

A New Objective Penalty Function Approach for Solving Constrained Minimax Problems

Jueyou Li · Zhiyou Wu · Qiang Long

Received: 23 May 2013 / Revised: 27 January 2014 / Accepted: 27 February 2014 /
Published online: 20 March 2014
© Operations Research Society of China, Periodicals Agency of Shanghai University,
and Springer-Verlag Berlin Heidelberg 2014

Abstract In this paper, a new objective penalty function approach is proposed for solving minimax programming problems with equality and inequality constraints. This new objective penalty function combines the objective penalty and constraint penalty. By the new objective penalty function, a constrained minimax problem is converted to minimizations of a sequence of continuously differentiable functions with a simple box constraint. One can thus apply any efficient gradient minimization methods to solve the minimizations with box constraint at each step of the sequence. Some relationships between the original constrained minimax problem and the corresponding minimization problems with box constraint are established. Based on these results, an algorithm for finding a global solution of the constrained minimax problems is proposed by integrating the particular structure of minimax problems and its global convergence is proved under some conditions. Furthermore, an algorithm is developed for finding a local solution of the constrained minimax problems, with its convergence proved under certain conditions. Preliminary results

This research was supported by Natural Science Foundation of Chongqing (Nos. cstc2013jjB00001 and cstc2011jjA00010), by Chongqing Municipal Education Commission (No.KJ120616).

J. Li
SITE, Federation University Australia, Ballarat, VIC 3353, Australia
e-mail: lijueyou@163.com

J. Li · Z. Wu (✉)
Department of Mathematics, Chongqing Normal University, Chongqing 400047, China
e-mail: zywu@cqnu.edu.cn

Q. Long
School of Science, Southwest University of Science and Technology, Mianyang, Sichuan 621010,
China

of numerical experiments with well-known test problems show that satisfactorily approximate solutions for some constrained minimax problems can be obtained.

Keywords Minimax problem · Constrained minimization · Objective penalty function · Approximate solution

1 Introduction

Consider the following constrained minimax programming problem:

$$\begin{aligned} \min_x \quad & F(x) \\ \text{s.t.} \quad & g_j(x) \leq 0, \quad j \in J, \\ & h_l(x) = 0, \quad l \in L. \end{aligned} \tag{P1}$$

where $F(x) = \max_{i \in I} f_i(x)$, $I = \{1, \dots, m\}$, J, L are a finite set of integers. In this paper, we assume that $f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i \in I$, $g_j(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $j \in J$, $h_l(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $l \in L$ are continuously differentiable, and denote its feasible set by $X = \{x \in \mathbb{R}^n \mid g_j(x) \leq 0, j \in J, h_l(x) = 0, l \in L\}$.

The objective function $F(x)$ is not necessarily differentiable, even when all $f_i(x)$, $i \in I$ are differentiable. Thus, problem (P1) is, in nature, a nondifferentiable optimization problem. In real-world applications, many problems of interest can be modeled as the above finite minimax problem (P1) such as in curve fitting, l_1 and l_∞ approximation problems, systems of nonlinear equations, nonlinear programming problems, multiobjective problems, engineering design, optimal control, and many other situations (see, for examples, [1–4]).

Over the past few decades, the finite minimax problem has attracted attention from more and more researchers and many algorithms have been developed, which can be divided into three classes in general. The first class of algorithms regards problem (P1) as a constrained nonsmooth optimization problem. Thus, it can be solved by general nonsmooth optimization methods including subgradient methods, bundle methods, and cutting plane methods (see [1, 4, 5]). Due to the particular structure of the objective function, the second class of algorithms tries to smooth approximately the maximum objective function $F(x)$ (named as smooth methods). Thus, problem (P1) can be converted into a smoothed optimization problem being solved by any powerful smooth optimization methods. Zang [6] firstly introduced a function to smooth the finite maximum function locally in some neighborhood of nondifferentiable point of $F(x)$. Some other smoothing techniques attempt to smooth $F(x)$ globally, including exponential (entropic) functions (see [7–10]) and hyperbolic smoothing functions (see [11]). However, these smooth methods need to solve a corresponding nonlinear optimization problem at each iteration to generate a new approximation solution by controlling the precision parameter(s). For the third class algorithms of solving problem (P1), according to the following well-known fact that problem (P1) is equivalent to a nonlinear smooth optimization problem through introducing a new variable $t \in \mathbb{R}$,

$$\begin{aligned}
 & \min_{(x,t)} t \\
 \text{s.t. } & f_i(x) \leq t, \quad i \in I, \\
 & g_j(x) \leq 0, \quad j \in J, \\
 & h_l(x) = 0, \quad l \in L.
 \end{aligned} \tag{P2}$$

The feasible set of (P2) is denoted by $\bar{X} = \{(x, t) \in \mathbb{R}^{n+1} | f_i(x) \leq t, i \in I, g_j(x) \leq 0, j \in J, h_l(x) = 0, l \in L\}$.

Obviously, solving the finite minimax problem (P1) is equivalent to solving the problem (P2). Many successful algorithms can be exploited to solve this equivalent reformulation (P2), such as sequential quadratic programming methods (see [12, 13]) and trust-region methods (see [14–16]). Di Pillo et al. [17] proposed an approach which reduces to solving a unconstrained nonlinear programming problem in which its objective function is formulated by a continuously differentiable exact penalty function. However, it needs to choose the suitable penalty parameters in each iteration in implementation. Rustem et al. [18] discussed an approach using an augmented Lagrangian formulation to solve the finite minimax problems with inequality constraints. But, this formulation seems a little complex and enlarges greatly the number of variables because of introducing the Lagrange multipliers. Recently, for dealing with finite constrained minimax problems, Obasanjo et al. [19] presented a primal-dual interior-point method to solve the equivalent problem (P2). Ma et al. [20] introduced a new exact and smooth penalty function to tackle minimax problems with equality constraints, in which the penalty parameter needed to be increased gradually to infinity. Based on the equivalent reformulation (P2) and a novel continuously differentiable exact objective penalty function, we propose a penalty function method to solve the minimax problem (P1) by taking a finite penalty parameter. The proposed approach in this paper should fall into the third class.

