Primal Interior-Point Method for Large Sparse Inequality Constrained Optimization

Lukšan, Ladislav
2005

Institute of Computer Science
Academy of Sciences of the Czech Republic

Primal interior-point method for large sparse inequality constrained optimization

L.Lukšan, J. Stuchlý, J. Vlček

Technical report No. 956

December 2005

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

Primal interior-point method for large sparse inequality constrained optimization

L.Lukšan, J. Stuchlý, J. Vlček [1]

Technical report No. 956

December 2005

Abstract:

In this paper, we propose an interior-point method for large sparse inequality constrained optimization. After a short introduction, the complete algorithm is introduced and some implementation details are given. We prove that this algorithm is globally convergent under standard mild assumptions. Thus nonconvex problems can be solved successfully. The results of computational experiments given in this paper confirm efficiency and robustness of the proposed method.

Keywords:
Constrained optimization, large-scale optimization, nonlinear programming, inequality constraints, interior-point methods, modified Newton methods, computational experiments.

[1]This work was supported by the Grant Agency of the Czech Academy of Sciences, project code IAA1030405, and the institutional research plan No. AV0Z10300504. L.Lukšan is also from the Technical University of Liberec, Hálkova 6, 461 17 Liberec.

1 Introduction

Consider the nonlinear programming problem:

$$\text{Minimize} \quad f(x) \quad \text{subject to} \quad c_i(x) \le 0, \quad 1 \le i \le m, \tag{1}$$

where $f : R^n \to R$ is a smooth function with a sparse Hessian matrix and $c_i : R^n \to R$, $0 \le i \le m$, are smooth functions depending on a small number of variables (n_i, say). We will use the following assumptions.

Assumption 1. Function $f(x)$ is bounded from below on R^n, i.e., there is $\underline{F} \in R$ such that $f(x) \ge \underline{F}$ for all $x \in R^n$.

Assumption 2. Functions $f(x)$ and $c_i(x)$, $1 \le i \le m$, are twice continuously differentiable on a sufficiently large convex set $\mathcal{D}$. Moreover, constants $\overline{c}$, $\overline{g}$, $\overline{G}$ exist such that $\|\nabla f(x)\| \le \overline{g}$, $\|\nabla^2 f(x)\| \le \overline{G}$ and $|c_i(x)| \le \overline{c}$, $\|\nabla c_i(x)\| \le \overline{g}$, $\|\nabla^2 c_i(x)\| \le \overline{G}$, $1 \le i \le m$, for all $x \in \mathcal{D}$.

The choice of $\mathcal{D}$ will be discussed later (see Assumption 3).

Using slack variables s_i, $1 \le i \le m$, problem (1) can be transformed into the equivalent problem:

$$\text{Minimize} \quad f(x) \quad \text{subject to} \quad c_i(x) + s_i = 0, \quad s_i \ge 0, \quad 1 \le i \le m. \tag{2}$$

The necessary first-order (KKT) conditions for the solution of (2) have the form

$$\nabla f(x) + \sum_{i=1}^{m} u_i \nabla c_i(x) = 0, \tag{3}$$

$$c_i(x) + s_i = 0, \quad s_i \ge 0, \quad u_i \ge 0, \quad s_i u_i = 0, \quad 1 \le i \le m, \tag{4}$$

where u_i, $1 \le i \le m$, are Lagrange multipliers.

In this paper, we introduce a simple primal interior-point method for (2). This problem is replaced by a sequence of unconstrained problems

$$\text{minimize} \quad B(x, s; \mu) \quad = \quad f(x) - \mu \sum_{i=1}^{m} \log s_i + \frac{1}{2\mu} \sum_{i=1}^{m} (c_i(x) + s_i)^2 \tag{5}$$

with barrier parameter $\mu > 0$, where we assume that $s_i > 0$, $1 \le i \le m$, and $\mu \to 0$ monotonically. Here $B(x, s; \mu) : R^{n+m} \to R$ is a function of $n + m$ variables $x \in R^n$, $s \in R^m$.

The interior-point method described in this paper is iterative, i.e., it generates a sequence of points $x_k \in R^n$, $k \in N$ (N is the set of integers). For proving the global convergence, we need the following assumption concerning functions $f(x)$, $c_i(x)$, $1 \le i \le m$, and sequence $\{x_k\}_1^\infty$.

Assumption 3. Assumption 2 holds and $\{x_k\}_1^\infty \in \mathcal{D}$.

The interior-point method investigated in this paper is a trust-region modification of the Newton method. Approximation of the Hessian matrix is computed by the gradient differences which can be carried out efficiently if the Hessian matrix is sparse (see [2]). Since the Hessian matrix need not be positive definite in the non-convex case, the standard line-search realization cannot be used. There are two basic possibilities, either a trust-region approach or the line-search strategy with suitable restarts, which eliminate this insufficiency. We have implemented and tested both these possibilities and our tests have shown that the first possibility, used in Algorithm 1, is more efficient.

The paper is organized as follows. In Section 2, we introduce the interior-point method for large sparse inequality constrained optimization and describe the corresponding algorithm. Section 3 contains more details concerning this algorithm such as the trust-region strategy and the barrier parameter update. In Section 4 we study theoretical properties of the interior-point method and prove that this method is globally convergent if Assumption 1, Assumption 3 and LICQ constraint qualification hold. Finally, in Section 5 we present results of computational experiments confirming the efficiency of the proposed method.

