# SOLVING THE FINITE MIN-MAX PROBLEM VIA AN EXPONENTIAL PENALTY METHOD*

X. Li
*Department of Engineering Mechanics, Dalian University of Technology*
*Dalian, China*
e-mail: Lixs@dlut.edu.cn

S. Pan
*Department of Applied Mathematics, Dalian University of Technology*
*Dalian, China*
e-mail: pshcoral12@sina.com

В работе рассмотрен метод сглаживания для конечной задачи минимакса, которая возникает при применении метода экспоненциального штрафа к его эквивалентной нелинейной программе. В результате исходная задача может быть решена с помощью безусловной минимизации гладкой функции. Для полноты приведен формальный вывод для экспоненциальных функций штрафа и описаны некоторые важные приложения результирующих гладких функций. Предложенные результаты демонстрируют простоту и эффективность данного подхода.

## Introduction

Consider the finite min-max problem (P) defined as

$$\min_{\mathbf{x}\in\mathbb{R}^n} \phi(\mathbf{x}), \tag{1}$$

where

$$\phi(\mathbf{x}) := \max_{1\le i\le m} \{g_i(\mathbf{x})\}. \tag{2}$$

In this work, we assume that all component functions $g_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, $i = 1, 2, \dots, m$ are continuously differentiable. Nevertheless, this is a typical non-smooth optimization problem due to the nondifferentiability of the max function $\phi(\mathbf{x})$.

The problem defined in (1) is one frequently arisen in scientific and engineering computing [1−3], due to its wide applications such as data fitting, game theory and economic equilibria. Therefore, the study on its robust and efficient algorithms becomes a prolonged research subject.

One way to deal with this problem is to transform it into an equivalent nonlinear programming problem (NLP)

$$\begin{aligned} &\min_{w,\mathbf{x}} \ w \\ &s.t.\ \ g_i(\mathbf{x}) - w \le 0,\ i = 1, 2, \dots, m, \end{aligned} \tag{3}$$

*This work is supported by special fund for basic research under grant G1999032805.

which is then solved by certain minimization algorithms [4 − 8]. In contrast, smoothing techniques [9 − 11] try to find a smooth function to approximate the non-smooth max-type function. The latter approach has apparent advantages over the former one because it is not only applicable to solving the min-max problem (P) itself, but also applicable to a variety of contexts, especially in the recent hot smoothing algorithms for the solution of complementarity, variational inequality and MPEC (mathematical programming with equilibrium constraints) problems.

The main purpose for this paper is to show that a smooth function with a uniform approximation property could be derived by means of an ordinary penalty method. This approach is surprisingly simple, but is not appeared in the literature from our search. At the same time, this paper is to show that a penalty function could be "derived", not just understood with "naive" sense. Such an approach could provide a new insight into penalty function methods. To this end, we add a formal derivation for the exponential penalty functions before applying it to the min-max problem.

The paper is organized as follows. Section 1 is devoted to deriving exponential penalty functions for the inequality constrained optimization problem by Lagrangian perturbation. In Section 2, we exploit the exponential penalty methods to solve the finite min-max problem (P) and discuss some properties of resulted smooth functions. In Section 3, several important applications of the smooth functions are described. In Section 4, we give a very simple algorithm and report the numerical results for some test problems from the literature. Finally, brief conclusions are drawn in Section 5.

# 1. Derivation of exponential penalty functions

The exponential penalty approach is one of the important optimization methods both in theoretical and algorithmic developments as an instrument for converting constrained problems into unconstrained ones. Nevertheless, it is rarely to make inquires about the origin of exponential penalty functions and their relations to other means that play the same role, for example the Lagrangian function. By deriving the exponential penalty functions through a Lagrangian perturbation procedure in this section, we are intended to establish a formal link of exponential penalty functions with Lagrangian function and give a new insight into exponential penalty methods.

Consider a general inequality constrained optimization problem (GNLP)

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) \\ &s.t. \quad g_i(\mathbf{x}) \le 0, \ i = 1, 2, \ldots, m, \end{aligned} \tag{4}$$

where $f(\mathbf{x})$ and $g_i(\mathbf{x}),\ i = 1, 2, \ldots, m$ are assumed to be continuously differentiable functions from $\mathbb{R}^n$ to $\mathbb{R}$.

As a matter of fact, the above problem (GNLP) can be equivalently written as an unconstrained optimization problem in the following form:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \Psi(\mathbf{x}) := \sup_{\boldsymbol{\lambda}\geq\mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) \tag{5}$$

due to the following fact

$$\sup_{\boldsymbol{\lambda}\geq\mathbf{0}} L(\mathbf{x}, \boldsymbol{\lambda}) = \begin{cases} f(\mathbf{x}) & \mathbf{x} \in X, \\ +\infty & \mathbf{x} \notin X, \end{cases}$$

where $L(\mathbf{x}, \boldsymbol{\lambda})$ is the ordinary Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n, \boldsymbol{\lambda} \in \mathbb{R}^m_+$$

and $X := \{\mathbf{x} \in \mathbb{R}^n \big| g_i(\mathbf{x}) \le 0,\ i = 1, 2, \ldots, m\}$ represents the feasible set of (GNLP). Although the formulation of (5) is very attractive, unfortunately, the linear form of $L(\mathbf{x}, \boldsymbol{\lambda})$ in dual variable $\boldsymbol{\lambda}$ prevents from doing this to get a continuous and analytical solution $\boldsymbol{\lambda}(\mathbf{x})$.

