Institute of Computer Science
Academy of Sciences of the Czech Republic

Interior point methods for large-scale nonlinear programming

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček

Technical report No. 917

September 23, 2004

Pod Vodárenskou věží 2, 182 07 Prague 8 phone: +420 2 688 42 44, fax: +420 2 858 57 89,
e-mail:e-mail:ics@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Interior point methods for large-scale nonlinear programming

Ladislav Lukšan, Ctirad Matonoha, Jan Vlček [1]

Technical report No. 917

September 23, 2004

Abstract:

In the contribution, we describe an algorithm for solving nonlinear nonconvex programming problems, which is based on the interior point approach. The main theoretical results concern direction determination and step-length selection. We split inequality constraints into active and inactive parts to overcome problems with instability. Inactive constraints are eliminated directly, while active constraints are used for defining a symmetric indefinite linear system. Inexact solution of this system is obtained iteratively using indefinitely preconditioned conjugate gradient method. Theorems confirming efficiency of the indefinite preconditioner are introduced. Furthermore, a new merit function is defined and a filter principle is used for step-length selection. The algorithm was implemented in the interactive system for universal functional optimization UFO. Results of numerical experiments are reported.

Keywords:
Nonlinear programming, Interior point methods, KKT systems, Indefinite preconditioners, Filter methods, Algorithms.

[1]This work was supported by the Grant Agency of the Czech Academy of Sciences, project code IAA1030405, and by the Ministry of Education of the Czech Republic, project code MSM 242200002.

Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8 and Technical University of Liberec, Hálkova 6, 461 17 Liberec

1 Introduction

Consider the general nonlinear programming problem

$$x = \arg \min_{x \in \mathcal{F}_{NP}} f(x), \tag{NP}$$

where

$$\mathcal{F}_{NP} = \{x \in R^n : c_I(x) \le 0,\ c_E(x) = 0\}.$$

Here $c_I(x) = [c_i(x) : i \in I]^T$, $c_E(x) = [c_i(x) : i \in E]^T$, where $I = \{1, \dots, m_I\}$, $E = \{m_I + 1, \dots, m_I + m_E = m\}$. We assume that the functions $f(x) : R^n \to R$, $c_I(x) : R^n \to R^{m_I}$, $c_E(x) : R^n \to R^{m_E}$ are twice continuously differentiable.

The necessary KKT (Karush-Kuhn-Tucker) conditions for the solution of problem (NP) (if the standard constraint qualifications hold) have the following form

$$\begin{aligned} g(x,u) &= 0, \\ c_I(x) &\le 0, \quad u_I \ge 0, \quad u_I^T c_I(x) = 0, \\ c_E(x) &= 0, \end{aligned}$$

where

$$g(x,u) = \nabla f(x) + A_I(x) u_I + A_E(x) u_E,$$

and $A_I(x) = [\nabla c_i(x) : i \in I]$, $A_E(x) = [\nabla c_i(x) : i \in E]$. Here $u_I = [u_i(x) : i \in I]^T$, $u_E = [u_i(x) : i \in E]^T$ are vectors of Lagrange multipliers.

We use the idea of interior point methods, which is based on the introduction of a slack vector $s_I = [s_i(x) : i \in I]^T$ and the transformation of the original problem to the sequence of problems with the logarithmic barrier function

$$x = \arg \min_{(x, s_I) \in \mathcal{F}_{IP}} \left(f(x) - \mu e^T \ln(S_I) e \right), \tag{IP}$$

where $\mu > 0$ is a barrier parameter and

$$\mathcal{F}_{IP} = \{(x, s_I) \in R^n \times R^{m_I} : c_I(x) + s_I = 0,\ c_E(x) = 0\}.$$

Here e is the vector with unit elements and $S_I = \mathrm{diag}(s_i : i \in I)$. The logarithmic barrier term is used to ensure the inequality $s_I \ge 0$ implicitly.

The choice of a barrier parameter μ plays an essential role in interior point methods. If $\mu = 0$, then the KKT conditions for (IP) coincide with the KKT conditions for (NP). Therefore $\mu \to 0$ is assumed. At the same time, the efficiency of interior point methods strongly depends on the rate at which μ tends to zero. This leads to special strategies, which are shortly discussed in Section 4.

The necessary KKT conditions for the solution of problem (IP) (if the standard constraint qualifications hold) are usually written in the primal or primal-dual formulations.

Primal formulation:	Primal-dual formulation:
$g(x,u) = 0,$	$g(x,u) = 0,$
$U_I e - \mu S_I^{-1} e = 0,$	$S_I U_I e - \mu e = 0,$
$c_I(x) + s_I = 0,$	$c_I(x) + s_I = 0,$
$c_E(x) = 0,$	$c_E(x) = 0,$

where $g(x, u)$ and S_I have the same meaning as above and $U_I = \mathrm{diag}(u_i : i \in I)$. Inequalities $s_i > 0$ and $u_i > 0$ are required in all iterations. Condition $s_I > 0$ is necessary for the definition of the logarithmic barrier function and condition $u_I > 0$ improves the properties of the linear system solved and is necessary for the construction of an efficient preconditioner. The primal-dual formulation leads to more effective algorithms. One reason for this claim is the fact that primal-dual equations are better scaled: the right hand side μe approaches zero as $\mu \to 0$, but $\mu S_I^{-1} e$ can have elements bounded from zero, since $s_i \to 0$ if the i-th inequality constraint is active at the solution point.

Linearizing the primal-dual equations, we get one step of the Newton method

$$\begin{bmatrix} G & 0 & A_I & A_E \\ 0 & U_I & S_I & 0 \\ A_I^T & I & 0 & 0 \\ A_E^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \\ \Delta u_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} g \\ S_I U_I e - \mu e \\ c_I + s_I \\ c_E \end{bmatrix}, \tag{1}$$

where $g = g(x, u)$ and $G = G(x, u) = \nabla^2 f(x) + \sum_{i \in I} u_i \nabla^2 c_i(x) + \sum_{i \in E} u_i \nabla^2 c_i(x)$. In practice, the Hessian matrix $G(x, u)$ is not usually given analytically, but automatic or numerical differentiation is used instead. If $G(x, u)$ is sparse, the numerical differentiation based on the method given in [9] is very effective. We assume that the matrix of system (1) is nonsingular (this is practically always the case if the standard constraint qualifications hold).

