \varepsilon \in (0, 1]$; M; $\epsilon_{tol} > 0 \, (\ll 1)$.

1. *Initialize.* Initialize the filter (using (9)), the iteration counter $k \leftarrow 0$ and $m(0) \leftarrow 0$.
2. *Check convergence.* Stop if the relative measures of primal and dual infeasibilities are less or equal to a given error tolerance ϵ_{tol}.
3. *Compute search direction.* Compute the search direction $\boldsymbol{\Delta}_k$ from the linear system (4).

4. *Backtracking line search.*

   4.1 *Initialize line search.* Compute the longest step size $\overline{\alpha}_k^{\max}$ to ensure positivity of slack and dual variables. Set $l \leftarrow 0$, $\overline{\alpha}_{k,l} = \overline{\alpha}_k^{\max}$.

   4.2 *Compute new trial point.* If the trial step size becomes too small, i.e., $\overline{\alpha}_{k,l} < \overline{\alpha}_k^{\min}$ with $\overline{\alpha}_k^{\min}$ defined by (11), go to feasibility restoration phase in step 8. Otherwise, compute the trial point $\boldsymbol{u}_k(\overline{\alpha}_{k,l})$ and recalculate μ using (5).

   4.3 *Check acceptability to the filter.* If $\left(\theta_{feas}(\boldsymbol{u}_k^1(\overline{\alpha}_{k,l})), \theta_c(\boldsymbol{u}_k^2(\overline{\alpha}_{k,l})), \theta_{op}(\boldsymbol{u}_k^3(\overline{\alpha}_{k,l}))\right) \in \overline{F}_k$, reject the trial step size and go to step 4.6.

   4.4 *Check sufficient decrease with respect to last $m(k)$ iterates.*
       If $\overline{\alpha}_{k,l}$ is a θ_{op}-step size ((7) holds) and the generalization of the Armijo condition (8) for the θ_{op} function holds, accept the trial step and go to step 5.

   4.5 *Check sufficient decrease with respect to last $m(k)$ iterates.* If (6) holds, accept the trial step and go to step 5. Otherwise go to step 4.6.

   4.6 *Choose new trial step size.* Set $\overline{\alpha}_{k,l+1} = \overline{\alpha}_{k,l}/2$, $l \leftarrow l+1$, and go back to step 4.2.

5. *Accept trial point.* Set $\overline{\alpha}_k \leftarrow \overline{\alpha}_{k,l}$ and $\boldsymbol{u}_k \leftarrow \boldsymbol{u}_k(\overline{\alpha}_k)$.

6. *Augment the filter if necessary.* If k is not a θ_{op}-type iteration, augment the filter using (10). Otherwise, leave the filter unchanged.

7. *Continue with next iteration.* Increase the iteration counter $k \leftarrow k+1$, $m(k) = \min\{m(k-1)+1, M\}$ and go back to step 2.

8. *Feasibility restoration phase.* Use a restoration algorithm to produce a point $\boldsymbol{u}_{k+1}$ that is acceptable to the filter, i.e., $\left(\theta_{feas}(\boldsymbol{u}_{k+1}^1), \theta_c(\boldsymbol{u}_{k+1}^2), \theta_{op}(\boldsymbol{u}_{k+1}^3)\right) \notin \overline{F}_k$. Augment the filter using (10) and continue with the regular iteration in step 7.

3.2. *Feasibility restoration phase*

In this section we present a restoration algorithm. The task of the restoration phase is to compute a new iterate acceptable to the filter by decreasing either the infeasibility or the centrality, whenever the nonmonotone backtracking line search procedure cannot make sufficient progress and the step size becomes too small. Thus, the purpose of a restoration algorithm is to decrease the value of $\theta_{feas}(\boldsymbol{u}^1)$ or $\theta_c(\boldsymbol{u}^2)$. To achieve this goal we introduce the functions