To circumvent this drawback, we introduce an entropy function $\xi(\boldsymbol{\lambda}) = \sum_{i=1}^m \lambda_i \ln \lambda_i$ as a perturbing term into the Lagrangian $L(\mathbf{x}, \boldsymbol{\lambda})$ to formulate a perturbed problem

$$\sup_{\boldsymbol{\lambda} \ge \mathbf{0}} \Big\{ L_p(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) - p^{-1} \xi(\boldsymbol{\lambda}) \Big\}. \tag{6}$$

For this perturbation problem, we have the following theorem:

**Theorem 1.** *The perturbation problem (6) yields a smooth function $\Psi_p(\mathbf{x})$ that pointwisely approximates to $\Psi(\mathbf{x})$ in $\mathbb{R}^n$ as $p \to +\infty$, so that the original problem* (GNLP) *is transformed into a smooth unconstrained optimization problem*

$$\min_{\mathbf{x} \in \mathbb{R}^n} \Psi_p(\mathbf{x}). \tag{7}$$

**Proof.** As the entropy function $\xi(\boldsymbol{\lambda})$ is strictly convex in the closed convex set $\mathbb{R}^m_+$ and coercive, i.e., for every sequence $\{\boldsymbol{\lambda}^k\} \in \mathbb{R}^m_+$ such that $\|\boldsymbol{\lambda}^k\| \to +\infty$, it always holds

$$\lim_{k \to \infty} \xi(\boldsymbol{\lambda}^k) = +\infty$$

the problem (6) must have a unique solution $\boldsymbol{\lambda}(\mathbf{x}, p)$ for any $\mathbf{x} \in \mathbb{R}^n$ and the objective function $L_p(\mathbf{x}, \boldsymbol{\lambda})$ attains a finite maximum at this point, thereby defines a function of variable $\mathbf{x}$

$$\Psi_p(\mathbf{x}) := L\big(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}, p)\big) = f(\mathbf{x}) + p^{-1} \xi^*\big(p\mathbf{g}(\mathbf{x})\big), \tag{8}$$

where the last equality comes from the definition of conjugate function, $\xi^*$ is the convex conjugate of $\xi$ and $\mathbf{g}(\mathbf{x}) = \big(g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})\big)^{\mathrm{T}}$. Since $\xi(\boldsymbol{\lambda})$ is a Legendre type convex function, i.e., $\xi \in L(\mathbb{R}^m_+)$, it follows from [12] that $\xi^* \in L(\mathrm{int}(\mathrm{dom}\, \xi^*))$ is also a Legendre type convex function. Thus, considering $\Psi_p(\mathbf{x})$ is a real-valued function on $\mathbb{R}^n$, we can infer that $\Psi_p(\mathbf{x})$ is smooth on the whole space $\mathbb{R}^n$.

Furthermore, $\xi(\boldsymbol{\lambda})$ is low-bounded in $\mathbb{R}^m_+$ and $\xi(\mathbf{0}) = 0$, so there holds

$$0 \le \sup_{\boldsymbol{\lambda} \ge \mathbf{0}} \Big\{ \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) - p^{-1} \xi(\boldsymbol{\lambda}) \Big\} \le \sup_{\boldsymbol{\lambda} \ge \mathbf{0}} \Big\{ \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) \Big\} + \frac{m}{pe}. \tag{9}$$

In view of (5), this demonstrates

$$\lim_{p \to +\infty} \Psi_p(\mathbf{x}) = \Psi(\mathbf{x}), \quad \forall \mathbf{x} \in X.$$

However, for $\mathbf{x} \notin X$, i.e., there at least exists an index $j$ such that $g_j(\mathbf{x}) > 0$,

$$\Psi_p(\mathbf{x}) = f(\mathbf{x}) + p^{-1} \sum_{i \ne j} \sup_{\lambda_i \ge 0} \Big\{ \lambda_i \big(p g_i(\mathbf{x})\big) - \lambda_i \ln \lambda_i \Big\} + $$
$$+ p^{-1} \sup_{\lambda_j \ge 0} \Big\{ \lambda_j \big(p g_j(\mathbf{x})\big) - \lambda_j \ln \lambda_j \Big\},$$

where as $p \to +\infty$, the last term in the right hand of last equation approaches to $+\infty$ due to the boundedness of $\xi(\boldsymbol{\lambda})$, and the second term to 0 in terms of (9). Hence, $\Psi_p(\mathbf{x})$ approximates pointwise to $\Psi(\mathbf{x})$ as $p \to +\infty$.
This completes the proof. □

By a simple computation, $\xi^*(\boldsymbol{\lambda}^*) = \sum_{i=1}^m \exp(\boldsymbol{\lambda_i}^* - 1)$. So, following the equation (8), we have the smooth function $\Psi_p(\mathbf{x})$ in the following form:

$$\Psi_p(\mathbf{x}) = f(\mathbf{x}) + p^{-1} \sum_{i=1}^m \exp\big[pg_i(\mathbf{x}) - 1\big], \tag{10}$$

which is just the familiar exponential penalty function. As the unique solution of perturbed problem (6),

$$\lambda_i(\mathbf{x}, p) = \exp\big[pg_i(\mathbf{x}) - 1\big],\ i = 1, 2, \ldots, m, \tag{11}$$

which provides an estimate for the Lagrange multipliers of the original problem (GNLP). Other exponential penalty functions, up to a constant difference with that given by (10), are often encountered in the literature and could be similarly derived through slight changes of $\xi(\boldsymbol{\lambda})$.

If Kullback — Leibler cross entropy $\xi(\boldsymbol{\lambda}, \boldsymbol{\mu}) := \sum_{i=1}^m \lambda_i \ln \dfrac{\lambda_i}{\mu_i}$ is adopted as the perturbing term, the exponential multiplier penalty function

$$\Psi_p(\mathbf{x}, \boldsymbol{\mu}) = f(\mathbf{x}) + p^{-1} \sum_{i=1}^m \mu_i \exp\big[pg_i(\mathbf{x}) - 1\big] \tag{12}$$

can be derived in a very similar manner. The vector $\boldsymbol{\mu}$ ($\geq \mathbf{0}$) of Lagrange multipliers has to be updated as do the other methods of multipliers.

