KYBERNETIKA — VOLUME 45 (2009), NUMBER 5, PAGES 841 – 864

PRIMAL INTERIOR–POINT METHOD FOR LARGE SPARSE MINIMAX OPTIMIZATION

Ladislav Lukšan, Ctirad Matonoha and Jan Vlček

In this paper, we propose a primal interior-point method for large sparse minimax optimization. After a short introduction, 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 paper confirm efficiency and robustness of the proposed method.

Keywords: unconstrained optimization, large-scale optimization, minimax optimization, nonsmooth optimization, interior-point methods, modified Newton methods, variable metric methods, computational experiments

AMS Subject Classification: 9K35, 90C06, 90C47, 90C51

1. INTRODUCTION

Consider the minimax problem: Minimize a function

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

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

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

Assumption 2. Functions $f_i(x)$, $1 \le i \le m$, are twice continuously differentiable on the convex hull of the level set $\mathcal{L}(\overline{F}) = \{x \in \mathbb{R}^n : F(x) \le \overline{F}\}$ 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 $x \in \mathrm{conv}\mathcal{L}(\overline{F})$ where $\|.\|$ denotes the Euclidean norm.

In this paper, 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, where n_i is a small number independent of n).

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

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

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 [4, 12]; sequential quadratic programming [7, 11]; interior-point [15, 21]; nonsmooth equation [5, 16]). In this paper, we introduce a feasible primal interior-point method that utilizes a special structure of minimax problem (1). Constrained problem (2) is replaced by a sequence of unconstrained problems

$$\text{minimize} \quad B(x, z; \mu) = z - \mu \sum_{i=1}^m \log(z - f_i(x)) \quad \text{subject to} \quad z > F(x), \tag{4}$$

where $0 < \mu \le \overline{\mu}$ (we assume that $\mu \to 0$ monotonically). Inequality $z > F(x)$ is assured algorithmically.

A primal interior-point method investigated in this paper is based on line search minimization of a logarithmic barrier function derived from the minimax problem structure. Approximation of the Hessian matrix of this barrier function is obtained by partitioned variable metric updates [10]. The resulting algorithm whose efficiency is confirmed by extensive computational experiments is described in detail.

The paper 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, 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 PDIP proposed in [15] and the KNITRO software [2] for a comparison. The last two methods, intended for solving general nonlinear programming problems, were applied to 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

$$\sum_{i=1}^m \frac{\mu}{z - f_i(x)} \nabla f_i(x) = 0, \quad 1 - \sum_{i=1}^m \frac{\mu}{z - f_i(x)} = 0. \tag{5}$$

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

$$f(x) = \begin{bmatrix} f_1(x) \\ \ldots \\ f_m(x) \end{bmatrix}, \quad u(x,z;\mu) = \begin{bmatrix} \mu/(z - f_1(x)) \\ \ldots \\ \mu/(z - f_m(x)) \end{bmatrix}, \quad e = \begin{bmatrix} 1 \\ \ldots \\ 1 \end{bmatrix}, \tag{6}$$

we can write (5) in the form

$$g(x,z;\mu) \triangleq A(x)\,u(x,z;\mu) = 0, \quad \gamma(x,z;\mu) \triangleq 1 - e^T u(x,z;\mu) = 0. \tag{7}$$

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

$$\begin{aligned}
\frac{\partial g(x,z;\mu)}{\partial x} &= \sum_{i=1}^m u_i(x,z;\mu) G_i(x) + \sum_{i=1}^m \frac{\mu}{(z - f_i(x))^2}\, g_i(x) g_i^T(x) \\
&= G(x,z;\mu) + A(x) V(x,z;\mu) A^T(x), \\
\frac{\partial g(x,z;\mu)}{\partial z} &= -\sum_{i=1}^m \frac{\mu}{(z - f_i(x))^2}\, g_i(x) = -A(x) V(x,z;\mu) e, \\
\frac{\partial \gamma(x,z;\mu)}{\partial x} &= -\sum_{i=1}^m \frac{\mu}{(z - f_i(x))^2}\, g_i^T(x) = -e^T V(x,z;\mu) A^T(x), \\
\frac{\partial \gamma(x,z;\mu)}{\partial z} &= \sum_{i=1}^m \frac{\mu}{(z - f_i(x))^2} = e^T V(x,z;\mu) e,
\end{aligned}$$

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

$$V(x,z;\mu) = \mathrm{diag}(\mu/(z - f_1(x))^2, \ldots, \mu/(z - f_m(x))^2).$$

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

$$\begin{bmatrix} G(x,z;\mu) + A(x)V(x,z;\mu)A^T(x) & -A(x)V(x,z;\mu)e \\ -e^T V(x,z;\mu) A^T(x) & e^T V(x,z;\mu) e \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} = - \begin{bmatrix} g(x,z;\mu) \\ \gamma(x,z;\mu) \end{bmatrix}. \tag{8}$$

Note that matrix $V(x,z;\mu)$ is positive definite.

Increments Δx and Δz determined from (8) 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 $\tilde{B}(z;\mu) = B(x,z;\mu)$ with $x \in \mathbb{R}^n$ fixed.