$$\theta_{2,f}(\boldsymbol{u}^1) = \tfrac{1}{2}\left(\|\boldsymbol{\rho}\|_2^2 + \|\boldsymbol{\alpha}\|_2^2 + \|\boldsymbol{v}\|_2^2 + \|\boldsymbol{\tau}\|_2^2\right),$$
$$\theta_{2,c}(\boldsymbol{u}^2) = \tfrac{1}{2}\left(\|\boldsymbol{\gamma}_w\|_2^2 + \|\boldsymbol{\gamma}_q\|_2^2 + \|\boldsymbol{\gamma}_z\|_2^2 + \|\boldsymbol{\gamma}_s\|_2^2\right).$$

The restoration algorithm herein presented works with the steps $\boldsymbol{\Delta}^1$ and $\boldsymbol{\Delta}^2$, computed from (4), that should be descent directions for $\theta_{2,f}(\boldsymbol{u}^1)$ and $\theta_{2,c}(\boldsymbol{u}^2)$, respectively. In fact,

$$\begin{aligned}
\boldsymbol{\nabla}\theta_{2,f}(\boldsymbol{u}^1)^{\mathrm{T}}\boldsymbol{\Delta}^1 &= -\boldsymbol{\rho}^{\mathrm{T}}\left(\boldsymbol{\nabla}h\boldsymbol{\Delta}x - \boldsymbol{\Delta}w\right) - \boldsymbol{v}^{\mathrm{T}}\left(\boldsymbol{\Delta}x - \boldsymbol{\Delta}g\right) + \boldsymbol{\tau}^{\mathrm{T}}\left(-\boldsymbol{\Delta}x - \boldsymbol{\Delta}t\right) + \\
&\quad + \boldsymbol{\alpha}^{\mathrm{T}}\left(-\boldsymbol{\Delta}w - \boldsymbol{\Delta}p\right) \\
&= -\boldsymbol{\rho}^{\mathrm{T}}\boldsymbol{\rho} - \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} - \boldsymbol{\tau}^{\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{\alpha} = -2\theta_{2,f}(\boldsymbol{u}^1) < 0
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{\nabla}\theta_{2,c}(\boldsymbol{u}^2)^{\mathrm{T}}\boldsymbol{\Delta}^2 &= \boldsymbol{\gamma}_w^{\mathrm{T}}(-\boldsymbol{\Delta}w - \mathbf{W}\mathbf{V}^{-1}\boldsymbol{\Delta}v) + \boldsymbol{\gamma}_q^{\mathrm{T}}\left(-\mathbf{Q}\mathbf{P}^{-1}\boldsymbol{\Delta}p - \boldsymbol{\Delta}q\right) + \\
&\quad + \boldsymbol{\gamma}_z^{\mathrm{T}}\left(-\mathbf{Z}\mathbf{G}^{-1}\boldsymbol{\Delta}g - \boldsymbol{\Delta}z\right) + \boldsymbol{\gamma}_s^{\mathrm{T}}\left(-\mathbf{S}\mathbf{T}^{-1}\boldsymbol{\Delta}t - \boldsymbol{\Delta}s\right) \\
&= -\boldsymbol{\gamma}_w^{\mathrm{T}}\boldsymbol{\gamma}_w - \boldsymbol{\gamma}_q^{\mathrm{T}}\boldsymbol{\gamma}_q - \boldsymbol{\gamma}_z^{\mathrm{T}}\boldsymbol{\gamma}_z - \boldsymbol{\gamma}_s^{\mathrm{T}}\boldsymbol{\gamma}_s = -2\theta_{2,c}(\boldsymbol{u}^2) < 0.
\end{aligned}$$

Additionally, we also ensure that the new iterate $\boldsymbol{x}_{k+1}$ does not deviate too much from the current iterate $\boldsymbol{x}_k$ (see step 5 in Algorithm *2*).

