http://communications.science.ankara.edu.tr

Commun.Fac.Sci.Univ.Ank.Ser. A1 Math. Stat.
Volume 72, Number 3, Pages 761–777 (2023)
DOI:10.31801/cfsuasmas.1150659
ISSN 1303-5991 E-ISSN 2618-6470

Research Article; Received:July 29, 2022; Accepted: April 26, 2023

AN EXACT PENALTY FUNCTION APPROACH FOR INEQUALITY CONSTRAINED OPTIMIZATION PROBLEMS BASED ON A NEW SMOOTHING TECHNIQUE

Nurullah YILMAZ[1] and Hatice OGUT[2]

[1,2]Department of Mathematics, Suleyman Demirel University, Isparta, TÜRKİYE

Abstract. Exact penalty methods are one of the effective tools to solve nonlinear programming problems with inequality constraints. In this study, a new class of exact penalty functions is defined and a new family of smoothing techniques to exact penalty functions is introduced. Error estimations are presented among the original, non-smooth exact penalty and smoothed exact penalty problems. It is proved that an optimal solution of smoothed penalty problem is an optimal solution of original problem. A smoothing penalty algorithm based on the the new smoothing technique is proposed and the convergence of the algorithm is discussed. Finally, the efficiency of the algorithm on some numerical examples is illustrated.

1. Introduction

We consider the following continuous constrained optimization problem

$$(P) \qquad \begin{aligned} &\min_{x\in\mathbb{R}^n} f(x) \\ &s.t.\ \ c_j(x) \le 0, \quad j = 1, 2, \dots, m, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c_j(x) : \mathbb{R}^n \to \mathbb{R}$, $j \in J = \{1, 2, ..., m\}$ are continuously differentiable functions. The set of feasible solutions is defined by $C_0 := \{x \in \mathbb{R}^n : c_j(x) \le 0,\ j = 1, 2, \dots, m\}$ and we assume that C_0 is not empty.

One of the most important methods in solving this problem is the penalty function approach. The penalty function approach is based on transforming the constrained optimization problem into an unconstrained problem. When the penalty function approach is applied to problem (P), it turns into the following unconstrained optimization problem:

2020 *Mathematics Subject Classification.* Primary 90C30, 65K05; Secondary 65D15.
Keywords. Constrained optimization, smoothing technique, exact penalty function.
[1] nurullahyilmaz@sdu.edu.tr-Corresponding Author; 0000-0001-6429-7518;
[2] haticeeogutt@gmail.com; 0000-0001-8640-0619.

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

where $F(x,\rho) = f(x) + \rho\sum_j G(c_j(x))$ and $\rho > 0$ parameter. The most common G functions are $G(t) = \max\{0,t\}^2$, $G(t) = \max\{0,t\}$, $G(t) = \max\{0,t\}^p$ $(0 < p \leq 1)$, $G(t) = \log(1 + \max\{0,t\})$ etc [4, 24]. Moreover, as the parameter ρ increases, the solution of the problem (1) gets closer to the solution of the problem (P). One of the desirable properties of penalty functions is precision. $F(x,\rho)$ is called as exact penalty function for problem (P) if there is appropriate parameter choice such that the optimal solution to the penalty problem is an optimal solution to the original problem [17, 26, 27]. We refer the following studies for more details [25, 28].

One of the well-known penalty function is called as l_2-penalty function and it is defined as

$$F_2(x,\rho) = f(x) + \rho\sum_j \max\{c_j(x), 0\}^2.$$

When f and c_j $(j = 1, 2, \ldots, m)$ are continuously differentiable, the l_2 penalty function is smooth, but it is not exact [17]. One of the most popular exact penalty function is called as l_1 penalty function which is defined as

$$F_1(x,\rho) = f(x) + \rho\sum_j \max\{c_j(x), 0\},$$

by Eremin [1] and Zangwill [2]. l_1 penalty function is exact but not differentiable. This is the main disadvantage of the l_1 exact penalty function, because it prevents some efficient algorithms (Steepest Descent, Newton, Quasi-Newton, etc.) from being used to solve the penalty problem. On the other hand, in order to increase the effectiveness of the exact penalty function, lower-order exact penalty functions have come to the fore in the literature [3, 4]. The lower order l_p-exact penalty function is defined as

$$F_p(x,\rho) = f(x) + \rho\sum_j \max\{c_j(x), 0\}^p,$$

where $0 < p < 1$ in [5, 6]. Similar to l_1, l_p penalty function is also exact but not differentiable and l_p penalty function is non-Lipschitz when $0 < p < 1$. Moreover, non-smooth penalty function can cause numerical instability in the solution process when the penalty parameter is large. For this reason the smoothing approaches for the penalty function have been emerged [7]. The smoothing approach can be expressed as the representation of the non-differentiable function with a family of smooth functions. A smoothing function is defined as follows:

Definition 1. *[8] A function $\tilde{f} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is called a smoothing function of a non-smooth function $f : \mathbb{R}^n \to \mathbb{R}^m$ if, for any $\varepsilon > 0$, $\tilde{f}(x,\varepsilon)$ is continuously differentiable and*

$$\lim_{z\to x,\varepsilon\downarrow 0} \tilde{f}(z,\varepsilon) = f(x)$$

for any $x \in \mathbb{R}^n$.