Lemma 2.1. Function $\tilde{B}(z;\mu) : (F(x),\infty) \to R$ has a unique stationary point, which is its global minimizer. This stationary point is characterized by the equation

$$e^T u(x,z;\mu) = 1. \tag{9}$$

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

$$F(x) + \mu = \underline{z}(x;\mu) \leq z(x;\mu) \leq \overline{z}(x;\mu) = F(x) + m\mu. \tag{10}$$

Moreover,

$$e^T u(x,\overline{z}(x;\mu);\mu) \leq 1 \leq e^T u(x,\underline{z}(x;\mu);\mu). \tag{11}$$

Proof. Function $\tilde{B}(z;\mu) : (F(x),\infty) \to \mathbb{R}$ is strictly convex in $(F(x),\infty)$, since it is a sum of linear and strictly convex functions. Thus if a stationary point of $\tilde{B}(z;\mu)$ exists, it is its unique global minimizer. Let $z(x;\mu)$ be the solution of equation (9) (which has the form (5)). Then

$$\frac{\mu}{z(x;\mu) - F(x)} \leq \sum_{i=1}^{m} \frac{\mu}{z(x;\mu) - f_i(x)} = 1 \leq \sum_{i=1}^{m} \frac{\mu}{z(x;\mu) - F(x)} = \frac{m\mu}{z(x;\mu) - F(x)}.$$

Thus $F(x) + \mu \leq z(x;\mu) \leq F(x) + m\mu$, which proves (10), and

$$\sum_{i=1}^{m} \frac{\mu}{\overline{z}(x;\mu) - f_i(x)} \leq 1 \leq \sum_{i=1}^{m} \frac{\mu}{\underline{z}(x;\mu) - F(x)},$$

which proves (11). Inequalities (11) imply that the solution $z(x;\mu)$ of (9) (the stationary point of $\tilde{B}(z;\mu)$) exists. □

Solution $z(x;\mu)$ of nonlinear equation (9) can be obtained by efficient methods proposed in [13, 14], which use localization inequalities (10) – (11). Therefore, we can assume to have $z = z(x;\mu)$ with a sufficient precision, which implies that the last element of the right-hand side in (8) is negligible. Assuming $z = z(x;\mu)$, we denote

$$B(x;\mu) = B(x,z(x;\mu);\mu) = z(x;\mu) - \mu \sum_{i=1}^{m} \log(z(x;\mu) - f_i(x)), \tag{12}$$

$u(x;\mu) = u(x,z(x;\mu);\mu)$, $V(x;\mu) = V(x,z(x;\mu);\mu)$ 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 $(x,z) \in \mathbb{R}^{n+1}$ of $B(x,z;\mu)$, it suffices to minimize $B(x;\mu)$ over $\mathbb{R}^n$.

Lemma 2.2. Consider barrier function (12). Then

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

and

$$\nabla^2 B(x;\mu) = G(x;\mu) + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^TV(x;\mu)A^T(x)}{e^TV(x;\mu)e}. \tag{14}$$

Solution Δx of the Newton equation

$$\nabla^2 B(x;\mu)\Delta x = -\nabla B(x;\mu) \tag{15}$$

is equal to the corresponding vector obtained by solving (8) with $z = z(x;\mu)$.

P r o o f. Differentiating $B(x;\mu)$, we obtain

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

since

$$1 - e^T u(x;\mu) = 1 - \sum_{i=1}^{m} \frac{\mu}{z(x;\mu) - f_i(x)} = 0.$$

Differentiating the last equality, one has

$$\sum_{i=1}^{m} \frac{\mu}{(z(x;\mu) - f_i(x))^2} \left(\nabla z(x;\mu) - g_i(x)\right) = 0,$$

which gives

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

Thus

$$\begin{aligned}
\nabla^2 B(x;\mu) &= \sum_{i=1}^{m} u_i(x;\mu)\, G_i(x) + \sum_{i=1}^{m} \frac{\mu}{(z(x;\mu) - f_i(x))^2} \left(g_i(x) - \nabla z(x;\mu)\right) g_i^T(x) \\
&= G(x;\mu) + A(x)V(x;\mu)A^T(x) - \frac{A(x)V(x;\mu)ee^T V(x;\mu)A^T(x)}{e^T V(x;\mu)e}.
\end{aligned}$$

Using the second equation of (8) 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 (15) □

Note that we use (8) rather than (15) for a direction determination since nonlinear equation (9) is solved with precision $\underline{\delta}$ and, therefore, in general $1 - e^T u(x;\mu)$ differs from zero.

Lemma 2.3. Let Δx solve (15) (or (8) 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 (15) 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 $v \in \mathbb{R}^n$ by (14) 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. □

Equality

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

where $U(x;\mu) = \mathrm{diag}(u_1(x;\mu), \ldots, u_m(x;\mu))$, 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 (14)). 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 $\|\nabla^2 B(x;\mu)\|$.

Lemma 2.4. Let Assumption 2 be satisfied. If $\mu \geq \underline{\mu} > 0$, then

$$\|\nabla^2 B(x;\mu)\| \leq m(\overline{G} + \overline{g}^2 \|V(x;\mu)\|) \leq m\left(\overline{G} + \frac{\overline{g}^2}{\underline{\mu}}\right).$$

Proof. Using (14) and Assumption 2, we obtain

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

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

$$\|V(x;\mu)\| = \frac{\mu}{(z(x;\mu) - F(x))^2} \leq \frac{1}{\mu} \leq \frac{1}{\underline{\mu}}$$

using (10). □

Now we return to the direction determination. To simplify the notation, we write equation (8) in the form

$$\begin{bmatrix} H & -a \\ -a^T & \alpha \end{bmatrix} \begin{bmatrix} d \\ \delta \end{bmatrix} = - \begin{bmatrix} g \\ \gamma \end{bmatrix} \tag{16}$$

where

$$H = G + A(x)V(x,z;\mu)A^T(x), \quad G = G(x,z;\mu), \tag{17}$$

$a = A(x)V(x,z;\mu)e$, $\alpha = e^T V(x,z;\mu)e$, and $d = \Delta x$, $\delta = \Delta z$. Since

