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

L. Lukšan, C. Matonoha, J. Vlček

Technical report No. 942

November 2005
Trust-region interior-point method for large sparse $l_1$ optimization

L. Lukšan, C. Matonoha, J. Vlček

Technical report No. 942

November 2005

Abstract:

In this paper, 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 nonconvex problems can be solved successfully. The results of computational experiments given in this paper 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, Hálková 6, 461 17 Liberec.
1 Introduction

Consider the $l_1$ optimization problem: Minimize function

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

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $0 \leq i \leq m$ ($m$ is usually large), are smooth functions depending on a small number of variables ($n_i$, say) satisfying either Assumption 1 or Assumption 2.

**Assumption 1.** Functions $f_i(x)$, $1 \leq i \leq m$, are twice continuously differentiable on the convex hull of level set $\mathcal{L}(\mathcal{F}) = \{x \in \mathbb{R}^n : F(x) \leq \mathcal{F}\}$ for a sufficiently large upper bound $\mathcal{F}$ and they have bounded the first and second-order derivatives on $\text{conv} \mathcal{L}(\mathcal{F})$, i.e., constants $\underline{g}$ and $\overline{G}$ exist such that $\|\nabla f_i(x)\| \leq \underline{g}$ and $\|\nabla^2 f_i(x)\| \leq \overline{G}$ for all $1 \leq i \leq m$ and $x \in \text{conv} \mathcal{L}(\mathcal{F})$.

**Assumption 2** Functions $f_i(x)$, $1 \leq i \leq m$, 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 $\mathcal{F}$, $\underline{g}$ and $\overline{G}$ exist such that $f_i(x) \leq \mathcal{F}$, $\|g_i(x)\| \leq \underline{g}$ and $\|G_i(x)\| \leq \overline{G}$ for all $x \in \mathcal{D}$. The choice of $\mathcal{F}$ and $\mathcal{D}$ will be discussed later (see Assumption 3). Note that set $\text{conv} \mathcal{L}(\mathcal{F})$ used in Assumption 1 need not be compact.

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

$$\text{minimize } \sum_{i=1}^{m} z_i \text{ subject to } -z_i \leq f_i(x) \leq z_i, \hspace{0.5cm} 1 \leq i \leq m.$$ \hspace{1cm} (2)

Problem (2) can be solved by an arbitrary nonlinear programming method utilizing sparsity (sequential linear programming [7], sequential quadratic programming [10], interior-point [1], [11], [24] and nonsmooth equation [12]). In this paper, we introduce a trust-region interior-point method that utilizes a special structure of the $l_1$ problem (1). The constrained problem (2) is replaced by a sequence of unconstrained problems

$$\text{minimize } B(x, z; \mu) = \sum_{i=1}^{m} z_i - \mu \sum_{i=1}^{m} \log(z_i - f_i(x)) - \mu \sum_{i=1}^{m} \log(z_i + f_i(x))$$

$$= \sum_{i=1}^{m} z_i - \mu \sum_{i=1}^{m} \log(z_i^2 - f_i^2(x))$$ \hspace{1cm} (3)

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

Barrier function (3) remains unchanged if we replace problem (2) by equivalent problem

$$\text{minimize } \sum_{i=1}^{m} z_i \text{ subject to } f_i^2(x) \leq z_i^2, \hspace{0.5cm} 1 \leq i \leq m.$$ \hspace{1cm} (4)
The necessary first-order (KKT) conditions for the solution of (4) have the form

$$\sum_{i=1}^{m} 2w_i f_i(x) \nabla f_i(x) = 0, \quad 2w_iz_i = 1, \quad w_i \geq 0, \quad w_i(z_i^2 - f_i^2(x)) = 0, \quad 1 \leq i \leq m, \quad (5)$$

where $w_i$, $1 \leq i \leq m$, are Lagrange multipliers. Since $z_i = |f_i(x)|$, $1 \leq i \leq m$, at the solution of (4), we can write (5) in the simpler equivalent form

$$\sum_{i=1}^{m} u_i \nabla f_i(x) = 0, \quad \frac{u_i z_i}{f_i(x)} = 1, \quad z_i^2 - f_i^2(x) = 0, \quad 1 \leq i \leq m, \quad (6)$$

where $u_i = 2w_i f_i(x)$ for $1 \leq i \leq m$.

