Trust-Region Interior-Point Method for Large Sparse l1 Optimization

Lukšan, Ladislav
2005

Institute of Computer Science
Academy of Sciences of the Czech Republic

Trust-region interior-point method for large sparse l_1 optimization

L.Lukšan, C. Matonoha, J. Vlček

Technical report No. 942

November 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

Trust-region interior-point method for large sparse l_1 optimization

L.Lukšan, C. Matonoha, J. Vlček [1]

Technical report No. 942

November 2005

Abstract:

In this report, we propose an interior-point method for large sparse l_1 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 relatively difficult l_1 optimization problems can be solved successfully. The results of computational experiments given in this report confirm efficiency and robustness of the proposed method.

Keywords:
Unconstrained optimization, large-scale optimization, nonsmooth optimization, minimax optimization, 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 l_1 optimization problem: Minimize the function

$$F(x) = \sum_{i=1}^{m} |f_i(x)|, \tag{1}$$

where $f_i : R^n \to R$, $0 \le i \le m$, are smooth functions depending on n_i variables and satisfying either Assumption 1 or Assumption 2. We assume that the function $F(x)$ is partially separable, which means that n, $m = \mathcal{O}(n)$ are large and $n_i = \mathcal{O}(1)$, $0 \le i \le m$, are small.

Assumption 1. Functions $f_i(x)$, $1 \le i \le m$, are twice continuously differentiable on $\mathrm{conv}\mathcal{L}(\overline{F})$ for a sufficiently large upper bound $\overline{F}$, where $\mathcal{L}(\overline{F}) = \{x \in R^n : F(x) \le \overline{F}\}$, and they have bounded first and second-order derivatives on $\mathrm{conv}\mathcal{L}(\overline{F})$. Thus constants $\overline{g}$ and $\overline{G}$ exist such that $\|\nabla f_i(x)\| \le \overline{g}$ and $\|\nabla^2 f_i(x)\| \le \overline{G}$ for all $1 \le i \le m$ and $x \in \mathrm{conv}\mathcal{L}(\overline{F})$.

Assumption 2. Functions $f_i(x)$, $1 \le i \le m$, are twice continuously differentiable on a sufficiently large convex compact set $\mathcal{D}$.

Since continuous functions attain their maxima on a compact set, Assumption 2 guarantees that constants $\overline{F}$, $\overline{g}$ and $\overline{G}$ exist such that $f_i(x) \le \overline{F}$, $\|\nabla f_i(x)\| \le \overline{g}$ and $\|\nabla^2 f_i(x)\| \le \overline{G}$ for all $1 \le i \le m$ and $x \in \mathcal{D}$. The choice of $\mathcal{L}(\overline{F})$ and $\mathcal{D}$ will be discussed later (see Assumption 3). Note that the set $\mathcal{L}(\overline{F})$ used in Assumption 1 need not be compact.

The minimization of F is equivalent to the sparse nonlinear programming problem with $n + m$ variables $x \in R^n$, $z \in R^m$:

$$\text{minimize} \quad \sum_{i=1}^{m} z_i \quad \text{subject to} \quad -z_i \le f_i(x) \le z_i, \quad 1 \le i \le m. \tag{2}$$

This problem satisfies the Mangasarian-Fromowitz constraint qualification conditions and the necessary first-order (KKT) conditions have the form

$$\sum_{i=1}^{m} u_i \nabla f_i(x) = 0, \quad z_i = |f_i(x)|, \quad |u_i| \le 1 \quad \text{and} \quad u_i = \frac{f_i(x)}{|f_i(x)|} \quad \text{if} \quad |f_i(x)| > 0. \tag{3}$$

Problem (2) can be solved by an arbitrary nonlinear programming method utilizing sparsity (sequential linear programming [8], sequential quadratic programming [10], interior-point [1], [12], [25] and nonsmooth equation [13]). The original problem (1) is a special case of the polyhedral composite nonsmooth problem (see [7]) and it can be also solved by the trust-region methods described in [20] and [26].

In this report, we introduce a trust-region interior-point method that utilizes a special structure of the l_1 optimization problem. The constrained problem (2) is replaced

by a sequence of unconstrained problems

$$\begin{aligned} \text{minimize} \quad B(x,z;\mu) &= \sum_{i=1}^{m} z_i - \mu \sum_{i=1}^{m} \log(z_i - f_i(x)) - \mu \sum_{i=1}^{m} \log(z_i + f_i(x)) \\ &= \sum_{i=1}^{m} z_i - \mu \sum_{i=1}^{m} \log(z_i^2 - f_i^2(x)) \end{aligned} \tag{4}$$

with a barrier parameter $0 < \mu \leq \overline{\mu}$, where we assume that $z_i > |f_i(x)|$, $1 \leq i \leq m$, and $\mu \to 0$ monotonically. Here $B(x,z;\mu) : R^{n+m} \to R$ is a function of $n+m$ variables $x \in R^n$, $z \in R^m$.

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

Assumption 3. Either Assumption 1 holds and $\{x_k\}_1^\infty \in \mathcal{L}(\overline{F})$ or Assumption 2 holds and $\{x_k\}_1^\infty \in \mathcal{D}$.

The interior-point method investigated in this report is a trust-region modification of the Newton method. Approximation of the Hessian matrix is computed by gradient differences which can be carried out efficiently if the Hessian matrix is sparse (see [3]). Since the Hessian matrix need not be positive definite in a non-convex case, a standard line-search realization cannot be used. There are two basic possibilities, either a trust-region approach or a 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 report is organized as follows. In Section 2, we introduce the interior-point method for large sparse l_1 optimization and describe a corresponding algorithm. Section 3 contains more details concerning this algorithm such as a trust-region strategy and a 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 3 holds. Finally, in Section 5 we present results of computational experiments confirming the efficiency of the proposed method.