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

where $\eta = a^T H^{-1} a - \alpha$ we can write

$$\begin{bmatrix} d \\ \delta \end{bmatrix} = -\begin{bmatrix} H & -a \\ -a^T & \alpha \end{bmatrix}^{-1} \begin{bmatrix} g \\ \gamma \end{bmatrix} = \begin{bmatrix} H^{-1}(a\,\delta - g) \\ \delta \end{bmatrix}, \tag{18}$$

where

$$\delta = \eta^{-1}(a^T H^{-1} g + \gamma).$$

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 [6] of the form

$$H + E = LDL^T, \tag{19}$$

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

$$LDL^T c = a, \quad LDL^T v = g \tag{20}$$

and set

$$\delta = \frac{a^T v + \gamma}{a^T c - \alpha}, \quad d = c\,\delta - v. \tag{21}$$

The above analysis gives a reason for high efficiency of the primal interior point method for partially separable minimax optimization. We decompose a sparse matrix $H = G + AVA^T$ of order n and use back substitution (20). If a primal-dual interior point method is applied to equivalent problem (2), an equation with matrix

$$\begin{bmatrix} \tilde{G} & \tilde{A} \\ \tilde{A}^T & -M \end{bmatrix}, \quad \tilde{G} = \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{A} = \begin{bmatrix} A \\ -e^T \end{bmatrix}$$

of order $n + m + 1$ has to be solved. Further elimination is impossible, since matrix $\tilde{G}$ is singular, $\tilde{A}$ has a dense row, and some elements of the diagonal matrix M tends to zero. Therefore, we have to use either the Bunch–Parlett decomposition [1] (often with a large fill-in) or the iterative CG algorithm with a suitable (indefinite) preconditioner. For this reason, the number of operations and the computational time per iteration are much larger if a primal-dual interior point method is used.

In (17), we assume that $G = G(x,z;\mu)$, where $G(x,z;\mu)$ is given either 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 this paper, we confine our attention to partitioned variable metric updates described in [10]. In this case, G is defined by the expression

$$G = \sum_{i=1}^{m} u_i(x;\mu) G_i, \tag{22}$$

where approximations G_i of $\nabla^2 f_i(x)$ are successively updated (more details are given in the next section).

Now we are in a position to describe the basic algorithm. This algorithm provides restarts if the direction vector $d \in \mathbb{R}^n$ does not satisfy the inequality

$$-g^T d \geq \varepsilon_0 \|g\| \|d\| \tag{23}$$

(with a suitable constant $0 < \varepsilon_0 < 1$). The step-length $\alpha > 0$ is obtained by the Goldstein rule [8] in such a way that

$$B(x + \alpha d;\mu) \leq B(x;\mu) + \varepsilon_1 \alpha g^T d \tag{24}$$

(with a suitable constant $0 < \varepsilon_1 < 1/2$) and either

$$B(x + \alpha d;\mu) \geq B(x;\mu) + (1 - \varepsilon_1) \alpha g^T d \tag{25}$$

or $\alpha = \overline{\Delta}/\|d\|$ (if (25) implies $\alpha > \overline{\Delta}/\|d\|$) hold.

Algorithm 1.

Data: Termination parameter $\underline{\varepsilon} > 0$, precision of 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 parameter $\overline{\varepsilon}_0 > 0$, 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. Choose $\mu \leq \overline{\mu}$. 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 $r := 1$ (restart indicator).

Step 2: Termination. Solve nonlinear equation (9) 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 Hessian matrices. If $r = 0$, update matrices G_i, $1 \leq i \leq m$, by using partitioned variable metric updates and compute G by (22). If $r = 1$, set $G_i := I_i$, $1 \leq i \leq m$, (I_i are unit matrices of desired orders) and compute G by (22). If $r = 2$, set $G := I$.

Step 4: Direction determination. Determine matrix H by (17). Determine vector d from (20)–(21) by using the Gill–Murray decomposition (19) of matrix H.

Step 5: Restart. If $r < 2$ and (23) does not hold, then set $r := r + 1$ and go to Step 3, else set $r := 0$.

Step 6: Step-length selection. Select step-length α in such a way that (24) and either (25) or $\alpha = \overline{\Delta}/\|d\|$ hold (note that nonlinear equation (9) has to be solved at the point $x + \alpha d$). Set $x := x + \alpha d$. 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 has only theoretical significance since it is necessary for proving the global convergence of Algorithm 1. Nevertheless, restarts appear rarely (approximately one restart after 2500 iterations as it is shown in Table 3 – Table 5 in Section 6) and have no practical influence on the efficiency of Algorithm 1.

Since function $B(x; \mu)$ is smooth, the line search utilized in Step 6 always finds a step-length satisfying the Goldstein conditions (24) and either (25) or $\alpha = \overline{\Delta}/\|d\|$. The use of $\overline{\Delta}$ 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 an upper bound for 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 $\overline{\Delta}$ can cause a better local solution to be reached. Therefore, $\overline{\Delta}$ is a parameter, which has to be sometimes adjusted.

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 2.3) that direction vector d obtained by solving equation (16) is descent for $B(x; \mu)$ if matrix $G(x; \mu)$ is positive definite. Matrix (22) obtained by partitioned variable metric updates is always positive definite, but it can be ill conditioned. Thus inequality (23) may not be satisfied. If (23) does not hold after the first restart, we set $G = I$. In this case, the approximation B of $\nabla^2 B(x; \mu)$ has the form

$$B = I + A(x)V(x; \mu)A^T(x) - \frac{A(x)V(x; \mu)ee^T V(x; \mu)A^T(x)}{e^T V(x; \mu)e} \tag{26}$$

and the minimum eigenvalue of B is not less than 1 (see proof of Lemma 2.3). Using the same way as in the proof of Lemma 2.4, we can write

$$\kappa(B) \leq \|B\| \leq \left\|I + A(x)V(x; \mu)A^T(x)\right\| \leq 1 + m\overline{g}^2\|V(x; \mu)\| \leq 1 + \frac{m\overline{g}^2}{\underline{\mu}} \tag{27}$$

(procedure used in Step 7 of Algorithm 1 assures that $\mu \geq \underline{\mu}$). If d solves equation $Bd + g = 0$, then (27) implies the inequality

$$-g^T d \geq \frac{1}{\kappa(B)} \|g\| \|d\| \geq \frac{\underline{\mu}}{\underline{\mu} + m\overline{g}^2} \|g\| \|d\|. \tag{28}$$

Thus inequalities (23) and (28) imply that direction vectors are uniformly descent and the following lemma (proved, e. g., in [3]), holds.

