Institute of Computer Science
Academy of Sciences of the Czech Republic

Primal interior-point method for large sparse minimax optimization

L.Lukšan, C. Matonoha, J. Vlček

Technical report No. 941

October 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 minimax optimization

L.Lukšan, C. Matonoha, J. Vlček [1]

Technical report No. 941

October 2005

Abstract:

In this report, we propose a primal interior-point method for large sparse minimax optimization. After a short introduction, where various barrier functions are discussed, the complete algorithm is introduced and important implementation details are given. We prove that this algorithm is globally convergent under standard mild assumptions. Thus the large sparse nonconvex minimax optimization problems can be solved successfully. The results of extensive 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 minimax problem: Minimize function

$$F(x) = \max_{1 \le i \le m} f_i(x), \tag{1}$$

where $f_i : R^n \to R$, $1 \le i \le m$, are smooth functions satisfying the following two assumptions:

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

Assumption 2. Functions $f_i(x)$, $1 \le i \le m$, are twice continuously differentiable on the convex hull of the level set $\mathcal{L}(\overline{F}) = \{x \in R^n : F(x) \le \overline{F}\}$ for a sufficiently large upper bound $\overline{F}$ and they have bounded the first and second-order derivatives on $\mathrm{conv}\mathcal{L}(\overline{F})$, i.e., 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 $x \in \mathrm{conv}\mathcal{L}(\overline{F})$.

In this report, we assume that problem (1) is partially separable, which means that functions $f_i(x)$, $1 \le i \le m$, depend on a small number of variables (n_i, say, with $n_i = \mathcal{O}(1)$, $1 \le i \le m$).

Minimization of F is equivalent to the nonlinear programming problem with $n+1$ variables $x \in R^n$, $z \in R$:

$$\text{minimize} \quad z \quad \text{subject to} \quad f_i(x) \le z, \quad 1 \le i \le m. \tag{2}$$

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

$$\sum_{i=1}^m u_i \nabla f_i(x) = 0, \quad \sum_{i=1}^m u_i = 1, \quad u_i \ge 0, \quad z - f_i(x) \ge 0, \quad u_i(z - f_i(x)) = 0, \tag{3}$$

where u_i, $1 \le i \le m$, are Lagrange multipliers. Problem (2) can be solved by an arbitrary nonlinear programming method utilizing sparsity (sequential linear programming [5], [11]; sequential quadratic programming [7], [10]; interior-point [14], [21]; nonsmooth equation [6], [15]). In this report, we introduce a feasible primal interior-point method that utilizes a special structure of the minimax problem (1). The constrained problem (2) is replaced by a sequence of unconstrained problems

$$\text{minimize} \quad B(x, z; \mu) = z + \mu \sum_{i=1}^m \varphi(z - f_i(x)), \tag{4}$$

where $\varphi : (0, \infty) \to R$ is a barrier function, $z > F(x)$ and $0 < \mu \le \overline{\mu}$ (we assume that $\mu \to 0$ monotonically). In connection with barrier functions, we will consider the following conditions.

Condition 1. $\varphi(t)$, $t \in (0, \infty)$, is a twice continuously differentiable function such that $\varphi(t)$ is decreasing, strictly convex, with $\lim_{t \to 0} \varphi(t) = \infty$ and $\varphi'(t)$ is increasing, strictly concave, with $\lim_{t \to \infty} \varphi'(t) = 0$.

Condition 2. $\varphi(t)$, $t \in (0, \infty)$, has a negative third-order derivative and $\varphi'(t)\varphi'''(t) > \varphi''(t)^2$ for $t > 0$.

Condition 3. $\varphi(t)$, $t \in (0, \infty)$, is a positive function.

Condition 1 is essential, we assume its validity for every barrier functions. Condition 2 serves for the estimation of norms of the Hessian matrices. Condition 3 is useful for investigation of the global convergence.

The most known and frequently used logarithmic barrier function

$$\varphi(t) = \log t^{-1} = -\log t, \tag{5}$$

satisfies Condition 1 and Condition 2, but does not satisfy Condition 3, since it is nonpositive for $t \geq 1$. Therefore, additional barrier functions have been studied. In [1], a truncated logarithmic barrier function is considered such that $\varphi(t)$ is given by (5) for $t \leq \tau$ and $\varphi(t) = a/t^2 + b/t + c$ for $t > \tau$, where a, b, c are chosen in such a way that $\varphi(t)$ is twice continuously differentiable in $(0, \infty)$, which implies that $a = -\tau^2/2$, $b = 2\tau$, $c = -\log\tau - 3/2$. Choosing $\tau = \overline{\tau}$ where $\overline{\tau} = \exp(-3/2)$ one has $\lim_{t\to\infty} \varphi(t) = c = 0$ and Condition 3 is satisfied. In this report we consider four particular barrier functions, which are introduced in Table 1 together with their derivatives.

	$\varphi(t)$	$\varphi'(t)$	$\varphi''(t)$	
B1	$-\log t$	$-1/t$	$1/t^2$	
B2	$-\log t$	$-1/t$	$1/t^2$	$t \leq \overline{\tau}$
	$2\overline{\tau}/t - \overline{\tau}^2/(2t^2)$	$-2\overline{\tau}/t^2 + \overline{\tau}^2/t^3$	$4\overline{\tau}/t^3 - 3\overline{\tau}^2/t^4$	$t > \overline{\tau}$
B3	$\log(t^{-1} + 1)$	$-1/(t(t+1))$	$(2t+1)/(t(t+1))^2$	
B4	$1/t$	$-1/t^2$	$2/t^3$	

Table 1: Barrier functions ($\overline{\tau} = \exp(-3/2)$).

All these barrier functions satisfy Condition 1, functions B1, B3, B4 satisfy Condition 2 and functions B2, B3, B4 satisfy Condition 3. Validity of Condition 2 can be easily proved by using Table 1. For example, considering B3 we can write

$$\varphi'''(t) = -2\frac{3t^2 + 3t + 1}{t^3(t+1)^3} < 0$$

and

$$\varphi'(t)\varphi'''(t) - \varphi''(t)^2 = 2\frac{3t^2 + 3t + 1}{t^4(t+1)^4} - \frac{4t^2 + 4t + 1}{t^4(t+1)^4} = \frac{2t^2 + 2t + 1}{t^4(t+1)^4} > 0$$

for $t \in (0, \infty)$. Note that our theory refers to all barrier functions satisfying Condition 1 (not only B1–B4).

A primal interior-point method investigated in this report is based on line search minimization of a special barrier function derived from the minimax problem structure. Approximation of the Hessian matrix of this barrier function is obtained by either gradient differences (Newton's method, NM) or partitioned variable metric updates (variable metric method, VM). A special restart scheme is developed to guarantee the convergence in the NM case. Furthermore, a great attention to the barrier parameter update is devoted. The resulting algorithm whose efficiency is confirmed by extensive computational experiments is described in detail.

The report is organized as follows. In Section 2, we introduce a primal interior-point method (i.e. interior point method that uses explicitly computed approximations of Lagrange multipliers instead of their updates) and describe the corresponding algorithm. Section 3 contains more details concerning this algorithm such as a restart strategy, numerical differentiation, variable metric updates, and a barrier parameter decrease. In Section 4 we study theoretical properties of the primal interior-point method and prove that this method is globally convergent if Assumption 1 and Assumption 2 hold. Section 5 contains a short description of a smoothing method SM described in [20] and [22] (and in other papers quoted therein), which is used for a comparison. Finally, in Section 6 we present results of computational experiments confirming the efficiency of the proposed method. Besides the SM method, we have used a primal-dual interior point method IP proposed in [14] and a nonsmooth equation method NE described in [15] for a comparison. The last two methods, intended for solving general nonlinear programming problems, were applied to the equivalent problem (2).

2 Description of the method

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

$$-\mu\sum_{i=1}^{m}\varphi'(z-f_i(x))\,\nabla f_i(x)=0, \quad 1+\mu\sum_{i=1}^{m}\varphi'(z-f_i(x))=0, \tag{6}$$

where $\varphi'(z-f_i(x))<0$ for all $1\le i\le m$. Denoting $g_i(x)=\nabla f_i(x)$, $1\le i\le m$, $A(x)=[g_1(x),\dots,g_m(x)]$ and

$$f(x)=\begin{bmatrix} f_1(x)\\ \dots\\ f_m(x)\end{bmatrix}, \quad u(x,z;\mu)=\begin{bmatrix} -\mu\,\varphi'(z-f_1(x))\\ \dots\\ -\mu\,\varphi'(z-f_m(x))\end{bmatrix}, \quad e=\begin{bmatrix} 1\\ \dots\\ 1\end{bmatrix}, \tag{7}$$

we can write (6) in the form

$$g(x,z;\mu)=A(x)\,u(x,z;\mu)=0, \quad \gamma(x,z;\mu)=1-e^Tu(x,z;\mu)=0. \tag{8}$$

These nonlinear equations can be solved by the Newton method. For this purpose, we need second-order derivatives of $B(x,z;\mu)$. One has

$$\begin{aligned}\frac{\partial A(x)u(x,z;\mu)}{\partial x} &= \sum_{i=1}^{m}u_i(x,z;\mu)G_i(x)+\mu\sum_{i=1}^{m}\varphi''(z-f_i(x))\,g_i(x)g_i^T(x)\\ &= G(x,z;\mu)+A(x)V(x,z;\mu)A^T(x),\end{aligned}$$

$$\begin{aligned}\frac{\partial A(x)u(x,z;\mu)}{\partial z} &= -\mu\sum_{i=1}^{m}\varphi''(z-f_i(x))\,g_i(x) = -A(x)V(x,z;\mu)e,\\ \frac{\partial(1-e^Tu(x,z;\mu))}{\partial x} &= -\mu\sum_{i=1}^{m}\varphi''(z-f_i(x))\,g_i^T(x) = -e^TV(x,z;\mu)A^T(x),\\ \frac{\partial(1-e^Tu(x,z;\mu))}{\partial z} &= \mu\sum_{i=1}^{m}\varphi''(z-f_i(x)) = e^TV(x,z;\mu)e,\end{aligned}$$

where $G_i(x) = \nabla^2 f_i(x)$, $1 \le i \le m$, $G(x,z;\mu) = \sum_{i=1}^m u_i(x,z;\mu)G_i(x)$, and

$$V(x,z;\mu) = \mu\,\mathrm{diag}(\varphi''(z-f_1(x)),\ldots,\varphi''(z-f_m(x))).$$

Using these expressions, we obtain a set of linear equations corresponding to a step of the Newton method

$$\begin{bmatrix} G(x,z;\mu)+A(x)V(x,z;\mu)A^T(x) & -A(x)V(x,z;\mu)e \\ -e^TV(x,z;\mu)A^T(x) & e^TV(x,z;\mu)e \end{bmatrix}\begin{bmatrix}\Delta x\\ \Delta z\end{bmatrix} = -\begin{bmatrix} g(x,z;\mu)\\ \gamma(x,z;\mu)\end{bmatrix} \tag{9}$$

or equivalently

$$\left(\begin{bmatrix} G(x,z;\mu) & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix} A(x)\\ -e^T\end{bmatrix} V(x,z;\mu)\,[\,A^T(x)\ \ -e\,]\right)\begin{bmatrix}\Delta x\\ \Delta z\end{bmatrix} = -\begin{bmatrix} g(x,z;\mu)\\ \gamma(x,z;\mu)\end{bmatrix}. \tag{10}$$

Note that matrix $V(x,z;\mu)$ is positive definite since $\varphi''(t) > 0$ for $t \in (0,\infty)$ by Condition 1.

Increments Δx and Δz determined from (9) 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, which 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 a minimizer $z(x;\mu) \in R$ of the function $B(x,z;\mu)$ for a given $x \in R^n$.