Although classical penalty function methods are popular, there are some shortcomings. For instances, the penalty function is not necessarily smooth if it is exact; the penalty function is not necessarily exact if it is smooth. Now, we take a general constrained minimization problem to show these as follows:

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t. } g_j(x) \leq 0, j \in J, h_l(x) = 0, \quad l \in L. \tag{1.1}$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable and $g_j(x), h_l(x)$ are given in (P1). The penalty functions are (see [29]):

$$F_1(x; \rho) = f(x) + \rho \left[\sum_{j \in J} g_j^+(x) + \sum_{l \in L} |h_l(x)| \right] \tag{1.2}$$

$$F_2(x; \rho) = f(x) + \rho \left[\sum_{j \in J} (g_j^+(x))^2 + \sum_{l \in L} h_l^2(x) \right] \tag{1.3}$$

where $g_j^+(x) = \max\{g_j(x), 0\}$. Under some conditions, $F_1(x; \rho)$ is an exact penalty function, but it is not smooth in general. $F_2(x; \rho)$ is a smooth penalty function, but it

is not exact. In addition, the penalty parameter ρ for all penalty function algorithms needs to be increased gradually. So does the exact penalty function algorithm, because we do not know exactly how big the penalty parameter ρ is needed. In fact, it is impossible to take a very big value of the penalty parameter ρ , which can cause numerical difficulties and ill-conditioning effects [29].

To circumvent difficulties from classical penalty functions, an objective penalty function has been proposed in [21–23] for solving problem (1.1) with the inequality constraints, and defined as follows:

$$F(x; M) = [f(x) - M]^p + \sum_{j \in J} g_j(x)^p \quad (1.4)$$

with $p > 0$. Suppose that x^* is an optimal solution of problem (1.1) with the equality constraints, and $f(x^*)$ is the optimal value of the objective function. By updating suitably the penalty parameter $\{M_k\}$ and solving a sequential unconstrained minimizations of $F(x; M_k)$, then there are the solution sequence $\{x(M_k)\} \rightarrow x^*$ as well as $\{M_k\} \rightarrow f(x^*)$ (see [22, 23]). Recently, Meng et al. [24, 25] introduced more general objective penalty functions. They proved that under certain conditions, objective penalty functions were exact and differentiable and proposed efficient objective penalty function algorithms to solve problem (1.1) with inequality constraints [26, 27]. This means that it only needs to solve single unconstrained optimization problem when the penalty parameter M is an exact penalty parameter. Using an objective penalty function similar to (1.4), Dutta et al. [28] firstly solved a class of minimax problems. However, the rule of updating penalty parameters $\{M_k\}$ was not easy to control and its initial value was difficult to choose without considering the particular structure of minimax problems in their papers.

Motivated by [21, 26, 28], we introduce a new smooth penalty function for solving minimax problems in this paper. The main feature of our new smooth penalty function combines the objective penalty function and constraint penalty function, which shares good properties of smoothness and some kinds of exactness and means that any efficient gradient minimization methods available can be used. It will be shown that the sequential solution of minimizations with a simple box constraint can approximate progressively the optimal solutions of original minimax problem. We first propose a global penalty function algorithm and prove its global convergence. Then, an algorithm for finding the approximately local solutions is proposed and its convergence is proved under certain conditions.

The rest of this paper is organized as follows. In Sect. 2, we obtain some relations between the minimax and its corresponding minimization problems with simple box constraints. Then, we propose a new penalty function algorithm for finding approximate solutions to minimax problems and prove the convergence under some conditions. In Sect. 3, we report some numerical results with 23 test problems. Finally, we conclude the paper.

2 A New Penalty Function Approach

Based on the equivalence between (P1) and (P2), the correspondence of local and global solutions of problem (P1) with local and global solutions of problem (P2) is established below.

Proposition 2.1 *Let \bar{x} be a global (local) solution of (P1) with $\bar{t} = F(\bar{x})$, then (\bar{x}, \bar{t}) is a global (local) solution of (P2). Conversely, if (\bar{x}, \bar{t}) is a global (local) solution of (P2), then the point \bar{x} is a global (local) solution of (P1) and $\bar{t} = F(\bar{x})$.*

In terms of Proposition 2.1, next we will focus attention on solving problem (P2) and construct the following penalty function for problem (P2):

$$E(x, t; M, \rho) = \frac{1}{2}[(t - M)^+]^2 + \frac{\rho}{2} \left\{ \sum_{i \in I} [(f_i(x) - t)^+]^2 + \sum_{j \in J} [g_j^+(x)]^2 + \sum_{l \in L} h_l^2(x) \right\}$$

where $M \in \mathbb{R}$ is called an objective penalty parameter and $\rho > 0$ is called constraint penalty parameter. It can easily be shown that, for each parameter M and ρ , $E(x, t; M, \rho)$ is continuously differentiable with respect to (x, t) .

2.1 A New Penalty Function Algorithm for a Global Solution

In this subsection, we assume that the parameter $\rho > 0$ is a pre-specified constant and consider the following optimization problem:

$$\min_{(x,t) \in Y} E(x, t; M, \rho) \tag{PM}$$

where $Y \subseteq \mathbb{R}^{n+1}$ is a simple box set satisfying $\bar{X} \subset Y = \prod_{i=1}^{n+1} [u_i, v_i]$, $-\infty < u_i \leq v_i < +\infty$. Hence, for any M and ρ , problem (PM) exists a minimizer. For some M , if an optimal solution (x_M^*, t_M^*) to (PM) is also an optimal solution to (P2), then M is called an exact value of the objective penalty parameter (see, [26]).

Theorem 2.1 *The following assertions hold:*

- (i) *If (x^*, t^*) is a global solution of (P2) and $M \geq t^*$, then (x^*, t^*) is also a global solution of (PM) with $E(x^*, t^*; M, \rho) = 0$ for any $\rho > 0$.*
- (ii) *Let (x^*, t^*) be a global solution of (P2). If (\bar{x}, \bar{t}) is a global solution of (PM) for some $\rho > 0$ and $M = t^*$, then (\bar{x}, \bar{t}) is also a global solution of (P2).*

Proof

- (i) Since (x^*, t^*) is a global solution to (P2) and $M \geq t^*$, we have $E(x^*, t^*; M, \rho) = 0$ for any $\rho > 0$. It follows from $E(x, t; M, \rho) \geq 0$ for any $(x, t) \in \mathbb{R}^{n+1}$ that (x^*, t^*) is a global solution to (PM).