Lemma 3.1. If Assumption 1 and Assumption 2 hold, the Goldstein line search (Step 6 of Algorithm 1) assures that a constant c exists such that

$$B(x_{k+1}; \mu_k) - B(x_k; \mu_k) \leq -c \|g(x_k; \mu_k)\|^2 \quad \forall k \in \mathbb{N} \tag{29}$$

(note that $g(x_k; \mu_k) = \nabla B(x_k; \mu_k)$ by Lemma 2.2).

If only Assumption 1 hold, the Goldstein line search implies weaker inequality

$$B(x_{k+1}; \mu_k) - B(x_k; \mu_k) \leq 0 \quad \forall k \in \mathbb{N}. \tag{30}$$

Matrix G appearing in Step 3 of Algorithm 1 is computed by using partitioned variable metric updates described in [10]. In our implementation, we use safeguarded scaled BFGS updates. In this case, G is given by (22). Let $\mathbb{R}_i^n \subset \mathbb{R}^n$, $1 \leq i \leq m$, be subspaces defined by independent variables of functions f_i and $Z_i \in \mathbb{R}^{n \times n_i}$ 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 $\tilde{G}_i = Z_i^T G_i Z_i$, $1 \leq i \leq m$. New reduced approximations of the Hessian matrices, used in the next iteration, are computed by the formulas

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

where

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

and where either $\tilde{\gamma}_i = 1$ or $\tilde{\gamma}_i = \tilde{s}_i^T \tilde{G}_i \tilde{s}_i / \tilde{s}_i^T \tilde{y}_i$. (we denote quantities from the next iteration by $+$). The particular choice of $\tilde{\gamma}_i$ is determined by the controlled scaling strategy described in [18]. In the first iteration we set $\tilde{G}_i = I_i$, $1 \leq i \leq m$, where I_i are unit matrices of suitable orders. Finally, $G_i^+ = Z_i \tilde{G}_i^+ Z_i^T$, $1 \leq i \leq m$.

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 $F(x) + \mu \leq z(x; \mu) \leq F(x) + m\mu$ and $\mu \to 0$, 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. Denoting by $s(x;\mu) = z(x;\mu)e - f(x)$ vector of slack variables, we can see from (6) 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

$$\mu_{k+1} = \max\left(\tilde{\mu}_{k+1}, \underline{\mu}\right), \tag{31}$$

where

$$\tilde{\mu}_{k+1} = \min\left[\max(\lambda\mu_k, \mu_k/(\sigma\mu_k + 1)), \max(\|g(x_k;\mu_k)\|^2, 10^{-2k})\right]. \tag{32}$$

Values $\underline{\mu} = 10^{-10}$, $\lambda = 0.85$, $\sigma = 100$ 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

$$\mu_{k+1} = \max(\underline{\mu}, \|g_k\|^2). \tag{33}$$

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. Formula (32) 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. Note that sequence $\{\mu_k\}_1^\infty$ obtained by Procedure A or Procedure B is non-increasing and bounded by $\mu_1 \le \overline{\mu}$.

4. GLOBAL CONVERGENCE

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

Lemma 4.1. Let $z(x;\mu)$ be a solution of equation (9) (for fixed x and variable μ), i. e., $1 - e^T u(x, z(x;\mu);\mu) = 0$. Then

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

P r o o f. Differentiating the second equation in (5), which has the form

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

by μ, we obtain

$$-\sum_{i=1}^{m} \frac{1}{z(x;\mu) - f_i(x)} + \sum_{i=1}^{m} \frac{\mu}{(z(x;\mu) - f_i(x))^2} \frac{\partial z(x;\mu)}{\partial \mu} = 0,$$

which gives

$$\frac{\partial z(x;\mu)}{\partial \mu} = \left(\sum_{i=1}^{m} \frac{\mu^2}{(z(x;\mu) - f_i(x))^2} \right)^{-1} = \left(\sum_{i=1}^{m} u_i^2(x;\mu) \right)^{-1} > 0.$$

Differentiating function

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

one has

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

□

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

$$B(x_{k+1};\mu_{k+1}) \leq B(x_{k+1};\mu_k) + L(\mu_k - \mu_{k+1}) \tag{34}$$

for all $k \in \mathbb{N}$.

P r o o f. (a) Using (12) and Assumption 1, we can write

$$\begin{aligned} B(x;\mu) - \underline{F} &= z(x;\mu) - \underline{F} - \mu \sum_{i=1}^{m} \log(z(x;\mu) - f_i(x)) \\ &\geq z(x;\mu) - \underline{F} - m\mu \log(z(x;\mu) - \underline{F}). \end{aligned}$$

Convex function $\psi(t) = t - m\mu\log(t)$ has a unique minimum at a point $t = m\mu$, since $\psi'(m\mu) = 1 - m\mu/m\mu = 0$. Thus

$$\begin{aligned} B(x;\mu) &\geq \underline{F} + m\mu - m\mu\log(m\mu) \geq \underline{F} + \min\,(0,\, m\mu_1(1 - \log(m\mu_1)) \\ &\geq \underline{F} + \min\,(0,\, m\mu_1(1 - \log(2m\mu_1)) \triangleq \underline{B}. \end{aligned} \tag{35}$$

Boundedness of $z(x;\mu)$ and $F(x)$ from below follows from Assumption 1 and inequalities (10).
(b) As in (a), we can write

$$B(x;\mu) - \underline{F} \geq \frac{z(x;\mu) - \underline{F}}{2} + \frac{z(x;\mu) - \underline{F}}{2} - m\mu\log(z(x;\mu) - \underline{F}).$$