2 Description of the method

Differentiating $B(x,z;\mu)$ given by (4), we obtain necessary conditions for a minimum in the form

$$\sum_{i=1}^{m} \frac{2\mu f_i(x)}{z_i^2 - f_i^2(x)} \nabla f_i(x) \triangleq \sum_{i=1}^{m} u_i(x,z_i;\mu) \nabla f_i(x) = 0 \tag{5}$$

and

$$1 - \frac{2\mu z_i}{z_i^2 - f_i^2(x)} = 1 - u_i(x,z_i;\mu)\frac{z_i}{f_i(x)} = 0, \quad 1 \leq i \leq m. \tag{6}$$

Denoting $g_i(x) = \nabla f_i(x)$, $1 \le i \le m$, $A(x) = [g_1(x), \ldots, g_m(x)]$,

$$f(x) = \begin{bmatrix} f_1(x) \\ \ldots \\ f_m(x) \end{bmatrix}, \quad z = \begin{bmatrix} z_1 \\ \ldots \\ z_m \end{bmatrix}, \quad u(x,z;\mu) = \begin{bmatrix} u_1(x,z_1;\mu) \\ \ldots \\ u_m(x,z_m;\mu) \end{bmatrix} \tag{7}$$

and $Z = \mathrm{diag}(z_1, \ldots, z_m)$, we can write (5)–(6) in the form

$$A(x)u(x,z;\mu) = 0, \quad u(x,z;\mu) = Z^{-1}f(x). \tag{8}$$

The system of $n+m$ nonlinear equations (8) 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 Δz of x and z, respectively. These increments can be used for obtaining new quantities

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

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

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

$$\frac{2\mu z_i(x;\mu)}{z_i^2(x;\mu) - f_i^2(x)} = 1 \quad \textit{or} \quad z_i^2(x;\mu) - f_i^2(x) = 2\mu z_i(x;\mu), \quad 1 \le i \le m, \tag{9}$$

which have the solutions

$$z_i(x;\mu) = \mu + \sqrt{\mu^2 + f_i^2(x)}, \quad 1 \le i \le m. \tag{10}$$

Proof. The function $B(x,z;\mu)$ (with x fixed) is convex for $z_i > |f_i(x)|$, $1 \le i \le m$, since it is a sum of convex functions. Thus if a stationary point of $B(x,z;\mu)$ exists, it is its unique global minimizer. Differentiating $B(x,z;\mu)$ by z (see (6)), we obtain the quadratic equations (9), which define its unique stationary point. □

Assuming $z = z(x;\mu)$ and using (6) and (10), we denote

$$u_i(x;\mu) \;=\; u_i(x,z(x;\mu);\mu) = \frac{f_i(x)}{z_i(x;\mu)} = \frac{f_i(x)}{\mu + \sqrt{\mu^2 + f_i^2(x)}}, \quad 1 \le i \le m, \tag{11}$$

and

$$\begin{aligned} B(x;\mu) \;&=\; B(x,z(x;\mu);\mu) = \sum_{i=1}^m z_i(x;\mu) - \mu \sum_{i=1}^m \log(z_i^2(x;\mu) - f_i^2(x)) \\ &=\; \sum_{i=1}^m [z_i(x;\mu) - \mu \log(z_i(x;\mu))] - \mu m \log(2\mu). \end{aligned} \tag{12}$$

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

Lemma 2. *Consider the barrier function (12). Then*

$$\nabla B(x;\mu) = A(x)u(x;\mu) \tag{13}$$

and

$$\nabla^2 B(x;\mu) = G(x;\mu) + A(x)V(x;\mu)A^T(x), \tag{14}$$

where

$$G(x;\mu) = \sum_{i=1}^{m} u_i(x;\mu)G_i(x) \tag{15}$$

with $G_i(x) = \nabla^2 f_i(x)$, $1 \le i \le m$, *and* $V(x;\mu) = \mathrm{diag}(v_1(x;\mu),\ldots,v_m(x;\mu))$ *with*

$$v_i(x;\mu) = \frac{2\mu}{z_i^2(x;\mu) + f_i^2(x)}, \quad 1 \le i \le m. \tag{16}$$

Proof. Differentiating (12), we obtain

$$\begin{aligned}
\nabla B(x;\mu) &= \sum_{i=1}^{m} \nabla z_i(x;\mu) - 2\mu \sum_{i=1}^{m} \frac{z_i(x;\mu)\nabla z_i(x;\mu) - f_i(x)g_i(x)}{z_i^2(x;\mu) - f_i^2(x)} \\
&= \sum_{i=1}^{m} \left(1 - \frac{2\mu z_i(x;\mu)}{z_i^2(x;\mu) - f_i^2(x)}\right) \nabla z_i(x;\mu) + \sum_{i=1}^{m} \frac{2\mu f_i(x)g_i(x)}{z_i^2(x;\mu) - f_i^2(x)} \\
&= \sum_{i=1}^{m} u_i(x;\mu)g_i(x) = A(x)u(x;\mu)
\end{aligned}$$

by (9) and (5). Differentiating (9), one has

$$\frac{\nabla z_i(x;\mu)}{z_i^2(x;\mu) - f_i^2(x)} - \frac{2z_i(x;\mu)(z_i(x;\mu)\nabla z_i(x;\mu) - f_i(x)g_i(x))}{(z_i^2(x;\mu) - f_i^2(x))^2} = 0$$

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

$$\nabla z_i(x;\mu) = \frac{2z_i(x;\mu)f_i(x)g_i(x)}{z_i^2(x;\mu) + f_i^2(x)} \tag{17}$$

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

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

by (11), (17), (9) and (16). Differentiating (13) and using the previous expression, we obtain

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

which is equation (14). □

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

$$\nabla^2 B(x;\mu) d = -g(x;\mu), \tag{18}$$

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

Proof. Equation (18) implies

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

since $V(x;\mu)$ is positive definite by (16). Thus $d^T g(x;\mu) < 0$ if $G(x;\mu)$ is positive definite. □

Expression (16) implies that $v_i(x;\mu)$ is bounded if $f_i^2(x)$ is bounded from zero. If $f_i^2(x)$ tends to zero faster than μ then $v_i(x;\mu)$ can tend to infinity and $\nabla^2 B(x;\mu)$ can be ill-conditioned (see (14)). The following lemma gives the upper bound for $\|\nabla^2 B(x;\mu)\|$.

Lemma 4. *If Assumption 3 holds, then*

$$\|\nabla^2 B(x;\mu)\| \leq m(\overline{G} + \overline{g}^2 \|V(x;\mu)\|) \leq \frac{\overline{C}}{\mu},$$

where $\overline{C} = m(2\overline{\mu}\overline{G} + \overline{g}^2)/2$.