The algorithm for an interior point method can be roughly described in the following form. For given vectors $x \in R^n$, $s_I \in R^{m_I}$, $u_I \in R^{m_I}$, $u_E \in R^{m_E}$ such that $s_I > 0$, $u_I > 0$ and a given barrier parameter $\mu > 0$, we determine direction vectors Δx, Δs_I, Δu_I, Δu_E by solving a linear system equivalent to (1). Furthermore, we choose step-length $0 < \alpha \le \overline{\alpha}$, and set $x := x + \alpha \Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha \Delta u_E$, where $s_I(\alpha, \Delta s_I) > 0$ and $u_I(\alpha, \Delta u_I) > 0$ are functions of α depending on Δs_I and Δu_I, which are chosen by a suitable strategy. Finally, we determine a new barrier parameter $\mu > 0$.

This contribution contains a survey of results proved in [19], new results concerning the use of filters described in [4] and a new algorithm based on these results. This algorithm was tested and compared with the algorithm proposed in [19] by using three collections of large scale nonlinear programming problems.

2 Direction determination

System (1) is nonsymmetric with the dimension $n + m_E + 2m_I$. This system can be symmetrized and reduced by the elimination of the vector Δs_I. One has

$$\Delta s_I = -U_I^{-1} S_I (u_I + \Delta u_I) + \mu U_I^{-1} e \tag{2}$$

so that

$$\begin{bmatrix} G & A_I & A_E \\ A_I^T & -U_I^{-1} S_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} g \\ c_I + \mu U_I^{-1} e \\ c_E \end{bmatrix}. \tag{3}$$

This system has one disadvantage: elements of matrix $U_I^{-1} S_I$ can be unbounded, since $u_i \to 0$ if the i-th inequality constraint is inactive at the solution point. To avoid this situation, we split the set of inequality constraints to the active subset, where $\hat{s}_I \le \varepsilon_I \hat{u}_I$, and the inactive subset, where $\check{s}_I > \varepsilon_I \check{u}_I$ (here $\varepsilon_I > 0$ is a suitable parameter). In the same way, we split and denote other quantities corresponding to inequality constraints. By elimination of inactive equations we obtain

$$\Delta \check{u}_I = \check{S}_I^{-1} \check{U}_I (\check{c}_I + \check{A}_I^T \Delta x) + \mu \check{S}_I^{-1} e \tag{4}$$

so that

$$\begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{U}_I^{-1}\hat{S}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \hat{u}_I \\ \Delta u_E \end{bmatrix} = - \begin{bmatrix} \hat{g} \\ \hat{c}_I + \mu \hat{U}_I^{-1} e \\ c_E \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} \hat{G} &= G + \check{A}_I \check{S}_I^{-1} \check{U}_I \check{A}_I^T, \\ \hat{g} &= g + \check{A}_I \check{S}_I^{-1} \check{U}_I \check{c}_I + \mu \check{A}_I \check{S}_I^{-1} e. \end{aligned}$$

Both matrices $\hat{G}$ and $\hat{U}_I^{-1}\hat{S}_I$ are bounded (if G and A are bounded) and if the strict complementarity conditions $\lim_{\mu\to 0}(s_i + u_i) > 0$, $i \in I$, hold (recall that $s_i > 0$ and $u_i > 0$), then one has $\lim_{\mu\to 0} \hat{U}_I^{-1}\hat{S}_I = 0$. Similarly, we can split equality (2) into two equalities to obtain

$$\begin{aligned} \Delta \hat{s}_I &= -\hat{U}_I^{-1}\hat{S}_I(\hat{u}_I + \Delta \hat{u}_I) + \mu \hat{U}_I^{-1} e, && (6) \\ \Delta \check{s}_I &= -(\check{c}_I + \check{A}_I^T \Delta x + \check{s}_I) && (7) \end{aligned}$$

after re-arrangements. Vector $\Delta \hat{u}_I$ is determined by solving system (5) and vector $\Delta \check{u}_I$ is computed from (4). Matrix $\check{S}_I^{-1}\check{U}_I$ is bounded and if the strict complementarity conditions hold, then $\lim_{\mu\to 0} \check{S}_I^{-1}\check{U}_I = 0$. It can be seen the elimination of inactive constraints is not necessary if (3) is solved by a direct method or if we use a suitable preconditioner. At any rate, this elimination can decrease the number of operations in an iterative method.

To simplify the notation in the subsequent analysis, we rewrite system (5) in the form

$$K\bar{d} = \begin{bmatrix} \hat{G} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix} \begin{bmatrix} d \\ \hat{d} \end{bmatrix} = \begin{bmatrix} b \\ \hat{b} \end{bmatrix} = \bar{b}, \tag{8}$$

where $\hat{A} = [\hat{A}_I, A_E]$ and $\hat{M} = \mathrm{diag}(\hat{M}_I, 0)$. Here $\hat{M}_I = \hat{U}_I^{-1}\hat{S}_I$ is a positive definite diagonal matrix. We assume that matrix K is nonsingular, which implies that A_E has a full column rank.

System (8) is symmetric but indefinite. It can be solved either directly by using the sparse Bunch-Parlett decomposition or iteratively by using Krylov-subspace methods for symmetric indefinite systems. Motivated by [17] (see also [2], [3], [5], [11], [16], [20], [21]) we use the preconditioner

$$C = \begin{bmatrix} \hat{D} & \hat{A} \\ \hat{A}^T & -\hat{M} \end{bmatrix}, \tag{9}$$

where $\hat{D}$ is a positive definite diagonal matrix derived from the diagonal of $\hat{G}$ (C is nonsingular if A_E has a full column rank). Then

$$C^{-1} = \begin{bmatrix} \hat{P} & \hat{Q} \\ \hat{Q}^T & -(\hat{A}^T \hat{D}^{-1} \hat{A} + \hat{M})^{-1} \end{bmatrix} \tag{10}$$

and

$$KC^{-1} = \left[\begin{array}{cc} I + (\hat{G} - \hat{D})\hat{P} & (\hat{G} - \hat{D})\hat{Q} \\ 0 & I \end{array} \right], \tag{11}$$

where $\hat{P} = \hat{D}^{-1} - \hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}\hat{A}^T\hat{D}^{-1}$, $\hat{Q} = \hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}$. This preconditioner has an important property. If it is used, the preconditioned conjugate gradient method can be efficiently applied to (8), even if matrix K is indefinite. This fact follows from the three theorems below, which are proved in [19] (we restrict to the situation when the matrix $\hat{G} - \hat{D}$ is nonsingular, which is a common situation and also the worst case in some sense, see [16]). Notice that the preconditioned conjugate gradient method can be written in the following algorithmic form.