The convex function $\tilde{\psi}(t) = t/2 - m\mu\log(t)$ has a unique minimum at a point $t = 2m\mu$. Thus

$$B(x;\mu) \geq \frac{z(x;\mu) - \underline{F}}{2} + \underline{F} + \min\,(0,\, m\mu_1(1 - \log(2m\mu_1)) = \frac{z(x;\mu) - \underline{F}}{2} + \underline{B}$$

or

$$z(x;\mu) - \underline{F} \leq 2(B(x;\mu) - \underline{B}) \tag{36}$$

(c) Using the mean value theorem and Lemma 4.1, we obtain

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k) &= \sum_{i=1}^{m} \log(z(x_{k+1},\tilde{\mu}_k) - f_i(x_{k+1}))(\mu_k - \mu_{k+1}) \\ &\leq \sum_{i=1}^{m} \log(z(x_{k+1};\mu_k) - f_i(x_{k+1}))(\mu_k - \mu_{k+1}) \\ &\leq m\log(z(x_{k+1};\mu_k) - \underline{F})(\mu_k - \mu_{k+1}), \end{aligned} \tag{37}$$

where $0 < \mu_{k+1} \leq \tilde{\mu}_k \leq \mu_k$. Using the relation $\log(t) \leq t/e$ (where $e = \exp(1)$), inequalities (30), (36) and (37) imply

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - \underline{B} &\leq B(x_{k+1};\mu_k) - \underline{B} + m\log(z(x_{k+1};\mu_k) - \underline{B})(\mu_k - \mu_{k+1}) \\ &\leq B(x_{k+1};\mu_k) - \underline{B} + me^{-1}(z(x_{k+1};\mu_k) - \underline{B})(\mu_k - \mu_{k+1}) \\ &\leq B(x_{k+1};\mu_k) - \underline{B} + 2me^{-1}(B(x_{k+1};\mu_k) - \underline{B})(\mu_k - \mu_{k+1}) \\ &= (1 + \lambda\delta_k)(B(x_{k+1};\mu_k) - \underline{B}) \\ &\leq (1 + \lambda\delta_k)(B(x_k;\mu_k) - \underline{B}), \end{aligned}$$

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

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

and since

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

the above product is finite. This together with (10) and (36) 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.
(d) Using (37), we can write

$$\begin{aligned} B(x_{k+1};\mu_{k+1}) - B(x_{k+1};\mu_k) &\leq m\log(z(x_{k+1};\mu_k) - \underline{F})(\mu_k - \mu_{k+1}) \\ &\leq m\log(F(x_{k+1}) + m\mu_k - \underline{F})(\mu_k - \mu_{k+1}) \\ &\leq m\log(\overline{F} + m\mu_1 - \underline{F})(\mu_k - \mu_{k+1}) \\ &\triangleq L(\mu_k - \mu_{k+1}), \end{aligned}$$

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

The assertion of Lemma 4.2 does not depend on bounds $\overline{g}$ and $\overline{G}$, since Assumption 2 is not used. Thus an upper bound $\overline{F}$ (independent of $\overline{g}$ and $\overline{G}$) exists such that $F(x_k) \leq \overline{F}$ for all $k \in \mathbb{N}$. This bound can be used for the definition of the level set in Assumption 2.

Lemma 4.3. Let Assumption 1 and Assumption 2 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. In Phase 1, the value of μ is fixed. Since the function $B(x;\mu)$ is continuous, bounded from below by Lemma 4.2, and since (29) (with $\mu_k = \mu$) holds, it can be proved (see [3]) 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 $\|g(x_l;\mu)\|^2 < \rho\mu$ in Procedure B. This is a contradiction with the definition of Phase 1. □

Theorem 4.4. Let Assumption 1 and Assumption 2 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

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

for $1 \leq i \leq m$.

Proof. (a) Since $\underline{\delta} = 0$, it holds $1 - e^T u(x_k;\mu_k) = 0$. Inequalities $u_i(x_k;\mu_k) \geq 0$, $z(x_k;\mu_k) - f_i(x_k) \geq 0$ follow from (6) and from Lemma 2.1.

(b) Using (29) and (34), we can write

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

which implies

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

where $\underline{B} = \underline{F} + \min(0,\, m\mu_1(1-\log(2m\mu_1))$ (see proof of Lemma 4.2). 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 $g(x_k;\mu_k) = \sum_{i=1}^{m} u_i(x_k;\mu_k)g_i(x_k) \to 0$.

(c) Let $1 \leq i \leq m$ be chosen arbitrarily. Using the definition of $u_i(x_k;\mu_k)$, we obtain

$$u_i(x_k;\mu_k)(z_i(x_k;\mu_k) - f_i(x_k)) = \frac{\mu_k}{z_i(x_k;\mu_k) - f_i(x_k)}(z_i(x_k;\mu_k) - f_i(x_k)) = \mu_k \to 0$$

by Lemma 4.3. □

Corollary 4.5. Let assumptions of Theorem 4.4 hold. Then every cluster point $x \in \mathbb{R}^n$ of sequence $\{x_k\}_1^\infty$ satisfies KKT conditions (3), where $u \in \mathbb{R}^m$ is a cluster point of sequence $\{u(x_k;\mu_k)\}_1^\infty$.

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

Theorem 4.6. Consider sequence $\{x_k\}_1^\infty$ generated by Algorithm 1. 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 $1 \leq i \leq m$.