Proof. Using (14) and Assumption 3, we obtain

$$\begin{aligned}\|\nabla^2 B(x;\mu)\| &= \left\| G(x;\mu) + A(x) V(x;\mu) A^T(x) \right\| \\ &\leq \left\| \sum_{i=1}^{m} u_i(x;\mu) G_i(x) \right\| + \left\| \sum_{i=1}^{m} v_i(x;\mu) g_i(x) g_i^T(x) \right\| \\ &\leq m\overline{G} + m\overline{g}^2 \|V(x;\mu)\|,\end{aligned}$$

since $|u_i(x;\mu)| \le 1$, $1 \le i \le m$, by (11). Since $V(x;\mu)$ is diagonal, one has

$$\|V(x;\mu)\| = \max_{1\le i\le m} |v_i(x;\mu)| = \max_{1\le i\le m} \left(\frac{2\mu}{z_i^2(x;\mu)+f_i^2(x)}\right) \tag{19}$$

by (16). Using (10), we can write

$$\begin{aligned} z_i^2(x;\mu)+f_i^2(x) &= \left(\mu+\sqrt{\mu^2+f_i^2(x)}\right)^2 + f_i^2(x) \\ &= 2\left(\mu^2+\mu\sqrt{\mu^2+f_i^2(x)}+f_i^2(x)\right) \ge 4\mu^2 \end{aligned}$$

for all $1 \le i \le m$, which together with (19) proves the lemma. □

The vector $d \in R^n$ obtained by solving (18) is descent for $B(x;\mu)$ if the matrix $G(x;\mu)$ is positive definite. Unfortunately, the positive definiteness of this matrix is not assured, which causes that the standard line-search methods cannot be used. For this reason, the trust-region methods were developed. These methods use a direction vector obtained as an approximate minimizer of the quadratic subproblem

$$\text{minimize} \quad Q(d) = \frac{1}{2}d^T\nabla^2 B(x;\mu)d + g^T(x;\mu)d \quad \text{subject to} \quad \|d\| \le \Delta, \tag{20}$$

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

$$\rho(d) = \frac{B(x+d;\mu)-B(x;\mu)}{Q(d)}, \tag{21}$$

we set

$$x^+ = x \quad \text{if} \quad \rho(d) < \underline{\rho} \quad \text{or} \quad x^+ = x+d \quad \text{if} \quad \rho(d) \ge \underline{\rho} \tag{22}$$

and update the trust region radius in such a way that

$$\underline{\beta}\|d\| \le \Delta^+ \le \overline{\beta}\|d\| \quad \text{if} \quad \rho(d) < \overline{\rho} \quad \text{or} \quad \Delta \le \Delta^+ \le \overline{\gamma}\Delta \quad \text{if} \quad \rho(d) \ge \overline{\rho}, \tag{23}$$

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

In (20), we assume that $\nabla^2 B(x;\mu) = G + A(x)V(x;\mu)A^T(x)$, where $G = G(x;\mu)$ (see (14)). In practical computations, G is an approximation of $G(x;\mu)$ obtained by using either gradient differences or variable metric updates. In the first case, G is computed by differences $A(x+\delta v_j)u(x;\mu) - A(x)u(x;\mu)$ for a suitable set of vectors v_j, $j = 1, 2, \ldots, \underline{n}$, where $\underline{n} \ll n$. Determination of vectors v_j, $j = 1, 2, \ldots, \underline{n}$, is equivalent to a graph coloring problem (see [3]). The corresponding code is proposed in [2]. In the second case, G is defined by the expression

$$G = \sum_{i=1}^{m} u_i(x;\mu)G_i,$$

where approximations G_i of $\nabla^2 f_i(x)$ are computed by using variable metric updates described in [11]. In this case, we assume that problem (1) is "partially separable", which means that functions $f_i(x)$, $1 \leq i \leq m$, depend on a small number of variables (n_i, say, with $n_i = \mathcal{O}(1)$, $1 \leq i \leq m$). More details are given in the next section.

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

Algorithm 1.

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

Input: A sparsity pattern of the matrix A. An initial estimation of the vector x.

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

Step 2: *Termination.* Determine the vector $z(x; \mu)$ by (10) and the vector $u(x; \mu)$ by (8). Compute the matrix $A(x)$ and the vector $g(x; \mu) = A(x)u(x; \mu)$. If $\mu \leq \underline{\mu}$ and $\|g(x; \mu)\| \leq \underline{\varepsilon}$, then terminate the computation. Otherwise set $k := k + 1$.

Step 3: *Approximation of the Hessian matrix.* Compute approximation G of Hessian matrix $G(x; \mu)$ by using either gradient differences or variable metric updates. Determine the Hessian matrix $\nabla^2 B(x; \mu)$ by (14).

Step 4: *Direction determination.* Determine the vector d as an approximate solution of the trust-region subproblem (20).

Step 5: *Step-length selection.* Set $x := x^+$, where x^+ is a point defined by (22). Compute the values $f_i(x)$, $1 \leq i \leq m$, and $F(x) = \sum_{1 \leq i \leq m} |f_i(x)|$.

Step 6: *Trust-region update.* Determine a new trust-region radius Δ satisfying (23) and set $\Delta := \min(\Delta, \overline{\Delta})$.

Step 7: *Barrier parameter update.* If $\rho(d) \geq \underline{\rho}$ (where $\rho(d)$ is given by (21)), determine a new value of the barrier parameter $\mu \geq \underline{\mu}$ (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, the maximum step-length $\overline{\Delta}$ is a parameter, which is most frequently tuned.

An important part of Algorithm 1 is an update of the 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 the direction vector $d \in R^n$ should be a solution of the quadratic subproblem (20). Usually, an inexact approximate solution suffices. There are several ways for computing suitable approximate solutions (see, e.g., [5], [14], [18], [19], [22], [23], [24]). We have used two approaches based on direct decompositions of the matrix $\nabla^2 B$ (to simplify the notation, we omit arguments x and μ in the subsequent considerations, i.e., we will write $\nabla^2 B$ and g instead of $\nabla^2 B(x;\mu)$ and $g(x;\mu)$).

The first strategy, the dog-leg method described in [5], [19], 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 is computed by using either the sparse Gill-Murray decomposition [9] or the sparse Bunch-Parlett decomposition [6]. The sparse Gill-Murray decomposition has the form $\nabla^2 B + E = LDL^T = R^T R$, 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, D is a positive definite diagonal matrix and R is an upper triangular matrix. The sparse Bunch-Parlett decomposition has the form $\nabla^2 B = PLML^T P^T$, where P is a permutation matrix, L is a lower triangular matrix and M is a block-diagonal matrix with 1×1 or 2×2 blocks (which is indefinite when $\nabla^2 B$ is indefinite). The following algorithm is a typical implementation of the dog-leg method.

Algorithm A: Data $\Delta > 0$.

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

Step 2: Compute the Cauchy step $d_C = -(g^T g / g^T \nabla^2 B g) g$. If $\|d_C\| \geq \Delta$, 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 $(d_N - d_C)^T d_C \geq 0$ and $\|d_N\| \leq \Delta$, set $d := d_N$ and terminate the computation.