- (ii) If (\bar{x}, \bar{t}) is a global solution of (PM) for some $\rho > 0$, then $E(\bar{x}, \bar{t}; M, \rho) \leq E(x, t; M, \rho)$ for any $(x, t) \in \mathbb{R}^{n+1}$. In particular, $E(\bar{x}, \bar{t}; M, \rho) \leq E(x^*, t^*; M, \rho) = 0$ due to $M = t^*$. Hence, $E(\bar{x}, \bar{t}; M, \rho) = 0$ and (\bar{x}, \bar{t}) is feasible of (P2). Moreover, $\bar{t} \leq M = t^* \leq t$ for any feasible point (x, t) of (P2). This shows that (\bar{x}, \bar{t}) is a global solution of (P2). □

Theorem 2.2 *Let (x^*, t^*) be a global solution of (P2). Suppose that for some M and some $\rho > 0$, (x_M^*, t_M^*) is a global solution of (PM). Then, the following relations hold:*

- (i) *If $E(x_M^*, t_M^*; M, \rho) = 0$, then (x_M^*, t_M^*) is feasible to (P2) and $t^* \leq t_M^* \leq M$.*
- (ii) *If $E(x_M^*, t_M^*; M, \rho) > 0$ and (x_M^*, t_M^*) is not feasible to (P2), then $M < t^*$ and $t_M^* < t^*$.*
- (iii) *If $E(x_M^*, t_M^*; M, \rho) > 0$ and (x_M^*, t_M^*) is feasible to (P2), then (x_M^*, t_M^*) is a global solution to (P2).*

Proof

- (i) The result is obvious from the definition of $E(x, t; M, \rho)$.
- (ii) Since (x_M^*, t_M^*) and (x^*, t^*) are global solutions of (PM) and (P2), respectively, we have

$$0 < E(x_M^*, t_M^*; M, \rho) \leq E(x^*, t^*; M, \rho) = \frac{1}{2}[(t^* - M)^+]^2,$$

which implies that $M < t^*$. If $t_M^* \leq M$, then $t_M^* \leq M < t^*$. Otherwise $t_M^* > M$, we have

$$0 < \frac{1}{2}[(t_M^* - M)^+]^2 < E(x_M^*, t_M^*; M, \rho) \leq E(x^*, t^*; M, \rho) = \frac{1}{2}[(t^* - M)^+]^2.$$

Hence, $t_M^* < t^*$.

- (iii) Since $E(x_M^*, t_M^*; M, \rho) > 0$ and (x_M^*, t_M^*) is a feasible point to (P2), we have

$$0 < \frac{1}{2}[(t_M^* - M)^+]^2 = E(x_M^*, t_M^*; M, \rho) \leq E(x, t; M, \rho) = \frac{1}{2}[(t - M)^+]^2, \forall (x, t) \in \bar{X},$$

which implies that

$$t_M^* - M \leq t - M, \forall (x, t) \in \bar{X}.$$

Thus, (x_M^*, t_M^*) is a global solution to (P2). □

Remark 2.1 From Theorem 2.2, the M satisfied (iii) of Theorem 2.2 is an exact objective penalty parameter. Theorem 2.2 provides a good way to solve problem (P2) for finding a global solution.

In terms of this theorem, we propose an algorithm to compute a globally optimal solution to problem (P2) by solving problem (PM) sequentially. We call the following algorithm as a global objective penalty function (for short, GOPF) algorithm.

GOPF Algorithm:

- Step 0** Give a constant $\rho > 0$ fixed. Choose a point $x_0 \in X$ and $a_1 < \min_{x \in X} F(x)$. Let $b_1 = t_0 = F(x_0)$ and $M_1 = \frac{a_1 + b_1}{2}$. Let $k := 1$.
- Step 1** Solve $\min_{(x,t) \in Y} E(x, t; M_k, \rho)$ starting at (x_{k-1}, t_{k-1}) , where $Y = X_0 \times [a_1, b_1] \subseteq \mathbb{R}^{n+1}$, X_0 is a simple box containing X . Let (x_k, t_k) be its global solution.
- Step 2** If $E(x_k, t_k; M_k, \rho) = 0$, let $a_{k+1} := a_k, b_{k+1} := t_k, M_{k+1} := \frac{a_{k+1} + b_{k+1}}{2}$, and go to Step 1. Otherwise, $E(x_k, t_k; M_k) > 0$, go to Step 3.
- Step 3** If (x_k, t_k) is not feasible to (P2), let $a_{k+1} := \max\{t_k, M_k\}, b_{k+1} := b_k, M_{k+1} := \frac{a_{k+1} + b_{k+1}}{2}$, $k := k + 1$, and go to Step 1. Otherwise, stop, (x_k, t_k) is a global solution of (P2).

In GOPF algorithm, it is assumed that there exists a global minimization algorithm available to solve the problem with simple box constraint in Step 1, and one can always get $a_1 < \min_{x \in X} F(x)$.

Theorem 2.3 Let $t^* = \min_{(x,t) \in \bar{X}} t$ and $\{(x_k, t_k)\}$ be the sequence generated by GOPF algorithm.

- (i) If $\{(x_k, t_k)\}$ is a finite sequence, i.e., GOPF algorithm stops at the \bar{k} th iteration, then $(x_{\bar{k}}, t_{\bar{k}})$ is a global solution to (P2).
- (ii) If $\{(x_k, t_k)\}$ is an infinite sequence, then $\{(x_k, t_k)\}$ is bounded and any limit point of $\{(x_k, t_k)\}$ is a global solution to (P2).

Proof

- (i) If GOPF algorithm terminates at the \bar{k} th iteration and the second situation of Step 3 occurs, by (iii) of Theorem 2.2, thus, $(x_{\bar{k}}, t_{\bar{k}})$ is a global solution to (P2).
- (ii) Suppose that $\{(x_k, t_k)\}$ is an infinite sequence generated by GOPF algorithm. Firstly, we prove the sequence $\{a_k\}$ increases and the sequence $\{b_k\}$ decreases with

$$a_k \leq t^* \leq b_k, \quad k = 1, 2, \dots \tag{2.1}$$

and

$$b_{k+1} - a_{k+1} \leq \frac{b_k - a_k}{2}, \quad k = 1, 2, \dots \tag{2.2}$$

by the induction method.

In fact, for $k = 1$, it follows from GOPF algorithm that $a_1 \leq a_2, b_2 \geq b_1, a_1 \leq t^* \leq b_1, b_2 - a_2 \leq \frac{b_1 - a_1}{2}$.