The above derivation shows that the traditional exponential penalty functions are conjugate to some specific perturbed Lagrangian functions in nature, and incorporate some excellent properties such as strict convexity, coercive and barrier role, which play a central role in the convergence analysis of exponential penalty methods [13, 14].

## 2. Smoothing methods for the finite min-max problem (P)

Now we are going to the main scheme of this paper, using the exponential penalty functions to solve the finite min-max problem (P). Here, we apply the exponential penalty function $\Psi_p(\mathbf{x})$ given by (10) to its equivalent formulation (NLP), whereby transform the constrained problem into the following unconstrained one:

$$\min_{w,\mathbf{x}} \Psi_p(w, \mathbf{x}) = w + p^{-1} \sum_{i=1}^m \exp\Big[p\big(g_i(\mathbf{x}) - w\big) - 1\Big]. \tag{13}$$

Since $\Psi_p(w, \mathbf{x})$ is a convex function of variable $w$ for any $\mathbf{x} \in \mathbb{R}^n$, the optimality condition for $w$ to be a solution of (NLP) is

$$\partial \Psi_p(w, \mathbf{x}) / \partial w = 1 - \sum_{i=1}^m \exp\Big[p\big(g_i(\mathbf{x}) - w\big) - 1\Big] = 0,$$

which means the following relationship

$$w^* = p^{-1} \ln \sum_{i=1}^{m} \exp \big[ p g_i(\mathbf{x}) \big] - p^{-1} \tag{14}$$

holds between the optimal variables $w^*$ and $\mathbf{x}$. Substituting it into $\Psi_p(w, \mathbf{x})$ to eliminate the variable $w$ yields

$$\phi_p(\mathbf{x}) := \Psi_p(w^*, \mathbf{x}) = p^{-1} \ln \sum_{i=1}^{m} \exp \big[ p g_i(\mathbf{x}) \big]. \tag{15}$$

Accordingly, the problem (13), hence the min-max problem (P), reduces to

$$\min_{\mathbf{x} \in \mathbb{R}^n} \phi_p(\mathbf{x}) \tag{16}$$

due to the equivalence between (NLP) and (P). Thus, the original non-smooth problem (P) is converted into an unconstrained minimization problem with a smooth objective function $\phi_p(\mathbf{x})$. This greatly facilitates the numerical solution.

Concerning the difference between $\phi(\mathbf{x})$ and $\phi_p(\mathbf{x})$, we give the following properties [11]:

1) **error bound:** $\phi(\mathbf{x}) \leq \phi_p(\mathbf{x}) \leq \phi(\mathbf{x}) + \ln(m)/p$ ;
2) **monotonicity:** $\phi_p(\mathbf{x}) \geq \phi_q(\mathbf{x}), \quad p \leq q$;
3) **approximation:** $\lim_{p \to +\infty} \phi_p(\mathbf{x}) = \phi(\mathbf{x})$;
4) **convexity:** $\phi(\mathbf{x})$ is convex if all $g_i(\mathbf{x}),\ i = 1, 2, \ldots, m$ are convex.

The above property of uniform approximation ensures that an $\varepsilon-$optimal solution of the min-max problem (P) can be found by solving the problem (16) as long as $p \geq \ln(m)/\varepsilon$. Meanwhile, the following formula

$$\lambda_i(\mathbf{x}) = \frac{\exp \big[ p g_i(\mathbf{x}) \big]}{\sum_{i=1}^{m} \exp \big[ p g_i(\mathbf{x}) \big]}, \ \ i = 1, 2, \ldots, m$$

provides an adaptive estimate for Lagrange multipliers of the problem (P) during iterations.

In addition, when applying the exponential multiplier penalty function $\Psi_p(\mathbf{x}, \boldsymbol{\mu})$ defined in (12) to the problem (NLP), we can obtain, by a similar procedure, another smooth function

$$\phi_p(\mathbf{x}, \boldsymbol{\mu}) = p^{-1} \ln \sum_{i=1}^{m} \mu_i \exp \big[ p g_i(\mathbf{x}) \big],$$

where $\sum_{i=1}^{m} \mu_i = 1,\ \mu_i \geq 0,\ i = 1, 2, \ldots, m$.

Though the smooth functions $\phi_p(\mathbf{x})$ and $\phi_p(\mathbf{x}, \boldsymbol{\mu})$ could be derived by some other procedures (see [9 − 11]), the derivation given in this paper is very elegant and does not resort to any complex concept.

Compared with $\phi_p(\mathbf{x})$, the smooth function $\phi_p(\mathbf{x}, \boldsymbol{\mu})$ has an advantage of avoiding the ill-conditioning caused when $p \to +\infty$, with extra cost of multiplier updates. To our experiences, however, there exists a "zero" pitfall for using $\phi_p(\mathbf{x}, \boldsymbol{\mu})$, which may result in the instability even premature termination of algorithms at a non-optimal solution. Therefore, use of this smooth function needs very sophisticated care for algorithmic implementation.

# 3. Some applications of smoothing function $\phi_p(\mathbf{x})$

As aforementioned, the smoothing method has wide applications other than min-max problem. In this section we describe some applications which are directly or indirectly related to the non-smooth max function $\phi(\mathbf{x})$. By the word "indirectly", we mean that there are certain problems where the max function stems from some reformulations to meet theoretical or computational requirements. In most of circumstances, the smooth functions $\phi_p(\mathbf{x})$ is more tractable than $\phi_p(\mathbf{x}, \boldsymbol{\mu})$ as commented in the end of last section and its appealing features have made it popular in many recent smoothing algorithms.