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

**Assumption 3** Either Assumption 1 holds and $\{x_k\}_1^\infty \in \mathcal{L}(F)$ or 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 $l_1$ 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 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 (3), we obtain necessary conditions for minimum in the form

$$\sum_{i=1}^{m} \frac{2\mu f_i(x)}{z_i^2 - f_i^2(x)} \nabla f_i(x) \Delta \sum_{i=1}^{m} u_i(x, z_i; \mu) \nabla f_i(x) = 0 \quad (7)$$

and

$$1 - \frac{2\mu z_i}{z_i^2 - f_i^2(x)} = 1 - u_i(x, z_i; \mu) \frac{z_i}{f_i(x)} = 0, \quad 1 \leq i \leq m. \quad (8)$$
Denoting \( g_i(x) = \nabla f_i(x) \), \( 1 \leq i \leq m \), \( A(x) = [g_1(x), \ldots, g_m(x)] \),

\[
    f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix}, \quad z = \begin{bmatrix} z_1 \\ \vdots \\ z_m \end{bmatrix}, \quad u(x, z; \mu) = \begin{bmatrix} u_1(x, z_1; \mu) \\ \vdots \\ u_m(x, z_m; \mu) \end{bmatrix}
\]

and \( Z = \text{diag}(z_1, \ldots, z_m) \), we can write (7)–(8) in the form

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

The system of \( n + m \) nonlinear equations (10) 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 \( \Delta x \) and \( \Delta z \) 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 special nonlinear programming problem (2), the structure of \( B(x, z; \mu) \) allows us to obtain minimizer \( z(x; \mu) \in R \) of function \( B(x, z; \mu) \) for a given \( x \in R^n \).

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

\[
    \frac{2 \mu z_i(x; \mu)}{z_i^2(x; \mu) - f_i^2(x)} = 1 \quad \text{or} \quad \frac{z_i^2(x; \mu) - f_i^2(x)}{2 \mu z_i(x; \mu)} = 2, \quad 1 \leq i \leq m,
\]

which have solutions

\[
    z_i(x; \mu) = \mu + \sqrt{\mu^2 + f_i^2(x)}, \quad 1 \leq i \leq m.
\]

**Proof.** Function \( B(x, z; \mu) \) (with \( x \) fixed) is convex for \( z_i > |f_i(x)| \), \( 1 \leq i \leq m \), 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 (8)), we obtain quadratic equations (11), which define its unique stationary point. \( \square \)

Assuming \( z = z(x; \mu) \), we denote

\[
    B(x; \mu) = \sum_{i=1}^{m} z_i(x; \mu) - \mu \sum_{i=1}^{m} \log(z_i^2(x; \mu) - f_i^2(x))
\]

and \( u(x; \mu) = u(x, z(x; \mu); \mu) \). In this case, barrier function \( B(x; \mu) \) depends only on \( x \). In order to obtain 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 barrier function (13). Then
\[ \nabla B(x; \mu) = A(x)u(x; \mu) \] (14)
and
\[ \nabla^2 B(x; \mu) = G(x; \mu) + A(x)V(x; \mu)A^T(x), \] (15)
where
\[ G(x; \mu) = \sum_{i=1}^{m} u_i(x; \mu)g_i(x) \] (16)
with \( G_i(x) = \nabla^2 f_i(x), \) 1 \( \leq \) \( i \) \( \leq \) \( m \), and \( V(x; \mu) = \text{diag}(v_1(x; \mu), \ldots, v_m(x; \mu)) \) with
\[ v_i(x; \mu) = \frac{2\mu}{z_i^2(x; \mu) + f_i^2(x)}, \quad 1 \leq i \leq m. \] (17)