Although other restoration algorithms could be used, the following one seems appropriate considering the step calculation of our interior point nonmonotone line search filter method. Note that in general, this type

of algorithm does not switch to the feasibility restoration phase if the current iterate is almost a feasible point. However, the restoration phase might be called at an iterate with very small θ_{feas} which would not lead to a sufficient reduction on the $\theta_{2,f}$ measure. Thus, to promote some progress our restoration algorithm tries to enforce some reduction of the centrality measure $\theta_{2,c}$.

Algorithm 2 (restoration algorithm)

0. Set $\overline{\alpha}_{k,0}^{\max} = \overline{\alpha}_k^{\max}$, $\boldsymbol{u}_{k,0} = \boldsymbol{u}_k$, $l = 0$, $m(0) = 0$ and start with step 4.
1. If $\boldsymbol{u}_{k,l}$ is acceptable to the filter (6) then set $\boldsymbol{u}_{k+1} = \boldsymbol{u}_{k,l}$ and stop restoration.
2. Compute $\boldsymbol{\Delta}_{k,l}$ by solving (4), with $\boldsymbol{u}_k = \boldsymbol{u}_{k,l}$.
3. Define the vectors $\boldsymbol{\Delta}_{k,l}^1$, $\boldsymbol{\Delta}_{k,l}^2$ which are used as search directions for the variables $\boldsymbol{u}_{k,l}^1$, $\boldsymbol{u}_{k,l}^2$. Compute $\overline{\alpha}_{k,l}^{\max}$.
4. Set $\overline{\alpha}_k = \overline{\alpha}_{k,l}^{\max}$.
5. Compute the trial point $\boldsymbol{u}_k(\overline{\alpha}_k)$ and recalculate μ using (5).

$$
\text{If } \left(\theta_{2,f}(\boldsymbol{u}_{k,l}^1(\overline{\alpha}_k)) \le \max_{0 \le \overline{j} \le m(l)} \theta_{2,f}(\boldsymbol{u}_{k,l-\overline{j}}^1) + \overline{\alpha}_k \eta_{\theta_{2,f}} \boldsymbol{\nabla}\theta_{2,f}(\boldsymbol{u}_{k,l}^1)^{\mathrm{T}} \boldsymbol{\Delta}_{k,l}^1 \right.
$$
$$
\text{or } \theta_{2,c}(\boldsymbol{u}_{k,l}^2(\overline{\alpha}_k)) \le \max_{0 \le \overline{j} \le m(l)} \theta_{2,c}(\boldsymbol{u}_{k,l-\overline{j}}^2) + \overline{\alpha}_k \eta_{\theta_{2,c}} \boldsymbol{\nabla}\theta_{2,c}(\boldsymbol{u}_{k,l}^2)^{\mathrm{T}} \boldsymbol{\Delta}_{k,l}^2 \left. \right)
$$
$$
\text{and } \left\| \boldsymbol{x}_{k,l}(\overline{\alpha}_k) - \boldsymbol{x}_k \right\|_2 \le \varepsilon \left(1 + \left\| \boldsymbol{x}_{k,l}(\overline{\alpha}_k) \right\|_2 \right)
$$

then set $\boldsymbol{u}_{k,l+1} = \boldsymbol{u}_{k,l}(\overline{\alpha}_k)$, $l = l + 1$, $m(l) = \min\{m(l-1)+1, M\}$ and return to step 1. Otherwise $\overline{\alpha}_k \leftarrow \overline{\alpha}_k/2$, and repeat step 5.

4. Numerical experiments

To test this interior point framework with a nonmonotone line search filter technique we selected 50 constrained problems from the Hock and Schittkowski (HS) collection [12]. This preliminary selection aims to consider small and simple to code problems mainly with inequality constraints and/or simple bounds, although there is also a set of problems with equality constraints. The problems were coded in Fortran according to the general formulation (1). The tests were done in double precision arithmetic with a Pentium 4 and Fortran 90.