Lemma 1. *Let Condition 1 be satisfied. Then function $B(x,z;\mu) : (F(x),\infty) \to R$ (with x fixed) has a unique stationary point, which is its global minimizer. This stationary point is characterized by the equation*

$$e^Tu(x,z;\mu) = 1. \tag{11}$$

Solution $z(x;\mu)$ of this equation satisfies inequalities

$$F(x) + \underline{t}(\mu) = \underline{z}(x;\mu) \le z(x;\mu) \le \overline{z}(x;\mu) = F(x) + \overline{t}(\mu), \tag{12}$$

where values $0 < \underline{t}(\mu) \le \overline{t}(\mu)$, independent of x, can be obtained as unique solutions of equations

$$1 + \mu\varphi'(\underline{t}(\mu)) = 0, \quad 1 + m\mu\varphi'(\overline{t}(\mu)) = 0. \tag{13}$$

Moreover

$$e^Tu(x,\overline{z}(x;\mu);\mu) \le 1 \le e^Tu(x,\underline{z}(x;\mu);\mu). \tag{14}$$

Proof. Denote $\tilde{B}(z) = B(x, z; \mu)$. Function $\tilde{B}(z) : (F(x), \infty) \to R$ is convex in $(F(x), \infty)$, since it is a sum of convex functions. Thus if a stationary point of $\tilde{B}(z)$ exists, it is its unique global minimizer. Since $\varphi'(z(x;\mu) - f_i(x)) < 0$ and $\varphi''(z(x;\mu) - f_i(x)) > 0$ for all $1 \le i \le m$ by Condition 1, we can write

$$\begin{aligned}\varphi'(z(x;\mu) - F(x)) &\ge \sum_{i=1}^m \varphi'(z(x;\mu) - f_i(x)) \ge \sum_{i=1}^m \varphi'(z(x;\mu) - F(x)) \\ &= m\varphi'(z(x;\mu) - F(x)). \end{aligned} \tag{15}$$

Thus if we choose $\underline{z}(x;\mu) = F(x) + \underline{t}(\mu)$, $\overline{z}(x;\mu) = F(x) + \overline{t}(\mu)$ in such a way that (13) hold, we obtain inequalities

$$1 + \sum_{i=1}^m \mu\varphi'(\underline{z}(x;\mu) - f_i(x)) \le 0 \le 1 + \sum_{i=1}^m \mu\varphi'(\overline{z}(x;\mu) - f_i(x)), \tag{16}$$

which are equivalent to (14). Inequalities (14) imply that the solution $z(x;\mu)$ of (11) (the stationary point of $\tilde{B}(z)$) exists. Since function $\varphi'(t)$ is increasing, we obtain $F(x) < \underline{z}(x;\mu) \le z(x;\mu) \le \overline{z}(x;\mu)$. The above considerations are correct, since continuous function $\varphi'(t)$ maps $(0, \infty)$ onto $(-\infty, 0)$, which implies that equations (13) have unique solutions. □

Corollary 1. *Bounds $\underline{t}(\mu)$ and $\overline{t}(\mu)$ for $t(x;\mu) = z(x;\mu) - F(x)$, corresponding to barrier functions B1, B3 and B4, are given in Table 2. For barrier function B2, we can use bounds $\underline{t}(\mu) \ge \min(\mu, \overline{\tau})$ and $\overline{t}(\mu) \le m\mu$.*

	$\underline{t}(\mu) = \underline{z}(x;\mu) - F(x)$	$\overline{t}(\mu) = \overline{z}(x;\mu) - F(x)$
B1	μ	$m\mu$
B3	$2\mu/(1 + \sqrt{1 + 4\mu})$	$2m\mu/(1 + \sqrt{1 + 4m\mu})$
B4	$\sqrt{\mu}$	$\sqrt{m\mu}$

Table 2: Bounds for $t(x;\mu) = z(x;\mu) - F(x)$.

Proof. (a) Consider the logarithmic barrier function B1. Then $\varphi'(t) = -1/t$, which together with (13) gives $\underline{t}(\mu) = \mu$ and $\overline{t}(\mu) = m\mu$.

(b) Consider the barrier function B2. Since $\varphi'(t) = -1/t$ for $t \le \overline{\tau}$ and

$$\varphi'(t) + \frac{1}{t} = \left(\frac{\overline{\tau}^2}{t^3} - \frac{2\overline{\tau}}{t^2} + \frac{1}{t}\right) = \frac{1}{t^3}(t^2 - 2\overline{\tau}t + \overline{\tau}^2) = \frac{1}{t^3}(t - \overline{\tau})^2 \ge 0$$

for $t > \overline{\tau}$, we can conclude that $\varphi'(t)$ of B2 is not less than $\varphi'(t)$ of B1, which implies that $m\mu$, the upper bound of B1, is also the upper bound of B2. Since $\varphi'(t)$ of B2 is equal to $\varphi'(t)$ of B1 for $t \le \overline{\tau}$, we can set $\underline{t}(\mu) = \mu$ if $\mu \le \overline{\tau}$. At the same time, $\overline{\tau}$ is a suitable lower bound of B2 if $\mu > \overline{\tau}$.

(c) Consider the barrier function B3. Then $\varphi'(t) = -1/(t^2+t)$ which together with (13) gives $\mu/(\underline{t}(\mu)^2 + \underline{t}(\mu)) = 1$. Thus $\underline{t}(\mu)$ is a positive solution of the quadratic equation $t^2 + t - \mu = 0$ which can be written in the form

$$\underline{t}(\mu) = \frac{\sqrt{1+4\mu}-1}{2} = \frac{2\mu}{1+\sqrt{1+4\mu}}.$$

The upper bound can be obtained by the same way.

(d) Consider the barrier function B4. Then $\varphi'(t) = -1/t^2$ which together with (13) gives $\underline{t}(\mu) = \sqrt{\mu}$ and $\overline{t}(\mu) = \sqrt{m\mu}$. □

Solution $z(x;\mu)$ of nonlinear equation (11) can be obtained by efficient methods proposed in [12], [13], which use localization inequalities (14). Therefore, we can assume $z = z(x;\mu)$ with a sufficient precision, which implies that the last elements of the right-hand sides in (9) – (10) are negligible. Assuming $z = z(x;\mu)$, we denote

$$B(x;\mu) = B(x, z(x;\mu); \mu) = z(x;\mu) + \mu \sum_{i=1}^{m} \varphi(z(x;\mu) - f_i(x)), \tag{17}$$

$u(x;\mu) = u(x, z(x;\mu); \mu)$, $V(x;\mu) = V(x, z(x;\mu); \mu)$ and $G(x;\mu) = G(x, z(x;\mu); \mu)$. In this case, barrier function $B(x;\mu)$ depends only on x. In order to obtain a minimizer $(x, z) \in R^{n+1}$ of $B(x, z; \mu)$, it suffices to minimize $B(x;\mu)$ over R^n.

Lemma 2. *Consider barrier function (17). Then*

$$\nabla B(x;\mu) = A(x)u(x;\mu) \tag{18}$$

and

$$\nabla^2 B(x;\mu) = G(x;\mu) + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^T V(x;\mu)A^T(x)}{e^T V(x;\mu)e}. \tag{19}$$

Solution Δx of the Newton equation

$$\nabla^2 B(x;\mu)\Delta x = -\nabla B(x;\mu) \tag{20}$$

is equal to the corresponding vector obtained by solving (9) with $z = z(x;\mu)$.

Proof. Differentiating $B(x;\mu)$, we obtain

$$\begin{aligned}
\nabla B(x;\mu) &= \nabla z(x;\mu) + \mu \sum_{i=1}^{m} \varphi'(z(x;\mu) - f_i(x))\,(\nabla z(x;\mu) - g_i(x)) \\
&= \nabla z(x;\mu)\left(1 + \mu \sum_{i=1}^{m} \varphi'(z(x;\mu) - f_i(x))\right) - \mu \sum_{i=1}^{m} \varphi'(z(x;\mu) - f_i(x))\, g_i(x) \\
&= -\mu \sum_{i=1}^{m} \varphi'(z(x;\mu) - f_i(x))\, g_i(x) = A(x)u(x;\mu)
\end{aligned}$$

since

$$1 - e^T u(x;\mu) = 1 + \mu \sum_{i=1}^{m} \varphi'(z(x;\mu) - f_i(x)) = 0.$$

Differentiating the last equality, one has

$$\mu \sum_{i=1}^{m} \varphi''(z(x;\mu) - f_i(x)) \, (\nabla z(x;\mu) - g_i(x)) = 0,$$

which gives

$$\nabla z(x;\mu) = \frac{A(x)V(x;\mu)e}{e^T V(x;\mu)e}.$$

Thus

$$\begin{aligned} \nabla^2 B(x;\mu) &= \sum_{i=1}^{m} u_i(x;\mu) \, G_i(x) + \mu \sum_{i=1}^{m} \varphi''(z(x;\mu) - f_i(x)) \, (g_i(x) - \nabla z(x;\mu)) \, g_i^T(x) \\ &= G(x;\mu) + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^T V(x;\mu)A^T(x)}{e^T V(x;\mu)e}. \end{aligned}$$

Using the second equation of (9) with $e^T u(x;\mu) = 1$, we obtain

$$\Delta z = \frac{e^T V(x;\mu)A^T(x)}{e^T V(x;\mu)e} \Delta x,$$

which after substituting into the first equation gives

$$\left(G(x;\mu) + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^T V(x;\mu)A^T(x)}{e^T V(x;\mu)e} \right) \Delta x = -A(x)u(x;\mu).$$

This is exactly equation (20). □

Note that we use (9) rather than (20) for a direction determination since nonlinear equation (11) is solved with precision $\underline{\delta}$ and, therefore, in general $1 - e^T u(x;\mu)$ differs from zero.

Lemma 3. *Let Δx solve (20) (or (9) with $z = z(x;\mu)$). If matrix $G(x;\mu)$ is positive definite, then $(\Delta x)^T \nabla B(x;\mu) < 0$ (direction vector Δx is descent for $B(x;\mu)$).*

Proof. Equation (20) implies

$$(\Delta x)^T \nabla^2 B(x;\mu) \Delta x = -(\Delta x)^T \nabla B(x;\mu).$$

Thus $(\Delta x)^T \nabla B(x;\mu) < 0$ if $\nabla^2 B(x;\mu)$ is positive definite. But

$$\begin{aligned} v^T \nabla^2 B(x;\mu) v &= v^T G(x;\mu) v + \left(v^T A(x)V(x;\mu)A^T(x)v - \frac{(v^T A(x)V(x;\mu)e)^2}{e^T V(x;\mu)e} \right) \\ &\geq v^T G(x;\mu) v \end{aligned}$$

for an arbitrary $v \in R^n$ by (19) and by the Schwarz inequality (since $V(x;\mu)$ is positive definite). Thus $(\Delta x)^T \nabla B(x;\mu) < 0$ if $G(x;\mu)$ is positive definite. $\square$

Consider the logarithmic barrier function B1. Then

$$V(x;\mu) = \frac{1}{\mu} U^2(x;\mu),$$

where $U(x;\mu) = \mathrm{diag}(u(x;\mu)_1, \dots, u(x;\mu)_m)$, which implies that $\|V(x;\mu)\| \to \infty$ as $\mu \to 0$. Thus $\nabla^2 B(x;\mu)$ can be ill-conditioned for μ small enough (see (19)). For this reason, it is necessary to use a lower bound $\underline{\mu}$ for μ (more details are given in Section 3). The following lemma gives upper bounds for $\|\nabla^2 B(x;\mu)\|$ if Condition 2 holds.