2 Description of the method

Differentiating $B(x, s; \mu)$ given by (5), we obtain necessary conditions for minimum in the form

$$\nabla f(x) + \sum_{i=1}^{m} \frac{c_i(x) + s_i}{\mu} \nabla c_i(x) \triangleq \nabla f(x) + \sum_{i=1}^{m} u_i(x, s_i; \mu) \nabla c_i(x) = 0 \tag{6}$$

and

$$-\frac{\mu}{s_i} + \frac{1}{\mu}(c_i(x) + s_i) = 0, \quad 1 \le i \le m. \tag{7}$$

Denoting $a_i(x) = \nabla c_i(x)$, $1 \le i \le m$, $A(x) = [a_1(x), \dots, a_m(x)]$,

$$c(x) = \begin{bmatrix} c_1(x) \\ \dots \\ c_m(x) \end{bmatrix}, \quad s = \begin{bmatrix} s_1 \\ \dots \\ s_m \end{bmatrix}, \quad e = \begin{bmatrix} 1 \\ \dots \\ 1 \end{bmatrix}, \quad u(x, s; \mu) = \begin{bmatrix} u_1(x, s_1; \mu) \\ \dots \\ u_m(x, s_m; \mu) \end{bmatrix} \tag{8}$$

and $S = \mathrm{diag}(s_1, \dots, s_m)$, we can write (6)–(7) in the form

$$\nabla f(x) + A(x)u(x, s; \mu) = 0, \quad u(x, s; \mu) = \mu S^{-1} e. \tag{9}$$

The system of $n+m$ nonlinear equations (9) can be solved by the Newton method, which uses second-order derivatives. In every step of the Newton method, we solve a set of $n+m$ linear equations to obtain increments Δx and Δs of x and s, respectively. These increments can be used for obtaining new quantities

$$x^+ = x + \alpha \Delta x, \quad s^+ = s + \alpha \Delta s,$$

where $\alpha > 0$ is a suitable step-size. This is a standard way for solving general nonlinear programming problems. Using a special problem (2), the structure of $B(x, s; \mu)$ allows us to obtain minimizer $s(x; \mu) \in R$ of function $B(x, s; \mu)$ for a given $x \in R^n$.

Lemma 1. *Function $B(x, s; \mu)$ (with x fixed) has the unique stationary point, which is its global minimizer. This stationary point is characterized by equations*

$$\frac{\mu}{s_i(x;\mu)} = \frac{1}{\mu}(c_i(x) + s_i(x;\mu)) \quad \textit{or} \quad s_i(x;\mu)(c_i(x) + s_i(x;\mu)) = \mu^2, \quad 1 \le i \le m, \tag{10}$$

which have solutions

$$
\begin{aligned}
s_i(x;\mu) &= \frac{\sqrt{c_i^2(x)+4\mu^2}-c_i(x)}{2} \quad \text{if} \quad c_i(x) \le 0, &(11)\\
s_i(x;\mu) &= \frac{2\mu^2}{\sqrt{c_i^2(x)+4\mu^2}+c_i(x)} \quad \text{if} \quad c_i(x) > 0. &(12)
\end{aligned}
$$

Proof. Function $B(x, s; \mu)$ (with x fixed) is convex for $s_i > 0$, $1 \le i \le m$, since it is a sum of convex functions. Thus if a stationary point of $B(x, s; \mu)$ exists, it is its unique global minimizer. Differentiating $B(x, s; \mu)$ by s (see (7)), we obtain quadratic equation (10), which define its unique stationary point. Roots of equation (10) are given by formulas (11) or (12) (we use the formula, which is less sensitive to the round-off errors). □

Assuming $s = s(x; \mu)$, we denote

$$
B(x;\mu) = f(x) - \mu \sum_{i=1}^{m} \log s_i(x;\mu) + \frac{1}{2\mu}\sum_{i=1}^{m}(c_i(x)+s_i(x;\mu))^2 \tag{13}
$$

and $u(x; \mu) = u(x, s(x; \mu); \mu)$. In this case, barrier function $B(x; \mu)$ depends only on x. In order to obtain minimizer $(x, s) \in R^{n+m}$ of $B(x, s; \mu)$, it suffices to minimize $B(x; \mu)$ over R^n.

Lemma 2. *Consider barrier function (13). Then*

$$
\nabla B(x;\mu) = g(x;\mu) \tag{14}
$$

and

$$
\nabla^2 B(x;\mu) = G(x;\mu) + A(x)V(x;\mu)A^T(x), \tag{15}
$$

where

$$
\begin{aligned}
g(x;\mu) &= \nabla f(x) + \sum_{i=1}^{m} u_i(x;\mu)\nabla c_i(x), &(16)\\
G(x;\mu) &= \nabla^2 f(x) + \sum_{i=1}^{m} u_i(x;\mu)\nabla^2 c_i(x) &(17)
\end{aligned}
$$

with

$$
u_i(x;\mu) = \frac{1}{\mu}(c_i(x)+s_i(x;\mu)) = \frac{\mu}{s_i(x;\mu)}, \quad 1 \le i \le m, \tag{18}
$$

and $V(x; \mu) = \mathrm{diag}(v_1(x; \mu), \ldots, v_m(x; \mu))$ *with*

$$
v_i(x;\mu) = \frac{1}{\mu}\,\frac{c_i(x)+s_i(x;\mu)}{c_i(x)+2s_i(x;\mu)}, \quad 1 \le i \le m. \tag{19}
$$