Proof. Differentiating (13), we obtain
\[ \begin{align*}
\nabla B(x; \mu) &= \sum_{i=1}^{m} \nabla z_i(x; \mu) - 2\mu \sum_{i=1}^{m} \frac{z_i(x; \mu) \nabla z_i(x; \mu) - f_i(x)g_i(x)}{z_i^2(x; \mu) - f_i^2(x)} \\
&= \sum_{i=1}^{m} \left(1 - \frac{2\mu z_i(x; \mu)}{z_i^2(x; \mu) - f_i^2(x)} \right) \nabla z_i(x; \mu) + \sum_{i=1}^{m} \frac{2\mu f_i(x)g_i(x)}{z_i^2(x; \mu) - f_i^2(x)} \\
&= \sum_{i=1}^{m} u_i(x; \mu)g_i(x) = A(x)u(x; \mu)
\end{align*} \]
by (11) and (7). Differentiating (11), 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 \( \leq \) \( i \) \( \leq \) \( m \), which gives
\[ \nabla z_i(x; \mu) = \frac{2z_i(x; \mu)f_i(x)g_i(x)}{z_i^2(x; \mu) + f_i^2(x)} \] (18)
for 1 \( \leq \) \( i \) \( \leq \) \( m \) after arrangements. Thus
\[ \begin{align*}
\nabla u_i(x; \mu) &= \nabla \left( \frac{f_i(x)}{z_i(x; \mu)} \right) = \frac{z_i(x; \mu)g_i(x) - f_i(x)\nabla z_i(x; \mu)}{z_i^2(x; \mu)} \\
&= \left(1 - \frac{2f_i^2(x)}{z_i^2(x; \mu) + f_i^2(x)} \right) \frac{g_i(x)}{z_i(x; \mu)} \\
&= \frac{z_i^2(x; \mu) - f_i^2(x)}{z_i^2(x; \mu) + f_i^2(x)} \frac{g_i(x)}{z_i(x; \mu)} \\
&= \frac{2\mu}{z_i^2(x; \mu) + f_i^2(x)} g_i(x) = v_i(x; \mu)g_i(x)
\end{align*} \]
by (8), (18), (11) and (17). Differentiating (14) and using the previous expression, we obtain

\[ \nabla^2 B(x; \mu) = \nabla \sum_{i=1}^{m} u_i(x; \mu) g_i(x) \]

\[ = \sum_{i=1}^{m} u_i(x; \mu) G_i(x) + \sum_{i=1}^{m} \nabla u_i(x; \mu) g_i^T(x) \]

\[ = \sum_{i=1}^{m} u_i(x; \mu) G_i(x) + \sum_{i=1}^{m} v_i(x; \mu) g_i(x) g_i^T(x), \]

which is equation (15).

Lemma 3. Let vector \( d \in \mathbb{R}^n \) solve equation

\[ \nabla^2 B(x; \mu) d = -g(x; \mu), \] (19)

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. Equation (19) 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 (17). Thus \( d^T g(x; \mu) < 0 \) if \( G(x; \mu) \) is positive definite.

Expression (17) 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 \( \mu \) then \( v_i(x; \mu) \) can tend to infinity and \( \nabla^2 B(x; \mu) \) can be ill-conditioned (see (15)). The following lemma gives the upper bound for \( \| \nabla^2 B(x; \mu) \| \).

Lemma 4. If Assumption 3 holds, then

\[ \| \nabla^2 B(x; \mu) \| \leq m(G + \bar{g}^2 \| V(x; \mu) \|) \leq m(G + \bar{g}^2/2\mu). \]

Proof. Using (15) and Assumption 3, we obtain

\[ \| \nabla^2 B(x; \mu) \| \leq \| G(x; \mu) + A(x) V(x; \mu) A^T(x) \| \]

\[ \leq \left\| \sum_{i=1}^{m} u_i(x; \mu) G_i(x) \right\| + \left\| \sum_{i=1}^{m} v_i(x; \mu) g_i(x) g_i^T(x) \right\| \]

\[ \leq mG + m\bar{g}^2 \| V(x; \mu) \|, \]

since inequalities \( z_i(x; \mu) \geq |f_i(x)| \) and (8) imply that \( |u_i| = |f_i(x)|/z_i(x; \mu) \leq 1 \) for all \( 1 \leq i \leq m \). Since \( V_\mu(x) \) is diagonal, one has

\[ \| V(x; \mu) \| = \max_{1 \leq i \leq m} v_i = \max_{1 \leq i \leq m} \left( \frac{2\mu}{z_i^2(x; \mu) + f_i^2(x)} \right), \] (20)
by (17). Using (12), we can write
\[
    r^2_i(x; \mu) + f^2_i(x) = \left( \mu + \sqrt{\mu^2 + f^2_i(x)} \right)^2 + f^2_i(x)
\]
\[
    = 2 \left( \mu^2 + \mu \sqrt{\mu^2 + f^2_i(x)} + f^2_i(x) \right) \geq 4 \mu^2
\]
for all \(1 \leq i \leq m\), which together with (20) proves the lemma. \(\square\)