The chosen values for the constants are similar to the ones used in [6]: $\theta_{feas}^{\min} = 10^{-4} \max\left\{1, \theta_{feas}(\boldsymbol{u}_0^1)\right\}$, $\theta_{feas}^{\max} = 10^4 \max\left\{1, \theta_{feas}(\boldsymbol{u}_0^1)\right\}$, $\theta_c^{\max} = 10^4 \max\left\{1, \theta_c(\boldsymbol{u}_0^2)\right\}$, $\theta_{op}^{\max} = 10^4 \max\left\{1, \theta_{op}(\boldsymbol{u}_0^3)\right\}$, $\gamma_{\theta_f} = \gamma_{\theta_c} = \gamma_{\theta_o} = 10^{-5}$, $\delta = 1$, $\xi = 0.05$, $s_{feas} = 1.1$, $s_{op} = 2.3$, $\eta_{op} = \eta_{\theta_{2,f}} = \eta_{\theta_{2,c}} = 10^{-4}$. We also used $\delta_\mu = 0.1$, $\varepsilon = 0.1$, $M = 5$ and $\epsilon_{tol} = 10^{-4}$.

We exercised the algorithm using a symmetric positive definite quasi-Newton BFGS approximation to the matrix $\mathbf{H}$. The iterative process uses a positive definite approximation to $\boldsymbol{\nabla}^2 F(\boldsymbol{x}_0)$ in the first iteration, except on those problems where singularity or ill-conditioning was encountered on that matrix and the identity matrix was used instead.

4.1. Comparative results

Table 1 summarizes the results under the "Filter Method" indication. Results inside parentheses were obtained with the identity matrix in the first iteration. We compare the nonmonotone line search filter method with a merit function approach based on the l_2 merit function ("Merit Function" in the table). Each set of two columns contains the number of iterations (N_{it}) needed to achieve a solution according to the convergence tolerance referred to in Algorithm *1*, and the number of θ_{op} evaluations (N_{op}) in the filter method or the number of merit function evaluations (N_{mf}) in the merit function approach.

Except with the problems HS1, HS5, HS16 and HS38, we managed to obtain a significant reduction in

Table 1. Numerical results

Problem	Filter Method N_{it}	N_{op}	Merit Function N_{it}	N_{mf}	Problem	Filter Method N_{it}	N_{op}	Merit Function N_{it}	N_{mf}
HS1	35	36	24	27	HS34	10	11	10	12
HS2	42	43	42	44	HS35	2	3	2	3
HS3	1	2	1	2	HS36	(9)	(10)	(9)	(10)
HS4	5	6	5	6	HS37	(13)	(14)	(13)	(14)
HS5	15	16	9	26	HS38	43	44	25	38
HS6	8	9	8	9	HS41	(9)	(10)	(9)	(11)
HS10	(13)	(14)	(13)	(14)	HS42	12	13	12	13
HS11	8	9	8	9	HS43	12	13	12	13
HS12	15	16	15	16	HS44	19	20	19	20
HS14	8	9	8	9	HS45	4	5	4	5
HS15	(17)	(18)	(18)	(20)	HS46	15	16	15	194
HS16	(12)	(15)	(11)	(12)	HS48	7	8	7	8
HS17	5	6	7	10	HS49	14	15	14	15
HS18	11	12	12	76	HS50	14	15	14	64
HS19	25	26	31	48	HS51	6	7	6	57
HS20	(22)	(23)	(22)	(25)	HS52	9	10	9	10
HS21	5	6	5	6	HS53	8	9	9	14
HS23	27	28	31	78	HS55	11	12	11	12
HS24	15	16	16	19	HS60	12	13	11	18
HS27	16	17	17	70	HS63	11	12	11	12
HS28	6	7	6	7	HS64	(46)	(108)	(55)	(235)
HS30	8	9	8	9	HS65	9	10	9	10
HS31	13	14	13	14	HS76	4	5	7	8
HS32	8	9	8	34	HS77	(18)	(19)	(18)	(48)
HS33	10	11	10	11	HS79	13	14	11	65

the number of N_{op} when compared with N_{mf} in 17 problems and a slight improvement in 4 problems. For the remaining 25 problems the two methods gave identical results.