Lemma 4. *Let Assumption 2, Condition 1 and Condition 2 be satisfied. If $\mu \ge \underline{\mu} > 0$, then*

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

where $\overline{V} = \underline{\mu}\varphi''(\underline{t}(\underline{\mu}))$.

Proof. Using (19) and Assumption 4, we obtain

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

Since $V(x;\mu)$ is diagonal and $f_i(x) \le F(x)$ for all $1 \le i \le m$, one has

$$\|V(x;\mu)\| = \mu\varphi''(z(x;\mu) - F(x)) \le \mu\varphi''(\underline{t}(\mu)).$$

Now we prove that $\mu\varphi''(\underline{t}(\mu))$ is a non-increasing function of μ, which implies that $\mu\varphi''(\underline{t}(\mu)) \le \underline{\mu}\varphi''(\underline{t}(\underline{\mu}))$. Differentiating (13) by μ, we obtain

$$\varphi'(\underline{t}(\mu)) + \mu\varphi''(\underline{t}(\mu))\underline{t}'(\mu) = 0 \quad \Rightarrow \quad \underline{t}'(\mu) = -\frac{\varphi'(\underline{t}(\mu))}{\mu\varphi''(\underline{t}(\mu))} > 0, \tag{21}$$

where $\underline{t}'(\mu)$ is a derivative of $\underline{t}(\mu)$ by μ. Thus, using Condition 2 and the fact that $\varphi''(\underline{t}(\mu)) > 0$, we can write

$$\frac{\mathrm{d}(\mu\varphi''(\underline{t}(\mu)))}{\mathrm{d}\mu} = \varphi''(\underline{t}(\mu)) + \mu\varphi'''(\underline{t}(\mu))\underline{t}'(\mu) = \varphi''(\underline{t}(\mu)) - \varphi'''(\underline{t}(\mu))\frac{\varphi'(\underline{t}(\mu))}{\varphi''(\underline{t}(\mu))} \le 0.$$

$\square$

Corollary 2. *If $\underline{\mu}$ is sufficiently small, we can use bounds $\overline{V} = \underline{\mu}^{-1}$ for B1 and B2, $\overline{V} = 2\underline{\mu}^{-1}$ for B3 and $\overline{V} = 2\underline{\mu}^{-1/2}$ for B4.*

Proof. We use expressions for $\varphi''(t)$ given in Table 1 and bounds for $\underline{t}(\mu)$ proposed in Corollary 1.

(a) Consider barrier function B1. In this case $\mu\varphi''(\underline{t}(\mu)) = \mu\varphi''(\mu) = \mu^{-1}$.

(b) Consider barrier function B2. If $\mu < \overline{\tau}$, then $\underline{t}(\mu) < \overline{\tau}$. Thus B2 is equal to B1 and we can use the previous bound.

(c) Consider barrier function B3. Assuming $\mu \le 3/4$, we can write

$$\mu\varphi''(\underline{t}(\mu)) = \mu\frac{2\underline{t}(\mu)+1}{(\underline{t}(\mu)^2+\underline{t}(\mu))^2} = \frac{1}{\mu}\left(1+2\frac{\sqrt{1+4\mu}-1}{2}\right) \le \frac{2}{\mu},$$

since $\underline{t}(\mu)^2+\underline{t}(\mu)=\mu$ (see proof of Corollary 1).

(d) Consider barrier function B4. In this case $\mu\varphi''(\underline{t}(\mu)) = \mu\varphi''(\mu^{1/2}) = 2\mu/\mu^{3/2} = 2\mu^{-1/2}$. □

As we can deduce from Corollary 1 and Corollary 2, properties of barrier function B4 depend on $\mu^{1/2}$ instead of μ. For this reason, we have used μ^2 instead of μ in barrier function B4 in our computational experiments.

Now we return to the direction determination. To simplify the notation, we write equation (9) in the form

$$\begin{bmatrix} H & -a \\ -a^T & \alpha \end{bmatrix}\begin{bmatrix} d \\ \delta \end{bmatrix} = -\begin{bmatrix} g \\ \gamma \end{bmatrix} \tag{22}$$

where

$$H = G + A(x)V(x,z;\mu)A^T(x), \quad G = G(x,z;\mu), \tag{23}$$

and $a = A(x)V(x,z;\mu)e$, $\alpha = e^TV(x,z;\mu)e$, $g = A(x)u(x,z;\mu)$, $\gamma = 1 - e^Tu(x,z;\mu)$. Since

$$\begin{bmatrix} H & -a \\ -a^T & \alpha \end{bmatrix}^{-1} = \begin{bmatrix} H^{-1} - H^{-1}a(a^TH^{-1}a-\alpha)^{-1}a^TH^{-1} & -H^{-1}a(a^TH^{-1}a-\alpha)^{-1} \\ -(a^TH^{-1}a-\alpha)^{-1}a^TH^{-1} & -(a^TH^{-1}a-\alpha)^{-1} \end{bmatrix},$$

we can write

$$\begin{bmatrix} d \\ \delta \end{bmatrix} = -\begin{bmatrix} H & -a \\ -a^T & \alpha \end{bmatrix}^{-1}\begin{bmatrix} g \\ \gamma \end{bmatrix} = \begin{bmatrix} H^{-1}(\delta a - g) \\ \delta \end{bmatrix}, \tag{24}$$

where

$$\delta = (a^TH^{-1}a-\alpha)^{-1}(a^TH^{-1}g+\gamma).$$

Matrix H is sparse if $A(x)$ has sparse columns. If H is not positive definite, it is advantageous to change it before a computation of the direction vector. Thus we use the sparse Gill-Murray decomposition

$$H + E = LDL^T, \tag{25}$$

where E is a positive semidefinite diagonal matrix that assures positive definiteness of LDL^T. Using the Gill-Murray decomposition, we solve two equations

$$LDL^Tc = a, \quad LDL^Tv = g \tag{26}$$

and set

$$\delta = \frac{a^T v + \gamma}{a^T c - \alpha}, \quad d = \delta c - v. \tag{27}$$

In (23), we assume that $G = G(x, z; \mu)$, where $G(x, z; \mu)$ is either given analytically or determined by using automatic differentiation, see [9]. In practical computations, G is frequently an approximation of $G(x, z; \mu)$ 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 $\underline{n} \ll n$ if G is sparse. 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

$$G = \sum_{i=1}^{m} u_i(x; \mu) G_i, \tag{28}$$

where approximations G_i of $\nabla^2 f_i(x)$ are computed by using variable metric updates described in [8]. More details are given in the next section.

Now we are in a position to describe the basic algorithm, in which the direction vector is modified in such a way that

$$-g^T d \geq \varepsilon_0 \|g\| \|d\|, \quad \underline{c}\|g\| \leq \|d\| \leq \overline{c}\|g\|, \tag{29}$$

where ε_0, $\underline{c}$, $\overline{c}$ are suitable constants.

Algorithm 1.

Data: Termination parameter $\underline{\varepsilon} > 0$, precision for the nonlinear equation solver $\underline{\delta} > 0$, bounds for the barrier parameter $0 < \underline{\mu} < \overline{\mu}$, rate of the barrier parameter decrease $0 < \lambda < 1$, restart parameters $0 < \underline{c} < \overline{c}$ and $\varepsilon_0 > 0$, line search parameter $\varepsilon_1 > 0$, rate of the step-size decrease $0 < \beta < 1$, step bound $\overline{\Delta} > 0$.

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

Step 1: *Initiation.* Set $\mu = \overline{\mu}$. Determine the sparsity pattern of matrix $H(x)$ from the sparsity pattern of matrix $A(x)$. Carry out a symbolic decomposition of $H(x)$. Compute values $f_i(x)$, $1 \leq i \leq m$, and $F(x) = \max_{1 \leq i \leq m} f_i(x)$. Set $k := 0$ (iteration count) and $r := 0$ (restart indicator).

Step 2: *Termination.* Solve nonlinear equation (11) with a precision $\underline{\delta}$ to obtain value $z(x; \mu)$ and vector $u(x; \mu) = u(x, z(x; \mu); \mu)$. Compute matrix $A := A(x)$ and vector $g := g(x; \mu) = A(x)u(x; \mu)$. If $\mu \leq \underline{\mu}$ and $\|g\| \leq \underline{\varepsilon}$, then terminate the computation. Otherwise set $k := k + 1$.

Step 3: *Approximation of the Hessian matrix.* Set $G = G(x; \mu)$ or compute an approximation G of the Hessian matrix $G(x; \mu)$ by using either gradient differences or variable metric updates.

Step 4: *Direction determination.* Determine matrix H by (23). Determine vector d from (26)-(27) by using the Gill-Murray decomposition (25) of matrix H.

Step 5: *Restart.* If $r = 0$ and (29) does not hold, choose a positive definite diagonal matrix D by formula (32) introduced in Section 3, set $G := D$, $r := 1$ and go to Step 4. If $r = 1$ and (29) does not hold, set $d := -g$ (steepest descent direction) and $r := 0$.

Step 6: *Step-length selection.* Define maximum step-length $\overline{\alpha} = \min(1, \overline{\Delta}/\|d\|)$. Find a minimum integer $l \geq 0$ such that $B(x+\beta^l \overline{\alpha} d; \mu) \leq B(x; \mu) + \varepsilon_1 \beta^l \overline{\alpha} g^T d$ (note that nonlinear equation (11) has to be solved at all points $x + \beta^j \overline{\alpha} d$, $0 \leq j \leq l$). Set $x := x + \beta^l \overline{\alpha} d$. Compute values $f_i(x)$, $1 \leq i \leq m$, and $F(x) = \max_{1 \leq i \leq m} f_i(x)$.

Step 7: *Barrier parameter update.* Determine a new value of the barrier parameter $\mu \geq \underline{\mu}$ (not greater than the current one) by one of the procedures described in Section 3. Go to Step 2.

The above algorithm requires several notes. The restart strategy in Step 5 implies that the direction vector d is uniformly descent and gradient-related (see (29)). Since function $B(x; \mu)$ is smooth, the line search utilized in Step 6 always finds a step size satisfying the Armijo condition $B(x + \alpha d; \mu) \leq B(x; \mu) + \varepsilon_1 \alpha g^T d$. 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 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 better local solution 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 the barrier parameter update. There are several influences that should be taken into account, which make updating procedures rather complicated. More details are given in Section 3.

Finally, note that the proposed interior-point method is very similar algorithmically (but not theoretically) to the smoothing method described in [20] and [22]. Thus Algorithm 1 can be easily adapted to an algorithm implementing the smoothing method (see Section 5). These methods are compared in Section 6.

3 Implementation details

In Section 2, we have proved (Lemma 3) that direction vector d obtained by solving equation (22) is descent for $\nabla B(x; \mu)$ if matrix $G(x; \mu)$ is positive definite. Unfortunately, positive definiteness of this matrix is not assured in general. A similar problem appears in a connection with the Newton method for smooth unconstrained optimization. Therefore, trust-region methods were developed for this reason. We have tested several trust-region methods, but the line-search approach with suitable restarts turns to be more efficient. In this case, matrix $G \approx G(x; \mu)$ is replaced by a positive definite diagonal matrix $D = \mathrm{diag}(D_{ii})$ if (29) (with $g = g(x; \mu) = A(x)u(x; \mu)$) does not hold. Thus the Hessian matrix

$\nabla^2 B(x;\mu)$ is replaced by the matrix

$$B = D + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^TV(x;\mu)A^T(x)}{e^TV(x;\mu)e} \tag{30}$$

(see (19)). Let $0 < \underline{D} \le D_{ii} \le \overline{D}$ for all $1 \le i \le n$. Then the minimum eigenvalue of B is not less than $\underline{D}$ (see proof of Lemma 3) and, using the same way as in the proof of Lemma 4, we can write

$$\|B\| \le \left\|D + A(x)V(x;\mu)A^T(x)\right\| \le \overline{D} + m\overline{g}^2\|V(x;\mu)\| \le \overline{D} + m\overline{g}^2\overline{V}, \tag{31}$$

where bounds $\overline{V}$ for individual barrier functions are given by Corollary 2 (procedure used in Step 7 of Algorithm 1 assures that $\mu \ge \underline{\mu}$). Using (31), we can write

$$\kappa(B) \le (\overline{D} + m\overline{g}^2\overline{V})/\underline{D}.$$

If d solves equation $Bd + g = 0$, then (29) hold with $\varepsilon_0 \le 1/\kappa(B)$, $\underline{c} \le \underline{D}$ and $\overline{c} \ge \underline{D}\kappa(B)$ (see [4]). If these inequalities are not satisfied, the case when (29) does not hold can appear. In this case we simply set $d = -g$ (this situation appears rarely when ε_0, $\underline{c}$ and $1/\overline{c}$ are sufficiently small).