Algorithm PCG

d – given, $\quad \hat{d} := 0,$
$r := b - \hat{G}d - \hat{A}\hat{d}, \quad \hat{r} := \hat{b} - \hat{A}^T d + \hat{M}\hat{d},$
$\beta := 0,$

while $\|r\| > \omega\|b\|$ or $\|\hat{r}\| > \omega \min(\|\hat{b}\|, \|\hat{c}\|)$ **do**

$\hat{t} := (\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}(\hat{A}^T\hat{D}^{-1}r - \hat{r}),$
$t := \hat{D}^{-1}(r - \hat{A}\hat{t}),$
$\gamma := r^T t + \hat{r}^T\hat{t}, \quad \beta := \beta\gamma,$
$p := t + \beta p, \quad \hat{p} := \hat{t} + \beta\hat{p},$
$q := \hat{G}p + \hat{A}\hat{p}, \quad \hat{q} := \hat{A}^T p - \hat{M}\hat{p},$
$\alpha := p^T q + \hat{p}^T\hat{q}, \quad \alpha := \gamma/\alpha,$
$d := d + \alpha p, \quad \hat{d} := \hat{d} + \alpha\hat{p},$
$r := r - \alpha q, \quad \hat{r} := \hat{r} - \alpha\hat{q},$
$\beta := 1/\gamma$

end while.

In this algorithm, the parameter ω represents precision of the inner iteration. According to the theory proposed in [10], this parameter should satisfy the inequality $0 \le \omega \le \overline{\omega} < 1$, which is necessary for the global convergence, and also $\omega \to 0$ as $\|\bar{b}\| \to 0$ should hold for assuring the superlinear rate of convergence. Algorithm PCG terminates if $\|r\| \le \omega\|b\|$, $\|\hat{r}\| \le \omega\|\hat{b}\|$, $\|\hat{r}\| \le \omega\|\hat{c}\|$ hold simultaneously, where

$$\hat{c} = \begin{bmatrix} \hat{c}_I + \hat{s}_I \\ \hat{c}_E \end{bmatrix}.$$

Inequality $\|\hat{r}\| \le \omega\|\hat{c}\|$ plays an essential role if ε_I is large. In this case, elements of $\hat{u}_I$ can be small enough, implying a large norm of $\hat{c}_I + \mu\hat{U}_I^{-1}e$. Thus the resulting equations are badly scaled and the precision $\|\hat{r}\| \le \omega\|\hat{b}\|$ is insufficient. Moreover, inequality $\|\hat{r}\| \le \omega\|\hat{c}\|$ is used in the proof of Theorem 6.

Theorem 1. Consider preconditioner C applied to system $K\bar{d} = \bar{b}$ and assume that $\hat{G} - \hat{D}$ is nonsingular. Then matrix KC^{-1} has at least $\hat{m}_I + 2m_E$ unit eigenvalues but at most $\hat{m}_I + m_E$ linearly independent eigenvectors corresponding to these eigenvalues exist. The other eigenvalues of matrix KC^{-1} are exactly eigenvalues of matrix $Z_E^T\tilde{G}Z_E(Z_E^T\tilde{D}Z_E)^{-1}$, where $[Z_E, A_E]$ is a nonsingular square matrix, $Z_E^T A_E = 0$, $Z_E^T Z_E = I$ and where $\tilde{G} =$

$\hat{G}+\hat{A}_I\hat{M}_I^{-1}\hat{A}_I^T$ and $\tilde{D} = \hat{D}+\hat{A}_I\hat{M}_I^{-1}\hat{A}_I^T$. If $Z_E^T\tilde{G}Z_E$ is positive definite, then all eigenvalues are positive.

Theorem 2. Consider preconditioner C applied to system $K\bar{d} = \bar{b}$ and assume that $\hat{G} - \hat{D}$ is nonsingular. Then the Krylov subspace $\mathcal{K}$ defined by matrix KC^{-1} and vector $\bar{r} \in R^{n+\hat{m}}$, where $\hat{m} = \hat{m}_I + m_E$, has a dimension of at most $\min(n+1, n-m_E+2)$.

Theorem 3. Consider the conjugate gradient method with preconditioner C applied to system $K\bar{d} = \bar{b}$. Assume that the initial $\bar{d}$ is chosen in such a way that $\hat{r} = 0$ at the start of the algorithm. Let matrix $Z_E^T\tilde{G}Z_E$ be positive definite. Then:

(a) Vector d^* (the first part of vector $\bar{d}^*$ which solves equation $K\bar{d} = \bar{b}$) is found after $n - m_E$ iterations at most.

(b) The algorithm cannot break down before d^* is found.

(c) Error $\|d - d^*\|$ converges to zero at least R-linearly with quotient

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1},$$

where κ is the spectral condition number of matrix $Z_E^T\tilde{G}Z_E(Z_E^T\tilde{D}Z_E)^{-1}$.

(d) If $d = d^*$, then also $\hat{d}_I = \hat{d}_I^*$ and d_E^* can be determined by the formula

$$d_E^* = d_E + (A_E^T\tilde{D}^{-1}A_E)^{-1}A_E^T\tilde{D}^{-1}r.$$

Theorem 3 assumes that the initial $\bar{d}$ is chosen in such a way that $\hat{r} = 0$ at the start of the algorithm. Equation (8) implies that this condition is satisfied if we set $\hat{d} = 0$ and

$$d = \hat{D}^{-1}\hat{A}(\hat{A}^T\hat{D}^{-1}\hat{A})^{-1}\hat{b}.$$

Matrix $(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1}$ used in Algorithm PCG is not computed, but the sparse Gill-Murray [15] decomposition (complete or incomplete) is used instead. Unfortunately, the matrix $\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M}$ can be dense when $\hat{A}$ has dense rows. To avoid this situation, we assume (without loss of generality) that $\hat{A}^T = [\hat{A}_s^T, \hat{A}_d^T]$ and $\hat{D} = \mathrm{diag}(\hat{D}_s, \hat{D}_d)$, where $\hat{M}_s = \hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M}$ is sparse and $\hat{A}_d$ consists of dense rows. Then