The obtained improvements are mainly due to the filter approach. The proposed nonmonotone strategy which aims to relax the acceptability criteria for the step size accounts only for changes in a small number of problems. In fact, if $M = 0$ is considered in the algorithm, meaning that a monotone sufficient reduction is imposed by the criteria (6) and (8) as well as by the conditions presented in the step 5 of the Algorithm 2, the results are identical to the ones presented in the table, except for the following problems: HS5 ($N_{it} = 13, N_{op} = 15$), HS16 ($N_{it} = 13, N_{op} = 19$), HS24 ($N_{it} = 18, N_{op} = 148$), HS27 ($N_{it} = 19, N_{op} = 29$), HS37 ($N_{it} = 17, N_{op} = 76$) and HS64 ($N_{it} = 45, N_{op} = 104$).

4.2.   *Tuning some algorithm parameters*

This primal-dual interior point algorithm has some parameters that can be tuned so that the best solution for a problem is obtained. In what follows we briefly describe what we have observed when three particular parameters were altered. Two of them aim to provide initial approximations to the algorithm (*recompute* and δ_{pos}), and the other is the convergence error tolerance (ϵ_{tol}).

(1) When an initial approximation to the variable $\boldsymbol{x}$ is provided, $\boldsymbol{x}_0$, the algorithm either recomputes a better $\tilde{\boldsymbol{x}}_0$ and the variable $\boldsymbol{y}_0$ using a simplified reduced KKT system (see [10]) or uses the given $\boldsymbol{x}_0$ and sets $\boldsymbol{y}_0 = \mathbf{1}$. In the first case, the parameter *recompute* is set to `on` (the default value), and in the second one, the parameter has to be `off`.
(2) Besides the initial values for $\boldsymbol{x}$ and $\boldsymbol{y}$, the algorithm has to initialize the dual and slack variables. In order to guarantee that all these variables are positive, the algorithm relies on the parameter δ_{pos}, whose default value is 0.01.

The results shown in Table 1 were obtained with *recompute*=`on` and $\delta_{pos} = 0.01$. The obtained local optima for the problems HS45, HS46 and HS60 are not as good as the ones in [12]. However, if we set *recompute*=`off` then the solution in [12] is obtained for problem HS45 with $N_{it} = 34$ and $N_{op} = 67$. Furthermore, if we set $\delta_{pos} = 1$ the algorithm also converges to the solution with $N_{it} = 18$ and $N_{op} = 23$. For the problem HS60, when *recompute* is set to `off`, the algorithm reaches the solution in [12] with $N_{it} = 15$ and $N_{op} = 16$ with either $\delta_{pos} = 0.01$ or $\delta_{pos} = 1$. Finally, for the problem HS46, with

recompute=off the algorithm still converges to the previous solution.

(3) We now consider the convergence error tolerance ϵ_{tol}. The default value is 10^{-4}.

We rerun all problems setting $\epsilon_{tol} = 10^{-6}$. With the default values for *recompute* and δ_{pos}, the obtained results were similar to the ones in the Table 1. We observed some slightly better approximations and very few differences on N_{it} and N_{op} except for the problems HS41, HS45, HS46 and HS49. In these cases, the level of accuracy (either primal or dual) was not attained and the algorithm continues until the maximum number of iterations is reached (in our case 100). However, if we set *recompute* =off and $\delta_{pos} = 1$, problems HS41, HS45 and HS49 converge to the solution in [12]. Problem HS46 was the only one that did not converge within 100 iterations.

4.3. *Open issues*

We are aware that using the acceptability conditions (6) or (7) and (8), this filter algorithm while promoting convergence to a stationary point does not enforce a sufficient decrease in a measure of optimality, for example in the barrier function Φ_μ.