Vector \(d \in \mathbb{R}^n\) obtained by solving (19) 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

\[
    \min_{d} Q(d) = \frac{1}{2} d^T \nabla^2 B(x; \mu) d + g^T(x; \mu) d \quad \text{subject to} \quad \|d\| \leq \Delta,
\]

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

\[
    \rho(d) = \frac{B(x + d; \mu) - B(x; \mu)}{Q(d)},
\]

we set

\[
    x^+ = x \quad \text{if} \quad \rho(d) \leq 0, \quad \text{or} \quad x^+ = x + d \quad \text{if} \quad \rho(d) > 0.
\]

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

\[
    \Delta^+ = \beta \Delta \quad \text{if} \quad \rho(d) < \underline{\rho},
\]
\[
    \Delta^+ = \Delta \quad \text{if} \quad \underline{\rho} \leq \rho(d) \leq \overline{\rho},
\]
\[
    \Delta^+ = \beta \Delta \quad \text{if} \quad \rho(d) < \overline{\rho},
\]

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

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

**Algorithm 1.**

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

**Input:** Sparsity pattern of matrix \(A\). Initial estimation of vector \(x\).

**Step 1:** *Initiation.* Choose initial barrier parameter \(\mu > 0\) and initial trust-region radius \(0 < \Delta \leq \Delta\). Determine the sparsity pattern of matrix \(\nabla^2 B\) from the sparsity pattern of matrix \(A\). Carry out symbolic decomposition of \(\nabla^2 B\). Compute values \(f_i(x)\), \(1 \leq i \leq m\), and \(F(x) = \sum_{1 \leq i \leq m} |f_i(x)|\). Set \(k := 0\) (iteration count).
Step 2: **Termination.** Determine vector \( z(x; \mu) \) by (12) and vector \( u(x; \mu) \) by (10). Compute matrix \( A(x) \) and vector \( g(x; \mu) = A(x)u(x; \mu) \). If \( \mu \leq \mu_{\text{max}} \) and \( \| g(x; \mu) \| \leq \varepsilon \), 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 \( A(x + \delta v)u(x; \mu) - g(x; \mu) \) 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 (21).

Step 5: **Step-length selection.** Determine step-length \( \alpha \) by (23) and set \( x := x + \alpha d \). Compute 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 new trust-region radius \( \Delta \) by (24) and set \( \Delta := \min(\Delta, \bar{\Delta}) \).

Step 7: **Barrier parameter update.** If \( \rho(d) \geq \rho \) (where \( \rho(d) \) is given by (22)), determine a new value of barrier parameter \( \mu \geq \mu_{\text{max}} \) (not greater than the current one) by the procedure described in Section 3. Go to Step 2.

The use of the maximum step-length \( \bar{\Delta} \) 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 \( \bar{\Delta} \) is a parameter, which is most frequently tuned.

The important part of Algorithm 1 is the update of barrier parameter \( \mu \). 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 \( d \in \mathbb{R}^n \) should be a solution of the quadratic subproblem (21). 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 two approaches based on direct decompositions of matrix \( \nabla^2 B \) (we omit arguments \( x \) and \( \mu \) in the subsequent considerations).

The first strategy, 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 is computed by using either sparse Gill-Murray decomposition [8] or sparse Bunch-Parlett decomposition [5]. 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 $\nabla^2 B = PLML^TP^T$, where $P$ is a permutation matrix, $L$ is a lower triangular matrix and $M$ is a block-diagonal matrix with $1 \times 1$ or $2 \times 2$ 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 $g^T\nabla^2 Bg \leq 0$, set $s := -(\Delta/\|g\|)g$ and terminate the computation.