Proof. Differentiating (13), we obtain

$$
\begin{aligned}
\nabla B(x;\mu) &= \nabla f(x) - \sum_{i=1}^{m} \frac{\mu}{s_i(x;\mu)}\nabla s_i(x;\mu)\\
&\quad + \sum_{i=1}^{m} \frac{1}{\mu}(c_i(x)+s_i(x;\mu))(\nabla c_i(x)+\nabla s_i(x;\mu))
\end{aligned}
$$

$$
\begin{aligned}
&= \nabla f(x) + \sum_{i=1}^m \left(-\frac{\mu}{s_i(x;\mu)} + \frac{c_i(x)+s_i(x;\mu)}{\mu} \right) \nabla s_i(x;\mu) \\
&\quad + \sum_{i=1}^m \frac{c_i(x)+s_i(x;\mu)}{\mu} \nabla c_i(x) \\
&= \nabla f(x) + \sum_{i=1}^m \frac{c_i(x)+s_i(x;\mu)}{\mu} \nabla c_i(x) \\
&= \nabla f(x) + \sum_{i=1}^m u_i(x;\mu) \nabla c_i(x)
\end{aligned}
$$

by (10) and (6). Differentiating (10), one has

$$
\nabla s_i(x;\mu)(c_i(x)+s_i(x;\mu)) + s_i(x;\mu)(\nabla c_i(x) + \nabla s_i(x;\mu)) = 0
$$

for $1 \le i \le m$, which gives

$$
\nabla s_i(x;\mu) = -\frac{s_i(x;\mu)}{c_i(x)+2s_i(x;\mu)} \nabla c_i(x) \tag{20}
$$

for $1 \le i \le m$. Thus

$$
\begin{aligned}
\nabla u_i(x;\mu) &= \frac{1}{\mu}(\nabla c_i(x) + \nabla s_i(x;\mu)) = \frac{1}{\mu}\left(1 - \frac{s_i(x;\mu)}{c_i(x)+2s_i(x;\mu)}\right) \nabla c_i(x) \\
&= \frac{1}{\mu} \frac{c_i(x)+s_i(x;\mu)}{c_i(x)+2s_i(x;\mu)} \nabla c_i(x) = v_i(x;\mu) \nabla c_i(x)
\end{aligned}
$$

by (18), (20) and (19). Differentiating (16) and using the previous expression, we obtain

$$
\begin{aligned}
\nabla^2 B(x;\mu) &= \nabla^2 f(x) + \sum_{i=1}^m u_i(x;\mu) \nabla^2 c_i(x) + \sum_{i=1}^m \nabla u_i(x;\mu) a_i^T(x) \\
&= \nabla^2 f(x) + \sum_{i=1}^m u_i(x;\mu) \nabla^2 c_i(x) + \sum_{i=1}^m v_i(x;\mu) a_i(x) a_i^T(x)
\end{aligned}
$$

(recall that $a_i(x) = \nabla c_i(x)$, $1 \le i \le m$), which is equation (15). □

Lemma 3. *Let vector $d \in R^n$ solve equation*

$$
\nabla^2 B(x;\mu) d = -g(x;\mu), \tag{21}
$$

where $g(x;\mu) = \nabla B(x;\mu) \neq 0$. If matrix $G(x;\mu)$ is positive definite, then $d^T g(x;\mu) < 0$ (direction vector d is descent for $B(x;\mu)$).

Proof. Equations (15) and (21) imply

$$
d^T g(x;\mu) = -d^T \nabla^2 B(x;\mu) d = -d^T G(x;\mu) d - d^T A(x) V(x;\mu) A^T(x) d \le -d^T G(x;\mu) d,
$$

since $V(x;\mu)$ is positive definite by (10) and (19). Thus $d^T g(x;\mu) < 0$ if $G(x;\mu)$ is positive definite. □

Expression (19) implies that $v_i(x;\mu)$ can tend to infinity and, therefore, $\nabla^2 B(x;\mu)$ can be ill-conditioned if $\mu \to 0$ (see (15)). The following lemma gives the upper bound for $\|\nabla^2 B(x;\mu)\|$.