It has been proven for this type of interior point method that if the matrix $\mathbf{N} = \mathbf{H}(\boldsymbol{x}, \boldsymbol{y}) + \mathbf{D} + \boldsymbol{\nabla} h(\boldsymbol{x})^{\mathrm{T}} \mathbf{E}^{-1} \boldsymbol{\nabla} h(\boldsymbol{x})$, known as the dual normal matrix in this paradigm, where $\mathbf{D} = \mathbf{G}^{-1} \mathbf{Z} + \mathbf{T}^{-1} \mathbf{S}$ and $\mathbf{E}^{-1} = \mathbf{V} \mathbf{W}^{-1} + \mathbf{Q} \mathbf{P}^{-1}$, is positive definite then the search direction $\boldsymbol{\Delta}_k^1$, computed from system (4), is a descent direction for Φ_μ at a feasible $\boldsymbol{u}_k^1$ [13]. Our algorithm uses a positive definite approximation to the matrix $\mathbf{H}$ to be able to guarantee descent directions. So, it is likely that Φ_μ is reduced although the step size selection is based on the violation of the optimality measure θ_{op}.

To see how the algorithm behaves on some problems where saddle points and/or maximizers are present, we selected the following problems. We set *recompute*=off and the other two parameters mentioned in the previous subsection maintain their default values.

Example 1: A concave quadratic in two variables

$$\min a(-x_1^2 - x_2^2 + bx_1) + 2 \ \ \text{s.t.} \ \ -1 \le x_i \le 1, i = 1, 2$$

where first we consider $a = 2$ and $b = 0.5$. The problem has a maximizer at $(0.25, 0)$ inside the box constraints. With the initial $(1, 1)$ the algorithm converges to $\boldsymbol{x}^* \approx (1, 1)^{\mathrm{T}} (f^* \approx -1)$ with $N_{it} = 6$ and $N_{op} = 7$. From the initial $(0, 2)$ the algorithm gives the minimizer $(-1, 1)$ $(f^* \approx -3)$ with $N_{it} = 10$ and $N_{op} = 11$.

Then, if $a = 1$ and $b = 4$ are considered, the quadratic has a maximizer $(2, 0)$ outside the box. With both initial points, the algorithm converges to a minimizer $\boldsymbol{x}^* \approx (-1, 1)^{\mathrm{T}} (f^* \approx -4)$ (with $N_{it} = 9$ / $N_{op} = 10$ and $N_{it} = 8$ / $N_{op} = 9$ respectively).

Example 2: An indefinite quadratic in two variables

$$\min 3x_1^2 - 4x_1 x_2 - 4x_2^2 \ \ \text{s.t.} \ \ -5 \le x_i \le 5, i = 1, 2$$

with a saddle point at $(0, 0)$. From the initial point $(1, 1)$, the algorithm reaches the point $\boldsymbol{x}^* = (3.3333, 5)^{\mathrm{T}} (f^* = -133.3333)$ (with $N_{it} = 20$ and $N_{op} = 180$). When the variables are considered free (no bounds on $x_i, i = 1, 2$) the algorithm diverges.

Example 3: A function with a maximizer at $(-2, 0)$ and a saddle point at $(0, 0)$ in two variables

$$\min 3x_1^2 - x_2^2 + x_1^3 \ \ \text{s.t.} \ \ -5 \le x_i \le 5, i = 1, 2.$$

Starting from $(1, 1)$ the algorithm converges to the solution $\boldsymbol{x}^* \approx (0, 5)^{\mathrm{T}} (f^* \approx -25)$ (with $N_{it} = 9$ and $N_{op} = 10$); starting from $(-2, -2)$ the algorithm converges to $\boldsymbol{x}^* \approx (0, -5)^{\mathrm{T}} (f^* \approx -25)$ (with $N_{it} = 12$ and $N_{op} = 13$). If we consider the variables free, the algorithm diverges.