$$(\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M})^{-1} = (\hat{M}_s + \hat{A}_d^T\hat{D}_d^{-1}\hat{A}_d)^{-1} = \hat{M}_s^{-1} - \hat{M}_s^{-1}\hat{A}_d^T\hat{M}_d^{-1}\hat{A}_d\hat{M}_s^{-1},$$

where

$$\hat{M}_d = \hat{D}_d + \hat{A}_d\hat{M}_s^{-1}\hat{A}_d^T$$

is a (low-dimensional) dense matrix. Again the sparse Gill-Murray decomposition of matrix $\hat{M}_s$ is used instead of its inversion. Notice that this approach is not quite reliable, since matrix $\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M}$ can be singular, even if $\hat{A}^T\hat{D}^{-1}\hat{A} + \hat{M}$ is nonsingular. In this case, the Gill-Murray decomposition can change the matrix $\hat{A}_s^T\hat{D}_s^{-1}\hat{A}_s + \hat{M}$ considerably, which can deteriorate the properties of the method. However, we can use the Bunch-Parlett decomposition of matrix C alternatively if this situation arises.

3 Step-length selection

Let $\overline{\alpha} = \min(1, \overline{\Delta}/\|\Delta x\|)$ ($\overline{\Delta}$ serves as a safeguard and has a similar significance as a trust-region radius). Step-length $0 < \alpha < \overline{\alpha}$ and new vectors $x := x + \alpha\Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha\Delta u_E$ can be determined in many ways, but it is necessary to satisfy conditions $s_I > 0$ (which is necessary for definition of the logarithmic barrier function) and $u_I > 0$ (which guarantees positive definiteness of matrix $M_I = U_I^{-1}S_I$). We have used two simple strategies for computation of $s_I(\alpha, \Delta s_I)$ and $u_I(\alpha, \Delta u_I)$. Strategy 1 handles individual components separately setting $s_i(\alpha, \Delta s_I) = s_i + \alpha_{s_i}\Delta s_i$ and $u_i(\alpha, \Delta u_I) = u_i + \alpha_{u_i}\Delta u_i$, $i \in I$, where

$$
\begin{aligned}
\alpha_{s_i} &= \alpha, && \Delta s_i \ge 0,\\
\alpha_{s_i} &= \min\left(\alpha, -\gamma\frac{s_i}{\Delta s_i}\right), && \Delta s_i < 0,\\
\alpha_{u_i} &= \alpha, && \Delta u_i \ge 0,\\
\alpha_{u_i} &= \min\left(\alpha, -\gamma\frac{u_i}{\Delta u_i}\right), && \Delta u_i < 0,
\end{aligned}
$$

and $0 < \gamma < 1$ is a coefficient close to unity. Strategy 2 uses bounds

$$
\begin{aligned}
\overline{\alpha}_s &= \gamma \min_{i\in I, \Delta s_i < 0}\left(-\frac{s_i}{\Delta s_i}\right),\\
\overline{\alpha}_u &= \gamma \min_{i\in I, \Delta u_i < 0}\left(-\frac{u_i}{\Delta u_i}\right),
\end{aligned}
$$

where $0 < \gamma < 1$ is a coefficient close to unity, and defines $s_I(\alpha, \Delta s_I) = s_I + \min(\alpha, \overline{\alpha}_s)\Delta s_I$, $u_I(\alpha, \Delta u_I) = u_I + \min(\alpha, \overline{\alpha}_u)\Delta u_I$.

3.1 Line-search with a merit function

A further requirement for the selection of a step-length is satisfying a suitable goal criterion. This criterion is usually a merit function, which is a combination of the barrier function and a measure of constraint violation. Motivated by [17], we use the following function

$$
\begin{aligned}
P_{\mu,\sigma}(\alpha) &= f(x + \alpha\Delta x) - \mu e^T \ln(S_I(\alpha, \Delta s_I))e\\
&+ (u_I + \Delta u_I)^T(c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I))\\
&+ (u_E + \Delta u_E)^T c_E(x + \alpha\Delta x)\\
&+ \frac{\sigma}{2}\|c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I))\|^2\\
&+ \frac{\sigma}{2}\|c_E(x + \alpha\Delta x)\|^2, \qquad (12)
\end{aligned}
$$

where $\mu > 0$ and $\sigma \ge 0$. The following theorem is proved in [19].

Theorem 4. Let $s_I > 0$, $u_I > 0$ and let the triple Δx, $\Delta\hat{u}_I$, Δu_E be an inexact solution of system (5) so that

$$
\begin{bmatrix} \hat{G} & \hat{A}_I & A_E \\ \hat{A}_I^T & -\hat{U}_I^{-1}\hat{S}_I & 0 \\ A_E^T & 0 & 0 \end{bmatrix}
\begin{bmatrix} \Delta x \\ \Delta\hat{u}_I \\ \Delta u_E \end{bmatrix}
+ \begin{bmatrix} \hat{g} \\ \hat{c}_I + \mu\hat{U}_I^{-1}e \\ c_E \end{bmatrix}
= \begin{bmatrix} r \\ \hat{r}_I \\ r_E \end{bmatrix}, \qquad (13)
$$

where r, $\hat{r}_I$, r_E are parts of the residual vector, and let $\Delta\check{u}_I$ and Δs_I be given by (4) and (2), respectively. Then

$$\begin{aligned} P'_{\mu,\sigma}(0) &= -(\Delta x)^T G\Delta x - (\Delta s_I)^T S_I^{-1} U_I \Delta s_I - \sigma(\|c_I + s_I\|^2 + \|c_E\|^2) \\ &+ (\Delta x)^T r + \sigma((\hat{c}_I + \hat{s}_I)^T \hat{r}_I + c_E^T r_E). \end{aligned}$$

If

$$\sigma > -\frac{(\Delta x^T) G\Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I}{\|c_I + s_I\|^2 + \|c_E\|^2} \tag{14}$$

and if (5) is solved with a sufficient precision, namely if

$$\begin{aligned} (\Delta x)^T r + \sigma((\hat{c}_I + \hat{s}_I)^T \hat{r}_I + c_E^T r_E) &< (\Delta x)^T G\Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I \\ &+ \sigma(\|c_I + s_I\|^2 + \|c_E\|^2), \end{aligned} \tag{15}$$

then $P'_{\mu,\sigma}(0) < 0$.