The choice of matrix D in restarts strongly affects the efficiency of Algorithm 1 for problems with indefinite Hessian matrices (if $G = G(x;\mu)$ or G is computed by numerical differentiation). We have tested various possibilities including the simple choice $D = I$, which proved to be unsuitable. The best results were obtained by the heuristic procedure proposed in [19] for equality constrained optimization and used in [14] in a connection with interior-point methods for nonlinear programming. This procedure uses formulas

$$\begin{aligned} D_{ii} &= \underline{D}, \quad \text{if} \quad \frac{\|g\|}{10}|H_{ii}| < \underline{D}, \\ D_{ii} &= \frac{\|g\|}{10}|H_{ii}|, \quad \text{if} \quad \underline{D} \le \frac{\|g\|}{10}|H_{ii}| \le \overline{D}, \\ D_{ii} &= \overline{D}, \quad \text{if} \quad \overline{D} < \frac{\|g\|}{10}|H_{ii}|, \end{aligned} \tag{32}$$

where $\underline{D} = 0.005$, $\overline{D} = 500.0$ and H is given by (23).

Lemma 5. *Direction vectors d_k, $k \in N$, generated by Algorithm 1 are uniformly descent and gradient-related ((29) hold for all $k \in N$). If Assumption 1, Assumption 2, and Condition 1 hold, then the Armijo line search (Step 6 of Algorithm 1) assures that a constant c exists such that*

$$B(x_{k+1};\mu_k) - B(x_k;\mu_k) \le -c\|g(x_k;\mu_k)\|^2 \quad \forall k \in N. \tag{33}$$

Proof. Inequalities (29) are obvious (they are assured by the restart strategy) and inequality (33) is proved, e.g., in [4] (note that $\nabla B(x_k;\mu_k) = g(x_k;\mu_k)$ by Lemma 2). □

Matrix G appearing in Step 3 of Algorithm 1 can be computed by using partitioned variable metric updates described in [8]. This way assures that matrix G is positive definite so restarts are unnecessary. In our implementation, we use safeguarded scaled BFGS updates. In this case, G is given by (28). Let $R_i^n \subset R^n$, $1 \le i \le m$, be subspaces defined by independent variables of functions f_i and $Z_i \in R^{n \times n_i}$ be matrices whose columns form canonical orthonormal bases in these subspaces (they are columns of the unit matrix of order n). Then we can define reduced approximations of the Hessian matrices $\tilde{G}_i = Z_i^T G_i Z_i$, $1 \le i \le m$. New reduced approximations of the Hessian matrices, used in the next iteration, are computed by the formulas

$$\begin{aligned}
\tilde{G}_i^+ &= \frac{1}{\tilde{\gamma}_i}\left(\tilde{G}_i - \frac{\tilde{G}_i \tilde{s}_i \tilde{s}_i^T \tilde{G}_i}{\tilde{s}_i^T \tilde{G}_i \tilde{s}_i}\right) + \frac{\tilde{y}_i \tilde{y}_i^T}{\tilde{s}_i^T \tilde{y}_i}, & \tilde{s}_i^T \tilde{y}_i > 0, \\
\tilde{G}_i^+ &= \tilde{G}_i, & \tilde{s}_i^T \tilde{y}_i \le 0,
\end{aligned}$$

where

$$\tilde{s}_i = Z_i^T (x^+ - x), \quad \tilde{y}_i = Z_i^T (\nabla f_i(x^+) - \nabla f_i(x)), \quad 1 \le i \le m,$$

and where either $\tilde{\gamma}_i = 1$ or $\tilde{\gamma} = \tilde{s}_i^T \tilde{G}_i \tilde{s}_i / \tilde{s}_i^T \tilde{y}_i$. (we denote by + quantities from the next iteration). The particular choice of $\tilde{\gamma}_i$ is determined by the controlled scaling strategy described in [17]. In the first iteration we set $\tilde{G}_i = I$, $1 \le i \le m$, where I are unit matrices of suitable orders. Finally, $G_i^+ = Z_i \tilde{G}_i^+ Z_i^T$, $1 \le i \le m$.

A very important part of Algorithm 1 is the barrier parameter update. 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, round-off errors can cause that $z(x;\mu) = F(x)$ when μ is too small (since $F(x)+\underline{t}(\mu) \le z(x;\mu) \le F(x)+\overline{t}(\mu)$ and $\overline{t}(\mu) \to 0$ as $\mu \to 0$ by Lemma 1), which leads to a breakdown (division by $z(x;\mu)-F(x) = 0$). Thus a lower bound $\underline{\mu}$ for the barrier parameter has to be used (we recommend value $\underline{\mu} = 10^{-10}$ in double precision arithmetic).

Algorithm 1 is also sensitive to the way in which the barrier parameter decreases. Considering logarithmic barrier function B1 and denoting by $s(x;\mu) = z(x;\mu)e - f(x)$ vector of slack variables, we can see from (7) that $u_i(x;\mu)s_i(x;\mu) = \mu$, $1 \le i \le m$. In this case, interior-point methods assume that μ decreases linearly (see [21]). We have tested various possibilities for the barrier parameter update including simple geometric sequences, which proved to be unsuitable. Better results were obtained by the following two procedures:

Procedure A.

Phase 1: If $\|g(x_k;\mu_k)\| \ge \underline{g}$, we set $\mu_{k+1} = \mu_k$, i.e., the barrier parameter is not changed.

Phase 2: If $\|g(x_k;\mu_k)\| < \underline{g}$, we set

$$\mu_{k+1} = \max\left(\tilde{\mu}_{k+1},\, \underline{\mu},\, 10\,\varepsilon_M |F(x_{k+1})|\right), \tag{34}$$

where $F(x_{k+1}) = \max_{1\le i\le m} f_i(x_{k+1})$, ε_M is a machine precision, and

$$\tilde{\mu}_{k+1} = \min\left[\max(\lambda\mu_k,\, \mu_k/(\sigma\mu_k+1)),\, \max(\|g(x_k;\mu_k)\|^2,\, 10^{-2k})\right], \tag{35}$$

where $g(x_k;\mu_k) = A(x_k)u(x_k;\mu_k)$. Values $\underline{\mu} = 10^{-10}$, $\lambda = 0.85$, and $\sigma = 100$ are chosen as defaults.

Procedure B.

Phase 1: If $\|g(x_k;\mu_k)\|^2 \ge \rho\mu_k$, we set $\mu_{k+1} = \mu_k$, i.e., the barrier parameter is not changed.

Phase 2: If $\|g(x_k;\mu_k)\|^2 < \rho\mu_k$, we set

$$\mu_{k+1} = \max(\underline{\mu},\, \|g_k\|^2). \tag{36}$$

Values $\underline{\mu} = 10^{-10}$ and $\rho = 0.1$ are chosen as defaults.

The choice of $\underline{g}$ in Procedure A is not critical. We can set $\underline{g} = \infty$ but a lower value is sometimes more suitable (especially for smoothing methods described in Section 5). More details are given in Section 6. The reason for using formula (34) was mentioned above. Formula (35) requires several notes. The first argument of the minimum controls the rate of the barrier parameter decrease, which is linear (geometric sequence) for small k (term $\lambda\mu_k$) and sublinear (harmonic sequence) for large k (term $\mu_k/(\sigma\mu_k+1)$). Thus the second argument, which assures that μ is small in the neighborhood of the solution, plays an essential role for large k. Term 10^{-2k} assures that $\mu = \underline{\mu}$ does not hold for small k. This situation can arise when $\|g(x_k;\mu_k)\|$ is small, even if x_k is far from the solution. The idea of Procedure B follows from the requirement that $B(x;\mu)$ should be sufficiently minimized for a current value of μ. Thus parameter μ_k is changed only if $\|g(x_k;\mu_k)\|$ is sufficiently small.

4 Global convergence

In the subsequent considerations, we assume that $\underline{\delta} = \underline{\varepsilon} = \underline{\mu} = 0$ and all computations are exact ($\varepsilon_M = 0$ in (34)). We will investigate an infinite sequence $\{x_k\}_1^\infty$ generated by Algorithm 1.

Lemma 6 *Let Assumption 1, Assumption 2, and Condition 1 be satisfied. Then the values $\{\mu_k\}_1^\infty$, generated by Algorithm 1, form a non-increasing sequence such that $\mu_k \to 0$.*

Proof. We prove that the number of consecutive steps in Phase 1 of the procedure for the barrier parameter decrease is finite. Then the number of steps in Phase 2 is infinite and since μ_k decreases in these steps either by geometric or by harmonic sequence, one has $\mu_k \to 0$.

(a) First we prove that $B(x;\mu)$ is bounded from below if μ is fixed. This assertion holds trivially if Condition 3 is satisfied. If this is not the case, then

$$B(x;\mu) - \underline{F} = z(x;\mu) - \underline{F} + \mu \sum_{i=1}^{m} \varphi(z(x;\mu) - f_i(x)) \geq z(x;\mu) - \underline{F} + m\mu\varphi(z(x;\mu) - \underline{F}).$$

Convex function $\psi(t) = t + m\mu\varphi(t)$ has a unique minimum at the point $t = \bar{t}(\mu)$, since $\psi'(\bar{t}(\mu)) = 1 + m\mu\varphi'(\bar{t}(\mu)) = 0$ by (13). Thus

$$B(x;\mu) - \underline{F} \geq \bar{t}(\mu) + m\mu\varphi(\bar{t}(\mu))$$

or $B(x;\mu) \geq \underline{B}$ where $\underline{B} = \underline{F} + \bar{t}(\mu) + m\mu\varphi(\bar{t}(\mu))$.

(b) In Phase 1, the value of μ is fixed. Since function $B(x;\mu)$ is continuous, bounded from below by (a), and since (33) (with $\mu_k = \mu$) holds, it can be proved (see [4]) that $\|g(x_k;\mu)\| \to 0$ if Phase 1 contains an infinite number of consecutive steps. Thus a step (with index l) belonging to Phase 1 exists such that either $\|g(x_l;\mu)\| < \underline{g}$ in Procedure A or $\|g(x_l;\mu)\| < \rho\mu$ in Procedure B. This is a contradiction with the definition of Phase 1. □

Now we clarify the dependence of $z(x;\mu)$ and $B(x;\mu)$ on the parameter μ. For this purpose, we assume that $z(x;\mu)$ and $B(x;\mu)$ are functions of μ and write $z(x,\mu) = z(x;\mu)$ and $B(x,\mu) = B(x;\mu)$.