Lemma 4. *If Assumption 3 holds, then*

$$\|\nabla^2 B(x;\mu)\| \le (m+1)\overline{G} + m(\frac{\overline{c}\,\overline{G} + \overline{g}^2}{\mu}).$$

Proof. Using (11), one has

$$c_i(x) + s_i(x;\mu) = \frac{c_i(x) + \sqrt{c_i(x)^2 + 4\mu^2}}{2} \le \frac{c_i(x) + |c_i(x)| + 2\mu}{2} \tag{22}$$

for $1 \le i \le m$, which together with (18) gives

$$\begin{aligned} u_i(x;\mu) &\le 1, \quad \text{if} \quad c_i(x) < 0, \\ u_i(x;\mu) &\le \frac{c_i(x)}{\mu} + 1, \quad \text{if} \quad c_i(x) \ge 0. \end{aligned}$$

At the same time, (11) implies that $s_i(x;\mu) \ge 0$ for $1 \le i \le m$, which together with (10) gives $c_i(x) + s_i(x;\mu) \ge 0$ for $1 \le i \le m$. Thus (19) implies

$$v_i(x;\mu) = \frac{1}{\mu}\frac{c_i(x) + s_i(x;\mu)}{c_i(x) + 2s_i(x;\mu)} \le \frac{1}{\mu}, \quad 1 \le i \le m.$$

Using (15), (17) and Assumption 3, we obtain

$$\begin{aligned} \|\nabla^2 B(x;\mu)\| &\le \left\| G(x;\mu) + A(x)V(x;\mu)A^T(x) \right\| \\ &\le \|\nabla^2 f(x)\| + \left\| \sum_{i=1}^m u_i(x;\mu)\nabla^2 c_i(x) \right\| + \left\| \sum_{i=1}^m v_i(x;\mu) a_i(x) a_i^T(x) \right\| \\ &\le \overline{G} + m\overline{G} \max_{1\le i\le m} u_i(x;\mu) + m\overline{g}^2 \max_{1\le i\le m} v_i(x;\mu) \\ &\le (m+1)\overline{G} + m(\frac{\overline{c}\,\overline{G} + \overline{g}^2}{\mu}). \end{aligned}$$

□

Vector $d \in R^n$ obtained by solving (21) is descent for $B(x;\mu)$ if matrix $G(x;\mu)$ is positive definite. Unfortunately, positive definiteness of this matrix is not assured, which causes that standard line-search methods cannot be used. For this reason, trust-region methods were developed. These methods use the direction vector obtained as an approximate minimizer of the quadratic subproblem

$$\text{minimize} \quad Q(d) = \frac{1}{2} d^T \nabla^2 B(x;\mu) d + g^T(x;\mu) d \quad \text{subject to} \quad \|d\| \le \Delta, \tag{23}$$

where Δ is the trust region radius (more details are given in Section 3). Direction vector d serves for obtaining new point $x^+ \in R^n$. Denoting

$$\rho(d) = \frac{B(x+d;\mu) - B(x;\mu)}{Q(d)}, \tag{24}$$

we set

$$x^+ = x \quad \text{if} \quad \rho(d) \le 0, \quad \text{or} \quad x^+ = x + d \quad \text{if} \quad \rho(d) > 0. \tag{25}$$

Finally, we update the trust region radius in such a way that

$$\begin{aligned} \Delta^+ &= \underline{\beta}\Delta \quad \text{if} \quad \rho(d) < \underline{\rho}, \\ \Delta^+ &= \Delta \quad \text{if} \quad \underline{\rho} \le \rho(d) \le \overline{\rho}, \\ \Delta^+ &= \overline{\beta}\Delta \quad \text{if} \quad \overline{\rho} < \rho(d), \end{aligned} \tag{26}$$

where $0 < \underline{\rho} < \overline{\rho} < 1$ and $0 < \underline{\beta} < 1 < \overline{\beta}$.

Now we are in a position to describe the basic algorithm.

Algorithm 1.

Data: Termination parameter $\underline{\varepsilon} > 0$, minimum value of the barrier parameter $\underline{\mu} > 0$, rate of the barrier parameter decrease $0 < \tau < 1$, trust-region parameters $0 < \underline{\rho} < \overline{\rho} < 1$, trust-region coefficients $0 < \underline{\beta} < 1 < \overline{\beta}$, step bound $\overline{\Delta} > 0$.

Input: Sparsity pattern of matrices $\nabla^2 f(x)$ and A. Initial estimation of vector x.

Step 1: *Initiation.* Choose initial barrier parameter $\mu > 0$ and initial trust-region radius $0 < \Delta \le \overline{\Delta}$. Determine the sparsity pattern of matrix $\nabla^2 B$ from the sparsity pattern of matrices $\nabla^2 f(x)$ and A. Carry out symbolic decomposition of $\nabla^2 B$. Compute values $f(x)$ and $c_i(x)$, $1 \le i \le m$. Set $k := 0$ (iteration count).

Step 2: *Termination.* Determine vector $s(x;\mu)$ by (11) or (12) and vector $u(x;\mu)$ by (18). Compute vectors $\nabla f(x)$ and $\nabla c_i(x)$, $1 \le i \le m$, and set $g(x;\mu) := \nabla f(x) + A(x)u(x;\mu)$. If $\mu \le \underline{\mu}$ and $\|g(x;\mu)\| \le \underline{\varepsilon}$, then terminate the computation. Otherwise set $k := k+1$.

Step 3: *Approximation of the Hessian matrix.* Compute approximation of matrix $G(x;\mu)$ by using differences $\nabla f(x+\delta v) + A(x+\delta v)u(x;\mu) - g(x;\mu)$ for a suitable set of vectors v (see [2]). Determine Hessian matrix $\nabla^2 B(x;\mu)$ by (15).

Step 4: *Direction determination.* Determine vector d as an approximate solution of trust-region subproblem (23).