Condition (14) restricts the choice of the parameter σ weakly. If matrix G is positive semidefinite, any value $\sigma \geq 0$ satisfies this condition. In the opposite case, the second term, which is always positive, decreases the value of $P'_{\mu,\sigma}(0)$ and partially eliminates the influence of the first term.

Inequality (14) gives one possibility for the computation of the parameter σ, which implies that $P'_{\mu,\sigma}(0) < 0$ if (15) holds. But it is usually more efficient in practical computations to choose parameter σ as a constant and replace matrix G by a positive definite diagonal matrix D if the condition $P'_{\mu,\sigma}(0) < 0$ does not hold. If all constraints are active, then $\hat{G} = G = D = \hat{D}$ and the following theorem, proved in [19], can be used.

Theorem 5. Consider Algorithm PCG with preconditioner C applied to system $K\bar{d} = \bar{b}$ with $\hat{G}$ replaced by $\hat{D}$. Then this algorithm finds the exact solution of $K\bar{d} = \bar{b}$ in its first iteration and $P'_{\mu,\sigma}(0) < 0$ for any value $\sigma \geq 0$.

If $P'_{\mu,\sigma}(0) < 0$, we can use a line-search technique. In this case, we set $\alpha = \beta^l \overline{\alpha}$, where $0 < \beta < 1$ is a line-search parameter and $l \geq 0$ is a minimum nonnegative integer such that $P_{\mu,\sigma}(\beta^l \overline{\alpha}) < P_{\mu,\sigma}(0)$. After determination of $\alpha \leq \overline{\alpha}$, we set $x := x + \alpha\Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta u_I)$, $u_E := u_E + \alpha\Delta u_E$.

The above line-search technique is not always advantageous, since step-length $\alpha = \beta^l \overline{\alpha}$ can be too short. Therefore, we have tested an additional possibility. We have used the simple choice $\alpha = \overline{\alpha}$ (the first step accepted). In this case, the merit function (12) serves only as indication of restarts. Surprisingly, this simple choice is very efficient in practice as is demonstrated in Table 1 – Table 3.

3.2 Line-search with a filter structure

When solving problem (IP), we need to decrease values of functions

$$B_\mu(x, s_I) = f(x) - \mu e^T \ln(S_I) e$$

and

$$P(x, s_I) = \frac{1}{2}\left(\|c_I(x) + s_I\|^2 + \|c_E(x)\|^2\right).$$

The use of a suitable merit function guarantees that at least one of these values is decreased. Unfortunately, this simple criterion can lead to cycling. Therefore, Fletcher and Leyffer [13] proposed a new idea based on a filter principle. Suppose that $B_\mu(x^i, s_I^i) \le B_\mu(x^j, s_I^j)$ and $P(x^i, s_I^i) \le P(x^j, s_I^j)$ for $i \in N$ and $j \in N$. Then we say that pair (x^i, s_I^i) dominates pair (x^j, s_I^j). In the k-th iteration, the filter is a set of pairs (x^i, s_I^i), $i \in \mathcal{F}_k \subset \{1, 2, \dots, k\}$, such that no pair dominates any other. Letting $x^{k+1} = x^k + \alpha^k \Delta x^k$ and $s_I^{k+1} = s_I(\alpha^k, \Delta s_I^k)$ we seek a step-length $\alpha^k > 0$ in such a way that no pair in the filter dominates pair (x^{k+1}, s_I^{k+1}). If this is true, then pair (x^{k+1}, s_I^{k+1}) is acceptable and is added to the filter.

Our strategy is based on the following theorem.

Theorem 6. Let the assumptions of Theorem 4 be satisfied and let $B_\mu(\alpha) = B_\mu(x + \alpha\Delta x, s_I(\alpha, \Delta s_I))$ and $P(\alpha) = P(x + \alpha\Delta x, s_I(\alpha, \Delta s_I))$. Then

$$P'(0) \le -2(1-\omega)P(0)$$

and if $P(0) = 0$, then

$$B'_\mu(0) = -(\Delta x)^T G \Delta x - (\Delta s_I)^T S_I^{-1} U_I \Delta s_I + (\Delta x)^T r$$

If $(\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I > 0$ and (5) is solved with a sufficient precision, namely if

$$(\Delta x)^T r < (\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I,$$

then $B'_\mu(0) < 0$.

Proof. Since $s_I > 0$ and $u_I > 0$, one has $s_I(\alpha, \Delta s_I) = s_I + \alpha\Delta s_I$ and $u_I(\alpha, \Delta u_I) = u_I + \alpha \Delta u_I$ for sufficiently small values of α. Thus differentiating $P(\alpha)$, we obtain

$$P'(0) = (c_I + s_I)^T (A_I^T \Delta x + \Delta s_I) + c_E^T A_E^T \Delta x. \tag{16}$$

Using the equality

$$\begin{bmatrix} G & 0 & A_I & A_E \\ 0 & S_I^{-1}U_I & I & 0 \\ A_I^T & I & 0 & 0 \\ A_E^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \\ \Delta u_I \\ \Delta u_E \end{bmatrix} + \begin{bmatrix} g \\ U_I e - \mu S_I^{-1} e \\ c_I + s_I \\ c_E \end{bmatrix} = \begin{bmatrix} r \\ 0 \\ r_I \\ r_E \end{bmatrix}, \tag{17}$$

which is equivalent to (4)–(7), we obtain

$$(c_I + s_I)^T (A_I^T \Delta x + \Delta s_I) = -\|c_I + s_I\|^2 + (\hat{c}_I + \hat{s}_I)^T \hat{r}_I,$$

$$c_E^T A_E^T \Delta x = -\|c_E\|^2 + c_E^T r_E$$

(since $\check{r}_I = 0$ by (4)). Since Algorithm PCG and the Schwarz inequality guarantee that $(\hat{c}_I + \hat{s}_I)^T \hat{r}_I + c_E^T r_E = \hat{c}^T \hat{r} \le \omega \|\hat{c}\|^2 \le \omega(\|c_I + s_I\|^2 + \|c_E\|^2)$, equality (16) implies that

$$P'(0) \le -(1-\omega)(\|c_I + s_I\|^2 + \|c_E\|^2) = -2(1-\omega)P(0).$$

Similarly, differentiating $B_\mu(\alpha)$, we obtain

$$B'_\mu(0) = (\Delta x)^T \nabla f(x) - \mu(\Delta s_I)^T S_I^{-1} e. \tag{18}$$