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

**Step 4:** If $(d_N - d_C)^Td_C \geq 0$ and $\|d_N\| > \Delta$, determine number $\theta$ in such a way that $d_C^Td_C/d_N^Td_N \leq \theta \leq 1$, 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 $(d_N - d_C)^Td_C < 0$, 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 (21) by using the Newton method applied to the nonlinear equation

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

where $(\nabla^2 B + \lambda I)d(\lambda) = -g$. This way, described in [18] in more details, follows from the KKT conditions for (21). Since the Newton method applied to (25) can be unstable, safeguards (lower and upper bounds to $\lambda$) are usually used. The following algorithm is a typical implementation of the optimum step method.
Algorithm B: Data $0 < \tilde{g} < 1 < \delta$ (usually $\tilde{g} = 0.9$ and $\delta = 1.1$), $\Delta > 0$.

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

Set $l = 0$ (inner iteration count).

**Step 2:** If $l > 0$ and $\lambda \leq \nu$, set $\lambda := \sqrt{\Delta \bar{\lambda}}$.

**Step 3:** Determine 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 vector $v \in \mathbb{R}^n$ such that $\|v\| = 1$ and $v^T (\nabla^2 B + \lambda I) v \leq 0$, set $\nu := \lambda - v^T (\nabla^2 B + \lambda I) v$, $\bar{\lambda} := \max(\nu, \bar{\lambda})$, $l := l + 1$ and go to Step 2.

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

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

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

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

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

The above algorithms generate direction vectors such that

$$\|d\| \leq \delta \Delta,$$

$$\|d\| < \delta \Delta \Rightarrow \nabla^2 B d = -g,$$

$$-Q(d) \geq \frac{\sigma}{\|g\|} \min \left(\|d\|, \frac{\|g\|}{\|\nabla^2 B\|}\right),$$

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

$$\|d\| \geq \gamma \sqrt{\mathcal{B}},$$

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

$$B(x + d; \mu) - B(x; \mu) \leq \rho Q(d) \leq -\rho \sigma \frac{\gamma^2}{\mathcal{B}} \text{ if } \rho \geq \rho$$

(27)
A very important part of Algorithm 1 is the update of the barrier parameter $\mu$. There are two requirements, which play opposite roles. First $\mu \rightarrow 0$ should hold, since this is the main property of every interior point method. On the other hand, the convergence theory requires (27) to hold. Thus a lower bound $\mu$ for the barrier parameter has to be used (we recommend value $\mu = 10^{-6}$ 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

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

(28)

where $0 < \tau < 1$.

4 Global convergence

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

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

**Proof.** (a) First we prove that $B(x; \mu)$ is bounded from below if $\mu$ is fixed. Since $z_i(x; \mu) \geq 0$ and

$$z_i^2(x; \mu) - f_i^2(x) = 2\mu z_i(x; \mu) = 2\mu \left( \mu + \sqrt{\mu^2 + f_i^2(x)} \right) \leq 2\mu (2\mu + |f_i(x)|) \leq 2\mu (2\mu + \mathcal{F})$$

(29)

for all $1 \leq i \leq m$ by (11) and (12), we can write

$$B(x; \mu) = \sum_{i=1}^m z_i(x; \mu) - \mu \sum_{i=1}^m \log(z_i^2(x; \mu) - f_i^2(x)) \geq -\mu m \log \left( 2\mu (2\mu + \mathcal{F}) \right).$$

(30)

(b) Now we prove that the sequence of points in which $\mu_k$ is updated is infinite. If it was finite, an index $l \in N$ would exist such that $\mu_{k+1} = \mu_k = \mu_l \forall k \geq l$. Since function $B(x; \mu_l)$ is continuous, bounded from below by (a) and since (27) (with $\mu_k = \mu_l$) holds $\forall k \geq l$, it can be proved (see [20]) that $\liminf_{k \rightarrow \infty} \|g(x_k; \mu_l)\| = 0$. Thus an index $k \geq l$ exists such that $\|g(x_k; \mu_l)\|^2 \leq \tau \mu_l$ and, therefore, $\mu_{k+1} = \|g(x_k; \mu_l)\|^2 \leq \tau \mu_l < \mu_l$ by (28), which is a contradiction. Since the sequence of points where $\mu_{k+1} \leq \tau \mu_k$ is infinite, we can conclude that $\mu_k \rightarrow 0$. \qed

Now we will prove that

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

(31)
for some $L \in \mathbb{R}$. For this purpose, we consider that $z(x; \mu)$ and $B(x; \mu)$ are functions of $\mu$ and we write $z(x, \mu) = z(x; \mu)$ and $B(x, \mu) = B(x; \mu)$.