For $k \geq 1$, assume that (2.1) and (2.2) hold.

Consider the case of $k + 1$. From Step 2, we have $a_{k+1} = a_k, b_{k+1} = t_k, M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$. Thus, by (i) of Theorem 2.2,

$$a_{k+1} = a_k \leq t^* \leq t_k = b_{k+1} \quad \text{and} \quad b_{k+1} \leq M_k = \frac{a_k + b_k}{2} \leq b_k. \tag{2.3}$$

Hence, $b_{k+1} - a_{k+1} \leq M_k - a_k = \frac{b_k - a_k}{2}$. From Step 3, we have $b_{k+1} = b_k, a_{k+1} = \max\{t_k, M_k\}$. Thus, by (ii) of Theorem 2.2,

$$a_k \leq \frac{a_k + b_k}{2} = M_k \leq a_{k+1} \quad \text{and} \quad a_{k+1} \leq t^* \leq b_k = b_{k+1}. \tag{2.4}$$

Hence, $b_{k+1} - a_{k+1} = b_k - \max\{t_k, M_k\} \leq b_k - M_k = \frac{b_k - a_k}{2}$. By the induction method, (2.1) and (2.2) can hold. In addition, we can also obtain

$$a_k \leq M_k \leq b_k, \quad k = 1, 2, \dots \tag{2.5}$$

Since $\{a_k\}$ is increasing, $\{b_k\}$ is decreasing, and (2.1), we have $\{a_k\}$ and $\{b_k\}$ converge. Let $\lim_{k \rightarrow \infty} a_k = a^*$ and $\lim_{k \rightarrow \infty} b_k = b^*$. From (2.2), we have $a^* = b^*$. From (2.5), we can get $\lim_{k \rightarrow \infty} M_k = a^*$.

Since $\{(x_k, t_k)\} \subset Y$ and Y is a bounded box, the sequence $\{(x_k, t_k)\}$ is bounded. Without loss of generality, assume that $(x_k, t_k) \rightarrow (\bar{x}, \bar{t})$ as $k \rightarrow \infty$. Next, we prove (\bar{x}, \bar{t}) is a global solution to (P2). It follows from $a_k \leq t^* \leq b_k, k = 1, 2, \dots$ that, by letting $k \rightarrow \infty$, we can obtain $a^* = t^*$. Noting that $\{M_k\}$ converges to t^* , by (ii) of Theorem 2.1, we can conclude that (\bar{x}, \bar{t}) is a global solution to (P2). \square

GOPF algorithm provides an alternative method to solve (P1). In GOPF algorithm, it does not need to increase the parameter M to ∞ , which differs from other penalty function methods [17–20]. Theorem 2.3 is obviously of great theoretical interest, however, if we use standard optimization methods to solve the sequences of problem (PM) we may possibly find local solutions of problem (PM). Then, it is interesting to analyze the more practical algorithm scheme, where we suppose only that at each step we are able to find a local solution of (PM). Next, we consider how to find a local solution to (P2) and obtain some results which differ from those in previous subsection. To ensure the feasibility of (P2), we additionally need to adjust the constraint penalty parameter ρ under certain circumstance.

2.2 A New Penalty Function Algorithm for a Local Solution

In the following, a point $(x, t) \in \mathbb{R}^{n+1}$ is said to be an ε -feasible solution of (P2) if

$$f_i(x) - t \leq \varepsilon, \quad i \in I, \quad g_j(x) \leq \varepsilon, \quad j \in J, \quad |h_l(x)| \leq \varepsilon, \quad l \in L. \tag{2.6}$$

For any given $\varepsilon > 0$, a point (\bar{x}, \bar{t}) is said to be an ε -local solution of (P2) if (\bar{x}, \bar{t}) is an ε -feasible solution of (P2) and there exists a neighborhood $N_{(\bar{x}, \bar{t})}$ of (\bar{x}, \bar{t}) such that $\bar{t} \leq t + \varepsilon$ for any $(x, t) \in N_{(\bar{x}, \bar{t})} \cap \bar{X}$.

Theorem 2.4 *Let $x_0 \in X, t_0 = F(x_0)$, and $t^* = \min_{(x,t) \in \bar{X}} t$. For any given $\varepsilon > 0$, suppose that for some $M \leq t_0$ and $\rho > 0$, $(x_{M,\rho}^*, t_{M,\rho}^*)$ is a local solution of (PM) starting at (x_0, t_0) . Then, the following statements hold:*

- (i) *If $E(x_{M,\rho}^*, t_{M,\rho}^*; M, \rho) = 0$, then $(x_{M,\rho}^*, t_{M,\rho}^*)$ is feasible to (P2) and $t^* \leq t_M^* \leq M$.*

- (ii) If $E(x_{M,\rho}^*, t_{M,\rho}^*; M, \rho) > 0$ and $(x_{M,\rho}^*, t_{M,\rho}^*)$ is feasible to (P2), then $(x_{M,\rho}^*, t_{M,\rho}^*)$ is a local solution to (P2).
- (iii) If $E(x_{M,\rho}^*, t_{M,\rho}^*; M, \rho) > 0$ and $(x_{M,\rho}^*, t_{M,\rho}^*)$ is not feasible to (P2), let $\rho_1 = K\rho$ ($K > 0$) and suppose that $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is a local solution of (PM) with $\rho = \rho_1$ starting at (x_0, t_0) . Then, there exists a $K_0 > 0$ such that $K \geq K_0$, $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is an ε -local solution of (P2).

Proof

- (i) The result is obvious from the definition of $E(x, t; M, \rho)$.
- (ii) Since $(x_{M,\rho}^*, t_{M,\rho}^*)$ is a local solution to (PM) and a feasible point to (P2), there exists a neighborhood $N_{(x_{M,\rho}^*, t_{M,\rho}^*)}$ of $(x_{M,\rho}^*, t_{M,\rho}^*)$ such that, for all $(x, t) \in N_{(x_{M,\rho}^*, t_{M,\rho}^*)} \cap \bar{X}$

$$0 < \frac{1}{2} [(t_{M,\rho}^* - M)^+]^2 = E(x_{M,\rho}^*, t_{M,\rho}^*; M, \rho) \leq E(x, t; M, \rho) = \frac{1}{2} [(t - M)^+]^2,$$

which implies that

$$t_{M,\rho}^* - M \leq t - M, \quad \forall (x, t) \in N_{(x_{M,\rho}^*, t_{M,\rho}^*)} \cap \bar{X}.$$

Thus, $(x_{M,\rho}^*, t_{M,\rho}^*)$ is a local solution to (P2).