If $P(0) = 0$, then $\|\hat{c}\| = 0$ and, therefore, $\|\hat{r}\| \le \omega\|\hat{c}\| = 0$. Thus $\|\hat{r}_I\| = 0$, $\|r_E\| = 0$ and (17) implies that

$$\begin{bmatrix} \hat{A}_I^T & I \\ A_E^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \end{bmatrix} = 0$$

and

$$\begin{bmatrix} G & 0 \\ 0 & S_I^{-1}U_I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s_I \end{bmatrix} + \begin{bmatrix} A_I & A_E \\ I & 0 \end{bmatrix} \begin{bmatrix} u_I + \Delta u_I \\ u_E + \Delta u_E \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} - \begin{bmatrix} \nabla f(x) \\ -\mu S_I^{-1} e \end{bmatrix}$$

Multiplying the second equality by $\left[(\Delta x)^T, (\Delta s_I)^T\right]$ and using the first one, we obtain

$$(\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I = (\Delta x)^T r - (\Delta x)^T \nabla f(x) + \mu (\Delta s_I)^T S_I^{-1} e,$$

which after substituting into (18) proves the second assertion. □

Theorem 6 implies several conclusions, which can be used for the construction of the corresponding algorithm.

- Directions Δx, Δs_I obtained by Algorithm PCG imply the inequality $P'(0) \le -2(1-\omega)P(0) < 0$. Thus, choosing an arbitrary $0 < \varepsilon_1 < 1$, there is a number $0 < \overline{\alpha} \le 1$ such that $P(\alpha) \le P(0) - 2\alpha\varepsilon_1(1-\omega)P(0)$ for all $0 < \alpha \le \overline{\alpha}$ (since usually $\omega \le 1/2$, we can write $P(\alpha) \le P(0) - \alpha\varepsilon_1 P(0)$).

- If standard constraint qualifications (LICQ or MFCQ, see [14]) are satisfied, then the Lagrange multipliers are bounded and $(\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I > 0$ implies $B'_\mu(0) < 0$ also for small but nonzero $P(0)$. If $P(0)$ is small and $B'_\mu(0) \ge 0$, we can replace G by a positive definite diagonal matrix D. If all constraints are active, then $\hat{G} = G = D$ and if we use the preconditioner (9) with $\hat{D} = D$, then $(\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I > 0$ and $r = 0$ by Theorem 5 and we can expect that new Δx, Δs_I are descent directions for B_μ. Note that $(\Delta x)^T G \Delta x + (\Delta s_I)^T S_I^{-1} U_I \Delta s_I > 0$ holds in a neighborhood of the solution satisfying the second order sufficient conditions.

- The above considerations form a basis for a suitable restart strategy. We need to replace G by D if $P(0)$ is small and $B'_\mu(0) \ge 0$. These two conditions can be replaced by one condition using a suitable merit function. We have chosen function $P_{\mu,\sigma}$ described in the previous subsection, which is extremely suitable for the detection of indefiniteness (Theorem 4). Thus G is replaced by D in case $P_{\mu,\sigma}(0) \ge 0$.

- The above strategy guarantees that pair (x^{k+1}, s_I^{k+1}) is acceptable for the so-called Markov filter [4], which contains only one pair (x^k, s_I^k). For assuring the global convergence, it is necessary to use larger number of pairs. In this case, values $B_{\mu^k}(x^i, s_I^i)$, $i \le k$ have to be recomputed to $B_{\mu^{k+1}}(x^i, s_I^i)$, $i \le k$ after changing μ^k to μ^{k+1}. Therefore, we need to store values $f(x^i)$ and $e^T \ln(S_I^i) e$ for $i \in \mathcal{F}_k$.

4 Computation of the barrier parameter

Most implementations of interior point methods choose the value μ in such a way that $0 < \mu < s_I^T u_I / m_I$ (or $\mu = \lambda s_I^T u_I / m_I$, where $0 < \lambda < 1$). This case is analyzed in [12] and used in [22]. Computational experience has shown that the algorithm performs best when components $s_i u_i$ of the dot-product in the numerator approach zero at a uniform rate. The distance from uniformity can be measured by the ratio

$$\varrho = \frac{\min_{i \in I}(s_i u_i)}{s_I^T u_I / m_I}.$$

Clearly, $0 < \varrho \le 1$ and $\varrho = 1$ if and only if condition $S_I U_I e - \mu e = 0$ holds. The value λ is then computed from ϱ. Usually heuristic formulas are used for this purpose. In our implementation, we have used the formula

$$\lambda = 0.1 \min\left(0.05 \frac{1-\varrho}{\varrho}, 2\right)^3 \tag{19}$$

proposed in [22]. We have also tested other possibilities, e.g., formulas given in [1], but formula (19) has shown to be best.

Concerning the local convergence analysis of interior point methods with various choices of the barrier parameter, we refer to [7] and [12]. It is necessary to note that slow decrease of μ can lead to a considerable increase of the total number of iterations, i.e., to a long computational time, but its rapid decrease can lead to a failure of the method.

5 Description of the algorithm

The above considerations can be summarized in the algorithmic form. Algorithm based on a merit function is proposed in [19], here we propose an algorithm based on a filter structure.

Algorithm 1.

Data: Parameter for the active constraint definition ε_I (e.g. $\varepsilon_I = 0.1$). Minimum precision for the direction determination $0 < \overline{\omega} < 1$ (e.g. $\overline{\omega} = 0.9$). Line-search parameters $0 < \beta < 1$ (e.g. $\beta = 0.5$) and $\bar{l} > 0$ (e.g. $\bar{l} = 20$). Maximum step-length reduction $0 < \gamma < 1$ (e.g. $\gamma = 0.95$ when barrier function (12) is used and $\gamma = 0.99$ otherwise). Step bound $\overline{\Delta} > 0$ (e.g. $\overline{\Delta} = 1000$).