Lemma 7. *Let Condition 1 be satisfied and $z(x,\mu)$ be a solution of equation (11) (for fixed x and variable μ), i.e., $1 - e^T u(x, z(x,\mu)) = 0$. Then*

$$\frac{\partial z(x,\mu)}{\partial \mu} > 0, \qquad \frac{\partial B(x,\mu)}{\partial \mu} = \sum_{i=1}^{m} \varphi(z(x,\mu) - f_i(x)).$$

Proof. Differentiating equation (11), which has the form

$$1 + \mu \sum_{i=1}^{m} \varphi'(z(x,\mu) - f_i(x)) = 0,$$

we obtain

$$\sum_{i=1}^{m} \varphi'(z(x,\mu) - f_i(x)) + \mu \sum_{i=1}^{m} \varphi''(z(x,\mu) - f_i(x)) \frac{\partial z(x,\mu)}{\partial \mu} = 0,$$

which gives

$$\frac{\partial z(x,\mu)}{\partial \mu} = \frac{1}{\mu^2 \sum_{i=1}^{m} \varphi''(z(x,\mu) - f_i(x))} > 0.$$

Differentiating function

$$B(x,\mu) = z(x,\mu) + \mu \sum_{i=1}^{m} \varphi(z(x,\mu) - f_i(x)),$$

one has

$$\begin{aligned} \frac{\partial B(x,\mu)}{\partial \mu} &= \frac{\partial z(x,\mu)}{\partial \mu} + \sum_{i=1}^{m} \varphi(z(x,\mu) - f_i(x)) + \mu \sum_{i=1}^{m} \varphi'(z(x,\mu) - f_i(x)) \frac{\partial z(x,\mu)}{\partial \mu} \\ &= \frac{\partial z(x,\mu)}{\partial \mu} \left(1 + \mu \sum_{i=1}^{m} \varphi'(z(x,\mu) - f_i(x))\right) + \sum_{i=1}^{m} \varphi(z(x,\mu) - f_i(x)) \\ &= \sum_{i=1}^{m} \varphi(z(x,\mu) - f_i(x)). \end{aligned}$$

□

Now we prove that $B(x;\mu)$, $z(x;\mu)$, and $F(x)$ are bounded and $B(x,\mu)$ is a Lipschitz continuous function of μ.

Lemma 8. *Let Assumption 1, Assumption 2, and Condition 1 be satisfied. Let $\{x_k\}_1^\infty$ and $\{\mu_k\}_1^\infty$ be sequences generated by Algorithm 1. Then sequences $\{B(x_k;\mu_k)\}_1^\infty$, $\{z(x_k;\mu_k)\}_1^\infty$, and $\{F(x_k)\}_1^\infty$ are bounded. Moreover, there is $L \geq 0$ such that*

$$B(x_{k+1};\mu_{k+1}) \leq B(x_{k+1};\mu_k) - L(\mu_{k+1} - \mu_k) \tag{37}$$

for all $k \in N$.

Proof. Boundedness from below simply follows from Assumption 1, inequality (12) and the proof of Lemma 6. If Condition 3 holds, then boundedness from above and (37) with $L = 0$ follow from (12), Lemma 7 and (33), since barrier terms are nonnegative. Assume now that Condition 3 does not hold.
(a) Denote $\underline{C} = \min(\underline{B}, \underline{F})$. As in the proof of Lemma 6, we can write

$$B(x;\mu) - \underline{C} \geq (z(x;\mu) - \underline{C})/2 + (z(x;\mu) - \underline{C})/2 + m\mu\varphi(z(x;\mu) - \underline{C}).$$

The convex function $\tilde{\psi}(t) = t/2 + m\mu\varphi(t)$ has a unique minimum at a point $t = \tilde{t}(\mu) \geq \bar{t}(\mu)$ (this follows from Condition 1, since $\varphi'(t)$ is a negative, concave and increasing function such that $\lim_{t\to\infty} \varphi'(t) = 0$). Thus

$$B(x;\mu) - \underline{C} \geq (z(x;\mu) - \underline{C})/2 + \tilde{t}(\mu)/2 + m\mu\varphi(\tilde{t}(\mu))$$

or

$$z(x;\mu) - \underline{C} \leq 2(B(x;\mu) - \underline{C}) - \tilde{t}(\mu) - 2m\mu\varphi(\tilde{t}(\mu)) \leq 2(B(x;\mu) - \underline{C}) + \eta, \tag{38}$$

where $\eta = \max(0, -2m\overline{\mu}\varphi(\tilde{t}(\overline{\mu})))$. The formula for η follows from the fact that $\tilde{t}'(\mu)$ satisfies the same equation (21) as $\underline{t}'(\mu)$. Thus $\tilde{t}(\mu)$ increases as μ increases and since $\varphi'(\tilde{t}(\mu)) < 0$, we obtain $-\overline{\mu}\varphi(\tilde{t}(\overline{\mu})) \geq -\mu\varphi(\tilde{t}(\mu))$.
(b) Using the mean value theorem and Lemma 7, we obtain

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k) &= \sum_{i=1}^{m} \varphi(z(x_{k+1},\tilde{\mu}_k) - f_i(x_{k+1}))(\mu_{k+1} - \mu_k) \\ &\leq \sum_{i=1}^{m} \varphi(z(x_{k+1};\mu_k) - f_i(x_{k+1}))(\mu_{k+1} - \mu_k) \\ &\leq m\varphi(z(x_{k+1};\mu_k) - \underline{C})(\mu_{k+1} - \mu_k), \end{aligned} \tag{39}$$

where $0 < \mu_{k+1} \le \tilde{\mu}_k \le \mu_k \le \overline{\mu}$. Condition 1 assures an existence of a number $a > 0$ such that $\varphi(t) \ge -at$ (at is a tangent of $\varphi(t)$). Thus inequalities (33), (38) and (39) imply

$$\begin{aligned}
B(x_{k+1};\mu_{k+1}) - \underline{C} &\le B(x_{k+1};\mu_k) - \underline{C} + m\varphi(z(x_{k+1};\mu_k) - \underline{C})(\mu_{k+1} - \mu_k) \\
&\le B(x_{k+1};\mu_k) - \underline{C} + ma(z(x_{k+1};\mu_k) - \underline{C})(\mu_k - \mu_{k+1}) \\
&\le B(x_{k+1};\mu_k) - \underline{C} + ma[2(B(x_{k+1};\mu_k) - \underline{C}) + \eta](\mu_k - \mu_{k+1}) \\
&= (1 + \lambda\delta_k)(B(x_{k+1};\mu_k) - \underline{C}) + \lambda\delta_k\eta/2 \\
&\le (1 + \lambda\delta_k)(B(x_k;\mu_k) - \underline{C}) + \lambda\delta_k\eta/2,
\end{aligned}$$

where $\lambda = 2ma$ and $\delta_k = \mu_k - \mu_{k+1}$. Then

$$\begin{aligned}
B(x_{k+1};\mu_{k+1}) - \underline{C} + \frac{\eta}{2} &\le \prod_{i=1}^{k}(1 + \lambda\delta_i)(B(x_1;\mu_1) - \underline{C} + \frac{\eta}{2}) \\
&\le \prod_{i=1}^{\infty}(1 + \lambda\delta_i)(B(x_1;\mu_1) - \underline{C} + \frac{\eta}{2})
\end{aligned}$$

and since

$$\sum_{i=1}^{\infty}\lambda\delta_i = \lambda(\mu_1 - \lim_{k\to\infty}\mu_k) = \lambda\mu_1$$

the above product is finite. This together with (12) and (38) proves that sequences $\{B(x_k;\mu_k)\}_1^\infty$, $\{z(x_k;\mu_k)\}_1^\infty$, and $\{F(x_k)\}_1^\infty$ are bounded from above.
(c) Using (39), we can write

$$\begin{aligned}
B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k) &\le m\varphi(z(x_{k+1};\mu_k) - \underline{C})(\mu_{k+1} - \mu_k) \\
&\le m\varphi(F(x_{k+1}) + \bar{t}(\mu_k) - \underline{C})(\mu_{k+1} - \mu_k) \\
&\le m\varphi(\overline{F} + \bar{t}(\overline{\mu}) - \underline{C})(\mu_{k+1} - \mu_k) \\
&\triangleq -L(\mu_{k+1} - \mu_k),
\end{aligned}$$

for all $k \in N$, where existence of $\overline{F}$ follows from boundedness of $\{F(x_k)\}_1^\infty$. □

The proof of Lemma 8 does not depend on bounds $\overline{g}$ and $\overline{G}$, since we can use a weaker inequality $B(x_{k+1};\mu_k) \le B(x_k;\mu_k)$ instead of (33). Thus an upper bound $\overline{F}$ (independent of $\overline{g}$ and $\overline{G}$) exists such that $F(x_k) \le \overline{F}$ for all $k \in N$ and we can use $\overline{F}$ in Assumption 2. Note that we can set $\overline{F} = B(x_1;\mu_1)$ if Condition 3 holds.

Theorem 1. *Let Assumption 1, Assumption 2, and Condition 1 be satisfied. Consider a sequence $\{x_k\}_1^\infty$ generated by Algorithm 1 (with $\underline{\delta} = \underline{\varepsilon} = \underline{\mu} = 0$). Then*

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

and

$$u_i(x_k;\mu_k) \ge 0, \quad z(x_k;\mu_k) - f_i(x_k) \ge 0, \quad \lim_{k\to\infty}u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) = 0$$

for $1 \le i \le m$.

Proof. Equality $1 - e^T u(x_k;\mu_k) = 0$ holds since $\underline{\delta} = 0$. Inequalities $u_i(x_k;\mu_k) \geq 0$, $z(x_k;\mu_k) - f_i(x_k) \geq 0$ follow from (7) (since $\varphi'(t)$ is negative for $t > 0$) and from Lemma 1.

(a) Since (33) and (37) hold, we can write

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - B(x_k;\mu_k) &= (B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k)) + (B(x_{k+1};\mu_k) - B(x_k;\mu_k)) \\ &\leq -L(\mu_{k+1} - \mu_k) - c\|g(x_k;\mu_k)\|^2, \end{aligned}$$

which implies

$$\begin{aligned} \underline{B} &\leq \lim_{k\to\infty} B(x_k;\mu_k) \leq B(x_1;\mu_1) - L\sum_{k=1}^{\infty}(\mu_{k+1} - \mu_k) - c\sum_{k=1}^{\infty}\|g(x_k;\mu_k)\|^2 \\ &= B(x_1;\mu_1) + L\mu_1 - c\sum_{k=1}^{\infty}\|g(x_k;\mu_k)\|^2, \end{aligned}$$

where $\underline{B} = \underline{F} + \bar{t}(\mu) + m\mu\varphi(\bar{t}(\mu))$ (see proof of Lemma 6). Thus one has

$$\sum_{k=1}^{\infty}\|g(x_k;\mu_k)\|^2 \leq \frac{1}{c}(B(x_1;\mu_1) - \underline{B} + L\mu_1) < \infty,$$

which implies $g(x_k;\mu_k) = \sum_{i=1}^m u_i(x_k;\mu_k)g_i(x_k) \to 0$.

(b) Let $1 \leq i \leq m$ be chosen arbitrarily. Since $u_i(x_k;\mu_k) \geq 0$, $z(x_k;\mu_k) - f_i(x_k) \geq 0$, one has

$$\limsup_{k\to\infty} u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \geq \liminf_{k\to\infty} u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \geq 0.$$

It suffices to prove that these inequalities are satisfied as equalities. Assume on the contrary that there is an infinite subset $N_1 \subset N$ such that $u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \geq \varepsilon \; \forall k \in N_1$, where $\varepsilon > 0$. Since $\underline{F} \leq f_i(x_k) \leq F(x_k) \leq \overline{F} \; \forall k \in N_1$, there exists an infinite subset $N_2 \subset N_1$ such that $F(x_k) - f_i(x_k)$, $k \in N_2$, converge.