Step 4: If $(d_N - d_C)^T d_C \geq 0$ and $\|d_N\| > \Delta$, determine a number θ in such a way that $d_C^T d_C / d_C^T d_N \leq \theta \leq 1$, 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 second strategy, the optimum step method, computes a more accurate solution

of (20) by using the Newton method applied to the nonlinear equation

$$\frac{1}{\|d(\lambda)\|} - \frac{1}{\Delta} = 0, \tag{24}$$

where $(\nabla^2 B + \lambda I)d(\lambda) = -g$. This way, described in [18] in more details, follows from the KKT conditions for (20). Since the Newton method applied to (24) can be unstable, the safeguards (lower and upper bounds to λ) are usually used. The following algorithm is a typical implementation of the optimum step method.

Algorithm B: Data $0 < \underline{\delta} < 1 < \overline{\delta}$ (usually $\underline{\delta} = 0.9$ and $\overline{\delta} = 1.1$), $\Delta > 0$.

Step 1: Determine $\underline{\nu}$ as the maximum diagonal element of the matrix $-\nabla^2 B$. Compute $\overline{\lambda} = \|g\|/\Delta + \|\nabla^2 B\|$, $\underline{\lambda} = \|g\|/\Delta - \|\nabla^2 B\|$ and set $\underline{\lambda} := \max(0, \underline{\nu}, \underline{\lambda})$, $\lambda := \underline{\lambda}$. Set $l = 0$ (the inner iteration count).

Step 2: If $l > 0$ and $\lambda \le \underline{\nu}$, set $\lambda := \sqrt{\underline{\lambda}\overline{\lambda}}$.

Step 3: Determine the Gill-Murray decomposition $\nabla^2 B + \lambda I + E = R^T R$. If $E = 0$ (i.e. if $\nabla^2 B + \lambda I$ is positive definite), go to Step 4. In the opposite case, determine a vector $v \in R^n$ such that $\|v\| = 1$ and $v^T(\nabla^2 B + \lambda I)v \le 0$, set $\underline{\nu} := \lambda - v^T(\nabla^2 B + \lambda I)v$, $\underline{\lambda} := \max(\underline{\nu}, \underline{\lambda})$, $l := l + 1$ and go to Step 2.

Step 4: Determine a vector $d \in R^n$ as a solution of the equation $R^T R d + g = 0$. If $\|d\| > \overline{\delta}\Delta$, set $\underline{\lambda} := \lambda$ and go to Step 6. If $\underline{\delta}\Delta \le \|d\| \le \overline{\delta}\Delta$, terminate the computation. If $\|d\| < \underline{\delta}\Delta$ and $\lambda = 0$, terminate the computation. If $\|d\| < \underline{\delta}\Delta$ and $\lambda \ne 0$, set $\overline{\lambda} := \lambda$ and go to Step 5.

Step 5: Determine a vector $v \in R^n$ as a good approximation of the eigenvector corresponding to the minimum eigenvalue of $\nabla^2 B$ in such a way that $\|v\| = 1$ and $v^T d \ge 0$ (this vector can be determined from the decomposition $R^T R$ in the way used in subroutines of the LAPACK library). Determine a number $\alpha \ge 0$ such that $\|d + \alpha v\| = \Delta$. If $\alpha^2\|Rv\|^2 \le (1 - \underline{\delta}^2)(\|Rd\|^2 + \lambda\Delta^2)$, set $d := d + \alpha v$ and terminate the computation. In the opposite case, set $\underline{\nu} := \lambda - \|Rv\|^2$, $\underline{\lambda} := \max(\underline{\nu}, \underline{\lambda})$ and go to Step 6.

Step 6: Determine a vector $v \in R^n$ as a solution of the equation $R^T v = d$ and set

$$\lambda := \lambda + \frac{\|d\|^2}{\|v\|^2}\left(\frac{\|d\| - \Delta}{\Delta}\right).$$

If $\lambda < \underline{\lambda}$, set $\lambda := \underline{\lambda}$. If $\lambda > \overline{\lambda}$, set $\lambda := \overline{\lambda}$. Set $l := l + 1$ and go to Step 2.

The above algorithms generate the 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 and (23) imply (see [21]) that a constant $0 < \underline{c} < 1$ exists such that

$$\|d\| \geq \underline{c} \frac{\|\tilde{g}\|\mu}{\overline{C}}, \tag{25}$$

where $\|\tilde{g}\|$ is the minimum norm of all gradients that have been computed so far and $\overline{C}$ is a constant used in Lemma 4. Thus using (21), (22) and (25), one has

$$B(x+d;\mu) - B(x;\mu) \leq \underline{\rho} Q(d) \leq -\underline{C}\mu\|\tilde{g}\|^2 \quad \text{if} \quad \rho(d) \geq \underline{\rho}, \tag{26}$$

where $\underline{C} = \underline{\rho}\,\underline{\sigma}\,\underline{c}/\overline{C}$.

The update of the trust region radius satisfying (23) requires a more detailed specification. In our implementation, we have set

$$\begin{array}{rlll}
\Delta^+ = \underline{\beta}\|d\| & \text{if} & \rho(d) < \overline{\rho}_1 & \text{and} \quad \tilde{\Delta} < \underline{\beta}\|d\|, \\
\Delta^+ = \tilde{\Delta} & \text{if} & \rho(d) < \overline{\rho}_1 & \text{and} \quad \underline{\beta}\|d\| \leq \tilde{\Delta} \leq \overline{\beta}\|d\|, \\
\Delta^+ = \overline{\beta}\|d\| & \text{if} & \rho(d) < \overline{\rho}_1 & \text{and} \quad \overline{\beta}\|d\| \leq \tilde{\Delta}, \\
\Delta^+ = \Delta & \text{if} & \overline{\rho}_1 \leq \rho(d) \leq \overline{\rho}_2, & \\
\Delta^+ = \Delta & \text{if} & \overline{\rho}_2 < \rho(d) & \text{and} \quad \|d\| < \underline{\delta}\Delta, \\
\Delta^+ = \overline{\gamma}\Delta & \text{if} & \overline{\rho}_2 < \rho(d) & \text{and} \quad \|d\| \geq \underline{\delta}\Delta,
\end{array}$$

where $0 < \underline{\rho} < \overline{\rho} = \overline{\rho}_1 < \overline{\rho}_2 < 1$, $\underline{\delta}$ is a value used in Algorithm B ($\underline{\delta} = 1$ in Algorithm A) and $\tilde{\Delta}$ is a value obtained by the quadratic interpolation.