## 3.1. Linear programming

At first, we consider a linear programming problem in Karmarkar's standard form (KLP):

$$\min_{\mathbf{x}} \{\mathbf{c}^\intercal \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{0},\ \mathbf{e}^\intercal \mathbf{x} = 1,\ \mathbf{x} \geq \mathbf{0}\},$$

where $\mathbf{c} \in \mathbb{R}^n,\ \mathbf{e} = (1, 1, \ldots, 1)^\intercal \in \mathbb{R}^n$, and $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a full row rank matrix with elements $a_{ij},\ i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, n$. It is easy to show that its dual problem (KLD) is

$$\begin{aligned} &\max_{z, \mathbf{y}}\ z \\ &s.t.\ \mathbf{A}^\intercal \mathbf{y} + z \leq \mathbf{c}, \end{aligned}$$

which is in fact the following linear min-max problem

$$\min_{\mathbf{y}} \max_{1 \leq j \leq n} \left\{ \sum_{i=1}^{m} a_{ij} y_i - c_j \right\}$$

and therefore could be solved by the following unconstrained minimization

$$\min_{\mathbf{y}} p^{-1} \ln \sum_{j=1}^{n} \exp \left[ p \Big( \sum_{i=1}^{m} a_{ij} y_i - c_j \Big) \right].$$

Both dual variable $\mathbf{y}(p)$ and primal variable $\mathbf{x}(p)$, as Lagrange multipliers of the dual problem, recovered from the formula

$$x_j(p) = \frac{\exp \left[ p \big( \sum_{i=1}^{m} a_{ij} y_i - c_j \big) \right]}{\sum_{j=1}^{n} \exp \left[ p \big( \sum_{i=1}^{m} a_{ij} y_i - c_j \big) \right]},\ j = 1, 2, \ldots, n$$

converge, respectively, to the optimal solutions of dual problem (KLD) and primal problem (KLP), when the penalty factor $p \to +\infty$ (see [14]).

For the standard linear programming problem (LP):

$$\min_{\mathbf{x}} \{\mathbf{c}^\intercal \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$$

and its dual problem (LD):

$$\max_{\mathbf{y}, \mathbf{s}} \{\mathbf{b}^\intercal \mathbf{y} : \mathbf{A}^\intercal \mathbf{y} + \mathbf{s} = \mathbf{c},\ \mathbf{s} \geq \mathbf{0}\},$$

where $\mathbf{b} \in \mathbb{R}^m$, matrix $\mathbf{A}$ and vector $\mathbf{c}$ are the same as above. The K — K — T conditions for both primal and dual problems are

$$\begin{aligned}&\mathbf{Ax} = \mathbf{b},\\ &\mathbf{A}^{\intercal}\mathbf{y} + \mathbf{s} = \mathbf{c},\\ &\mathbf{xs} = \mathbf{0}, \mathbf{x} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0},\end{aligned} \tag{17}$$

where $\mathbf{xs}$ denotes the coordinate-wise product of vectors $\mathbf{x}$ and $\mathbf{s}$. With resort to NCP function $\min\{a, b\}$, the above complementary conditions can be replaced equivalently by the following equation

$$\min\{x_i, s_i\} = -\max\{-x_i, -s_i\} = 0,\ i = 1, 2, \dots, n,$$

which is further smoothened to

$$p^{-1} \ln\left[\exp(-px_i) + \exp(-ps_i)\right] = 0, \quad i = 1, 2, \dots, n$$

through the smooth function $\phi_p(\mathbf{x})$. The resulted smooth system of equations can be solved by the classical Newton's method. With such a reformulation, we have developed a primal-dual non-interior point algorithm.

## 3.2. Nonlinear programming

Consider the inequality constrained nonlinear programming problem (ICP):

$$\begin{aligned}&\min_{\mathbf{x}} f_0(\mathbf{x})\\ &s.t.\ f_i(\mathbf{x}) \leq 0,\ i = 1, 2, \dots, m,\end{aligned} \tag{18}$$

where $f_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R},\ i = 0, 1, \dots, m$ are assumed to be continuously differentiable. The presence of inequality constraints brings the main difficulty to its solution.

Contrast with general penalty methods that normally require to solve a sequence of unconstrained problems, the exact penalty methods need only one unconstrained minimization provided that the penalty factor exceeds a threshold. This means a lot of savings for computational efforts. However, smooth exact penalty functions either involve second derivatives or are very complicated whereas two simple exact penalty functions, $L_1$ and $L_\infty$ exact penalty functions, are not differentiable.

For the $L_1$ exact penalty method, (ICP) is transformed into the following non-smooth unconstrained optimization problem:

$$\min_{\mathbf{x}} f_0(\mathbf{x}) + \alpha \sum_{i=1}^{m} \max\left\{0, f_i(\mathbf{x})\right\},$$

whose solution becomes exactly optimal for the original problem as long as the penalty parameter $\alpha > \max\limits_{1 \leq i \leq m} \lambda_i^*$, where $\lambda_i^*,\ i = 1, 2, \dots, m$, denote the optimal Lagrange multipliers of original problem (18). For the $L_\infty$ exact penalty method, (ICP) is converted to

$$\min_{\mathbf{x}} f_0(\mathbf{x}) + \alpha \max\left\{0, f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})\right\}$$

and the penalty factor $\alpha > \sum_{i=1}^{m} \lambda_i^*$ is required in this case. Replacing the above max-valued functions with their smooth substitutes through $\phi_p(\mathbf{x})$, we can get an approximate solution of

(ICP) by solving the smooth optimization problem

$$\min_{\mathbf{x}} f_0(\mathbf{x}) + \frac{\alpha}{p} \sum_{i=1}^{m} \ln \left[1 + \exp\left(p f_i(\mathbf{x})\right)\right] \tag{19}$$

or

$$\min_{\mathbf{x}} f_0(\mathbf{x}) + \frac{\alpha}{p} \ln \left\{1 + \sum_{i=1}^{m} \exp[p f_i(\mathbf{x})]\right\}. \tag{20}$$

Thus, the smooth function $\phi_p(\mathbf{x})$ makes it possible to solve the nonlinear programming problem with these non-differentiable exact penalty functions in practice. Following this line, we have proposed a quasi-exact penalty function method for solving (ICP) in [15].