(c) Assume first that $F(x_k) - f_i(x_k) \overset{N_2}{\to} \delta > 0$. Since

$$z(x_k;\mu_k) - f_i(x_k) \geq F(x_k) - f_i(x_k) \geq \delta/2$$

for sufficiently large $k \in N_2$, one has

$$u_i(x_k;\mu_k) = -\mu_k\varphi'(z(x_k;\mu_k) - f_i(x_k)) \leq -\mu_k\varphi'(\delta/2) \overset{N_2}{\to} 0,$$

since $\mu_k \to 0$ by Lemma 6. Since values $z(x_k;\mu_k) - f_i(x_k)$, $k \in N_2$, are bounded by Assumption 1 and Lemma 8, we obtain $u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \overset{N_2}{\to} 0$, which is a contradiction.

(d) Assume now that $F(x_k) - f_i(x_k) \overset{N_2}{\to} 0$. Since $z(x_k;\mu_k) - F(x_k) \to 0$ as $\mu_k \to 0$ by Lemma 1, we can write

$$z(x_k;\mu_k) - f_i(x_k) = (z(x_k;\mu_k) - F(x_k)) + (F(x_k) - f_i(x_k)) \overset{N_2}{\to} 0.$$

At the same time, (6) and (7) imply that values $u_i(x_k;\mu_k)$, $k \in N_2$, are bounded. Thus $u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \overset{N_2}{\to} 0$, which is a contradiction. □

Corollary 3. *Let assumptions of Theorem 1 hold. Then every cluster point $x \in R^n$ of sequence $\{x_k\}_1^\infty$ satisfies KKT conditions (3), where $u \in R^m$ is a cluster point of sequence $\{u(x_k;\mu_k)\}_1^\infty$.*

Assuming that values $\underline{\delta}$, $\underline{\varepsilon}$, $\underline{\mu}$ are nonzero and logarithmic barrier function B1 is used, we can prove the following theorem informing us about the precision obtained, when Algorithm 1 terminates.

Theorem 2. *Consider sequence $\{x_k\}_1^\infty$ generated by Algorithm 1 with the logarithmic barrier function B1. Let Assumption 1 and Assumption 2 hold. Then, choosing $\underline{\delta} > 0$, $\underline{\varepsilon} > 0$, $\underline{\mu} > 0$ arbitrarily, there is an index $k \geq 1$ such that*

$$\|g(x_k;\mu_k)\| \leq \underline{\varepsilon}, \quad |1 - e^T u(x_k;\mu_k)| \leq \underline{\delta},$$

and

$$u_i(x_k;\mu_k) \geq 0, \quad z(x_k;\mu_k) - f_i(x_k) \geq 0, \quad u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) \leq \underline{\mu}$$

for all $1 \leq i \leq m$.

Proof. Equality $|1 - e^T u(x_k;\mu_k)| \leq \underline{\delta}$ follows immediately from the fact that equation $e^T u(x_k;\mu_k) = 1$ is solved with the precision $\underline{\delta}$. Inequalities $u_i(x_k;\mu_k) \geq 0$, $z(x_k;\mu_k) - f_i(x_k) \geq 0$ follow from (7) and Lemma 1 as in the proof of Theorem 1. Since $\mu_k \to 0$ by Lemma 6 and $g(x_k;\mu_k) \to 0$ by Theorem 1, there is an index $k \geq 1$ such that $\mu_k \leq \underline{\mu}$ and $\|g(x_k;\mu_k)\| \leq \underline{\varepsilon}$ (thus Algorithm 1 terminates at the k-th iteration). Using (7) and the fact that $\varphi'(t) = -1/t$ for B1, we obtain

$$u_i(x_k;\mu_k)(z(x_k;\mu_k) - f_i(x_k)) = \frac{\mu_k}{z(x_k;\mu_k) - f_i(x_k)}(z(x_k;\mu_k) - f_i(x_k)) = \mu_k \leq \underline{\mu}.$$

□

Theorem 2 also holds for B2 and B3, since $\varphi'(t) \geq \varphi'_{B1}(t)$ for these barrier functions (see proof of Corollary 1). For B4 the upper bound is proportional to $\sqrt{\underline{\mu}}$, which again indicates that we should use μ^2 instead of μ in (17) in this case.

5 Smoothing method for large sparse minimax optimization

In this section, we briefly describe a smoothing method for large sparse minimax optimization which is algorithmically very similar to the proposed interior-point method and which will be used for a comparison. This smoothing method investigated in [20] and [22] (and in other papers quoted therein) uses smoothing function

$$S(x;\mu) = \mu \log \sum_{i=1}^m \exp\left(\frac{f_i(x)}{\mu}\right) = F(x) + \mu \log \sum_{i=1}^m \exp\left(\frac{f_i(x) - F(x)}{\mu}\right), \tag{40}$$

where $F(x)$ is given by (1) and $\mu > 0$ (we assume that $\mu \to 0$ monotonically). The following result is proved in [20].

Lemma 9. *Consider smoothing function (40). Then*

$$\nabla S(x;\mu) = A(x)\tilde{U}(x;\mu)e \tag{41}$$

and

$$\nabla^2 S(x;\mu) = \tilde{G}(x;\mu) + \frac{1}{\mu}A(x)\tilde{U}(x;\mu)A^T(x) - \frac{1}{\mu}A(x)\tilde{U}(x;\mu)ee^T\tilde{U}(x;\mu)A^T(x), \tag{42}$$

where $\tilde{G}(x;\mu) = \sum_{i=1}^m \tilde{u}_i(x;\mu)G_i(x)$, $\tilde{U}(x;\mu) = \operatorname{diag}(\tilde{u}_1(x;\mu),\dots,\tilde{u}_m(x;\mu))$, *and*

$$\tilde{u}_i(x;\mu) = \frac{\exp(f_i(x)/\mu)}{\sum_{j=1}^m \exp(f_j(x)/\mu)} = \frac{\exp((f_i(x)-F(x))/\mu)}{\sum_{j=1}^m \exp((f_j(x)-F(x))/\mu)} \tag{43}$$

for $1 \le i \le m$, *which implies* $e^T\tilde{u}(x;\mu) = 1$.

Note that (42) together with the Schwarz inequality implies

$$\begin{aligned} v^T\nabla^2 S(x;\mu)v &= v^T\tilde{G}(x;\mu)v + \frac{1}{\mu}\left(v^TA(x)\tilde{U}(x;\mu)A^T(x)v - \frac{(v^TA(x)\tilde{U}(x;\mu)e)^2}{e^T\tilde{U}(x;\mu)e}\right) \\ &\ge v^T\tilde{G}(x;\mu)v. \end{aligned}$$

Thus $\nabla^2 S(x;\mu)$ is positive definite if $\tilde{G}(x;\mu)$ is positive definite.

Using Lemma 9, we can write one step of the Newton method in the form $x^+ = x + \alpha d$ where $\nabla^2 S(x;\mu)d = -\nabla S(x;\mu)$ or

$$\left(\tilde{H} - \frac{1}{\mu}\tilde{g}\tilde{g}^T\right)d = -\tilde{g}, \tag{44}$$

where

$$\tilde{H} = \tilde{G}(x;\mu) + \frac{1}{\mu}A(x)\tilde{U}(x;\mu)A^T(x) \tag{45}$$

and $\tilde{g} = A(x)\tilde{U}(x;\mu)e$. It is evident that matrix $\tilde{H}$ has the same sparsity pattern as H in (23). Since

$$\left(\tilde{H} - \frac{1}{\mu}\tilde{g}\tilde{g}^T\right)^{-1} = \tilde{H}^{-1} + \frac{\tilde{H}^{-1}\tilde{g}\tilde{g}^T\tilde{H}^{-1}}{\mu - \tilde{g}^T\tilde{H}^{-1}\tilde{g}},$$

the solution of (44) can be written in the form

$$d = \frac{\mu}{\tilde{g}^T\tilde{H}^{-1}\tilde{g} - \mu}\tilde{H}^{-1}\tilde{g}. \tag{46}$$

If $\tilde{H}$ is not positive definite, it is advantageous to change it before computation of the direction vector. Thus we use the sparse Gill-Murray decomposition $\tilde{H} + \tilde{E} = \tilde{L}\tilde{D}\tilde{L}^T$, solve equation

$$\tilde{L}\tilde{D}\tilde{L}^Tv = \tilde{g} \tag{47}$$

and set

$$d = \frac{\mu}{\tilde{g}^T v - \mu} v. \tag{48}$$

More details concerning the smoothing method can be found in [20] and [22], where the proof of its global convergence is introduced.

The above considerations and formulas form a basis for the algorithm, which is very similar to Algorithm 1. This algorithm differs from Algorithm 1 in Step 2, where no nonlinear equation is solved (since vector $\tilde{u}(x;\mu)$ is computed directly from (43)), in Step 4, where (26)-(27) are replaced by (47)-(48), and in Step 6, where $B(x;\mu)$ is replaced by $S(x;\mu)$. Note that μ in (40) has a different meaning from μ in (17) so procedures for updating these parameters need not be identical. Nevertheless, the procedure described in Section 3 was successful in connection with the smoothing method (we have also tested procedures proposed in [20] and [22], but they were less efficient). Note finally, that the smoothing method described in this section has also insufficiencies concerning finite precision computations. If μ is small, than many evaluations of exponentials lead to underflows. This effect decreases the precision of computed gradients, which brings a problem with the termination of the iterative process. For this reason, a lower bound $\underline{\mu}$ has to be used, which is usually greater than the corresponding bound in the interior point method (we recommend $\underline{\mu} = 10^{-6}$ for the smoothing method).

6 Computational experiments

The primal interior-point method was tested by using two collections of 22 relatively difficult problems with optional dimension chosen from [18], which can be downloaded (together with the above report) from `www.cs.cas.cz/~luksan/test.html` as Test 14 and Test 15. Functions $f_i(x)$, $1 \le i \le m$, given in [18], serve for defining objective functions

$$F(x) = \max_{1 \le i \le m} f_i(x) \tag{49}$$

and

$$F(x) = \max_{1 \le i \le m} |f_i(x)| = \max_{1 \le i \le m} \left[\max(f_i(x), -f_i(x))\right]. \tag{50}$$

Function (49) is not used in connection with Test 15, since Assumption 1 is not satisfied (sometimes $F(x) \to -\infty$) in this case.

In Algorithm 1, Procedure A, Procedure B, we have used parameters $\underline{\varepsilon} = 10^{-6}$, $\underline{\delta} = 10^{-6}$, $\underline{\mu} = 10^{-10}$, $\overline{\mu} = 1$, $\underline{g} = \infty$, $\lambda = 0.85$, $\sigma = 100$, $\rho = 0.1$, $\underline{c} = 10^{-10}$, $\overline{c} = 10^{10}$, $\varepsilon_0 = 10^{-8}$, $\varepsilon_1 = 10^{-4}$, $\beta = 0.5$, $\overline{\Delta} = 1000$ as defaults (values $\underline{\mu}$ and $\overline{\Delta}$ were sometimes tuned). In the implementation of the smoothing method described in Section 5, we have used the same values with the following three exceptions: $\underline{\mu} = 10^{-6}$, $\underline{g} = 1$, $\lambda = 0.95$.