Matrix G appearing in Step 3 of Algorithm 1 can be computed by using gradient differences as was mentioned in the previous section. Alternatively, safeguarded BFGS updates can be applied efficiently. In this case, $G^+ = \sum_{i=1}^m u_i(x^+;\mu^+)G_i^+$, where x^+ is the vector of variables in the next iteration and G_i^+, $1 \leq i \leq m$ are computed by the following way. Let $R_i^n \subset R^n$, $1 \leq i \leq m$, be subspaces defined by n_i variables on which each f_i depends and $Z_i \in R^{n \times n_i}$ be the matrix whose columns form the canonical orthonormal basis in R_i^n (i.e., they are columns of the unit matrix). Then we can define reduced approximations of the Hessian matrices $\tilde{G}_i = Z_i^T G_i Z_i$, $1 \leq i \leq m$. New reduced approximations of the Hessian matrices are computed by the formulas

$$\begin{array}{rcll}
\tilde{G}_i^+ & = & \tilde{G}_i + \dfrac{\tilde{y}_i\tilde{y}_i^T}{\tilde{s}_i^T\tilde{y}_i} - \dfrac{\tilde{G}_i\tilde{s}_i\tilde{s}_i^T\tilde{G}_i}{\tilde{s}_i^T\tilde{G}_i\tilde{s}_i}, & \tilde{s}_i^T\tilde{y}_i > 0, \\
\tilde{G}_i^+ & = & \tilde{G}_i, & \tilde{s}_i^T\tilde{y}_i \leq 0,
\end{array}$$

where $\tilde{s}_i = Z_i^T(x^+ - x)$, $\tilde{y}_i = Z_i^T(\nabla f_i(x^+) - \nabla f_i(x))$, $1 \leq i \leq m$. Finally $G_i^+ = Z_i\tilde{G}_i^+ Z_i^T$, $1 \leq i \leq m$. In the first iteration we set $\tilde{G}_i = I$, $1 \leq i \leq m$ (I is the unit matrix).

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, Lemma 4

implies that $\nabla^2 B(x;\mu)$ can be ill-conditioned if μ is too small. Thus the lower bound $\underline{\mu}$ for μ is used in Algorithm 1 (we recommend value 10^{-8} in the 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

$$\mu^+ = \max(\underline{\mu}, \|g\|^2) \quad \text{if} \quad \rho(d) \geq \underline{\rho} \quad \text{and} \quad \|g\|^2 \leq \tau\mu \tag{27}$$

(where $0 < \tau < 1$) and $\mu^+ = \mu$ otherwise.

4 Global convergence

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

Lemma 5. *Let Assumption 3 be satisfied. Then the values $\{\mu_k\}_1^\infty$, generated by Algorithm 1, form a non-increasing sequence such that $\mu_k \to 0$. Moreover, there is an infinite subset $K \subset N$ such that $\rho(d_k) \geq \underline{\rho}$ and $\|g(x_k;\mu_k)\|^2 \leq \tau\mu_k$ for all $k \in K$.*

Proof. (a) First, we prove that $B(x;\mu)$ is bounded from below if μ is fixed. Since $z_i(x;\mu) \geq 2\mu$ and

$$\frac{\partial B(x;\mu)}{\partial z_i} = 1 - \frac{\mu}{z_i(x;\mu)} \geq 1 - \frac{\mu}{2\mu} = \frac{1}{2}$$

for $1 \leq i \leq m$ by (10) and (12), function $B(x;\mu)$ attains its minimum if $z_i(x;\mu) = 2\mu$ for $1 \leq i \leq m$. Thus

$$\begin{aligned} B(x;\mu) &= \sum_{i=1}^{m} [z_i(x;\mu) - \mu \log(z_i(x;\mu))] - m\mu \log(2\mu) \\ &\geq m[2\mu - \mu\log(2\mu)] - m\mu\log(2\mu) = 2m\mu(1 - \log(2\mu)). \end{aligned} \tag{28}$$

(b) Now we prove that the set K of indices for which μ_k is updated is infinite. Note that K is the set of indices such that $\rho(d_k) \geq \underline{\rho}$ and $\|g(x_k;\mu_k)\|^2 \leq \tau\mu_k$ (see (27)). If K was finite, an index $\underline{k} \in N$ would exist such that $\mu_k = \mu_{\underline{k}}$ $\forall k \geq \underline{k}$. Since the function $B(x;\mu_{\underline{k}})$ is continuous, uniformly bounded from below by (a), $\|\nabla^2 B(x;\mu_{\underline{k}})\|$ is uniformly bounded from above by Lemma 4 and (22) holds for all $k \geq \underline{k}$, it can be proved (see [4]) that $\lim_{k\to\infty} \|g(x_k;\mu_{\underline{k}})\| = 0$. Thus an index $l \geq \underline{k}$ exists such that $\rho(d_l) \geq \underline{\rho}$ and $\|g(x_l;\mu_{\underline{k}})\|^2 \leq \tau\mu_{\underline{k}}$ (since $\rho(d_k) < \underline{\rho}$ implies that x_k and, therefore, $\|g(x_k;\mu_{\underline{k}})\|$ does not change). Thus $\mu_{l+1} = \|g(x_l;\mu_{\underline{k}})\|^2 \leq \tau\mu_{\underline{k}} < \mu_{\underline{k}}$ by (27), which is in a contradiction with the assumption that $\mu_k = \mu_{\underline{k}}$ $\forall k \geq \underline{k}$. Since we have proved that the set K of indices for which $\mu_{k+1} \leq \tau\mu_k$ is infinite, we can conclude that $\mu_k \to 0$. □

Now we will prove that

$$B(x_{k+1};\mu_{k+1}) \leq B(x_{k+1};\mu_k) - L(\mu_{k+1} - \mu_k) \tag{29}$$

for some constant $L \in R$. For this purpose, we consider that $z(x;\mu)$ and $B(x;\mu)$ are functions of μ and we write $z(x,\mu) = z(x;\mu)$ and $B(x,\mu) = B(x;\mu)$.