- (iii) We first prove that $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is an ε -feasible solution of (P2). By contraction, suppose that for any $n > 0$, there exists $K_n > n$ such that when $\rho_1 = K_n\rho$, $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is not an ε -feasible solution of (P2). Then, there exists at least $i_0 \in I$ or $j_0 \in J$ or $l_0 \in L$, such that

$$\sum_{i \in I} [(f_i(x_{M,\rho_1}^*) - t)^+]^2 + \sum_{j \in J} [g_j^+(x_{M,\rho_1}^*)]^2 + \sum_{l \in L} h_l^2(x_{M,\rho_1}^*) \geq \varepsilon^2.$$

Since $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is a local solution of (PM) starting from (x_0, t_0) , and (x_0, t_0) is a feasible point of (P2), we have

$$\frac{1}{2} [(t_0 - M)^+]^2 = E(x_0, t_0; M, \rho_1) \geq E(x_{M,\rho_1}^*, t_{M,\rho_1}^*; M, \rho_1) \geq \frac{K\rho}{2} \varepsilon^2.$$

Letting $K \rightarrow +\infty$ in the last inequality above, we have $\frac{1}{2} [(t_0 - M)^+]^2 \rightarrow +\infty$, which leads to a contract that $\frac{1}{2} [(t_0 - M)^+]^2$ is finite. Thus, there exists a $K_0 > 0$ such that $K \geq K_0$; we have $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ as an ε -feasible solution of (P2). Similar to the proof of (ii), we can obtain that $(x_{M,\rho_1}^*, t_{M,\rho_1}^*)$ is an ε -local solution of (P2). \square

Based on Theorem 2.4, we develop an algorithm to find an approximately local solution to (P2). The algorithm solves locally the sequence of minimizations of problem (PM) by adjusting the two parameters M and ρ , which is called a local objective penalty function (for short, LOPF) algorithm.

LOPF Algorithm:

- Step 0** Give $\rho \geq 1$. Choose $1 > \varepsilon > 0$, $x_0 \in X$, and $a_1 < \min_{x \in X} F(x)$. Let $b_1 = t_0 = F(x_0)$, and $M_1 = \frac{a_1 + b_1}{2}$. Let $\rho_1 = \rho$, $p_1 = 0$, and $k := 1$.
- Step 1** Solve $\min_{(x,t) \in Y} E(x, t; M_k, \rho_k)$ starting at (x_0, t_0) , where $Y = X_0 \times [a_1, b_1] \subseteq \mathbb{R}^{n+1}$, X_0 is a simple box containing X . Let (x_k, t_k) be its local solution, and go to Step 2.
- Step 2** If $E(x_k, t_k; M_k, \rho_k) = 0$, go to Step 3; otherwise, $E(x_k, t_k; M_k, \rho_k) > 0$, go to Step 4.
- Step 3** If $b_k - a_k \leq \varepsilon$, go to Step 5; otherwise, let $a_{k+1} := a_k$, $b_{k+1} := M_k$, $M_{k+1} := \frac{a_{k+1} + b_{k+1}}{2}$, $\rho_{k+1} := \rho$, $p_{k+1} := 0$, $k := k + 1$, and go to Step 1.
- Step 4** If (x_k, t_k) is not an ε -feasible solution of (P2), let $\rho_{k+1} := 10\rho_k$, $p_{k+1} := p_k + 1$, $k := k + 1$, and go to Step 1; otherwise, go to Step 5.
- Step 5** Stop and (x_k, t_k) is an ε -local solution to (P2).

Remark 2.2 Any efficient gradient methods available can be used in Step 1. The parameter ρ is increased gradually in Step 4 for guaranteeing that the point (x_k, t_k) is ε -feasible for problem (P2). In numerical experiments later, the parameter ρ does not need to be increased big enough.

Theorem 2.5

- (i) *The LOPF algorithm must stop in finite steps.*
- (ii) *For any given $\varepsilon > 0$ and $\rho \geq 1$, let $\{(x_K, t_K)\}$ be the final solution obtained by LOPF algorithm. Then, $\{(x_K, t_K)\}$ is a local solution or an ε -local solution of (P2).*

Proof Let t^* be the global optimal value of (P2) and (x_k, t_k) be a sequence generated by LOPF algorithm.

- (i) We prove that the LOPF algorithm must stop in finite steps.
 - (1) If $E(x_k, t_k; M_k, \rho_k) = 0$ and $(b_k - a_k) \leq \varepsilon$, then the algorithm stops; otherwise, let $a_{k+1} := a_k$, $b_{k+1} := M_k$, $M_{k+1} := \frac{a_{k+1} + b_{k+1}}{2}$, $\rho_{k+1} = \rho$, $k := k + 1$ and go to Step 1.
 - (2) $E(x_k, t_k; M_k, \rho_k) > 0$, and if (x_k, t_k) is an ε -feasible solution of (P2), then the algorithm stops; otherwise, let $\rho_{k+1} := 10\rho_k$, $p_{k+1} := p_k + 1$, $k := k + 1$ and go to Step 1. From (iii) of Theorem 2.4, there are at most $\lceil \lg \frac{(b_1 - a_1)^2}{4\varepsilon^2} \rceil + 1$ iteration times from Step 4 to Step 1, which means that when $p_k \geq \lceil \lg \frac{(b_1 - a_1)^2}{4\varepsilon^2} \rceil + 1$, (x_k, t_k) must be an ε -feasible solution of (P2), where $\lceil x \rceil$ represents the largest integer less than x . For any k , only if $F(x_k, M_k, \rho_k) = 0$ and $(b_k - a_k) > \varepsilon$, we will change k to $k + 1$ and let $a_{k+1} := a_k$, $b_{k+1} := M_k$, $M_{k+1} := \frac{a_{k+1} + b_{k+1}}{2}$, $\rho_{k+1} = \rho$, go to Step 1. Then, by (i) of Theorem 2.4, we have that $t^* \leq t_k \leq M_k$. In this case, (x_k, t_k) is also a global minimizer for the problem $\min_{(x,t) \in Y} E(x, t; M_k, \rho_k)$. From the proof of Theorem 2.3, we know that

$$a_{k+1} = a_k \leq t^* \leq b_{k+1} \leq b_k \quad \text{and} \quad a_k \leq M_k \leq b_k \quad \text{and}$$

$$b_{k+1} - a_{k+1} \leq \frac{b_k - a_k}{2}.$$

Hence, there exists a $k_0 > 0$, such that for any $k \geq k_0$, $(b_k - a_k) \leq \varepsilon$. Based the above proof, the LOPF algorithm must terminate in finite steps.