Step 5: *Step-length selection.* If $\rho(d) > 0$ (where $\rho(d)$ is given by (24)), set $x := x+d$ and compute values $f(x)$ and $c_i(x)$, $1 \le i \le m$.

Step 6: *Trust-region update.* Determine new trust-region radius Δ by (26) and set $\Delta := \min(\Delta, \overline{\Delta})$.

Step 7: *Barrier parameter update.* If $\rho(d) \ge \underline{\rho}$, determine a new value of barrier parameter $\mu \ge \underline{\mu}$ (not greater than the current one) by the procedure described in Section 3. Go to Step 2.

The use of the maximum step-length $\overline{\Delta}$ has no theoretical significance, but is very useful for practical computations. First, the problem functions can sometimes be evaluated only in a relatively small region (if they contain exponentials) so that the maximum step-length is necessary. Secondly, the problem can be very ill-conditioned far from the solution point, thus large steps are unsuitable. Finally, if the problem has more local solutions, a suitably chosen maximum step-length can cause a local solution with a lower value of f to be reached. Therefore, maximum step-length $\overline{\Delta}$ is a parameter, which is most frequently tuned.

The important part of Algorithm 1 is the update of barrier parameter μ. There are several influences that should be taken into account, which make the updating procedure rather complicated.

3 Implementation details

In Section 2, we have pointed out that direction vector $d \in R^n$ should be a solution of the quadratic subproblem (23). Usually, an inexact approximate solution suffices. There are several ways for computing a suitable approximate solutions (see, e.g., [19], [4], [22], [23], [18], [21], [13]). We have used the dog-leg method based on direct decompositions of matrix $\nabla^2 B$ (we omit arguments x and μ in the subsequent considerations).

The dog-leg method described in [19], [4], seeks d as a linear combination of the Cauchy step $d_C = -(g^T g / g^T \nabla^2 B g) g$ and the Newton step $d_N = -(\nabla^2 B)^{-1} g$. The Newton step can be computed by using the sparse Gill-Murray decomposition [8]. This decomposition has the form $\nabla^2 B + E = LDL^T$, where E is a positive semidefinite diagonal matrix (which is equal to zero when $\nabla^2 B$ is positive definite), L is a lower triangular matrix and D is a positive definite diagonal matrix. The following algorithm is a typical implementation of the dog-leg method.

Algorithm A: Data $\Delta > 0$.

Step 1: If $g^T \nabla^2 B g \le 0$, set $s := -(\Delta/\|g\|) g$ and terminate the computation.

Step 2: Compute the Cauchy step $d_C = -(g^T g / g^T \nabla^2 B g) g$. If $\|d_C\| \ge \Delta$, set $d := (\Delta/\|d_C\|) d_C$ and terminate the computation.

Step 3: Compute the Newton step $d_N = -(\nabla^2 B)^{-1} g$. If $(d_N - d_C)^T d_C \ge 0$ and $\|d_N\| \le \Delta$, set $d := d_N$ and terminate the computation.

Step 4: If $(d_N - d_C)^T d_C \ge 0$ and $\|d_N\| > \Delta$, determine number θ in such a way that $d_C^T d_C / d_C^T d_N \le \theta \le 1$, choose $\alpha > 0$ such that $\|d_C + \alpha(\theta d_N - d_C)\| = \Delta$, set $d := d_C + \alpha(\theta d_N - d_C)$ and terminate the computation.

Step 5: If $(d_N - d_C)^T d_C < 0$, choose $\alpha > 0$ such that $\|d_C + \alpha(d_C - d_N)\| = \Delta$, set $d := d_C + \alpha(d_C - d_N)$ and terminate the computation.

The above algorithm generates direction vectors such that

$$\begin{aligned} \|d\| &\le \overline{\delta}\Delta, \\ \|d\| &< \underline{\delta}\Delta \;\Rightarrow\; \nabla^2 B d = -g, \\ -Q(d) &\ge \underline{\sigma}\|g\| \min\left(\|d\|, \frac{\|g\|}{\|\nabla^2 B\|}\right), \end{aligned}$$

where $0 < \underline{\sigma} < 1$ is a constant depending on the particular algorithm. These inequalities imply (see [20]), that a constant $0 < \underline{c} < 1$ exists such that

$$\|d\| \ge \underline{c}\gamma/\overline{B}, \tag{27}$$

where γ is the minimum norm of gradients that have been computed and $\overline{B}$ is an upper bound for $\|\nabla^2 B\|$ (assuming $\mu \ge \underline{\mu} > 0$, we can set $\overline{B} = (m+1)\overline{G} + m(\overline{c}\,\overline{G} + \overline{g}^2)/\underline{\mu}$ by Lemma 4). Thus

$$B(x+d;\mu) - B(x;\mu) \le \underline{\rho} Q(d) \le -\underline{\rho}\,\underline{\sigma}\,\underline{c}\frac{\gamma^2}{\overline{B}} \quad \text{if} \quad \rho \ge \underline{\rho} \tag{28}$$

by (25) and (27).

A very important part of Algorithm 1 is the update of the barrier parameter μ. There are two requirements, which play opposite roles. First $\mu \to 0$ should hold, since this is the main property of every interior point method. On the other hand, the convergence theory requires (28) to hold. Thus a lower bound $\underline{\mu}$ for the barrier parameter has to be used (we recommend value $\underline{\mu} = 10^{-6}$ in double precision arithmetic).