P r o o f. 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 (6) and Lemma 2.1 as in the proof of Theorem 4.4. Since $\mu_k \to 0$ by Lemma 4.3 and $g(x_k;\mu_k) \to 0$ by Theorem 4.4, 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 kth iteration). Using (6), we obtain

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

□

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

$$S(x;\mu) = \mu \log \sum_{i=1}^{m} \exp\left(\frac{f_i(x)}{\mu}\right) = F(x) + \mu \log \sum_{i=1}^{m} \exp\left(\frac{f_i(x) - F(x)}{\mu}\right), \quad (38)$$

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 5.1. Consider smoothing function (38). Then

$$\nabla S(x;\mu) = A(x)\tilde{U}(x;\mu)e \quad (39)$$

and

$$\nabla^2 S(x;\mu) = \tilde{G}(x;\mu) + \frac{1}{\mu}A(x)\tilde{U}(x;\mu)A^T(x) - \frac{1}{\mu}A(x)\tilde{U}(x;\mu)ee^T\tilde{U}(x;\mu)A^T(x), \quad (40)$$

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

$$\tilde{u}_i(x;\mu) = \frac{\exp(f_i(x)/\mu)}{\sum_{j=1}^{m}\exp(f_j(x)/\mu)} = \frac{\exp((f_i(x)-F(x))/\mu)}{\sum_{j=1}^{m}\exp((f_j(x)-F(x))/\mu)} \quad (41)$$

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

Note that (40) 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 5.1, 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

$$\left(\tilde{H} - \frac{1}{\mu}\tilde{g}\tilde{g}^T\right)d = -\tilde{g}, \quad (42)$$

where

$$\tilde{H} = \tilde{G}(x;\mu) + \frac{1}{\mu}A(x)\tilde{U}(x;\mu)A^T(x) \quad (43)$$

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 (17). 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 (42) can be written in the form

$$d = \frac{\mu}{\tilde{g}^T\tilde{H}^{-1}\tilde{g} - \mu}\tilde{H}^{-1}\tilde{g}. \tag{44}$$

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

$$\tilde{L}\tilde{D}\tilde{L}^T v = \tilde{g} \tag{45}$$

and set

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

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 (41)), in Step 4, where (20) – (21) are replaced by (45) – (46), and in Step 6, where $B(x;\mu)$ is replaced by $S(x;\mu)$. Note that μ in (38) has a different meaning from μ in (12) so procedures for updating these parameters need not be identical. Nevertheless, Procedure A 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). Finally, note 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 [19], 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 [19], serve for defining objective functions

$$F(x) = \max_{1\le i\le m} f_i(x) \tag{47}$$

and

$$F(x) = \max_{1\le i\le m} |f_i(x)| = \max_{1\le i\le m} \left[\max(f_i(x), -f_i(x))\right]. \tag{48}$$

Function (47) is not used in connection with Test 15, since Assumption 1 is not satisfied in this case (sometimes $F(x) \to -\infty$).

In Procedure A, Procedure B and Algorithm 1, we have used parameters $\underline{\varepsilon} = 10^{-6}$, $\underline{\delta} = 10^{-6}$, $\underline{\mu} = 10^{-10}$, $\overline{\mu} = 1$, $\underline{g} = \infty$, $\lambda = 0.85$, $\sigma = 100$, $\rho = 0.1$, $\varepsilon_0 = 10^{-8}$, $\varepsilon_1 = 10^{-4}$, $\overline{\Delta} = 1000$ as defaults. Value $\overline{\Delta}$ was sometimes decreased (in `NT` cases, see below). In the implementation of the smoothing method described in Section 5, we have used the same default 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 the primal interior point method PIP (Algorithm 1), with the smoothing method `SM` described in the previous section and the primal-dual interior point method `PDIP` described in [15]. All these methods are implemented in the interactive system for universal functional optimization UFO [17] as line-search subroutines for discrete minimax optimization and nonlinear programming. All mentioned subroutines use the same modules for numerical differentiation, stepsize selection, and variable metric updates, so the results are quite comparable. Method PDIP was used for solving equivalent nonlinear programming problem (2). For an additional comparison, we have used a Fortran version of the `KNITRO` software described in [2]. This code with options `hessopt=6` (`LBFGS`), `opttol`=10^{-8}, `xtol`=10^{-12} was also used for solving problem (2).

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 the default value of parameter $\overline{\Delta}$ was changed), `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 default value of parameter $\overline{\Delta}$ was changed (for `KNITRO`, `NT` corresponds to the number of problems for which artificial box constraints had to be added), 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 all cases.

Table 1. Test 14: Function (47) with 200 variables.

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
PIP	1615	2429	1637	–	–	–	1	1.50
SM	7244	21008	7266	–	1	–	8	9.14
PDIP	1790	3925	1790	5	1	–	9	4.63
KNITRO	6661	12364	6683	–	1	–	6	33.59

Table 2. Test 14: Function (48) with 200 variables.

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
PIP	3227	5279	3248	1	–	–	3	1.88
SM	2985	6311	3006	1	1	–	6	4.84
PDIP	2836	7023	2837	6	1	–	8	7.97
KNITRO	5868	8949	5890	–	1	–	5	36.28

Table 3. Test 15: Function (48) with 200 variables.

Method	NIT	NFV	NFG	NR	NL	NF	NT	Time
PIP	2435	5045	2457	1	1	–	1	2.56
SM	9326	18290	9348	1	2	–	5	38.06
PDIP	2056	3883	2056	–	2	–	4	7.00
KNITRO	1166	1934	1168	–	3	–	2	24.62

Results introduced in the above tables imply the following conclusions. Primal interior-point method PIP (Algorithm 1) seems to be more efficient than other methods tested. Smoothing method SM is more sensitive to the choice of its parameters, converges more slowly and requires greater CPU time (since computation of exponentials is time consuming). Primal-dual interior-point methods for general nonlinear programming problems (PDIP and KNITRO) convert the original problem to the problem with $n+1$ basic variables, m (or $2m$ for function (48)) 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 PDIP uses constant penalty parameter (see [15]), which had to be sometimes adjusted. Thus the number NT is slightly greater for PDIP.