Another way to resolve the difficulty of inequality constraints is that all constraints of (ICP) are replaced by a single constraint, through the maximum operator, to formulate an equivalent problem

$$\begin{aligned} &\min_{\mathbf{x}} f_0(\mathbf{x}), \\ &s.t.\ \phi(\mathbf{x}) := \max_{1\le i\le m} \{f_i(\mathbf{x})\} \le 0 \end{aligned} \tag{21}$$

and then the non-smooth $\phi(\mathbf{x})$ is replaced by its uniform approximation $\phi_p(\mathbf{x})$. As such, the problem (ICP) with multiple constraints can be approximately solved by solving the following singly-constrained smooth one:

$$\begin{aligned} &\min_{\mathbf{x}} f_0(\mathbf{x}), \\ &s.t.\ \phi_p(\mathbf{x}) \le 0. \end{aligned} \tag{22}$$

In [16] such an approach is termed as the aggregate function method.

## 3.3. Complementarity problem

Consider the vertical nonlinear complementarity problem (VNCP):

$$\mathbf{x} \ge \mathbf{0},\ F_1(\mathbf{x}) \ge \mathbf{0}, \cdots, F_m(\mathbf{x}) \ge \mathbf{0},\ x_j \prod_{i=1}^{m} F_i^j(\mathbf{x}) = 0,$$

where $F_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R},\ i = 1, 2, \ldots, m$ are vector-valued functions and $F_i^j(\mathbf{x})$ denotes the $j^{th}$ component of $F_i(\mathbf{x})$. Obviously, in a special case of $m = 1$, the problem (VNCP) reduces to a general nonlinear complementarity problem (NCP):

$$\mathbf{x} \ge \mathbf{0},\ F_1(\mathbf{x}) \ge \mathbf{0},\ \mathbf{x}^{\mathsf{T}} F_1(\mathbf{x}) = 0.$$

It is easy to verify that the problem (VNCP) is equivalent to the following non-smooth equations:

$$\begin{aligned} &\min\{x_j, F_1^j(\mathbf{x}), \ldots, F_m^j(\mathbf{x})\} = \\ &= -\max\{-x_j, -F_1^j(\mathbf{x}), \ldots, -F_m^j(\mathbf{x})\} = 0, \quad j = 1, 2, \ldots, n. \end{aligned}$$

Once again, one can replace the maximum operator here by smoothing approximation $\phi_p(\mathbf{x})$ and convert it into a smooth system of equations. A non-interior continuation method for generalized linear complementarity problems is developed in [17].

### 3.4. Box constrained variational inequality problem (BVIP)

This problem is to find an $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$ such that

$$(\mathbf{y} - \mathbf{x})^\intercal F(\mathbf{x}) \geq 0 \quad \forall \mathbf{y} \in [\mathbf{l}, \mathbf{u}],$$

where $[\mathbf{l}, \mathbf{u}]$ is a box constraint in $\mathbb{R}^n$ with $\mathbf{l} \leq \mathbf{u}$. It is easy to show that the problem (BVIP) is equivalent to the system of equations:

$$\mathbf{x} - \text{mid}\{\mathbf{l}, \mathbf{u}, \mathbf{x} - F(\mathbf{x})\} = \text{mid}\{\mathbf{x} - \mathbf{l}, \mathbf{x} - \mathbf{u}, F(\mathbf{x})\} = 0,$$

where the mid operator $\text{mid}\{a, b, c\}$ can be represented by

$$\text{mid}\{a, b, c\} = a + b + c - \min\{a, b, c\} - \max\{a, b, c\}.$$

Here the max and min operators could be replaced by smoothing approximations with $\phi_p(\mathbf{x})$ in appropriate forms.

Sample applications given above are all hot topics of recent research. In fact, the smooth function $\phi_p(\mathbf{x})$ has been extensively used in other areas such as multi-objective optimization, bi-level programming, random programming, etc.

## 4. Algorithmic implementation and numerical results

It should be stressed that, with our methods, the solution of min-max problem (P) becomes extremely simple because it is reduced to unconstrained minimization problem (16) with smooth function $\phi_p(\mathbf{x})$ and $\phi_p(\mathbf{x}, \boldsymbol{\mu})$. Thus, any efficient unconstrained optimization algorithms (see BFGS) can be applied to this solution. This will greatly facilitate practical usage for engineers. The computations of given examples serves as an illustration for the simplicity, efficiency, stability, high accuracy and easy implementation of our algorithm. In our computations, the parameter $p$ is deliberately set at a fixed constant for three reasons. At first, it is to show that a high accurate solution can be found by a single minimization with a sufficiently large $p$. Secondly, it is to demonstrate that the ill-conditioning of $\phi_p(\mathbf{x})$ with a quite large $p$ is not as severe as the theory claims. Third is to clarify a misunderstanding on exponential manipulations involved in the function $\phi_p(\mathbf{x})$ and its gradient $\nabla\phi_p(\mathbf{x})$. In fact, the computer overflow could be completely eliminated from computing through the following conversions:

$$\begin{aligned} \phi_p(\mathbf{x}) &= \phi(\mathbf{x}) + p^{-1} \ln \sum_{i=1}^{m} \exp\Big[p\big(g_i(\mathbf{x}) - \phi(\mathbf{x})\big)\Big], \\ \nabla\phi_p(\mathbf{x}) &= \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}), \end{aligned}$$

where $\phi(\mathbf{x})$ denotes the maximum among all the component functions at current iteration and

$$\lambda_i = \frac{\exp\Big[p\big(g_i(\mathbf{x}) - \phi(\mathbf{x})\big)\Big]}{\sum_{i=1}^{m} \exp\Big[p\big(g_i(\mathbf{x}) - \phi(\mathbf{x})\big)\Big]}.$$