Algorithm 1 is also sensitive on the way in which the barrier parameter decreases. We have tested various possibilities for the barrier parameter update including simple geometric sequences, which were proved to be unsuitable. Better results were obtained by setting

$$\mu_{k+1} = \mu_k \quad \text{if} \quad \|g_k\|^2 > \tau\mu_k \quad \text{or} \quad \mu_{k+1} = \max(\underline{\mu}, \|g_k\|^2) \quad \text{if} \quad \|g_k\|^2 \le \tau\mu_k, \tag{29}$$

where $0 < \tau < 1$.

4 Global convergence

In the subsequent considerations, we will assume that $\underline{\varepsilon} = \underline{\mu} = 0$ and all computations are exact. We will investigate infinite sequence $\{x_k\}_1^\infty$ generated by Algorithm 1.

Lemma 5. *Let Assumption 1 and Assumption 3 be satisfied. Then values $\{\mu_k\}_1^\infty$, generated by Algorithm 1, form a non-increasing sequence such that $\mu_k \to 0$. Moreover*

$$\liminf_{k\to\infty} \|\nabla B(x_k; \mu_k)\| = 0. \tag{30}$$

Proof. (a) First we prove that $B(x;\mu)$ is bounded from below if μ is fixed. Using Assumption 1, Assumption 3 and (11), one has

$$s_i(x;\mu) \le \frac{1}{2}\left(|c_i(x)| + \sqrt{c_i^2(x) + 4\mu^2}\right) \le |c_i(x)| + \mu \le \overline{c} + \mu.$$

Thus we can write

$$\begin{aligned} B(x;\mu) &= f(x) - \mu\sum_{i=1}^m \log s_i(x;\mu) + \frac{1}{2\mu}\sum_{i=1}^m (c_i(x) + s_i(x;\mu))^2 \\ &\ge \underline{F} - \mu\sum_{i=1}^m \log(\overline{c} + \mu) = \underline{F} - m\mu\log(\overline{c} + \mu). \end{aligned} \tag{31}$$

(b) Now we prove that the sequence of points in which μ_k is updated is infinite. If it was finite, an index $l \in N$ would exist such that $\mu_{k+1} = \mu_k = \mu_l \ \forall k \ge l$. Since function $B(x;\mu_l)$ is continuous, bounded from below by (a) and since (28) (with $\mu = \mu_l$) holds $\forall k \ge l$, it can be proved (see [20]) that $\liminf_{k\to\infty} \|g(x_k;\mu_l)\| = 0$. Thus an index $k \ge l$ exists such that $\|g(x_k;\mu_l)\|^2 \le \tau\mu_l$ and, therefore, $\mu_{k+1} = \|g(x_k;\mu_l)\|^2 \le \tau\mu_l < \mu_l$ by (29), which is a contradiction. Since the sequence of points where $\mu_{k+1} \le \tau\mu_k$ is infinite, we can conclude that $\mu_k \to 0$.

(c) Let $N_1 = \{k \in N : \|g(x_k;\mu_k)\|^2 \le \tau\mu_k\}$. This set is infinite by (b) and since $\mu_k \to 0$ one has $\|g(x_k;\mu_k)\| \xrightarrow{N_1} 0$. □

Remark 1. Since all quantities considered are bounded by Assumption 3, we can found a subset $N_2 \subset N_1 \subset N$ such that $g(x_k) \stackrel{N_2}{\to} g^*$ and $c_i(x_k) \stackrel{N_2}{\to} c_i^*$, $a_i(x_k) \stackrel{N_2}{\to} a_i^*$, $1 \le i \le m$. Let $M_1 = \{i \in \{1,\dots,m\} : c_i^* \ge 0\}$ and $M_2 = \{i \in \{1,\dots,m\} : c_i^* < 0\}$. We will assume that vectors a_i^*, $i \in M_1$, are linearly independent (the LICQ constraint qualification holds for the limiting Jacobian matrix). To simplify the notation, we use the multiindex M_1 with vectors and matrices corresponding to index set M_1 and multiindex M_2 with vectors and matrices corresponding to index set M_2, respectively.

Theorem 1. *Let Assumption 1 and Assumption 3 be satisfied. Consider sequence $\{x_k\}_1^\infty$, generated by Algorithm 1. Let $\{x_k\}_{N_2} \subset \{x_k\}_1^\infty$ be a subsequence mentioned in Remark 1. Then*

$$\|g(x_k;\mu_k)\| \stackrel{N_2}{\to} 0$$

and

$$s_i(x_k;\mu_k) \ge 0, \quad u_i(x_k;\mu_k) \ge 0, \quad s_i(x_k;\mu_k)u_i(x_k;\mu_k) \stackrel{N_2}{\to} 0$$

for $1 \le i \le m$. If the LICQ constraint qualification holds for the limiting Jacobian matrix, then also