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

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

and

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

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

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

Differentiating 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

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

by (11). \hfill \Box

**Lemma 7.** Let Assumption 3 be satisfied. Then (31) holds with some $L \in \mathbb{R}$.

**Proof.** Using Lemma 6, the mean value theorem and (29), we can write

$$B(x_{k+1}, \mu_{k+1}) - B(x_k, \mu_k) = -\sum_{i=1}^{m} \log(z_i^2(x_{k+1}, \tilde{\mu}_k) - f_i^2(x_{k+1}))(\mu_{k+1} - \mu_k)
\quad \leq -\sum_{i=1}^{m} \log \left(2\tilde{\mu}_k(2\tilde{\mu}_k + F)\right) (\mu_{k+1} - \mu_k)
\quad \leq -\sum_{i=1}^{m} \log \left(2\mu_1(2\mu_1 + F)\right) (\mu_{k+1} - \mu_k)
\quad = -L(\mu_{k+1} - \mu_k),$$

where $0 < \mu_{k+1} \leq \tilde{\mu}_k \leq \mu_k \leq \mu_1$. \hfill \Box
Theorem 1. Let Assumption 3 be satisfied. Consider sequence \( \{x_k\}_\infty \), 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
\[
u_i(x_k; \mu_k) = \frac{f_i(x_k)}{z_i(x_k; \mu_k)}, \quad \lim_{k \to \infty} (z_i^2(x_k; \mu_k) - f_i^2(x_k)) = 0
\]
for \( 1 \leq i \leq m \).

Proof. Equalities \( u_i(x_k; \mu_k) = f_i(x_k)/z_i(x_k; \mu_k) \), \( 1 \leq i \leq m \), follow from (8).

(a) Since (31) holds, we can write
\[
B(x_{k+1}; \mu_{k+1}) - B(x_k; \mu_k) = (B(x_{k+1}; \mu_{k+1}) - B(x_{k+1}; \mu_k)) + (B(x_{k+1}; \mu_k) - B(x_k; \mu_k)) \leq -L(\mu_{k+1} - \mu_k) + (B(x_{k+1}; \mu_k) - B(x_k; \mu_k),
\]
which together with (31), (27) and Lemma 5 implies
\[
\| g(x_k; \mu_k) \| \leq \sum_{i=1}^{m} \left( \frac{\rho \sigma c}{B} \right) \gamma_k \geq 0, \quad \forall k \in N.
\]

Then, using the previous inequality, we would obtain
\[
\| g(x_k; \mu_k) \| \geq \epsilon \quad \forall k \in N.
\]

It remains to prove that the sum on the right hand side is infinite, which gives a contradiction. If this sum was finite, an index \( l \in N \) would exist such that \( \rho_k(d_k) < \rho \quad \forall k \geq l \). Thus \( \Delta_k \to 0 \) by (24), which with inequality \( \Delta_k \geq \| d_k \| \geq \epsilon \gamma_k/B \) (see (26)) gives \( \gamma_k \to 0 \). But this is in contradiction with (32).
(b) Using (29), one has 

\[ z_i^2(x_k; \mu_k) - f_i^2(x_k) \leq 2\mu_k(2\mu_k + \bar{F}) \]

Thus \( z_i^2(x_k; \mu_k) - f_i^2(x_k) \to 0 \) as \( \mu_k \to 0 \).

\[ \square \]

**Remark 1.** If we replace (23) by

\[ x^+ = x \quad \text{if} \quad \rho(d) < \underline{\rho}, \quad \text{or} \quad x^+ = x + d \quad \text{if} \quad \rho(d) \geq \underline{\rho} \]

in Algorithm 1, then \( \lim_{k \to \infty} \|g(x_k; \mu_k)\| = 0 \). A proof of this assertion can be found, e.g., in [3].