At this point, it should be noted that our smoothing algorithms have a fundamental distinction with nonlinear re-scaling algorithms proposed by Polyak [18], where the finite min-max problem (P) is replaced by unconstrained minimization of

$$L(\mathbf{x}, \boldsymbol{\mu}, p) = p^{-1} \sum_{i=1}^{m} \mu_i \big[ \exp(p g_i(\mathbf{x})) - 1 \big],$$

which is defined as Lagrangian function for an equivalent problem obtained by certain exponential transformations on original constraint functions. Although both kinds of algorithms have mimic objective functions and same optimal solution $x^*$, the overflow of exponential operations is hard to be avoided without our logarithmic operator in $L(\mathbf{x}, \boldsymbol{\mu}, p)$. Furthermore, the "zero" pitfall on multiplier updates will also occur when using $L(\mathbf{x}, \boldsymbol{\mu}, p)$, leading to premature termination of algorithms. In the computation of all examples below, we only use the smooth function $\phi_p(\mathbf{x})$ which has favorable properties and is the most tractable one from our experiences.

Based on the above fact, we establish a simple algorithm for solving the min-max problem (P) as follows:

**Algorithm 1**

(S.0): Given any initial point $\mathbf{x}^0$ and set $p = \ln(m) \cdot (1.0e{+}5)$.

(S.1): Apply BFGS algorithm to performing a unconstrained minimization of $\phi_p(\mathbf{x})$ and let $\hat{\mathbf{x}}$ denote the minimizer.

The examples given below are all taken from some renowned authors' work, where they are calculated by some sophisticated algorithms. Preliminary numerical results of Algorithm 1 are reported in sequel, where $\mathbf{x}^*$ denotes the optimal point of problem and $\hat{\mathbf{x}}$ is the minimizer yielded by the above algorithm.

*Example 1*[19]
$g_1(\mathbf{x}) = x_1^2 + x_2^4,\ g_2(\mathbf{x}) = (2 - x_1)^2 + (2 - x_2)^2,\ g_3(\mathbf{x}) = 2\exp[-x_1 + x_2],$
$\mathbf{x}^* = (1.13904, 0.89956),\ \phi(\mathbf{x}^*) = 1.95222,\ \mathbf{x}^0 = (1, -0.1),$
$\hat{\mathbf{x}} = (1.139037261, 0.89955975),\ \phi_p(\hat{\mathbf{x}}) = 1.95223071.$
*Example 2*[19]
$g_1(\mathbf{x}) = x_1^4 + x_2^2,\ g_2(\mathbf{x}) = (2 - x_1)^2 + (2 - x_2)^2,\ g_3(\mathbf{x}) = 2\exp[-x_1 + x_2],$
$\mathbf{x}^* = (1, 1),\ \phi(\mathbf{x}^*) = 2,\ \mathbf{x}^0 = (2, 2),$
$\hat{\mathbf{x}} = (1.00000105, 0.99999750)^{\mathrm{T}},\ \phi_p(\hat{\mathbf{x}}) = 2.0.$
*Example 3*[19]
$g_1(\mathbf{x}) = 5x_1 + x_2,\ g_2(\mathbf{x}) = -5x_1 + x_2,\ g_3(\mathbf{x}) = x_1^2 + x_2^2 + 4x_2,$
$\mathbf{x}^* = (0, -3),\ \phi(\mathbf{x}^*) = -3,\ \mathbf{x}^0 = (1, 1),$
$\hat{\mathbf{x}} = (0.00000000, -3.00000000),\ \phi_p(\hat{\mathbf{x}}) = -2.99999000.$
*Example 4*[19]
$g_1(\mathbf{x}) = x_1^2 + x_2^2,\ g_2(\mathbf{x}) = g_1(\mathbf{x}) + 10(-4x_1 - x_2 + 4),\ g_3(\mathbf{x}) = g_1(\mathbf{x}) + 10(-x_1 - 2x_2 + 6),$
$\mathbf{x}^* = (1.2, 2.4),\ \phi(\mathbf{x}^*) = 7.2,\ \mathbf{x}^0 = (-1, 5),$
$\hat{\mathbf{x}} = (1.20000021, 2.40000042),\ \phi_p(\hat{\mathbf{x}}) = 7.20000502.$
*Example 5*[19]
$$g_1(\mathbf{x}) = \frac{x_1 + x_2 y_i}{1 + x_3 y_i + x_4 y_i^2 + x_5 y_i^3} - e^{y_i}\ (1 \le i \le 21),$$
$$g_2(\mathbf{x}) = -\Big(\frac{x_1 + x_2 y_i}{1 + x_3 y_i + x_4 y_i^2 + x_5 y_i^3} - e^{y_i}\Big)\ (22 \le i \le 42),$$
$y_i = -1 + 0.1(i - 1),$