$\mathbb{R}_+$ represents the non-negative real numbers. Smoothing functions are often used to solve non-smooth optimization problems [9–12]. In addition, there is quite a lot of work in the literature on smoothed penalty functions l_1 and l_p [13–20]. A comprehensive review is presented in [23].

As it is well-known that gradient based methods (e. g. Newtonian methods) which are powerful tools in nonlinear programming usually needs second-order continuously differentiability of an objective function. Therefore, it is essential to develop smoothing techniques which makes l_1 and l_p exact penalty functions the second order continuously differentiable. Although there are different smoothing studies for l_1, l_p and other penalty functions in the literature, there is no smoothing approach that includes all of them.

The aim of this study is to re-define the class of exact penalty functions for problem (P) and propose a new second-order continuously differentiable smoothing technique for a new exact penalty functions in general form. By applying the proposed smoothing technique to exact penalty functions, a smoothed penalty function and a smoothed penalty problem are obtained. The relationships among the solutions which are obtained for original constrained optimization problem, exact penalty problem and the smoothed penalty problem is investigated. Based on the smoothed penalty problem, it is aimed to create an algorithm to solve the problem (P). Numerical experiments are presented by applying this algorithm to test problems.

2. Main Results

2.1. **A New Exact Penalty Function.**

In this part of the study, we first re-define a class of exact penalty functions as follows:

$$h(t) = \begin{cases} 0, & t < 0, \\ g(t), & t \geq 0, \end{cases}$$

where $g : \mathbb{R}_+ \to \mathbb{R}_+$ second-order continuously differentiable function with (a) $g(0) = 0$ and (b) $g'(t) > 0$ and $g''(t) \leq 0$ for $t > 0$. Based on the above definition, the exact penalty function for problem (P) is defined by

$$F_g(x, \rho) = f(x) + \rho \sum_j h(c_j(x))$$

and the penalty problem is given by

$$(P_g) \quad \min_{x \in \mathbb{R}^n} F_g(x, \rho).$$

Moreover we have the following properties based on the function $g(t)$:

(i) if $g(t) = t$ then $F_g(x, \rho)$ become l_1-exact penalty function ([2]),

(ii) if $g(t) = t^p$ for $0 < p < 1$ is then $F_g(x, \rho)$ become l_p-lower order exact penalty function ([4, 5]),

(iii) if $g(t) = \log(1+t)$ is then $F_g(x,\rho)$ become logarithmic exact penalty function is obtained ([24]).

We need the following assumptions to state the exactness of our penalty function.
Assumption 1. $f(x)$ is a coercive function, i.e., $\lim_{||x||\to\infty} f(x) = \infty$.

Assumption 1 implies that there exist a compact set $Y \subset \mathbb{R}^n$ such that all local minimizer of problem (P) are included in int$Y$.
Assumption 2. The number of local minimizer of the problem (P) is finite.

Theorem 1. *Suppose that Assumptions 1 and 2 hold. Then, there exist a threshold value $\bar{\rho}$ such that $\rho \in [\bar{\rho}, \infty)$, every solution of (P_g) is a solution of (P).*

Proof. The proof is obtained by following the way at the proof of the Corollary 2.3 in [5]. □

2.2. **Smoothing Techniques.** As it is known that, the differentiability of the penalty functions established with the functions given by (i), (ii) and (iii) cannot always be guaranteed. Especially, when $g(t) = 0$, the function h is non differentiable. Therefore, we offer the following smoothing functions for the function h inspiring from the studies [21, 22].

The smoothing function of h is defined as

$$h_{1,\gamma}(t) = \begin{cases} 0, & t < 0, \\ \frac{\gamma g'(\gamma)-2g(\gamma)}{\gamma^3} t^3 - \frac{\gamma g'(\gamma)-3g(\gamma)}{\gamma^2} t^2, & 0 \le t \le \gamma, \\ g(t), & t > \gamma, \end{cases} \tag{2}$$

where $\gamma > 0$ is the smoothing parameter.