**Corollary 1.** Let assumptions of Theorem 1 and (33) hold. Then every cluster point \( x \in \mathbb{R}^n \) of sequence \( \{x_k\}_{k=1}^{\infty} \) satisfies KKT conditions (6), where \( u \in \mathbb{R}^m \) is a cluster point of sequence \( \{u(x_k; \mu_k)\}_{k=1}^{\infty} \).

5 Computational experiments

The primal interior-point method was tested by using the collection of relatively difficult problems with optional dimension chosen from [15], which can be downloaded (together with the above report) from www.cs.cas.cz/~luksan/test.html as Test 14 and Test 15. Functions \( f_i(x), 1 \leq i \leq m \), given in [15] serve for defining objective function

\[ F(x) = \sum_{1 \leq i \leq m} |f_i(x)| \tag{34} \]

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

The first set of the tests concerns comparison of interior-point methods with various trust-region and line-search strategies and the bundle variable metric method proposed in [17]. 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 M is the method used: T1 – Algorithm A with the Bunch-Parlett decomposition, T2 – Algorithm A with the Gill-Murray decomposition, T3 – Algorithm B with the Gill-Murray decomposition, L – line-search with restarts described in [14], B – bundle variable metric method described in [17]; 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, NF is the number of problems, for which the global minimizer was not found (either a worse local minimum was obtained or the method failed even if parameter \( \Sigma \) was tuned), NT is the number of problems for which parameter \( \Sigma \) was tuned and TIME is the total computational time in seconds.
The results introduced in these tables indicate that trust-region strategies are more efficient than 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 [14], where matrix $\nabla^2 B$ has a different structure. The trust-region interior-point method is less sensitive to the choice of parameters and requires a lower number of iterations and shorter computational time in comparison with the bundle variable metric method proposed in [17]. This method also finds the global minimum (if the $l_1$ problems has several local solutions) more frequently (see column NF in the above tables).

The second set of tests concerns a comparison of the interior-point method, realized as a 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 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. The bundle variable metric method was chosen for comparison, since it is based on a quite different principle and can also be used for large sparse $l_1$ optimization.

<table>
<thead>
<tr>
<th>M</th>
<th>NIT</th>
<th>NFV</th>
<th>NFG</th>
<th>NR</th>
<th>NF</th>
<th>NT</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>2263</td>
<td>2520</td>
<td>21133</td>
<td>-</td>
<td>1</td>
<td>10</td>
<td>4.14</td>
</tr>
<tr>
<td>T2</td>
<td>2411</td>
<td>2768</td>
<td>18417</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>2.67</td>
</tr>
<tr>
<td>T3</td>
<td>2441</td>
<td>2784</td>
<td>20143</td>
<td>2</td>
<td>-</td>
<td>6</td>
<td>4.02</td>
</tr>
<tr>
<td>L</td>
<td>11955</td>
<td>23625</td>
<td>62658</td>
<td>-</td>
<td>1</td>
<td>8</td>
<td>11.30</td>
</tr>
<tr>
<td>B</td>
<td>32608</td>
<td>32634</td>
<td>32634</td>
<td>22</td>
<td>2</td>
<td>11</td>
<td>24.72</td>
</tr>
</tbody>
</table>

Table 1: Test 14: Function (34) with 200 variables

<table>
<thead>
<tr>
<th>M</th>
<th>NIT</th>
<th>NFV</th>
<th>NFG</th>
<th>NR</th>
<th>NF</th>
<th>NT</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>3440</td>
<td>4050</td>
<td>19623</td>
<td>3</td>
<td>1</td>
<td>13</td>
<td>3.69</td>
</tr>
<tr>
<td>T2</td>
<td>2541</td>
<td>2986</td>
<td>13745</td>
<td>5</td>
<td>-</td>
<td>11</td>
<td>2.63</td>
</tr>
<tr>
<td>T3</td>
<td>4655</td>
<td>5477</td>
<td>27090</td>
<td>2</td>
<td>-</td>
<td>12</td>
<td>5.11</td>
</tr>
<tr>
<td>L</td>
<td>6886</td>
<td>17162</td>
<td>43339</td>
<td>28</td>
<td>1</td>
<td>16</td>
<td>12.17</td>
</tr>
<tr>
<td>B</td>
<td>34608</td>
<td>34854</td>
<td>34854</td>
<td>22</td>
<td>1</td>
<td>20</td>
<td>13.61</td>
</tr>
</tbody>
</table>