Input: Sparsity pattern of matrices $\nabla^2 F$ and A. Initial choice of vector x.

Step 1: *Initiation.* Choose the values $\mu > 0$ (e.g. $\mu = 1$) and $\sigma > 0$ (e.g. $\sigma = 1$). For $i \in I$ set $s_i := \max(-c_i(x), \delta_s)$ and $u_i := \delta_u$, where $\delta_s > 0$ (e.g. $\delta_s = 0.1$) and $\delta_u > 0$ (e.g. $\delta_u = 0.1$). For $i \in E$ set $u_i := 0$. Compute value $f(x)$ and vectors $c_I(x)$, $c_E(x)$. Include the pair (x, s_I) into the filter. Set $k := 0$.

Step 2: *Termination.* Compute matrix $A := A(x)$ and vector $g := g(x, u)$. If KKT conditions (derived by the primal-dual formulation) with μ sufficiently small are satisfied with a sufficient precision, then terminate the computation. Otherwise set $k := k + 1$.

Step 3: *Approximation of the Hessian matrix.* Compute approximation G of the Hessian matrix $G(x, u)$ by using differences of gradient $g(x, u)$ as in [9].

Step 4: *Direction determination.* Define active and inactive constraints, build linear system (8) and choose preconditioner (9). Determine positive definite diagonal matrix $\hat{D}$ as an approximation of the diagonal of $\hat{G}$ and factorize the matrix $\hat{A}^T\hat{D}^{-1}\hat{A}+\hat{M}$ by using the complete or incomplete Gill-Murray decomposition to obtain (10). Set $\omega = \min(\|\bar{b}\|, 1/k, \overline{\omega})$ and determine direction vector $\bar{d}$ (i.e. vectors Δx, $\Delta\hat{u}_I$ and Δu_E) as an inexact solution of (8) (with the precision ω) by Algorithm PCG. Compute vectors $\Delta\check{u}_I$, $\Delta\hat{s}_I$, $\Delta\check{s}_I$ by (4), (6), (7), respectively. Compute directional derivative $P'_{\mu,\sigma}(0)$ of the merit function $P_{\mu,\sigma}(\alpha)$. If $P'_{\mu,\sigma}(0) < 0$ go to Step 6.

Step 5: *Restart.* If $G = D$ (i.e., if restart was already used), then terminate the computation (the algorithm fails), else determine positive definite diagonal matrix D by the procedure given in [17], set $G = D$ and go to Step 4.

Step 6: *Step-length selection.* Define maximum step-length $\overline{\alpha}$ and functions $s_I(\alpha, \Delta s_I)$, $u_I(\alpha, \Delta u_I)$ by one of the strategies described in Section 3. Find the minimum integer $0 \le l \le \bar{l}$ such that the pair (x, s_I) is not dominated by any pair from the filter. If such l does not exist go to Step 5, else set $\alpha = \beta^l\overline{\alpha}$, $x := x + \alpha\Delta x$, $s_I := s_I(\alpha, \Delta s_I)$, $u_I := u_I(\alpha, \Delta s_I)$, $u_E := u_E + \alpha\Delta u_E$ and include the pair (x, s_I) into the filter.

Step 7: *Barrier parameter.* Determine parameter λ by (19), set $\mu = \lambda s_I^T u_I/m_I$, recompute barrier terms in the filter and go to Step 2.

Algorithm 1 can theoretically fail in Step 5 in case the filter contains more than one pair, but this situation never arose in our computational experiments.

6 Numerical experiments

The above algorithm was tested and compared with the algorithm proposed in [19] by using three sets of test problems. These sets were obtained as modifications of 18 test problems for equality constrained minimization given in [17], [18], which can be downloaded (together with [18]) from `http://www.cs.cas.cz/~luksan/test.html` (we excluded Problem 5.8 from the first two sets, since it consumed more than 50% of the total CPU time). In Set 1, equalities $c(x) = 0$ are replaced by inequalities $c(x) \ge 0$. In Set 2, equalities $c(x) = 0$ are replaced by inequalities $c(x) \le 0$ (i.e., this set contains problems LUKVLI1–LUKVLI18 from the CUTE collection [6]). Set 3 contains inequalities $-1 \le x \le 1$ and $-1 \le c(x) \le 1$. All problems used have optional dimension; we have chosen dimension with 1000 variables. We have tested six interior point methods differing by line-search strategies. The first three methods, which use various merit functions, are based on Algorithm 1 proposed in [19]. The last three methods, which use various filters introduced in [4], are based on Algorithm 1 described above.

MM: Line-search without any merit function (the first step accepted).

OM: Line-search with the merit function

$$\begin{aligned} P_\sigma(\alpha) &= f(x + \alpha\Delta x) + (u_I + \Delta u_I)^T(c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I)) \\ &+ (u_E + \Delta u_E)^T c_E(x + \alpha\Delta x) + \sigma P(\alpha) \end{aligned}$$

(barrier term excluded).

BM: Line-search with the merit function

$$\begin{aligned} P_{\mu,\sigma}(\alpha) &= B_\mu(\alpha) + (u_I + \Delta u_I)^T (c_I(x + \alpha\Delta x) + s_I(\alpha, \Delta s_I)) \\ &+ (u_E + \Delta u_E)^T c_E(x + \alpha\Delta x) + \sigma P(\alpha) \end{aligned}$$

MF: Line-search with the Markov filter containing only one pair (x, s) with the dominance defined by $B_\mu(x)$ and $P(x)$.

OF: Line-search with the objective filter containing at most 50 pairs with the dominance defined by $f(x)$ and $P(x)$.

BF: Line-search with the barrier filter containing at most 50 pairs with the dominance defined by $B_\mu(x)$ and $P(x)$.

The results of the tests are listed in three tables, where M is the method inroduced above (MM, OM, BM, MF, OF, BF), S is the strategy for the step-length restriction mentioned in Section 3, NIT is the total number of iterations, NFV is the total number of function evaluations, NFG is the total number of gradient evaluations (NFG is much greater than NIT, since the second order derivatives are computed by using gradient differences), NCG is the total number of CG iterations, NRS is the total number of restarts and NFAIL gives the number of failures for a given set (the number of problems which have not been solved). Each row of each table contains a summary of the results for all problems used.