Example 4: A function with two minimizers at $(-1,-1)$ and $(0,0)$ and a saddle point at $(-0.5,-0.5)$

$$\min 2x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + x_1^4 \quad \text{s.t.} \quad -5 \le x_i \le 5, i = 1, 2.$$

With the initial approximations $(1,1)$ and $(-2,-2)$, the algorithm converges to the minimizers $(0,0)$ and $(-1,-1)$ respectively with $N_{it} = 8$ and $N_{op} = 9$. If we do not use bounds on the variables, the algorithm behaves as before using $N_{it} = 10$ and $N_{op} = 11$ in the first case and $N_{it} = 9$ and $N_{op} = 10$ in the second one.

Example 5: A concave function in one variable [13] with a global maximum at 0.5

$$\min 4x(1-x) \quad \text{s.t.} \quad 0 \le x \le 1.$$

In this problem, the matrix $\mathbf{N}$ is indefinite and for any initial approximation inside the set $0.4 \le x \le 0.6$, an interior point method might converge to the maximizer 0.5. However, our quasi-Newton interior point implementation guarantees positive definite $\mathbf{N}$ matrices and, for both $x_0 = 0.6$ and $x_0 = 0.8$, converges to $x^* \approx 1, (f^* \approx 0)$ with $N_{it} = 11$ and $N_{op} = 17$.

Thus, in practice we may say that our algorithm has been able to escape from maximizers and saddle points and converges to stationary points that are minimizers.

4.4. *Results of some typical problems*

To further understand the behavior of this interior point filter method we consider four problems that might cause some numerical difficulties to some algorithms.

Example 6: This is a nonconvex problem where the system (3) may not have a real solution for some μ [14]

$$\min x - x^2 \quad \text{s.t.} \quad x \ge 0.$$

The problem has a local minimum at 0 and a global minimum at ∞. Our algorithm converges to $x^* \approx 0$ $(f^* \approx 0)$ with $N_{it} = 6$ and $N_{op} = 7$ for both $x_0 = 3$ and $x_0 = -2$. When we set *recompute*= off the algorithm converges to 0 if $x_0 = -2$ ($N_{it} = 7$ and $N_{op} = 8$) and diverges if $x_0 = 3$.

To see how the algorithm behaves with respect to the Maratos effect we solved the problem proposed in [15]:

Example 7:

$$\min 2(x_1^2 + x_2^2 - 1) - x_1 \quad \text{s.t.} \quad x_1^2 + x_2^2 = 1.$$

Considering an initial approximation $(\cos(\alpha), \sin(\alpha))$, which is feasible for any α, and in particular for $\alpha = \pi/2$, we managed to reach a good approximation to the solution $(1, 0.1206 \times 10^{-3}), f^* = -0.9998$ with $N_{it} = 7$ and $N_{op} = 8$. Exactly the same results are obtained with the monotone version of the algorithm $(M = 0)$. We also tested the merit function approach and obtained the same solution with $N_{it} = 7$ and $N_{mf} = 32$. Note that in the merit function implementation we set $\overline{\alpha}_k = \overline{\alpha}_k^{\max}$ whenever the trial step size $\overline{\alpha}_{k,l}$ in the backtracking line search procedure verifies $\overline{\alpha}_{k,l} \left\| \mathbf{\Delta}_k^1 \right\|_2 \le 10^{-8}$.

We also consider the Wächter-Biegler problem [16]. The problem-input format for the implementation of our algorithm is as follows:

Example 8:

$$\min x \quad \text{s.t.} \quad x^2 + a \ge 0, \quad x - b \ge 0$$

where $a, b \in \mathbb{R}$ and $b \ge 0$. For $a \ne -b^2$ the problem is well-posed. According to the conditions specified in [16], we consider $a = b = 1$ and the initial approximation $x_0 = -3$. Our interior point filter algorithm was able to successfully converge to $x^* = 1 \, (f^* = 1)$ with $N_{it} = 6$ and $N_{op} = 7$. If $a = -1, b = 1$ and

$x_0 = -2$ [17] the algorithm still converges to the same solution ($N_{it} = 22$ and $N_{op} = 91$). The success of the algorithm is certainly due to the implementation of a shift strategy on the slack variables that maintains these variables far away from zero. We refer to [17] for details.