Table 2: Test 15: Function (34) with 200 variables

The results introduced in these tables indicate that trust-region strategies are more efficient than 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 [14], where matrix $\nabla^2 B$ has a different structure. The trust-region interior-point method is less sensitive to the choice of parameters and requires a lower number of iterations and shorter computational time in comparison with the bundle variable metric method proposed in [17]. This method also finds the global minimum (if the $l_1$ problems has several local solutions) more frequently (see column NF in the above tables).

The second set of tests concerns a comparison of the interior-point method, realized as a 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 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. The bundle variable metric method was chosen for comparison, since it is based on a quite different principle and can also be used for large sparse $l_1$ optimization.
<table>
<thead>
<tr>
<th>P</th>
<th>Trust-region interior-point method</th>
<th>Bundle variable metric method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIT</td>
<td>NFV</td>
</tr>
<tr>
<td>1</td>
<td>1591</td>
<td>1595</td>
</tr>
<tr>
<td>2</td>
<td>411</td>
<td>503</td>
</tr>
<tr>
<td>3</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>19</td>
</tr>
<tr>
<td>9</td>
<td>120</td>
<td>135</td>
</tr>
<tr>
<td>10</td>
<td>855</td>
<td>1021</td>
</tr>
<tr>
<td>11</td>
<td>33</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>131</td>
<td>132</td>
</tr>
<tr>
<td>13</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>138</td>
<td>172</td>
</tr>
<tr>
<td>16</td>
<td>163</td>
<td>200</td>
</tr>
<tr>
<td>17</td>
<td>174</td>
<td>200</td>
</tr>
<tr>
<td>18</td>
<td>2043</td>
<td>2466</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>20</td>
<td>1441</td>
<td>1810</td>
</tr>
<tr>
<td>21</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>1531</td>
<td>1908</td>
</tr>
</tbody>
</table>

| Σ  | 8836 | 10418 | 62068 | TIME=42.24 | 45865 | 45954 | 45954 | TIME=132.83 |

Table 3: Test 14: Function (34) with 1000 variables
<table>
<thead>
<tr>
<th>P</th>
<th>Trust-region interior-point method</th>
<th>Bundle variable metric method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NIT</td>
<td>NFV</td>
</tr>
<tr>
<td>1</td>
<td>1464</td>
<td>1477</td>
</tr>
<tr>
<td>2</td>
<td>83</td>
<td>108</td>
</tr>
<tr>
<td>3</td>
<td>26</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>66</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>86</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>83</td>
<td>102</td>
</tr>
<tr>
<td>9</td>
<td>280</td>
<td>336</td>
</tr>
<tr>
<td>10</td>
<td>67</td>
<td>85</td>
</tr>
<tr>
<td>11</td>
<td>3435</td>
<td>3644</td>
</tr>
<tr>
<td>12</td>
<td>463</td>
<td>553</td>
</tr>
<tr>
<td>13</td>
<td>85</td>
<td>112</td>
</tr>
<tr>
<td>14</td>
<td>103</td>
<td>123</td>
</tr>
<tr>
<td>15</td>
<td>59</td>
<td>83</td>
</tr>
<tr>
<td>16</td>
<td>96</td>
<td>127</td>
</tr>
<tr>
<td>17</td>
<td>52</td>
<td>64</td>
</tr>
<tr>
<td>18</td>
<td>1109</td>
<td>1126</td>
</tr>
<tr>
<td>19</td>
<td>29</td>
<td>31</td>
</tr>
<tr>
<td>20</td>
<td>59</td>
<td>73</td>
</tr>
<tr>
<td>21</td>
<td>94</td>
<td>125</td>
</tr>
<tr>
<td>22</td>
<td>133</td>
<td>167</td>
</tr>
<tr>
<td>Σ</td>
<td>7886</td>
<td>8578</td>
</tr>
</tbody>
</table>

Table 4: Test 15: Function (34) with 1000 variables

The results introduced in these tables 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 global 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