The second set of tests concerns a comparison of PIP, SM, PDIP, and KNITRO by using large-scale test problems with 1000 variables. The results of computational experiments are given in three tables, where P is the problem number, NIT is the number of iterations, NFV is the number of function 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 the default value of parameter $\overline{\Delta}$ was changed and the total computational time in seconds.

The results introduced in these tables confirm conclusions following from the previous tables. Primal interior-point method PIP seems to be more efficient than smoothing method SM and primal-dual interior point methods PDIP and KNITRO in all indicators. The computational time is significantly shorter and also the number of iterations is usually smaller in comparison with other 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. Note that KNITRO, even if it uses sparse matrix techniques and efficient linear solvers, is not intended for partially separable minimax problems. Since partitioned variable metric updates are not implemented in KNITRO, we had to use LBFGS updates, which are usually less efficient (the other algorithmic options of KNITRO gave worse results).

	PIP			SM			PDIP			KNITRO		
P	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F
3	49	66	1.9879E-08	67	79	7.7863E-09	42	47	1.3635E-09	544	946	4.9900E-08
4	96	129	0.542761	811	1082	0.542761	101	178	0.542761	1289	2990	0.542762
5	52	76	8.2469E-08	40	60	4.5272E-08	24	24	6.6464E-10	32	45	1.0000E-07
6	128	161	4.2512E-08	58	76	1.2340E-07	62	62	3.6884E-08	70	93	1.0000E-07
7	56	115	0.260163	100	193	0.260162	59	67	0.260163	93	308	0.260163
8	12	22	1556.50	6	52	1556.50	126	576	1556.50	346	501	1556.50
11	76	101	5.3883E-02	109	111	5.3883E-02	110	145	5.3883E-02	222	360	5.3883E-02
12	185	300	0.996492	279	329	0.996492	77	87	0.996492	150	370	0.996492
13	3	5	3.5691E-27	7	9	3.7597E-14	30	76	1.4028E-10	37	65	2.0971E-06
14	1	2	1.5746E-11	4	10	4.2368E-12	11	11	9.7635E-08	12	13	1.0000E-07
15	76	544	3.9960E-03	34	47	3.9960E-03	112	155	3.9960E-03	230	675	3.9960E-03
16	139	549	-6.4083E-04	1102	4247	-6.3766E-04	157	225	-6.4093E-04	50	147	-6.4083E-04
17	24	57	1.9712E-09	43	70	5.8556E-07	78	140	-7.9809E-10	1921	2399	1.9084E-06
18	50	150	1.3605E-08	63	142	5.9387E-07	58	70	1.3856E-11	3696	3914	8.4782E-07
19	10	18	42.5076	4	8	42.5075	33	80	42.5076	168	281	42.5076
20	24	38	-9.9900E-04	67	74	-9.9900E-04	94	118	-9.9990E-04	2141	4284	-9.9890E-04
21	11	15	5.9940E-03	157	595	5.9940E-03	50	80	5.9940E-03	237	400	5.9940E-03
22	49	153	1.1563E-03	37	51	1.1563E-03	156	266	1.1564E-03	691	1222	1.1564E-03
Σ	1041	2501	NT = 3	2988	7235	NT = 6	1380	2407	NT = 5	11929	19013	NT = 6
	Time = 3.28			Time = 15.08			Time = 17.20			Time = 452.40		

Table 4. Test 14: Function (47) with 1000 variables.

	PIP			SM			PDIP			KNITRO		
P	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F
3	68	69	3.2154E-08	69	82	5.3346E-08	61	71	4.2724E-11	146	155	5.3426E-07
4	76	97	0.542761	565	705	0.542761	50	54	0.542761	1559	4687	0.542763
5	99	102	1.0340E-07	109	255	2.0875E-07	32	33	1.7087E-08	32	44	1.9999E-07
6	88	128	1.1565E-07	90	162	5.7907E-07	61	62	4.1932E-10	57	72	1.9999E-07
7	34	66	0.260162	144	416	0.260162	46	46	0.260163	63	96	0.260163
8	25	65	1556.50	6	52	1556.50	188	263	1556.50	143	215	1556.50
11	64	69	5.3883E-02	345	347	5.3883E-02	91	105	5.3883E-02	1160	2934	5.3883E-02
12	295	820	0.996492	168	305	0.996492	191	489	0.996492	117	259	0.996492
13	16	17	6.2330E-08	6	7	1.5698E-12	23	23	3.4757E-06	34	63	2.1588E-07
14	5	7	3.6173E-12	4	66	3.9124E-12	12	12	1.0604E-08	9	10	8.6949E-07
15	36	38	3.9960E-03	38	83	3.9960E-03	88	105	3.9960E-03	404	872	3.9960E-03
16	31	84	2.2706E-14	307	676	5.9528E-10	82	83	3.6565E-10	70	124	4.8801E-07
17	66	171	6.5729E-07	75	83	1.2438E-06	66	83	8.3629E-08	595	629	5.9618E-07
18	57	80	2.0579E-07	128	288	9.1744E-07	254	287	1.6106E-07	526	567	1.7520E-06
19	9	18	42.5076	4	8	42.5075	23	70	42.5076	149	250	42.5076
20	252	295	1.4999E-04	581	1741	5.9882E-05	104	136	9.9969E-06	978	1987	4.1502E-04
21	12	17	5.9940E-03	69	236	5.9940E-03	47	78	5.9940E-03	73	210	5.9940E-03
22	39	48	1.1563E-03	42	68	1.1563E-03	125	193	1.1564E-03	1596	3144	1.1564E-03
Σ	1272	2191	NT = 2	2750	5580	NT = 7	1544	2193	NT = 4	7711	16318	NT = 12
	Time = 4.06			Time = 18.81			Time = 30.36			Time = 708.06		

Table 5. Test 14: Function (48) with 1000 variables.