(ii) Suppose (x_K, t_K) is the final solution obtained by LOPF algorithm. We now prove that (x_K, t_K) must be a local solution or an ε -local solution of (P2).

By LOPF algorithm, we know that the final obtained solution has only one of the following two cases:

- (1) $E(x_K, t_K; M_K, \rho_K) = 0$ and $(b_K - a_K) \leq \varepsilon$, then $(x_K, t_K) \in \bar{X}$ and $a_K \leq t^* \leq t_K \leq b_K$. Hence, we have that $t_K \leq t^* + \varepsilon$. So, (x_K, t_K) is an ε -local solution of (P2).
- (2) $E(x_K, t_K; M_K, \rho_K) > 0$, and x_K is an ε -feasible solution of (P2). By the proof of Theorem 2.4, we know that (x_K, t_K) must be an ε -local solution of (P2). \square

2.3 How to Choose a_1

From the above algorithms, it can be seen that the initial value of a_1 is difficult to be chosen for finding the optimal solution of problem (P2). In [25], the authors have designed an objective penalty function algorithm for solving nonlinear programming problems with inequality constraints. However, they failed to point out which a_1 should be considered as a good start point. Indeed, when the solution is unknown, it is difficult to choose a suitable a_1 for nonlinear programming problems with inequality constraints. In minimax problems, choosing a suitable one is even more complicated. Next, we will discuss how to choose a suitable a_1 for minimax problems, which is a lower bound of the optimal objective value of (P2).

Let X_0 be a box set containing X , where X is the feasible set of (P1). Based on the particular structure of the minimax problem, consider the following problem:

$$\min_{x \in X_0} f_i(x) \tag{Pi}$$

where $f_i(x)$ is a member of $\{f_1(x), \dots, f_m(x)\}$.

Proposition 2.2 For any given $i \in I$, let \hat{x}_i be the optimal solution of (Pi). Let \bar{x} be the optimal solution of (P1). Then, $f_i(\hat{x}_i) \leq F(\bar{x})$.

Proof For any $x \in X$, we must have $f_i(x) \leq \max_{i \in I} \{f_i(x)\}$. Thus,

$$\min_{x \in X_0} f_i(x) \leq \min_{x \in X} f_i(x) \leq \min_{x \in X} \{ \max_{i \in I} \{f_i(x)\} \}.$$

The desired result can be obtained. \square

Remark 2.3 By Propositions 2.1 and 2.2, we can choose $a_1 \leq f_i(\hat{x}_i)$ for an $i \in I$. Obviously, $a_1 \leq f_i(\hat{x}_i) \leq F(\bar{x}) = \min_{(x,t) \in \bar{X}} t$. For problem (Pi), an objective function

$f_i(x) \in \{f_1(x), \dots, f_m(x)\}$, which is relatively simple or convex, should be chosen firstly.

3 Numerical Experiments

In this section, we consider some preliminary numerical results obtained with LOPF algorithm described in the above section. We perform the numerical experiments on testing problems taken from the literature [18, 30]. We coded LOPF algorithm in Matlab 2010b and ran it on a personal computer (CPU: Intel Core2 1.80 GHz, RAM: 1.99 GB). After a few preliminary tests, the following choices were made and used on all the test problems.

- The stopping parameters are set to $\varepsilon = 10^{-4} - 10^{-3}$, and all the initial constraint parameter is set to $\rho = 10^2$, except for problems 4.12 and 4.13. Set $\rho = 10^0$ and $\rho = 10^1$ for problems 4.12 and 4.13, respectively.
- The value of a_1 is chosen by solving a relatively simple constrained minimization problem (Pi). All the starting points x_0 are taken from the literature.
- All the minimization problems required to be solved in Step 1 are solved by implementing a sequential quadratic programming algorithm.

We ran the algorithm on 11 test problems used in [18] and 12 test problems given in [30]. A full analytical description of the test problems along with the starting points, the optimal solutions found, and the sources of the problems are given in [18, 30]. In Table 1, we outline the data of all these test problems. In some cases, more than one starting point is used. Note that we do not use all test problems from Chap. 4 of [30] because for some of them not all input data are available.

From the results reported in Table 2, we see that the algorithm is capable to solve all test problems considered. A comparison with the data reported in [18, 30] shows that our algorithm is capable of reaching at least the same accuracy (for example, 10^{-3}) of optimal objective values reported in [18, 30], except the problem 4.13. We note that using only the data reported in [18, 30], it is not possible to compare the efficiency of our scheme with those of [18] and [30]. To make future comparisons possible, we report in Table 2 some more data which give a more detailed picture of the computational effort required by our algorithm. More specifically, for each problem, we report the number of outer iterations (N), the total number of objective function evaluations (F), the optimal objective values found by LOPF algorithm (F^*), the optimal objective penalty parameters (M^*), the last constraint penalty parameter (ρ^*), and the CPU time.

By 23 numerical examples, we can obtain the following results:

- (I) For all test problems, numerical experiments show that an approximate solution can be found, except problem 4.13, by comparing with existing results (see Table 1). One significant property of the algorithm is the fact that the gradient of each function f_i need not be computed in every iteration of the