The first set of tests concerns a comparison of four primal interior point methods (Algorithm 1) based on barrier functions `B1-B4`, with the smoothing method `SM` (see (40)), the primal-dual interior point method `IP` described in [14], and the non-smooth equation method `NE` described in [15]. All these methods are implemented in the interactive system for universal functional optimization UFO [16] as line-search subroutines for discrete minimax optimization. There are two modifications: NM denotes the discrete Newton methods where the Hessian matrix is computed using gradient differences by the way described in [3] and VM denotes the variable metric methods with partitioned updates described in [8]. All mentioned subroutines use the same modules for numerical differentiation, stepsize selection, and variable metric updates. Thus the results are quite comparable. The methods listed above were tested by using medium-size test problems with 200 variables. The results of computational experiments are reported in three tables, where only summary results (over all 22 test problems) are given. Here `Method` is the method used, `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 the parameters were tuned, and `Time` is the total computational time in seconds. It is necessary to note that both the primal interior point and the smoothing algorithms used Procedure A in almost all cases. Only the variable metric versions of the primal interior point methods reported in Table 3 used Procedure B, which was more advantageous in this case.

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
B1-NM	1682	3771	11173	325	-	-	4	1.75
B2-NM	2145	6613	14333	627	-	-	8	2.42
B3-NM	2015	6825	12662	599	-	-	7	1.88
B4-NM	5650	10561	33675	648	1	-	8	4.19
SM-NM	4213	12632	32451	823	1	-	8	7.78
IP-NM	1715	3558	16943	74	1	-	10	6.05
NE-NM	5159	22195	42161	2363	2	-	14	32.86
B1-VM	1316	2873	1338	-	-	-	2	0.91
B2-VM	2225	3835	2247	2	-	-	3	1.34
B3-VM	1784	3443	1806	2	-	-	3	1.17
B4-VM	4638	8866	4658	4	-	-	3	2.05
SM-VM	7192	20710	7214	-	1	-	8	6.42
IP-VM	1805	4023	1805	12	1	-	9	5.25
NE-VM	2756	5667	2756	27	1	-	9	5.31

Table 3: Test 14: Function (49) with 200 variables

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
B1-NM	2026	5686	17096	358	-	-	6	3.52
B2-NM	3094	15109	24233	1704	-	-	6	4.89
B3-NM	3249	14141	24334	1374	-	-	7	4.72
B4-NM	5553	30984	47543	3082	-	-	9	9.31
SM-NM	3502	9497	31084	613	1	-	9	8.99
IP-NM	1792	3167	16531	158	2	-	13	8.84
NE-NM	3171	11074	25936	1256	2	-	16	18.25
B1-VM	3212	5262	3233	1	1	-	3	1.63
B2-VM	3261	5971	3283	1	1	-	3	2.49
B3-VM	2880	5491	2902	-	1	-	3	2.02
B4-VM	4612	10054	4634	3	1	-	4	2.19
SM-VM	3247	6865	3268	1	2	-	6	3.92
IP-VM	2860	7017	2861	6	2	-	8	8.28
NE-VM	3396	7009	3396	18	2	-	14	7.94

Table 4: Test 14: Function (50) with 200 variables

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
B1-NM	9031	12176	48216	2752	-	-	6	10.01
B2-NM	7973	13513	39626	5499	2	-	8	9.90
B3-NM	10787	15066	53645	7575	1	-	8	12.99
B4-NM	13865	21989	75100	10834	1	-	10	12.28
SM-NM	13186	26036	79195	4785	3	2	9	29.27
IP-NM	2046	3188	14327	1008	1	1	7	9.30
NE-NM	2914	5616	20241	1087	3	-	18	21.36
B1-VM	2432	5040	2454	1	1	-	1	2.27
B2-VM	2533	6662	2551	5	-	-	2	3.16
B3-VM	4277	11033	4298	1	-	-	3	4.28
B4-VM	5823	23001	5840	8	-	-	6	4.43
SM-VM	10769	19835	10791	1	2	-	4	23.06
IP-VM	2056	3883	2056	-	1	-	4	7.95
NE-VM	2425	3896	2425	18	3	-	12	11.77

Table 5: Test 15: Function (50) with 200 variables

The results introduced in these tables imply several conclusions. First, the use of variable metric updates is more advantageous for computing Hessian approximations than the application of gradient differences. This follows from the fact that the test problems used are strongly nonconvex. Thus the Newton methods are frequently restarted, which decreases their efficiency and, moreover, their parameters ($\underline{\mu}$ and $\overline{\Delta}$) have to be tuned more frequently. Variable metric implementations of the primal interior point method

(Algorithm 1) are very robust. They rarely require decrease of $\overline{\Delta}$ to obtain the lowest known local minimum (among other possible local minima).

Secondly, the use of barrier function B1 gives better results in a comparison with the use of B2, B3 and especially B4, even if the latter barrier functions have better theoretical properties (Condition 3 holds). Thus we restrict our considerations only to primal interior-point methods based on the logarithmic barrier function B1.

Finally, primal interior-point methods seem to be more efficient than other methods tested. Smoothing methods SM are more sensitive to the choice of their parameters, converge more slowly and require greater CPU time (since computation of exponentials is time consuming). Primal-dual interior-point methods for general nonlinear programming problems IP convert the original problem to the problem with $n+1$ basic variables, m (or $2m$ for function (50)) slack variables and the same number of equality constraints. Thus the size of linear algebra subproblems and the resulting CPU time is considerably larger. Note that our implementation of primal-dual interior-point methods for general nonlinear programming problems uses constant penalty parameter (see [14]), which has to be sometimes tuned. Thus the number NT is slightly greater for IP. The same considerations concern nonsmooth equation methods for general nonlinear programming problems NE, which are even less efficient than methods IP.

The second set of tests concerns a comparison of the primal interior-point methods (NM and VM) that use logarithmic barrier function B1 with the smoothing methods (NM and VM) and the primal-dual interior point methods (NM and VM). Large-scale test problems with 1000 variables are used. The results of computational experiments are given in six 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 two rows of every table contain summary results including the number of problems for which parameters were tuned and the total computational time in seconds.

The results introduced in these tables confirm conclusions following from the previous tables. Primal interior-point methods (especially the VM implementation) seem to be more efficient than smoothing methods and primal-dual interior point methods in all indicators. The computational time is significantly shorter, the number of the lowest known local minima attained is greater and also the number of iterations is much smaller in the case of primal interior-point methods. We believe that the efficiency of the primal interior-point methods could be even improved by using more sophisticated procedures for the barrier parameter decrease, more complicated variable metric updates, different strategies for restarts or suitable trust region realizations.

	B1-NM				SM-NM				IP-NM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	76	134	308	0.989896	737	1556	2952	0.989896	212	1310	1060	0.989896
2	80	282	567	0.127687E-24	119	187	840	0.513861E-08	647	2987	5176	0.163466E-11
3	22	23	161	0.456800E-07	75	104	532	0.105307E-10	33	35	264	0.112111E-09
4	42	54	301	0.542761	281	921	1974	0.542761	60	69	480	0.542762
5	9	10	60	0.235212E-09	55	67	336	0.148021E-07	22	22	154	0.136285E-09
6	107	116	1512	0.427133E-07	54	81	770	0.385220E-07	34	34	510	0.435320E-09
7	33	62	306	0.260162	142	459	1287	0.260162	46	46	460	0.260163
8	8	18	162	1556.50	5	93	90	1556.50	526	1390	10013	0.195540E+07
9	477	1047	8604	0.715894	1474	1489	26550	0.715983	482	3166	9158	0.753054
10	367	1755	6624	-0.512419E-01	190	608	3438	-0.340060E-01	45	166	855	0.220000
11	40	50	246	0.538829E-01	306	311	1842	0.538829E-01	47	53	329	0.538829E-01
12	205	229	824	0.996492	1141	2770	4568	0.996492	290	680	1450	0.996492
13	7	8	32	0.906020E-36	6	7	28	0.600735E-13	27	32	135	0.123786E-09
14	5	6	36	0.101420E-16	4	5	30	0.370107E-16	193	193	1351	0.500436E-04
15	13	20	56	0.399600E-02	132	263	532	0.404131E-02	403	406	2015	0.399600E-02
16	163	558	984	-0.640831E-03	2077	10414	12468	-0.637664E-03	86	97	602	-0.640931E-03
17	25	29	156	0.197131E-08	10	11	66	0.585579E-06	30	32	210	-0.894918E-09
18	32	88	198	0.136034E-07	381	1417	2292	0.590942E-06	40	42	280	0.214416E-10
19	6	16	42	42.5075	3	45	18	42.5075	20	52	140	42.5076
20	9	25	60	-0.999000E-03	61	70	372	-0.999000E-03	77	98	539	-0.998998E-03
21	12	18	78	0.599399E-02	204	851	1230	0.599399E-02	36	46	252	0.599399E-02
22	110	784	666	0.115634E-02	156	564	942	0.115634E-02	560	1926	3920	0.115635E-02
Σ	1848	5332	21983	NT = 5	7613	22293	63157	NT = 11	3916	12882	39353	NT = 9
	Time = 26.08				Time = 75.30				Time = 135.00			

Table 6: Test 14: Function (49) with 1000 variables

	B1-VM				SM-VM				IP-VM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	41	49	42	0.147225E-11	721	1582	722	0.989896	492	3357	492	0.989896
2	107	121	108	0.719942E-11	126	179	127	0.164782E-10	111	249	111	7.87703
3	49	66	50	0.198787E-07	67	79	68	0.778625E-08	42	47	42	0.135577E-08
4	96	129	97	0.542761	2024	4109	2025	0.542761	75	98	75	0.542761
5	63	91	64	0.792546E-07	46	70	47	0.104934E-07	24	24	24	0.655164E-09
6	129	162	130	0.268008E-07	58	76	59	0.123379E-06	70	70	70	0.665156E-09
7	52	86	53	0.260162	105	217	106	0.260162	59	67	59	0.260163
8	12	22	13	1556.50	6	52	6	1556.50	124	552	124	1556.50
9	79	127	80	0.712464	65	96	66	0.715983	298	989	298	0.720430
10	244	369	245	-0.404416E-01	513	630	514	-0.420237E-01	211	374	211	-0.446701E-01
11	78	100	79	0.538829E-01	109	111	110	0.538829E-01	138	326	138	0.538829E-01
12	194	346	195	0.996492	279	329	280	0.996492	98	133	98	0.996492
13	3	5	4	0.356911E-26	7	9	8	0.375969E-13	21	23	21	0.370713E-09
14	1	2	2	0.157461E-10	4	10	5	0.423678E-11	11	11	11	0.976187E-07
15	74	594	75	0.399600E-02	27	40	28	0.399600E-02	75	139	75	0.399600E-02
16	139	549	140	-0.640831E-03	1180	5610	1181	-0.637664E-03	151	195	151	-0.640931E-03
17	24	57	25	0.197120E-08	43	70	44	0.585560E-06	50	70	50	-0.896232E-09
18	50	150	51	0.136049E-07	63	142	64	0.591821E-06	63	79	63	0.308698E-10
19	10	14	11	42.5075	4	8	5	42.5075	33	80	33	42.5076
20	24	38	25	-0.999000E-03	67	74	68	-0.999000E-03	94	118	94	-0.998998E-03
21	11	15	12	0.599399E-02	157	595	158	0.599399E-02	50	80	50	0.599399E-02
22	49	153	50	0.115634E-02	37	51	38	0.115634E-02	156	266	156	0.115635E-02
Σ	1529	3245	1551	NT = 6	5708	14139	5729	NT = 8	2446	7347	2446	NT = 8
	Time = 5.64				Time = 21.55				Time = 53.97			