$\mathbf{x}^* = (0.999878, 0.253588, -0.746608, 0.245202, -0.037490),$
$\phi(\mathbf{x}^*) = 0,\ \mathbf{x}^0 = (0.5, 0, 0, 0, 0),$
$\hat{\mathbf{x}} = (0.99987755, 0.25362535, -0.74656767, 0.24515748,$
$-0.03747228),\ \phi_p(\hat{\mathbf{x}}) = 0.00012714.$

*Example 6*[19]

$\phi(\mathbf{x}) = -x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75 \max_{1\le i\le m}\{g_i(\mathbf{x})\},$
$g_1(\mathbf{x}) = x_1^2 + x_2^2 - 1,\ g_2(\mathbf{x}) = -(x_1^2 + x_2^2 - 1),$
$\mathbf{x}^* = (1, 0),\ \phi(\mathbf{x}^*) = -1,\ \mathbf{x}^0 = (-1, -1),$
$\hat{\mathbf{x}} = (0.99999471, 0.00000000),\ \phi_p(\hat{\mathbf{x}}) = -0,99999000.$

*Example 7*[2]

$g_1(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2,\ g_2(\mathbf{x}) = -g_1(\mathbf{x}),\ g_3(\mathbf{x}) = \sin(x_1),$
$g_4(\mathbf{x}) = -g_3(\mathbf{x}),\ g_5(\mathbf{x}) = \cos(x_2),\ g_6(\mathbf{x}) = -g_5(\mathbf{x}),$
$\mathbf{x}^* = (0.453296, -0, 906592),\ \phi(\mathbf{x}^*) = 0.61643246,\ \mathbf{x}^0 = (3, 1),$
$\hat{\mathbf{x}} = (0.45329553, -0.90659105),\ \phi_p(\hat{\mathbf{x}}) = 0.61643610.$

*Example 8*[2]

$g_1(\mathbf{x}) = (1/2)[x_1 + 10x_1/(x_1 + 0.1) + 2x_2^2],$
$g_2(\mathbf{x}) = (1/2)[-x_1 + 10x_1/(x_1 + 0.1) + 2x_2^2],$
$g_3(\mathbf{x}) = (1/2)[x_1 - 10x_1/(x_1 + 0.1) - 2x_2^2],$
$\mathbf{x}^* = (0, 0),\ \phi(\mathbf{x}^*) = 0,\ \mathbf{x}^0 = (3, 1),$
$\hat{\mathbf{x}} = (-0.00073970, 0.19207034),\ \phi_p(\hat{\mathbf{x}}) = 0.00000631.$

*Example 9*[2]

$g_1(\mathbf{x}) = \left[x_1 - \sqrt{x_1^2 + x_2^2}\cos\left(\sqrt{x_1^2 + x_2^2}\right)\right]^2 + 0.005(x_1^2 + x_2^2),$
$g_2(\mathbf{x}) = \left[x_2 - \sqrt{x_1^2 + x_2^2}\sin\left(\sqrt{x_1^2 + x_2^2}\right)\right]^2 + 0.005(x_1^2 + x_2^2),$
$\mathbf{x}^* = (0, 0),\ \phi(\mathbf{x}^*) = 0,\ \mathbf{x}^0 = (1.41831, -4.79462),$
$\hat{\mathbf{x}} = (0.00000152, 0.00000000),\ \phi_p(\hat{\mathbf{x}}) = 0.00001000.$

*Example 10*[8]

$g_1(\mathbf{x}) = \exp[x_1^2/1000 + (x_2 - 1)^2],$
$g_2(\mathbf{x}) = \exp[x_1^2/1000 + (x_2 + 1)^2],$
$\mathbf{x}^* = (0, 0),\ \phi(\mathbf{x}^*) = e,\ \mathbf{x}^0 = (1.5, 0.05),$
$\hat{\mathbf{x}} = (0.00000002, 0.00000000),\ \phi_p(\hat{\mathbf{x}}) = 2.71829183.$

*Example 11*[8]

$g_1(\mathbf{x}) = g(\mathbf{x} + 2\mathbf{e}_2),\ g_2(\mathbf{x}) = g(\mathbf{x} - 2\mathbf{e}_2),$
$g(\mathbf{x}) = \exp[0.0001x_1^2 + x_2^2 + x_3^2 + 2x_4^2 + x_5^2 + \cdots + x_{10}^2],$
$\mathbf{e}_2 = (0, 1, 0, 0, 0, 0, 0, 0, 0, 0),$
$\mathbf{x}^* = (0, \ldots, 0),\ \phi(\mathbf{x}^*) = 54.591846,\ \mathbf{x}^0 = (100, 0.1, \ldots, 0.1),$
$\hat{\mathbf{x}} = (0.00000000, -0.00000000, -0.00000000, -0.00000000,$
$0.00000000, 0.00000000, 0.00000000, -0.00000000,$
$-0.00000000, 0.00000000),\ \phi_p(\hat{\mathbf{x}}) = 54.59816003.$

*Example 12*[8]

$$g_i(\mathbf{x}) = \sum_{j=1}^{11} \frac{1}{j - 1 + i}\exp[x_i - \sin(i - 1 + 2(j - 1))]^2,\ (1 \le i \le 10),$$

$\mathbf{x}^* = (0.012427, 0.290377, -0.33466, -0.126514, 0.233137,$
$-0.276568, -0.166612, 0.229147, -0.185807, -0.170438,$
$0.240165),\ \phi(\mathbf{x}^*) = 3.703483,\ \mathbf{x}^0 = (1, 1, \ldots, 1),$
$\hat{\mathbf{x}} = (0.01244922, 0.29069248, -0.33460669, -0.12644597,$
$0.23281000, -0.27653988, -0.16658122, 0.22909022, -0.18291420,$