$$\|c_i(x_k;\mu_k) + s_i(x_k;\mu_k)\| \stackrel{N_2}{\to} 0.$$

Proof. (a) The first assertion follows directly from (30). Expressions (10)–(12) imply that $s_i(x_k;\mu_k) \ge 0$, $u_i(x_k;\mu_k) \ge 0$ and $s_i(x_k;\mu_k)u_i(x_k;\mu_k) = \mu_k$, which gives $s_i(x_k;\mu_k)u_i(x_k;\mu_k) \stackrel{N_2}{\to} 0$. Since functions $c_i(x)$, $1 \le i \le m$, are continuous by Assumption 3, there is an index $k_1 \in N_2$ such that $c_{M_2}(x_k) < 0$ for all $k \in N_2$, $k \ge k_1$. Thus $\|c_{M_2}(x_k) + s_{M_2}(x_k;\mu_k)\| \le \sqrt{m}\mu_k$ for all $k \in N_2$, $k \ge k_1$ by (22), which implies $\|c_{M_2}(x_k) + s_{M_2}(x_k;\mu_k)\| \stackrel{N_2}{\to} 0$.

(b) Assume that $M_1 \ne \emptyset$. Let N_2 be a subset mentioned in Remark 1 and $A_{M_1}(x_k) \stackrel{N_2}{\to} A_{M_1}^*$, where matrix $A_{M_1}^*$ has linearly independent columns. Then there is an index $k_2 \in N_2$, $k_2 \ge k_1$ such that $k \in N_2$, $k \ge k_2$ implies $\|A_{M_1}(x_k)v\| \ge \sigma^*\|v\|/2 \ \forall v \in R^n$, where $\sigma^* > 0$ is the minimum singular value of matrix $A_{M_1}^*$. Using (16) and (18) one has

$$\begin{aligned}
&\frac{1}{\mu_k}\|A_{M_1}(x_k)(c_{M_1}(x_k) + s_{M_1}(x_k;\mu_k))\| \\
\le\ & \|g(x_k;\mu_k)\| + \|\nabla f(x_k)\| + \frac{1}{\mu_k}\|A_{M_2}(x_k)(c_{M_2}(x_k) + s_{M_2}(x_k;\mu_k))\| \\
\le\ & \|g(x_k;\mu_k)\| + \|\nabla f(x_k)\| + \sqrt{m}\|A_{M_2}(x_k)\| \le \|g(x_k;\mu_k)\| + (m+1)\overline{g}
\end{aligned}$$

for $k \in N_2$, $k \ge k_2$, since $\|c_{M_2}(x_k) + s_{M_2}(x_k;\mu_k))\| \le \sqrt{m}\mu_k$ by (a). Thus

$$\begin{aligned}
\frac{\sigma^*}{2}\|c_{M_1}(x_k) + s_{M_1}(x_k;\mu_k))\| &\le \|A_{M_1}(x_k)(c_{M_1}(x_k) + s_{M_1}(x_k;\mu_k))\| \\
&\le \mu_k(\|g(x_k;\mu_k)\| + (m+1)\overline{g})
\end{aligned}$$

for $k \in N_2$, $k \ge k_2$ and since $\|g(x_k;\mu_k)\| \stackrel{N_2}{\to} 0$ and $\mu_k \stackrel{N_2}{\to} 0$, we can conclude that $\|c_{M_1}(x_k) + s_{M_1}(x_k;\mu_k)\| \stackrel{N_2}{\to} 0$. □

Corollary 1. *Let assumptions of Theorem 1 hold and sequence $\{x_k\}_1^\infty$ be bounded. Then there exists a cluster point $x \in R^n$ of sequence $\{x_k\}_1^\infty$ satisfying KKT conditions (3)-(4), where $u \in R^m$ is a cluster point of sequence $\{u(x_k;\mu_k)\}_1^\infty$.*

5 Computational experiments

The primal interior-point method was tested and compared with the primal-dual interior point method proposed in [11] by using two sets of test problems. These sets were obtained as modifications of test problems for equality constrained optimization given in [15] and [16], which can be downloaded (together with report [15]) from `http://www.cs.cas.cz/~luksan/test.html` (always two problems of the set were excluded, since they were not successfully solved by both methods compared). In Set 1, equalities $c(x) = 0$ are replaced by inequalities $c(x) \le 0$. Set 2 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.

The results of computational experiments are given in two tables, where P is the problem number, NIT is the number of iterations, NFV is the number of function evaluations, NFG is the number of gradient evaluations and F is the function value reached. The last row of every table contains summary results including the total computational time.

	Primal interior-point method				Primal-dual interior-point method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
2	42	65	588	24299.2	15	15	210	24299.2
3	21	22	132	6.5E-10	19	19	114	1.4E-09
4	135	193	810	399.738	30	36	180	399.732
5	19	20	200	2.8E-13	13	13	130	1.5E-11
6	30	31	434	1.3E-11	30	30	420	-2.6E-13
7	38	48	273	-385.269	40	41	280	-385.269
9	99	119	700	99.8950	37	39	259	99.8950
10	155	185	936	352.954	38	38	228	353.122
11	30	33	186	2.8E-07	23	24	138	1.4E-09
12	79	95	560	7.2E-10	18	18	126	7.2E-09
13	68	73	552	1.9E-08	28	28	224	1.8E-06
14	33	34	238	2.8E-08	49	49	343	4.9E-09
15	36	38	222	1.1E-09	33	33	198	2.6E-10
16	74	86	375	3.7E-08	27	27	135	5.9E-10
17	24	48	120	1196.18	13	13	65	1196.18
18	71	86	360	566.693	14	14	70	566.693
Σ	954	1176	6686	TIME=2.70	427	437	3120	TIME=1.55