Lemma 6. *Let $z_i(x,\mu)$, $1 \le i \le m$, be the values given by Lemma 1 (for fixed x and variable μ). Then*

$$\frac{\partial z_i(x,\mu)}{\partial \mu} > 1, \quad \forall 1 \le i \le m,$$

and

$$\frac{\partial B(x,\mu)}{\partial \mu} = -\sum_{i=1}^{m} \log(z_i^2(x,\mu) - f_i^2(x)) = -\sum_{i=1}^{m} \log(2\mu z_i(x,\mu)).$$

Proof. Differentiating the expressions $z_i(x,\mu) = \mu + \sqrt{\mu^2 + f_i^2(x)}$, $1 \le i \le m$, following from Lemma 1, we obtain

$$\frac{\partial z_i(x,\mu)}{\partial \mu} = 1 + \frac{\mu}{\sqrt{\mu^2 + f_i^2(x)}} > 1, \quad 1 \le i \le m.$$

Differentiating the function

$$B(x,\mu) = \sum_{i=1}^{m} z_i(x,\mu) - \mu \sum_{i=1}^{m} \log(z_i^2(x,\mu) - f_i^2(x)),$$

one has

$$\begin{aligned}
\frac{\partial B(x,\mu)}{\partial \mu} &= \sum_{i=1}^{m} \frac{\partial z_i(x,\mu)}{\partial \mu} - \sum_{i=1}^{m} \log(z_i^2(x,\mu) - f_i^2(x)) - \sum_{i=1}^{m} \frac{2\mu z_i(x,\mu)}{z_i^2(x,\mu) - f_i^2(x)} \frac{\partial z_i(x,\mu)}{\partial \mu} \\
&= \sum_{i=1}^{m} \frac{\partial z_i(x,\mu)}{\partial \mu} \left(1 - \frac{2\mu z_i(x,\mu)}{z_i^2(x,\mu) - f_i^2(x)}\right) - \sum_{i=1}^{m} \log(z_i^2(x,\mu) - f_i^2(x)) \\
&= -\sum_{i=1}^{m} \log(z_i^2(x,\mu) - f_i^2(x)) = -\sum_{i=1}^{m} \log(2\mu z_i(x,\mu))
\end{aligned}$$

by (9). □

Lemma 7. *Let Assumption 3 be satisfied. Then (29) holds with some constant $L \in R$.*

Proof. Using Lemma 6, the mean value theorem and (10), we can write

$$\begin{aligned}
B(x_{k+1},\mu_{k+1}) - B(x_{k+1},\mu_k) &= -\sum_{i=1}^{m} \log(2\tilde{\mu}_k z_i(x_{k+1},\tilde{\mu}_k))(\mu_{k+1} - \mu_k) \\
&\le -\sum_{i=1}^{m} \log\left(2\tilde{\mu}_k(2\tilde{\mu}_k + \overline{F})\right)(\mu_{k+1} - \mu_k) \\
&\le -\sum_{i=1}^{m} \log\left(2\overline{\mu}(2\overline{\mu} + \overline{F})\right)(\mu_{k+1} - \mu_k) \\
&\triangleq -L(\mu_{k+1} - \mu_k),
\end{aligned}$$

where $0 < \mu_{k+1} \le \tilde{\mu}_k \le \mu_k \le \overline{\mu}$. □

Theorem 1. *Let Assumption 3 be satisfied. Consider the sequence $\{x_k\}_1^\infty$, generated by Algorithm 1. Then*

$$\liminf_{k\to\infty} \sum_{i=1}^{m} u_i(x_k;\mu_k) g_i(x_k) = 0,$$

and

$$\lim_{k\to\infty} (z_i^2(x_k;\mu_k) - f_i^2(x_k)) = 0, \quad |u_i(x_k;\mu_k)| \le 1,$$
$$\lim_{k\to\infty} u_i(x_k;\mu_k) = 1 \quad \text{if} \quad \liminf_{k\to\infty} f_i(x_k) > 0,$$
$$\lim_{k\to\infty} u_i(x_k;\mu_k) = -1 \quad \text{if} \quad \limsup_{k\to\infty} f_i(x_k) < 0$$

for $1 \le i \le m$.

Proof. The inequalities $|u_i(x_k;\mu_k)| \le 1$, $1 \le i \le m$, and the last two relations follow immediately from (11).

(a) Since (29) holds, we can write

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - B(x_k;\mu_k) &= (B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k)) + (B(x_{k+1};\mu_k) - B(x_k;\mu_k)) \\ &\le -L(\mu_{k+1} - \mu_k) + (B(x_{k+1};\mu_k) - B(x_k;\mu_k)), \end{aligned}$$

which together with (29), (26) and Lemma 5 implies

$$\begin{aligned} \underline{B} &\le \lim_{k\to\infty} B(x_k;\mu_k) \le B(x_1;\mu_1) - L\sum_{k=1}^{\infty}(\mu_{k+1} - \mu_k) + \sum_{k=1}^{\infty}(B(x_{k+1};\mu_k) - B(x_k;\mu_k)) \\ &\le B(x_1;\mu_1) + L\overline{\mu} + \sum_{\rho(d_k)\ge\underline{\rho}} (B(x_{k+1};\mu_k) - B(x_k;\mu_k)) \\ &\le B(x_1;\mu_1) + L\overline{\mu} - \underline{C} \sum_{\rho(d_k)\ge\underline{\rho}} \mu_k \|\tilde{g}_k\|^2, \end{aligned}$$

where

$$\underline{B} = \min_{0\le\mu\le\overline{\mu}} 2m\mu(1 - \log(2\mu)) \ge 2m\overline{\mu}(1 - \log(2\overline{\mu}))$$

(see (28)) and $\|\tilde{g}_k\| = \min_{1\le j\le k} \|g(x_j;\mu_j)\|$. If $\liminf_{k\to\infty} \|g(x_k;\mu_k)\| = 0$ was not satisfied, then a number $\varepsilon > 0$ would exist such that

$$\|\tilde{g}_k\| \ge \varepsilon \quad \forall k \in N. \tag{30}$$

Then, using the last proved inequality, we obtain

$$B(x_1;\mu_1) + L\overline{\mu} - \underline{B} \ge \underline{C}\varepsilon^2 \sum_{\rho(d_k)\ge\underline{\rho}} \mu_k \ge \underline{C}\varepsilon^2 \sum_{k\in K} \mu_k \ge \frac{\underline{C}\varepsilon^2}{\tau} \sum_{k\in K} \|g(x_k;\mu_k)\|^2,$$

where K is the infinite set of indices defined in Lemma 5. Thus $\|g(x_k;\mu_k)\| \xrightarrow{K} 0$ which is in a contradiction with (30).

(b) Using (9) and (10), one has $z_i^2(x_k;\mu_k) - f_i^2(x_k) \le 2\mu_k(2\mu_k + \overline{F})$. Thus $z_i^2(x_k;\mu_k) - f_i^2(x_k) \to 0$ as $\mu_k \to 0$. □

Corollary 1. *Let the assumptions of Theorem 1 be satisfied and the sequence* $\{x_k\}_1^\infty$ *be bounded (either* $\mathcal{L}(\overline{F})$ *in Assumption 1 is compact or Assumption 2 holds). Then there is a cluster point* $x \in R^n$ *of the sequence* $\{x_k\}_1^\infty$ *that satisfies KKT conditions (3), where* $u \in R^m$ *is a cluster point of the sequence* $\{u(x_k; \mu_k)\}_1^\infty$.