	PIP			SM			PDIP			KNITRO		
P	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F	NIT	NFV	F
3	26	31	1.1284E-11	47	64	1.8337E-10	34	41	5.5665E-09	64	134	3.9891E-07
4	57	105	0.445072	256	814	0.445072	71	79	0.445072	68	123	0.445072
5	7	8	3.3307E-16	20	32	6.7130E-13	40	40	2.6410E-09	13	25	2.0000E-07
6	37	171	7.7716E-16	30	69	1.5614E-11	49	49	1.0807E-09	13	18	1.9994E-07
7	40	171	8.73314	106	524	8.73314	39	41	8.73314	46	76	8.73314
8	88	248	8.7944E-07	73	108	1.5697E-08	164	356	4.8816E-08	73	138	1.0000E-06
10	90	145	0.679030	2309	2363	0.679030	39	39	0.679030	103	209	0.679030
12	56	151	9.23541	329	1010	9.23541	50	91	9.23541	95	381	9.23541
13	66	155	21.9683	916	2001	21.9682	74	87	21.9683	65	155	21.9683
14	30	92	0.459990	36	108	0.459990	19	19	0.459990	16	17	0.459990
15	34	68	0.230608	262	285	0.230607	27	27	0.230608	11	16	0.230608
16	35	75	6.6667E-02	46	81	6.6667E-02	36	36	6.6667E-02	31	62	6.6667E-02
17	69	269	5.1876E-12	56	110	9.4369E-16	98	98	7.0381E-11	147	185	1.9985E-07
19	40	45	2.8866E-15	37	56	8.8818E-16	54	54	3.4805E-10	36	54	2.0001E-07
20	172	1287	7.1054E-14	107	121	5.6843E-13	175	175	6.5834E-09	36	53	4.4918E-07
21	47	178	0.900000	129	599	0.900000	51	98	0.900000	78	158	0.900000
22	83	233	2.35487	74	198	2.35487	47	57	2.35487	62	84	2.61783
Σ	977	3432	NT = 0	4833	8543	NT = 5	1067	1417	NT = 8	957	1888	NT = 6
	Time = 7.94			Time = 96.88			Time = 50.27			Time = 147.17		

Table 6. Test 15: Function (48) with 1000 variables.

ACKNOWLEDGEMENT

This work was supported by the Grant Agency of the Academy of Sciences of the Czech Republic, project No. IAA1030405, the Czech Science Foundation, project No. 201/06/P397, and the institutional research plan No. AV0Z10300504.

(Received October 2, 2008.)

REFERENCES

[1] J. R. Bunch and B. N. Parlett: Direct methods for solving symmetric indefinite systems of linear equations. SIAM J. Numer. Anal. *8* (1971), 639–655.

[2] R. H. Byrd, J. Nocedal, and R. A. Waltz: KNITRO: An integrated package for nonlinear optimization. In: Large-Scale Nonlinear Optimization (G. di Pillo and M. Roma, eds.), Springer, Berlin 2006, pp. 35–59.

[3] R. Fletcher: Practical Methods of Optimization. Second edition. Wiley, New York 1987.

[4] R. Fletcher and E. Sainz de la Maza: Nonlinear programming and nonsmooth optimization by successive linear programming. Math. Programming *43* (1989), 235–256.

[5] Y. Gao and X. Li: Nonsmooth equations approach to a constrained minimax problem. Appl. Math. *50* (2005), 115–130.

[6] P. E. Gill and W. Murray: Newton type methods for unconstrained and linearly constrained optimization. Math. Programming *7* (1974), 311–350.

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

[8] A. A. Goldstein: On steepest descent. SIAM J. Control *3* (1965), 147–151.

[9] A. Griewank: Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. SIAM, Philadelphia 2000.

[10] A. Griewank and P. L. Toint: Partitioned variable metric updates for large-scale structured optimization problems. Numer. Math. *39* (1982), 119–137.

[11] S. P. Han: Variable metric methods for minimizing a class of nondifferentiable functions. Math. Programming *20* (1981), 1–13.

[12] K. Jónasson and K. Madsen: Corrected sequential linear programming for sparse minimax optimization. BIT *34* (1994), 372–387.

[13] D. Le: Three new rapidly convergent algorithms for finding a zero of a function. SIAM J. Sci. Statist. Comput. *6* (1985), 193–208.

[14] D. Le: An efficient derivative-free method for solving nonlinear equations. ACM Trans. Math. Software *11* (1985), 250–262.

[15] L. Lukšan, C. Matonoha, and J. Vlček: Interior point method for nonlinear nonconvex optimization. Numer. Linear Algebra Appl. *11* (2004), 431–453.

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

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

[18] L. Lukšan and E. Spedicato: Variable metric methods for unconstrained optimization and nonlinear least squares. J. Comput. Appl. Math. *124* (2000), 61–93.

[19] L. Lukšan and J. Vlček: Sparse and Partially Separable Test Problems for Unconstrained and Equality Constrained Optimization, ICS AS CR Report No. V-767, Prague 1998.

[20] E. Polak, J. O. Royset, and R. S. Womersley: Algorithm with adaptive smoothing for finite minimax problems. J. Optim. Theory Appl. *119* (2003), 459–484.

[21] J. Vanderbei and D. F. Shanno: An interior point algorithm for nonconvex nonlinear programming. Comput. Optim. Appl. *13* (1999), 231–252.

[22] S. Xu: Smoothing methods for minimax problems. Comput. Optim. Appl. *20* (2001), 267–279.

Ladislav Lukšan, Ctirad Matonoha and Jan Vlček, Institute of Computer Science – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 182 07 Praha 8, Czech Republic (the first author is also from Technical University of Liberec, Hálkova 6, 461 17 Liberec, Czech Republic).
e-mails: luksan@cs.cas.cz, matonoha@cs.cas.cz, vlcek@cs.cas.cz