Table 1 Test problems from the literature

Problem	n	I	$J + L$	Optimal value	Starting points
1	2	3	2	1.952 22	(a) (1.0, -0.1); (b) (0.0, 0.0); (c) (2.0, 2.0); (d) (4.0, -4.0)
2	2	6	2	0.616 43	(a) (3.0, 1.0); (b) (1.0, 3.0)
3	2	3	2	0.0	(a) (3.0, 1.0); (b) (50.00, 0.05)
4	2	3	2	2.25	(a) (2.1, 1.9); (b) (1.9, 2.1); (c) (2.0, 4.0); (d) (4.0, 2.0)
5	4	4	3	-44.0	(a) (0.0, 0.0, 0.0, 0.0); (b) (-2.0, 1.0, -2.0, 1.0)
6	2	3	2	2.0	(a) (0.0, 1.0); (b) (3.0, 1.0)
7	7	5	4	680.630 06	(a) (1.0, 2.0, 0.0, 4.0, 0.0, 1.0, 1.0) (b) (3.0, 3.0, 0.0, 5.0, 1.0, 3.0, 0.0)
8	2	4	2	0.0	(a) (-1.2, 1.0)
9	2	2	2	2.718 28	(a) (50.00, 0.05); (b) (1.0, 1.1)
10	2	2	2	0.0	(a) (1.418 31, -4.794 62)
11	10	9	8	108.016 69	(a) (2.0, 3.0, 5.0, 5.0, 1.0, 2.0, 7.0, 3.0, 6.0, 10.0)
4.1	2	3	1	-0.389 659 52	(a) (1.0, 2.0)
4.2	2	3	1	-0.330 357 14	(a) (-2.0, -1.0)
4.3	2	3	1	-0.448 910 79	(a) (-1.00, 0.01)
4.4	2	3	1	-0.429 280 61	(a) (-1.0, 3.0)
4.5	6	3	15	-1.859 618 70	(a) (0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
4.6	7	163	9	0.101 830 89	(a) (0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5)
4.7	8	8	1	0.0	(a) (0.125, ..., 0.125)
4.8	10	6	3	24.306 209	(a) (2.0, 3.0, 5.0, 5.0, 1.0, 2.0, 7.0, 3.0, 6.0, 10.0)
4.10	20	38	10	0.506 947 99	(a) (100, ..., 100)
4.12	10	9	5	-1 768.807 0	(a) (1 745, 12 000, 110, 3 048, 1 974, 89.2, 92.8, 8.0, 3.6, 145)
4.13	7	13	2	1 227.226 0	(a) (1 745, 110, 3 048, 89, 92, 8, 145)
4.15	16	19	1	174.786 99	(a) (0.80, 0.83, 0.85, 0.87, 0.90, 0.10, 0.12, 0.19, 0.25, 0.29, 512, 13.10, 71.80, 640, 650, 5.70)

Optimal value: optimal objective value of $F(x)$ from the literature [18, 30], Start points: different starting points from the literature [18, 30]

n number of variables, I number of functions f_i , $J+L$ number of constrained functions g_j and h_l

- algorithm. In general, as M_k approaches F^* , fewer and fewer elements $f_i(x_k)$ will satisfy the condition $f_i(x_k) > M_k$.
- (II) The optimal solutions can be found from different starting points. The present algorithm shows the robustness. For problem 4, however, a special case occurs that a local solution was obtained from the initial point (b). The same case also happened in [18].
 - (III) It is important to choose the value of a_1 which has directly an effect on the number of function calls. From Table 2, we find that the more the function calls, the more the time on CPU. So, choosing a suitable a_1 by Remark 2.3 is necessary in practice. In addition, the number of outer iterations is closely

Table 2 Results for test problems

Problem	Start	N	F	F^*	M^*	ρ^*	CPU (s)
1	(a)	13	424	1.952 289 58	1.938 040	1.0e+3	0.406 250
	(b)	15	516	1.952 290 02	1.937 805	1.0e+3	0.484 375
	(c)	15	521	1.952 267 08	1.937 881	1.0e+3	0.515 625
	(d)	19	722	1.952 234 71	1.937 900	1.0e+3	0.640 625
2	(a)	14	562	0.616 443 68	0.602 631	1.0e+3	0.453 125
	(b)	14	850	0.616 439 61	0.602 631	1.0e+3	0.546 875
3	(a)	14	539	0.000 024 70	-0.013 392	1.0e+3	0.453 125
	(b)	15	579	0.000 008 89	-0.013 986	1.0e+3	0.515 625
4	(a)	13	449	2.249 810 47	2.235 555	1.0e+2	0.421 875
	(b)	13	448	4.044 562 38 ^A	4.033 986	1.0e+2	0.406 250
	(c)	15	619	2.249 827 05	2.236 023	1.0e+2	0.578 125
	(d)	15	574	2.250 102 47	2.236 223	1.0e+2	0.562 500
5	(a)	16	1 300	-43.999 965 52	-44.014 359	1.0e+2	0.734 375
	(b)	17	1 162	-43.999 864 48	-44.012 867	1.0e+2	0.718 750
6	(a)	13	549	2.000 061 45	1.985 952	1.0e+2	0.406 250
	(b)	17	570	2.000 006 77	1.986 000	1.0e+2	0.531 250
7	(a)	17	4 008	680.648 604 47	680.638 527	1.0e+4	1.515 625
	(b)	22	4 645	680.631 401 33	680.617 283	1.0e+4	1.937 500
8	(a)	13	919	0.000 006 31	-0.013 672	1.0e+3	0.578 125
9	(a)	16	747	2.718 306 35	2.704 320	1.0e+2	0.609 375
	(b)	17	585	2.718 314 64	2.703 864	1.0e+2	0.562 500
10	(a)	12	463	0.000 000 21	-0.014 298	1.0e+3	0.453 125
11	(a)	21	5 815	108.014 975 68	109.945 009	1.0e+3	2.406 250
4.1	(a)	13	586	-0.389 675 39	-0.403 809	1.0e+2	0.515 625
4.2	(a)	13	535	-0.330 406 17	-0.344 238	1.0e+2	0.578 125
4.3	(a)	13	594	-0.449 077 69	-0.462 889	1.0e+2	0.500 000
4.4	(a)	11	610	-0.429 268 65	-0.443 584	1.0e+2	0.406 250
4.5	(a)	13	1 304	-1.859 658 50	-1.873 474	1.0e+2	0.781 250
4.6	(a)	14	1 519	0.101 887 70	0.097 418	1.0e+3	1.125 000
4.7	(a)	14	2 930	0.000 193 37	0.011 793	1.0e+2	1.578 125
4.8	(a)	20	4 339	24.304 791 98	24.292 490	1.0e+2	2.109 375
4.10	(a)	25	5 482	0.507 053 82	0.493 070	1.0e+2	3.031 250
4.12	(a)	25	8 289	-1 768.805 541 48	-1 768.960 022	1.0e+0	3.531 250
4.13	(a)	22	1 113	1 228.123 059 15	1 228.277 284	1.0e+1	0.734 375
4.15	(a)	19	9 419	174.805 030 07	174.791 084	1.0e+2	4.515 625

^A Find a local solution from the initial point (b)

related to the number of variables. The more the variables, the more the outer iterations.