Table 1: Set 1 of 16 problems with 1000 variables

	Primal interior-point method				Primal-dual interior-point method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	53	63	324	4.9E-08	32	34	192	3.99778
2	32	55	448	15678.2	18	18	252	15678.2
3	34	59	204	14.9973	19	19	114	14.9974
4	90	117	540	938.570	41	41	246	981.816
6	138	146	1946	12511.0	46	47	644	12511.0
7	68	78	483	-348.499	77	83	539	-348.499
9	101	122	714	100.125	46	62	322	100.125
10	151	183	912	4.9E-08	47	91	282	9.5E-13
11	33	35	204	3.2E-08	24	26	144	1.4E-08
12	24	26	175	1.1E-09	21	21	147	5.1E-08
13	28	30	232	3.2E-08	31	32	248	3.8E-10
14	40	41	287	6.0E-08	35	35	245	2.9E-08
15	27	28	168	6.0E-10	36	40	216	2.1E-11
16	74	85	375	4.9E-08	30	30	150	1.0E-09
17	46	58	235	346.405	19	19	95	346.405
18	68	80	345	198.721	24	24	120	198.721
Σ	1007	1206	7592	TIME=4.44	546	622	3956	TIME=3.29

Table 2: Set 2 of 16 problems with 1000 variables

The results introduced in these tables show that primal-dual interior point method is more robust that primal interior method described in this paper. Nevertheless, sometimes a better solution can be obtained by the primal interior point method (Problem 10 in Table 1 and Problem 1 in Table 2).

References

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

[2] T.F.Coleman, J.J.Moré: Estimation of sparse Hessian matrices and graph coloring problems. Mathematical Programming 28 (1984) 243-270.

[3] A.R.Conn, N.I.M.Gould, P.L.Toint: Trust-region methods. SIAM 2000.

[4] J.E.Dennis, H.H.W.Mei: An unconstrained optimization algorithm which uses function and gradient values. Report No. TR 75-246, 1975.

[5] I.S.Duff, M.Munksgaard, H.B.Nielsen, J.K.Reid: Direct solution of sets of linear equations whose matrix is sparse and indefinite. J. Inst. Maths. Applics. 23 (1979) 235-250.

[6] R.Fletcher: Practical Methods of Optimization (second edition). Wiley, New York, 1987.

[7] R.Fletcher, E.Sainz de la Maza: Nonlinear programming and nonsmooth optimization by successive linear programming. Mathematical Programming (43) (1989) 235-256.

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

[9] N.I.M.Gould, S.Lucidi, M.Roma, P.L.Toint: Solving the trust-region subproblem using the Lanczos method. Report No. RAL-TR-97-028, 1997.

[10] P.E.Gill, W.Murray, M.A.Saunders: SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM Review 47 (2005) 99-131.

[11] L.Lukšan, C.Matonoha, J.Vlček: Interior point method for nonlinear nonconvex optimization. Numerical Linear Algebra with Applications 11 (2004) 431-453.

[12] L.Lukšan, C.Matonoha, J.Vlček: Nonsmooth equation method for nonlinear nonconvex optimization. In: Conjugate Gradient Algorithms and Finite Element Methods (M.Křížek, P.Neittaanmäki, R.Glovinski, S.Korotov eds.). Springer Verlag, Berlin 2004.

[13] L.Lukšan, C.Matonoha, J.Vlček: A shifted Steihaug-toint method for computing a trust-region step. Technical Report V-914. Prague, ICS AS CR, 2004. Submitted to BIT Numerical Mathematics.

[14] L.Lukšan, C.Matonoha, J.Vlček: Primal interior-point method for large sparse minimax optimization. Technical Report V-941. Prague, ICS AS CR, 2005.

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

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

[17] Lukšan L., Vlček J.: Variable metric method for minimization of partially separable nonsmooth functions. To appear in Pacific Journal on Optimization.

[18] J.J.Moré, D.C.Sorensen: Computing a trust region step. SIAM Journal on Scientific and Statistical Computations 4 (1983) 553-572.

[19] M.J.D.Powell: A new algorithm for unconstrained optimization. In: "Nonlinear Programming" (J.B.Rosen O.L.Mangasarian, K.Ritter, eds.) Academic Press, London 1970.

[20] M.J.D.Powell: On the global convergence of trust region algorithms for unconstrained optimization. Mathematical Programming 29 (1984) 297-303.

[21] M.Rojas, S.A.Santos, D.C.Sorensen: A new matrix-free algorithm for the large-scale trust-region subproblems. SIAM J. Optimization 11 (2000) 611-646.

[22] T.Steihaug: The conjugate gradient method and trust regions in large-scale optimization. SIAM Journal on Numerical Analysis 20 (1983) 626-637.

[23] P.L.Toint: Towards an efficient sparsity exploiting Newton method for minimization. In: Sparse Matrices and Their Uses (I.S.Duff, ed.), Academic Press, London 1981, 57-88.

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