Table 1: Set 1 of 17 problems with 1000 variables: $\varepsilon_I = 10^{-1}$

M	S	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
MM	1	567	567	4137	24969	20	4.92	-
OM	2	592	763	4218	19505	18	4.31	-
BM	2	550	593	3936	21806	14	4.70	-
MF	2	580	606	4121	12841	6	3.78	-
OF	2	572	665	4073	12874	7	3.77	-
BF	2	583	669	4199	12983	10	3.91	-

Table 2: Set 2 of 17 problems with 1000 variables: $\varepsilon_I = 10^{-1}$

M	S	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
MM	1	393	393	2823	10728	19	3.02	-
OM	2	392	939	2885	5322	31	2.58	1
BM	2	476	925	3403	5654	73	3.33	1
MF	2	419	457	3061	5722	10	2.42	-
OF	2	422	479	3074	5730	11	2.48	-
BF	2	417	469	3044	5691	10	2.41	-

Table 3: Set 3 of 18 problems with 1000 variables: $\varepsilon_I = 10^{-1}$

M	S	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
MM	1	571	571	3991	2731	13	6.83	1
OM	2	578	778	4195	3933	24	8.79	-
BM	2	572	699	4294	3670	25	7.55	-
MF	2	530	620	3882	2618	11	7.14	-
OF	2	534	640	3915	2668	12	6.66	-
BF	2	512	568	3772	2604	13	6.74	-

The last table demonstrates the influence of the parameter ε_I on the effectiveness of the method. It contains results for the third set of problems corresponding to the choice M=BF.

Table 4: Set 3 of 18 problems with 1000 variables: M=BF

ε_I	S	NIT	NFV	NFG	NCG	NRS	TIME	NFAIL
10^{-4}	2	1204	1367	8722	17435	217	13.20	-
10^{-2}	2	594	670	4302	3710	10	5.79	-
10^{-1}	2	512	568	3772	2604	13	6.74	-
1	2	532	598	3909	3021	13	8.78	-
10^{+8}	2	525	653	3907	2898	14	6.16	-

7 Conclusions

Result introduced in this contribution and similar results presented in [19] confirm our effort to develop effective iterative methods for solving linear KKT systems (more detailed conclusions are given in [19]). The above tables show that the use of a filter principle combined with restarts and with strategies assuring positivity of slack variables and Lagrange multipliers leads to robust and efficient algorithms. These methods (MF, OF, BF) are more efficient than methods based on an augmented-Lagrangian merit function (MM, OM, BM) described in [19]. Surprisingly, the simplest method MM with a single-pair filter is competitive with more sophisticated methods OF, BF that use standard multi-pair filters. The last table shows that the new termination criterion used in Algorithm PCG allows us to utilize large values of ε_I, so that the splitting on active and inactive constrains is no more important.

References

[1] M. Argaez, R.A. Tapia and L. Velasquez (1998). Numerical comparison of path-following strategies for a basic interior point method for nonlinear programming. Report No. CRPC-TR9777-S, Center for Research on Parallel Computation, Rice University, Houston.

[2] O. Axelsson and M. Neytcheva (2002). Preconditioning methods for linear systems with saddle point matrices. *Lecture Notes in Computer Science*, **2331**, 784-793.

[3] O. Axelsson and M. Neytcheva (2003). Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical Linear Algebra with Applications*, **10**, 1-26.

[4] H.Y. Benson, R.J. Vanderbei and D.F. Shanno (2002). Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions. *Computational Optimization and Applications*, **23**, 257-272.

[5] L. Bergamaschi, J. Gondzio and G. Zilli (2002). Preconditioning indefinite systems in interior point methods for optimization. Report No. MS-02-001, Department of Mathematics and Statistics, University of Edinburgh.

[6] I. Bongartz, A.R. Conn, N. Gould and P.L. Toint (1995). CUTE: constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, **21**, 123-160.

[7] R.H. Byrd, G. Liu and J. Nocedal (1998). On the local behavior of an interior point method for nonlinear programming. In: D.F.Griffiths and D.J.Higham (Eds.), *Numerical Analysis 1997*, pp.37-56. Addison Wesley Longman.

[8] R.H. Byrd, J. Nocedal and R.A. Waltz (2000). Feasible interior methods using slacks for nonlinear optimization. Report No. OTC 2000/11, Optimization Technology Center.

[9] A.R. Curtis, M.J.D. Powell and J.K. Reid (1974). On the estimation of sparse Jacobian matrices. *IMA Journal of Aplied Mathematics*, **13**, 117-119.

[10] R.S. Dembo and T. Steihaug (1983). Truncated Newton Algorithms for Large-Scale Optimization. *Mathematical Programming*, **26**, 190-212.

[11] C. Durazzi and V. Ruggiero: Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems. To appear in *Numerical Linear Algebra with Applications*.

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

[13] R. Fletcher and S. Leyffer (1997). Nonlinear programming without a penalty function. Technical Report NA/171, Department of Mathematics, University of Dundee.

[14] A. Forsgren, P.E. Gill and M.H. Wright (2002). Interior methods for nonlinear optimization. *SIAM Review*, **44**, 525-597.

[15] P.E. Gill and W. Murray (1974). Newton type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, **7**, 311-350.

[16] C. Keller, N.I.M. Gould and A.J. Wathen (2000). Constraint preconditioning for indefinite linear systems. *SIAM J. on Matrix Analysis and Applications*, **21**, 1300-1317.

[17] L. Lukšan and J. Vlček (1998). Indefinitely Preconditioned Inexact Newton Method for Large Sparse Equality Constrained Nonlinear Programming Problems. *Numerical Linear Algebra with Applications*, **5**, 219-247.

[18] L. Lukšan and J. Vlček (1998). Sparse and partially separable test problems for unconstrained and equality constrained optimization. Report V-767. Prague, ICS AS CR.

[19] L. Lukšan, C. Matonoha and J. Vlček (2004). Interior-point method for nonlinear nonconvex optimization. To appear in *Numerical Linear Algebra with Applications*.

[20] I. Perugia and V. Simoncini (2000). Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numerical Linear Algebra with Applications*, **7**, 1-32.

[21] M. Rozložník and V. Simoncini (2000). Short-term recurrences for indefinite preconditioning of saddle point problems. Report No. SAM 2000-08, SAM, ETH Zurich.

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