- (IV) The parameter M approximates the optimal objective value F^* . Thus, these numerical results accord with the results proven in previous section. It is worth stressing that the penalty parameter M converges to a finite value,

unlike other penalty parameters increasing to infinity. Moreover, from the last second column of Table 2, it can be observed that the constraint penalty parameter ρ increases gradually to a finite value, up to 10^4 .

In conclusion, the numerical results indicate that the algorithm is promising. The adaptive objective penalty parameter rule provides a useful mean of approximating the optimal objective value while the algorithm is ensured of progress toward the approximate solution.

4 Conclusion

We introduced a new objective penalty function method for the solution of minimax problems with equality and inequality constraints. The method is approximating an optimal objective value for an equivalent minimization reformulation of the minimax problem by a series of smooth, simple problems with only box constraints. Numerical results reported indicate that the method introduced in this paper can be numerically efficient. The objective penalty function approach is quite different from existing proposals for the same class of problems. We think that, besides its theoretical properties, the proposed algorithm has one distinct practical advantage: it only requires the use of standard gradient optimization methods for the solution of smooth optimization problems, and no other complex operations are necessary.

References

- [1] Demyanov, V.F., Malozemov, V.N.: Introduction to Minimax. Wiley, New York (1974)
- [2] Du, D.Z., Pardalos, P.M.: Minimax and Applications. Kluwer, Dordrecht (1995)
- [3] Charalambous, C., Conn, A.R.: An efficient method to solve the minimax problem directly. *SIAM J. Numer. Anal.* **15**, 162–187 (1978)
- [4] Overton, M.L.: Algorithms for nonlinear l_1 and l_∞ fitting. In: Powell, M.J.D. (ed.) *Nonlinear Optimization*, pp. 213–221. Academic Press, London (1981)
- [5] Ploak, E.: Basics of minimax algorithms. In: Clarke, F.H., Demyanov, V.F., Giannesi, F. (eds.) *Nonsmooth Optimization and Related Topics*, pp. 343–369. Plenum, New York (1989)
- [6] Zang, I.: A smoothing technique for min–max optimization. *Math. Program.* **19**, 61–77 (1980)
- [7] Li, X.S., Fang, S.C.: On the entropic regularization method for solving min–max problems with applications. *Math. Method Oper. Res.* **46**, 119–130 (1997)
- [8] Li, X.S., Pan, S.: Solving the finite min–max problem via an exponential penalty method. *Comput. Technol.* **8**, 3–15 (2003)
- [9] Xu, S.: Smoothing method for minimax problems. *Comput. Optim. Appl.* **20**, 267–279 (2001)
- [10] Polak, E., Royset, J.O., Womersley, R.S.: Algorithms with adaptive smoothing for finite minimax problems. *J. Optim. Theory Appl.* **119**, 459–484 (2003)
- [11] Bagirov, A.M., Al Nuaimat, A., Sultanova, N.: Hyperbolic smoothing function method for minimax problems. *Optimization.* (2012). doi:[10.1080/02331934.2012.675335](https://doi.org/10.1080/02331934.2012.675335)
- [12] Zhou, J.L., Tits, A.L.: Nonmonotone line search for minimax problems. *J. Optim. Theory Appl.* **76**, 455–476 (1993)
- [13] Zhu, Z., Cai, X., Jian, J.: An improved SQP algorithm for solving minimax problems. *Appl. Math. Lett.* **22**, 464–469 (2009)
- [14] Vardi, A.: New minimax algorithm. *J. Optim. Theory Appl.* **75**, 613–634 (1992)

- [15] Wang, F., Zhang, K.: A hybrid algorithm for nonlinear minimax problems. *Ann. Oper. Res.* **164**, 167–191 (2008)
- [16] Ye, F., Liu, H., Zhou, S., Liu, S.: A smoothing trust-region Newton-CG method for minimax problem. *Appl. Math. Comput.* **199**, 581–589 (2008)
- [17] Di Pillo, G., Grippo, L., Lucidi, S.: A smooth method for the finite minimax problem. *Math. Program.* **60**, 187–214 (1993)
- [18] Rustem, B., Nguyen, Q.: An algorithm for the inequality-constrained discrete min–max problem. *SIAM J. Optim.* **8**, 265–283 (1998)
- [19] Obasanjo, E., Tzallas-Regas, G., Rustem, B.: An interior-point algorithm for nonlinear minimax problems. *J. Optim. Theory Appl.* **144**, 291–318 (2010)
- [20] Ma, C., Li, X., Zhang, L.: New exact penalty function for solving constrained finite min–max problems. *Appl. Math. Mech.* **33**, 253–270 (2012)
- [21] Morrison, D.D.: Optimization by least squares. *SIAM J. Numer. Anal.* **5**, 83–88 (1968)
- [22] Fletcher, R.: *Practical Method of Optimization*. Wiley-Interscience, New York (1981)
- [23] Burke, J.V.: Calmness and exact penalization. *SIAM J. Control Optim.* **29**, 493–497 (1991)
- [24] Meng, Z., Hu, Q., Dang, C.: An objective penalty function method for nonlinear programming. *Appl. Math. Lett.* **17**(6), 683–689 (2004)
- [25] Meng, Z., Hu, Q., Dang, C.: A penalty function algorithm with objective parameters for nonlinear mathematical programming. *J. Ind. Manage. Optim.* **5**, 585–601 (2009)
- [26] Meng, Z., Dang, C., Jiang, M., Xu, X., Shen, R.: Exactness and algorithm of an objective penalty function. *J. Glob. Optim.* (2012). doi:[10.1007/s10898-012-9900-9](https://doi.org/10.1007/s10898-012-9900-9)
- [27] Liu, S., Meng, Z.: A new algorithm for solving constrained optimization on exact penalty with two parameters (in Chinese). *Math. Appl.* **22**(2), 346–351 (2009)
- [28] Dutta, S.R.K., Vidyasagar, M.: New algorithms for constrained minimax optimization. *Math. Program.* **13**, 140–155 (1977)
- [29] Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: *Nonlinear Programming: Theory and Algorithms*. Wiley-interscience, New York (2006)
- [30] Lukšan, L., Vlček, J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report No. 798, Institute of Computer Science, Academy of Sciences of the, Czech Republic, January (2000)