Table 7: Test 14: Function (49) with 1000 variables

	B1-NM				SM-NM				IP-NM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	333	519	1336	0.989896	576	960	2308	0.989896	327	3190	1635	0.989896
2	111	273	784	0.191795E-07	120	184	847	0.102769E-06	457	2259	3656	7.87703
3	43	44	308	0.402297E-07	67	99	476	0.416589E-07	108	353	864	0.144922E-09
4	46	59	329	0.542761	200	587	1407	0.542761	151	154	1208	0.542761
5	130	135	786	0.765345E-07	510	1663	3066	0.383590E-06	48	55	336	0.267541E-09
6	165	1426	2324	0.114623E-06	176	678	2478	0.405405E-06	39	40	585	0.351788E-08
7	36	78	333	0.260162	116	283	1053	0.260162	50	58	500	0.260163
8	7	8	144	1556.50	5	93	90	1556.50	137	403	2603	1794.58
9	566	897	10206	0.712465	1569	1573	28260	0.715983	70	98	1330	0.712464
10	758	5962	13662	0.164760E-02	608	1629	10962	0.322914E-02	99	399	1881	0.383609E-01
11	27	28	168	0.538829E-01	307	309	1848	0.538829E-01	100	406	700	0.538829E-01
12	243	278	976	0.996492	1632	5130	6532	0.996492	204	423	1020	0.996492
13	18	19	76	0.249715E-07	7	8	32	0.218524E-11	31	38	155	0.520073E-09
14	5	31	36	0.393648E-11	1	22	6	0.393649E-11	197	197	1379	0.116727E-04
15	71	271	288	0.399600E-02	471	960	1888	0.400051E-02	489	562	2445	0.399600E-02
16	115	359	696	0.177635E-14	298	566	1794	0.168724E-13	106	130	742	0.130652E-07
17	84	519	510	0.158367E-06	86	266	522	0.294551E-06	154	352	1078	0.998658E-07
18	234	1780	1410	0.421100E-07	221	757	1332	0.354858E-06	349	1562	2443	0.385079E-06
19	6	16	42	42.5075	3	45	18	42.5075	31	95	217	42.5076
20	298	2245	1794	0.912428E-04	729	1607	4380	0.937049E-04	365	837	2555	0.656297E-04
21	10	12	66	0.599399E-02	25	52	156	0.599399E-02	30	51	210	0.599399E-02
22	77	449	468	0.115634E-02	434	1560	2610	0.115634E-02	1094	4690	7658	0.115635E-02
Σ	3383	15408	36742	NT = 7	8161	19031	72065	NT = 13	4636	16352	35200	NT = 13
	Time = 50.05				Time = 100.08				Time = 164.92			

Table 8: Test 14: Function (50) with 1000 variables

	B1-VM				SM-VM				IP-VM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	42	50	43	0.989896	757	1654	758	0.989896	76	85	76	0.989896
2	94	120	95	0.262006E-07	139	188	140	0.113991E-06	150	211	150	7.87703
3	68	69	69	0.321536E-07	69	82	70	0.533455E-07	50	54	50	0.357158E-07
4	77	98	78	0.542761	567	707	568	0.542761	63	93	63	0.542761
5	99	102	100	0.103403E-06	109	255	110	0.208752E-06	41	41	41	0.693972E-09
6	88	128	89	0.115652E-06	90	162	91	0.579359E-06	68	69	68	0.117520E-08
7	34	63	35	0.260162	136	350	137	0.260162	42	42	42	0.260163
8	25	65	26	1556.50	6	52	6	1556.50	185	230	185	1556.50
9	78	137	79	0.712464	66	105	67	0.715983	180	572	180	0.713574
10	70	91	71	0.399467E-13	901	3138	902	0.166564E-02	322	433	322	0.533903E-03
11	68	75	69	0.538829E-01	345	347	346	0.538829E-01	135	277	135	0.538829E-01
12	304	829	305	0.996492	260	935	261	0.996492	272	967	272	0.996492
13	16	17	17	0.623304E-07	6	7	7	0.156979E-11	17	17	17	0.418384E-05
14	5	7	6	0.361730E-11	4	66	5	0.391237E-11	12	12	12	0.105667E-07
15	36	38	37	0.399600E-02	34	77	35	0.399600E-02	80	94	80	0.399600E-02
16	31	82	32	0.177635E-14	359	838	360	0.198753E-11	102	103	102	0.102185E-08
17	27	35	28	0.195210E-07	75	83	76	0.124384E-05	61	70	61	0.101459E-06
18	46	54	47	0.111068E-06	128	288	129	0.917438E-06	187	229	187	0.161298E-06
19	9	18	10	42.5075	4	8	5	42.5075	40	129	40	42.5076
20	417	443	418	0.149988E-03	464	932	465	0.667921E-04	222	285	222	0.256358E-04
21	12	17	13	0.599399E-02	69	236	70	0.599399E-02	48	78	48	0.599399E-02
22	39	45	40	0.115634E-02	42	68	43	0.115634E-02	131	212	131	0.115635E-02
Σ	1685	2583	1707	NT = 4	4630	10578	4651	NT = 8	2484	4303	2484	NT = 8
	Time = 5.39				Time = 31.05				Time = 55.22			

Table 9: Test 14: Function (50) with 1000 variables

	B1-NM				SM-NM				IP-NM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	3542	4253	14172	0.444089E-14	204	653	820	0.900000	39	41	195	0.900000
2	107	121	540	0.193178E-12	132	328	665	1.74188	47	95	282	1.74189
3	19	20	120	0.111676E-07	39	75	240	0.100165E-06	49	51	343	0.248945E-07
4	46	123	188	0.445071	144	504	580	0.445072	103	103	515	0.445072
5	8	12	54	0.258110E-11	17	21	108	0.124175E-09	36	36	252	0.754023E-09
6	13	15	196	0.333066E-15	30	37	434	0.134812E-09	40	44	600	0.460345E-08
7	39	80	160	8.73314	103	348	416	8.73314	29	32	145	8.73314
8	106	163	321	0.999446E-06	101	224	306	0.122089E-05	183	511	732	0.377079E-07
9	254	744	1785	1.86765	142	321	1001	1.91698	180	388	1440	1.93861
10	111	300	672	0.679030	2363	2401	14184	0.679030	31	79	217	0.679030
11	5006	6498	20028	0.216937E-11	279	896	1120	1.79469	69	290	345	0.125826E-08
12	111	163	672	9.23541	250	424	1506	9.23541	55	57	385	9.23541
13	129	473	780	18.7973	323	1116	1944	21.9682	67	70	469	21.9683
14	49	127	350	0.459990	38	75	273	0.459990	23	23	184	0.459990
15	54	91	385	0.230607	231	273	1624	0.230607	27	27	216	0.230608
16	105	261	742	0.666667E-01	75	135	532	0.666669E-01	39	39	312	0.666667E-01
17	1049	2035	9450	0.195263E-02	46	72	423	0.696491E-15	71	73	710	0.567312E-09
18	2553	2908	15324	0.310862E-14	5681	13878	34092	0.186517E-12	124	134	868	0.156724E-08
19	35	58	396	0.549671E-10	100	292	1111	0.321303E-09	53	53	636	0.399549E-09
20	283	284	1704	0.999966	720	1929	4326	0.999970	121	186	847	0.999796
21	32	82	198	0.900000	165	450	996	0.900000	58	143	406	0.900000
22	82	290	492	2.35487	590	634	3546	2.40743	65	65	455	2.35487
Σ	13733	19101	68729	NT = 8	11773	25086	70247	NT = 10	1509	2540	10554	NT = 15
	Time = 59.75				Time = 117.17				Time = 84.13			

Table 10: Test 15: Function (50) with 1000 variables

	B1-VM				SM-VM				IP-VM			
P	NIT	NFV	NFG	F	NIT	NFV	NFG	F	NIT	NFV	NFG	F
1	2760	3059	2761	0.00000	148	417	149	0.900000	73	121	73	0.763802E-09
2	56	88	57	1.74188	53	75	54	0.412225E-11	49	125	49	1.74189
3	26	30	27	0.229997E-10	47	64	48	0.183372E-09	45	57	45	0.227357E-08
4	57	105	58	0.445071	256	814	257	0.445072	67	68	67	0.445072
5	7	8	8	0.333066E-15	20	32	21	0.671296E-12	46	46	46	0.231353E-08
6	37	171	38	0.777156E-15	30	69	31	0.156140E-10	40	40	40	0.191487E-08
7	40	171	40	8.73314	106	524	107	8.73314	35	38	35	8.73314
8	88	248	89	0.879442E-06	87	135	88	0.254255E-06	144	145	144	0.175868E-07
9	267	336	268	1.93825	162	199	163	1.92755	63	157	63	170.723
10	87	137	88	0.679030	2303	2350	2303	0.679030	52	69	52	0.679030
11	3453	4631	3454	0.444089E-14	308	1130	309	1.79469	45	128	45	0.657288E-08
12	53	151	54	9.23541	329	1010	330	9.23541	63	134	63	9.23541
13	61	133	62	21.9682	917	2041	917	21.9682	280	2498	280	21.9683
14	30	92	31	0.459990	36	108	37	0.459990	22	22	22	0.459990
15	34	68	35	0.230607	262	285	263	0.230607	20	20	20	0.230608
16	35	73	36	0.666667E-01	45	74	46	0.666669E-01	46	46	46	0.666667E-01
17	69	269	70	0.518757E-11	56	110	57	0.943689E-15	93	95	93	0.158363E-09
18	260	340	261	0.532907E-14	1036	2055	1037	0.00000	327	1470	327	1.73669
19	48	52	49	0.264719E-10	46	83	47	0.866158E-10	71	71	71	0.217062E-08
20	265	1975	266	0.710542E-13	107	121	108	0.568434E-12	157	173	157	0.414677E-07
21	47	178	47	0.900000	129	599	130	0.900000	77	130	77	0.900000
22	80	174	81	2.35487	74	198	75	2.35487	41	44	41	2.35487
Σ	7860	12489	7880	NT = 3	6557	12493	6577	NT = 8	1856	5697	1856	NT = 7
	Time = 29.86				Time = 75.05				Time = 111.38			

Table 11: Test 15: Function (50) with 1000 variables

References

[1] A.Ben-Tal, G.Roth: A truncated log barrier algorithm for large-scale convex programming and minimax problems: Implementation and computational results. Optimization Methods and Software 6 (1996) 283-312.

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

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

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

[6] Y.Gao, X.Li: Nonsmooth equations approach to a constrained minimax problem. Applications of Mathematics 50 5-130.

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

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

[9] A.Griewank, f.corliss: Automatic Differentiation of Algorithms: Theory, Implementation, and Application. SIAM, Philadelphoa 1991.

[10] S.P.Han: Variable metric methods for minimizing a class of nondifferentiable functions. Mathematical programming 20 (1981) 1-13.

[11] K.Jónasson, K.Madsen: Corrected sequential linear programming for sparse minimax optimization. BIT 34 (1994) 372-387.

[12] D.Le: Three new rapidly convergent algorithms for finding a zero of a function. SIAM J. on Scientific and Statistical Computaions 6 (1985) 193-208.

[13] D.Le: An efficient derivative-free method for solving nonlinear equations. ACM Transactions on Mathematical Software 11 (1985) 250-262.

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

[15] 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.

[16] L.Lukšan, M.Tůma, J.Hartman, J.Vlček, N.Ramešová, M.Šiška, C.Matonoha: Interactive System for Universal Functional Optimization (UFO). Version 2006, Report No. V-977, Prague, ICS AS CR, 2006.

[17] L.Lukšan, E.Spedicato: Variable metric methods for unconstrained optimization and nonlinear least squares. Journal of Computational and Applied Mathematics 124 (2000) 61-93.

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

[19] 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.

[20] E.Polak, J.O.Royset, R.S.Womersley: Algorithm with adaptive smoothing for finite minimax problems. Journal of Optimization Theory and Applications 119 (2003) 459-484.

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

[22] S.Xu: Smoothing methods for minimax problems. Computational Optimization and Applications 20 (2001) 267-279.