$- 0.17044129, 0.24017750)$, $\phi_p(\hat{\mathbf{x}}) = 3.70348527$.
*Example 13*[8]
$t = x_1 - (x_4 + 1)^4$,
$g_1(\mathbf{x}) = t^2 + (x_2 - t^4)^2 + 2x_3^2 - 5t - 5(x_2 - t^4) - 21x_3 + 7x_4$,
$g_2(\mathbf{x}) = g_1(\mathbf{x}) + 10\big[t^2 + (x_2 - t^4)^2 + x_3^2 + x_4^2 + t - (x_2 - t^4) + x_3 + x_4 - 8\big]$,
$g_3(\mathbf{x}) = g_1(\mathbf{x}) + 10\big[t^2 + 2(x_2 - t^4)^2 + x_3^2 + 2x_4^2 - t - x_4 - 10\big]$,
$g_4(\mathbf{x}) = g_1(\mathbf{x}) + 10\big[t^2 + (x_2 - t^4)^2 + x_3^2 + 2t - (x_2 - t^4) - x_4 - 5\big]$,
$\mathbf{x}^* = (0, 1, 2, -1)$, $\phi(\mathbf{x}^*) = -44$, $\mathbf{x}^0 = (0, 0, 0, 0)$,
$\hat{\mathbf{x}} = (-0.00000004, 0.99999996, 1.99999986, -0.99999980)$,
$\phi_p(\hat{\mathbf{x}}) = -43.9999942$.

# Conclusion

In this paper we deals with a long-last research topic, the min-max problem. Due to its wide applications, many algorithms have been proposed. From previous sections, however, it is clear that the present algorithm is the simplest one compared with others and has other features favorable in numerical computation. Contributions of this work may be summarized in several aspects:

1. The derivation of exponential penalty function $\Psi_p(\mathbf{x})$ and the exponential multiplier penalty function $\Psi_p(\mathbf{x}, \boldsymbol{\mu})$ through our Lagrangian perturbation approach helps with establishing a formal link between two different kinds of important functions, Lagrangian and penalty functions. This does not only give a new insight into the penalty function methods, but also suggests a way to construct useful penalty functions.

2. The derivation of smooth functions $\phi_p(\mathbf{x})$ and $\phi_p(\mathbf{x}, \boldsymbol{\mu})$ for the finite min-max problem (P) through directly applying ordinary exponential penalty functions to its equivalent problem (NLP) is very simple and clean.

3. The algorithm based upon the smooth function $\phi_p(\mathbf{x})$ makes it very simple to solve the non-differentiable min-max problem. It is superior to other algorithms, for instance nonlinear re-scaling.

4. The description for some applications of smooth functions may have effect on widening reader's fields of vision.

# References

[1] Polak E. On the mathematical foundations of nondifferentiable optimization // SIAM Rev. 1987. Vol. 29. P. 21–89.

[2] Polak E., Higgins J.E., Maynes D.Q. A barrier function method for minimax problems // Math. Program. 1992. Vol. 54. P. 155–176.

[3] Rockafellar R.T. Computational schemes for large-scale problems in extended linear-quadratic programming // Math. Program. 1990. Vol. 48. P. 447–474.

[4] Charalambous C., Conn A.R. An efficient method to solve the minimax problem directly // SIAM J. Numer. Anal. 1978. Vol. 15. P. 162–187.

[5] Fletcher R. A model algorithm for composite nondifferentiable optimization problem // Math. Programming Study. 1982. Vol. 17. P. 67–76.

[6] Han S. Variable metric methods for minimizing a class of non-differentiable functions // Math. Program. 1981. Vol. 20. P. 1–13.

[7] Murry W., Overton M.L. A projected Lagrangian algorithm for nonlinear minimax optimization // SIAM J. Sci. Statist. Comput. 1980. Vol. 1. P. 345–370.

[8] Polak E., Maynes D.Q., Higgins J.E. Superlinear convergent algorithm for minmax problem // J. Optim. Theory Appl. 1991. Vol. 69. P. 407–439.

[9] Ben T.A., Teboulle M. A smoothing technique for non-differentiable optimization problems // Lecture Notes in Math. 1405. B.: Springer-Verl. 1989. P. 1–11.

[10] Bertsekas D.P. Approximation procedures based on the method of multipliers // J. Optim. Theory Appl. 1977. Vol. 23. P. 487–510.

[11] Li X. An entropy-based aggregate method for minimax optimization // Eng. Optim. 1992. Vol. 18. P. 277–285.

[12] Rockafellar R.T. Convex Analysis. Princeton: Princeton Univ. Press, 1970.

[13] Comimetti R., Dussault J.P. Stable exponential-penalty algorithm with superlinear convergence // J. Optim. Theory Appl. 1994. Vol. 2. P. 285–309.

[14] Comimetti R., Martin J.S. Asymptotic analysis of the exponential penalty trajectory in linear programming // Math. Program. 1994. Vol. 67. P. 169–187.

[15] Li X. A smooth quasi-exact penalty function for nonlinear programming // Chin. Sci. Bull. 1991. Vol. 37. P. 806–809.

[16] Li X. An aggregate function method for nonlinear programming // Sci. China Ser. A. 1991. Vol. 34. P. 1467–1473.

[17] Peng J., Lin Z. A non-interior-point continuation method for generalized linear complementarity problems // Math. Program. 1999. Vol. 86. P. 533–563.

[18] Polyak R., Griva I., Sobieszczanski-Sobieski J. Nonlinear rescaling in discrete minimax. Manuscript, Department of SEOR and Math. Sci. Jorge Mason Univ., 2001.

[19] Pillo G.D., Grippo L. A smooth method for the finite minimax problem // Math. Program. 1993. Vol. 60. P. 187–214.

*Received for publication November 4, 2002*