Now we return to the case when $\underline{\varepsilon} > 0$ and $\underline{\mu} > 0$. The following lemma specifies a precision with which the KKT conditions are satisfied if Algorithm 1 terminates.

Theorem 2 *Let Assumption 3 be satisfied. If Algorithm 1 terminates in the k-th iteration, then*

$$\left\| \sum_{i=1}^{m} u_i(x_k; \mu_k) g_i(x_k) \right\| \leq \underline{\varepsilon},$$

and

$$z_i(x_k; \mu_k) - |f_i(x_k)| \leq 2\underline{\mu}, \quad |u_i(x_k; \mu_k)| \leq 1,$$

$$\left| u_i(x_k; \mu_k) - \frac{f_i(x_k)}{|f_i(x_k)|} \right| \leq \frac{2\underline{\mu}}{|f_i(x_k)|}$$

for $1 \leq i \leq m$.

Proof. The first inequality is a consequence of the termination condition (Step 2 of Algorithm 1). The second one follows from (10), since

$$z_i(x_k; \mu_k) = \mu_k + \sqrt{\mu_k^2 + f_i^2(x_k)} \leq 2\underline{\mu} + |f_i(x_k)|$$

($\mu_k = \underline{\mu}$ if Algorithm 1 terminates). The third inequality follows immediately from (11) and the last inequality is a consequence of the second one, since (11) implies

$$\begin{aligned} \left| u_i(x_k; \mu_k) - \frac{f_i(x_k)}{|f_i(x_k)|} \right| &= \left| \frac{f_i(x_k)}{z_i(x_k; \mu_k)} - \frac{f_i(x_k)}{|f_i(x_k)|} \right| = \frac{z_i(x_k; \mu_k) - |f_i(x_k)|}{z_i(x_k; \mu_k)} \\ &\leq \frac{2\underline{\mu}}{z_i(x_k; \mu_k)} \leq \frac{2\underline{\mu}}{|f_i(x_k)|}. \end{aligned}$$

□

Theorem 2 demonstrates that the first three KKT conditions can be satisfied with a prescribed precision and indicates that the last one could be inaccurate if $|f_i(x_k)|$ is small. This indication was confirmed by our computational experiments. Using $\underline{\mu} = 10^{-8}$, we obtained $1 - |u_i| \approx 10^{-8}$ for $|f_i| \approx 1$ and $1 - |u_i| \approx 10^{-2}$ for $|f_i| \approx 10^{-6}$.

5 Computational experiments

The primal interior-point method was tested by using the two collections of 22 relatively difficult problems with an optional dimension chosen from [16], which can be downloaded (together with the above report) from `www.cs.cas.cz/~luksan/test.html` as

Test 14 and Test 15. The functions $f_i(x)$, $1 \leq i \leq m$, given in [16] serve for defining the objective function

$$F(x) = \sum_{1 \leq i \leq m} |f_i(x)|. \tag{31}$$

We have used the parameters $\underline{\varepsilon} = 10^{-6}$, $\underline{\mu} = 10^{-8}$, $\underline{\delta} = 0.9$, $\overline{\delta} = 1.1$, $\overline{\Delta} = 1000$, $\underline{\rho} = 10^{-4}$, $\overline{\rho}_1 = 0.1$, $\overline{\rho}_2 = 0.9$, $\underline{\beta} = 0.1$, $\overline{\beta} = 0.5$, $\overline{\gamma} = 2.0$, $\tau = 0.01$ in Algorithm 1 as defaults (the step bound $\overline{\Delta}$ was sometimes tuned).

The first set of the tests concerns a comparison of the primal interior point method (using various trust-region and line-search strategies) with several other methods for sparse minimax optimization. Medium-size test problems with 200 variables were used. The results of computational experiments are reported in two tables, where only summary results (over all 22 test problems) are given. Here `Method` is the method used: `T1` the primal interior-point method with Algorithm A and the Gill-Murray decomposition, `T2` the primal interior-point method with Algorithm A and the Bunch-Parlett decomposition, `T3` the primal interior-point method with Algorithm B and the Gill-Murray decomposition, `LS` the primal interior-point method with line search and restarts described in [15], `IP` the primal-dual interior point method described in [**?**], `BV` the bundle variable metric method described in [17]. All these methods were realized in two modifications: NM denotes the discrete Newton method with the Hessian matrix computed using the differences by the way described in [3] and VM denotes the variable metric method with the partitioned updates described in [11]. At the same time `NIT` is the total number of iterations, `NFV` is the total number of function evaluations, `NFG` is the total number of gradient evaluations, `NR` is the total number of restarts, `NL` is the number of problems, for which the lowest known local minimum was not found (even if parameters $\underline{\mu}$ and $\overline{\Delta}$ were tuned), `NF` is the number of problems, for which no local minimum was found (either a premature termination occurred or the number of function evaluations exceeded the upper bound), `NT` is the number of problems for which parameters were tuned and `Time` is the total computational time in seconds.

M	NIT	NFV	NFG	NR	NL	NF	NT	TIME
T1-NM	2784	3329	23741	1	-	-	4	3.70
T2-NM	2392	2755	19912	2	-	1	8	3.19
T3-NM	3655	4161	32421	4	1	1	7	6.52
LS-NM	5093	12659	30350	1	1	-	6	4.58
DI-NM	4565	6301	37310	212	2	-	12	30.63
T1-VM	2784	3329	23741	1	-	-	4	3.70
T2-VM	2392	2755	19912	2	-	1	8	3.19
T3-VM	3655	4161	32421	4	1	1	7	6.52
LS-VM	5093	12659	30350	1	1	-	6	4.58
DI-VM	4565	6301	37310	212	2	-	12	30.63
BV-VM	34079	34111	34111	22	1	1	11	25.72

Table 1: Test 14 – 22 problems with 200 variables

M	NIT	NFV	NFG	NR	NL	NF	NT	TIME
T1-NM	3331	4213	18989	17	-	-	6	3.74
T2-NM	3170	4027	17452	17	-	1	12	3.68
T3-NM	5424	6503	31722	11	1	1	10	7.83
LS-NM	8183	20245	52200	36	2	-	9	10.90
DI-NM	4565	6301	37310	212	2	-	12	30.63
T1-VM	2784	3329	23741	1	-	-	4	3.70
T2-VM	2392	2755	19912	2	-	1	8	3.19
T3-VM	3655	4161	32421	4	1	1	7	6.52
LS-VM	5093	12659	30350	1	1	-	6	4.58
DI-VM	4565	6301	37310	212	2	-	12	30.63
BV-NM	34499	34745	34745	22	1	-	11	13.14

Table 2: Test 15 – 22 problems with 200 variables

The results introduced in the above tables indicate that the trust-region strategies are more efficient than the restarted line-search strategies in connection with the interior-point method for l_1 optimization. These observations differ from conclusions concerning the interior-point method for minimax optimization proposed in [15], where the matrix $\nabla^2 B$ has a different structure. The trust-region interior-point method T1 is less sensitive to the choice of parameters and requires a lower number of iterations and a shorter computational time in comparison with the bundle variable metric method B proposed in [17]. Method T1 also finds the best known local minimum (if l_1 problems have several local solutions) more frequently (see the column NL in the above tables).