Finally we tried to solve the problem HS13 from [12]. The constraint qualification is not satisfied at the solution $(1, 0)$. This means that the vector $\boldsymbol{\sigma}$ is not zero at the solution. In this problem the vector $\boldsymbol{\beta}$ is zero. With the initial approximation $(-2, -2)$ our algorithm stopped at the point $(0.9997, -0.1529 \times 10^{-20})$ because the maximum number of iterations was reached. The relative measure of dual infeasibility in our stopping criterion is not satisfied. This is in fact an interesting topic for future research.

5. Conclusions

We present an interior point nonmonotone line search filter method for solving a nonlinear constrained optimization problem. The novelty here is that each entry in the filter has three components measuring feasibility, centrality and optimality. The use of the nonmonotone strategy aims to relax the conditions that are normally used to consider a trial point to be acceptable.

The algorithm was tested with a set of small problems and compared with a similar interior point method with a merit function approach. These preliminary numerical results seem encouraging in such a way that we intend to go on testing the algorithm with larger problems and perform a global convergence analysis of the algorithm. The extension of the nonmonotone strategy to the filter approach and the viability of a triple entry filter need adequate theoretical understanding.

References

[1] Fletcher, R. and Leyffer, S., 2002, Nonlinear programming without a penalty function. *Mathematical Programming,* **91**, 239–269.
[2] Ulbrich, M., Ulbrich, S. and Vicente, L.N., 2004, A globally convergent primal-dual interior-point filter method for nonlinear programming. *Mathematical Programming,* **100**, 379–410.
[3] Benson, H.Y., Vanderbei, R.J. and Shanno, D.F., 2002, Interior-point methods for nonconvex nonlinear programming: filter methods and merit functions. *Computational Optimization and Applications,* **23**, 257–272.
[4] Wächter, A. and Biegler, L.T., 2005, Line search filter methods for nonlinear programming: motivation and global convergence. *SIAM Journal on Optimization,* **16**, 1–31.
[5] Wächter, A. and Biegler, L.T., 2005, Line search filter methods for nonlinear programming: local convergence. *SIAM Journal on Optimization,* **16**, 32–48.
[6] Wächter, A. and Biegler, L.T., 2006, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming,* **106**, 25–57.
[7] Grippo, L., Lampariello, F. and Lucidi S., 1986, A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis,* **23**, 707–716.
[8] Grippo, L., Lampariello, F. and Lucidi S., 1991, A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Mathematik,* **59**, 779–805.
[9] Bonettini, L., 2005, A nonmonotone inexact Newton method. *Optimization Methods and Software,* **20**, 475–491.
[10] Vanderbei, R.J., 1998, LOQO: An interior-code for quadratic programming. Technical report SOR-94-15, Princeton University, Statistics and Operations Research.
[11] Fletcher, R., Gould, N.I.M., Leyffer, S., Toint, Ph.L. and Wächter, A., 2002, Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming. *SIAM Journal on Optimization,* **13**, 635–659.
[12] Hock, W. and Schittkowski, K., 1981, *Test Examples for Nonlinear Programming* (Springer-Verlag).
[13] Vanderbei, R.J. and Shanno, D.F., 1999, An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications,* **13**, 231–252.
[14] Shanno, D.F. and Vanderbei, R.J., 2000, Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. *Mathematical Programming B,* **87**, 303–316.
[15] Nocedal, J. and Wright, S.J., 1999, *Numerical Optimization* (Springer-Verlag).
[16] Wächter, A. and Biegler, L.T., 2000, Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming,* **88**, 565–574.
[17] Benson, H.Y., Shanno, D.F and Vanderbei, R.J., 2000, Interior-point methods for nonconvex nonlinear programming: jamming and comparative numerical testing. Technical report ORFE-00-02, Princeton University, Operations Research and Financial Engineering.