The second set of the tests concerns a comparison of the interior-point method, realized as the dog-leg method with the Gill-Murray decomposition, with the bundle variable metric method described in [17]. Large-scale test problems with 1000 variables are used. The results of computational experiments are given in Tables 3 and 4, 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 the summary results including the total computational time in seconds. The bundle variable metric method was chosen for the comparison, since it is based on a quite different principle and can also be used for the large sparse l_1 optimization.

	Trust-region interior-point method				Bundle variable metric method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	1594	1598	6380	0.166502E-09	7819	7842	7842	0.174023E-20
2	415	516	2912	0.106432E-08	127	130	130	0.735523E-17
3	32	33	231	0.604855E-07	89	89	89	0.359364E-14
4	27	39	196	269.499	81	81	81	269.499
5	30	31	186	0.107950E-06	39	39	39	0.122456E-14
6	32	33	462	0.611870E-07	100	100	100	0.110358E-12
7	18	20	171	336.937	211	211	211	336.937
8	18	19	342	761774.	36	39	39	761774.
9	212	259	3834	327.680	6181	6181	6181	327.682
10	970	1176	17460	0.386416E-01	14369	14369	14369	0.740271E-01
11	82	90	498	10.7765	319	319	319	10.7765
12	35	36	144	982.273	115	117	117	982.273
13	27	28	112	0.277182E-06	16	17	17	0.139178E-18
14	1	12	6	0.129382E-08	3	3	3	0.129382E-08
15	202	246	812	1.96106	3948	3957	3957	1.97013
16	161	169	972	0.435729E-15	4505	4556	4556	0.475529E-03
17	484	564	2910	0.165706E-11	441	443	443	0.857271E-06
18	2093	2538	12564	0.105340E-05	1206	1216	1216	0.129694E-03
19	15	16	96	59.5986	182	182	182	59.5986
20	1226	1529	7362	0.154869E-11	7828	7830	7830	0.102202E-04
21	21	22	132	2.13866	29	30	30	2.13866
22	1423	1770	8544	1.00000	337	341	341	1.00000
Σ	9118	10774	66332	TIME=42.56	47981	48092	48092	TIME=155.67

Table 3: Test 14 – 22 problems with 1000 variables

	Trust-region interior-point method				Bundle variable metric method			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	1464	1477	5860	0.123345E-12	359	540	540	0.815757E-08
2	121	181	605	4.00000	453	473	473	0.153343E-07
3	27	31	168	0.775716E-09	114	114	114	0.374913E-08
4	65	76	264	648.232	53	54	54	648.232
5	6	7	42	0.655031E-14	285	285	285	0.422724E-05
6	8	9	126	0.754396E-13	560	560	560	0.649530E-08
7	73	111	296	12029.9	542	650	650	12029.9
8	83	100	252	0.230723E-06	939	942	942	0.380433E-03
9	532	609	3731	2777.75	4428	4429	4429	2780.11
10	103	148	618	658.048	1389	1389	1389	658.048
11	3452	3674	13812	0.821565E-14	411	454	454	0.838373E-09
12	652	773	3918	3117.36	1879	1882	1882	3125.85
13	165	212	996	14808.8	727	728	728	14808.8
14	162	201	1134	566.112	514	514	514	566.112
15	67	93	476	181.926	654	654	654	181.926
16	268	328	1883	66.5333	1376	1376	1376	66.5333
17	122	147	1107	0.146536E-13	9092	9092	9092	0.337978E-08
18	78	89	474	0.619504E-13	3160	3160	3160	0.754900
19	29	31	330	0.382360E-12	15933	15944	15944	0.239244E-08
20	69	86	420	0.131734E-10	1509	1699	1699	0.756975E-08
21	118	195	708	1326.92	425	426	426	1327.95
22	80	112	486	2993.36	9875	9875	9875	2993.37
Σ	7744	8690	37706	TIME=30.03	54677	55240	55240	TIME=155.90

Table 4: Test 15 – 22 problems with 1000 variables

The results introduced in Tables 3 and 4 confirm conclusions following from the previous tables. The trust-region interior-point method seems to be more efficient than the bundle variable metric method in all indicators. Especially, the computational time is much shorter and also the number of the best known local minima attained is greater in the case of the trust-region interior-point method. We believe that the efficiency of the interior-point method could be improved by using a better procedure for the barrier parameter update.

References

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

[2] T.F.Coleman, B.S.Garbow, J.J.Moré: Software for estimating sparse Hessian matrices. ACM Transactions on Mathematical Software 11 (1985) 363-377.

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

[4] Conn A.R., Gould N.I.M., and Toint P.L., 2000, *Trust-region methods* (SIAM).

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

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

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

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

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

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

[11] A.Griewank, P.L.Toint: Partitioned variable metric updates for large-scale structured optimization problems. Numerical Mathematics 39 (1982) 119-137.

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

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

[14] Lukšan L., Matonoha C., and Vlček J., 2004, A shifted Steihaug-Toint method for computing a trust-region step. Technical Report V-914. Prague, ICS AS CR.

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

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

[17] Lukšan L. and Vlček J., 2006, Variable metric method for minimization of partially separable nonsmooth functions. *Pacific Journal on Optimization,* **2,**9-70.

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

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

[20] Powell M.J.D., 1983, In: G.K. Chui, L.L. Schumaker, and J.D. Ward (Eds.) *General Algorithms for Discrete Nonlinear Approximation Calculations* (Approximation Theory IV, New York: Academic Press).

[21] Powell M.J.D., 1984, On the global convergence of trust region algorithms for unconstrained optimization. *Mathematical Programming,* **29,** 297-303.

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

[23] Steihaug T., 1983, The conjugate gradient method and trust regions in large-scale optimization. *SIAM Journal on Numerical Analysis,* **20,** 626-637.

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

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

[26] Yuan Y., 1985, On the Superlinear Convergence of a Trust Region Algorithm for Nonsmooth Optimization. *Mathematical Programming,* **31,** 269-285.