Lemma 1. *For any $t \in \mathbb{R}$, the smoothing function $h_{1,\gamma}(t)$ satisfies that*

i. *$h_{1,\gamma}(t)$ is continuously differentiable,*
ii. *$\lim_{\gamma\to 0} h_{1,\gamma}(t) = h(t)$.*

Proof. i. For any $\gamma > 0$, we have

$$h'_{1,\gamma}(t) = \begin{cases} 0, & t < 0, \\ 3\frac{\gamma g'(\gamma)-2g(\gamma)}{\gamma^3} t^2 - 2\frac{\gamma g'(\gamma)-3g(\gamma)}{\gamma^2} t, & 0 \le t \le \gamma, \\ g'(t), & t > \gamma, \end{cases}$$

and it is easy to see that the function $h'_{1,\gamma}(t)$ is continuous at the transition points $t = 0$ and $t = \gamma$.

ii. The difference between $h(t)$ and $h_{1,\gamma}(t)$ is stated by

$$h(t) - h_{1,\gamma}(t) = \begin{cases} 0, & t < 0, \\ g(t) - \left[\frac{\gamma g'(\gamma)-2g(\gamma)}{\gamma^3} t^3 - \frac{\gamma g'(\gamma)-3g(\gamma)}{\gamma^2} t^2\right], & 0 \le t \le \gamma, \\ 0, & t > \gamma, \end{cases}$$

for any $\gamma > 0$. Therefore the maximum difference between $h(t)$ and $h_{1,\gamma}(t)$ arises when $0 \leq t \leq \gamma$. Let us define the following

$$l_{1,\gamma}(t) = \frac{\gamma g'(\gamma) - 2g(\gamma)}{\gamma^3} t^3 - \frac{\gamma g'(\gamma) - 3g(\gamma)}{\gamma^2} t^2,$$

then for $0 \leq t \leq \gamma$ we have

$$\begin{aligned} l'_{1,\gamma}(t) &= \frac{1}{\gamma^3}\left[\gamma g'(\gamma)\left(3t^2 - 2\gamma t\right) + g(\gamma)\left(6\gamma t - 6t^2\right)\right] \\ &\geq \frac{g(\gamma)}{\gamma^3}\left[4\gamma t - 3t^2\right] \\ &\geq 0. \end{aligned}$$

Since $l_{1,\gamma}(t) \geq 0$ and it is non-decreasing we obtain

$$h(t) - h_{1,\gamma}(t) = g(t) - l_{1,\gamma}(t) \leq g(\gamma). \tag{3}$$

By taking the limit as $\gamma \to 0$, the proof is obtained.

□

For different exact penalty function, the error estimation between $h_{1,\gamma}(t)$ and $h(t)$ can be calculated. For example, if we take $g(t) = t$, then by considering (3) we obtain

$$0 \leq h(t) - h_{1,\gamma}(t) \leq \gamma.$$

With a similar approach, a second order differentiable smoothing function of $h(t)$ can be generated as:

$$h_{2,\gamma}(t) = \begin{cases} 0, & t < 0, \\ l_{2,\gamma}(t), & 0 \leq t \leq \gamma, \\ g(t), & t > \gamma, \end{cases} \tag{4}$$

form is obtained. Here

$$\begin{aligned} l_{2,\gamma}(t) &= \frac{\gamma^2 g''(\gamma) - 6\gamma g'(\gamma) + 12g(\gamma)}{2\gamma^5} t^5 - \frac{\gamma^2 g''(\gamma) - 7\gamma g'(\gamma) + 15g(\gamma)}{\gamma^4} t^4 \\ &\quad + \frac{\gamma^2 g''(\gamma) - 8\gamma g'(\gamma) + 20g(\gamma)}{2\gamma^3} t^3, \end{aligned}$$

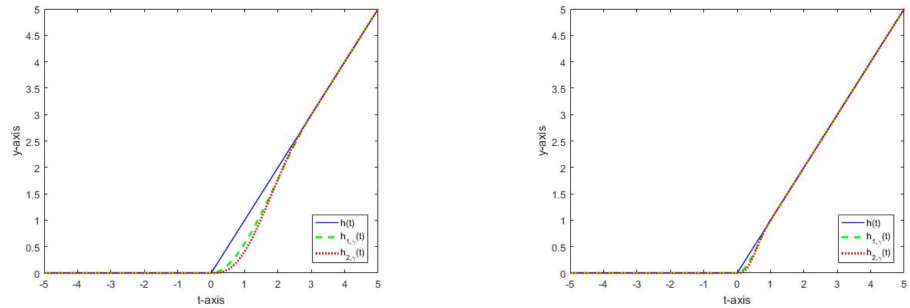
for $\gamma > 0$.

Lemma 2. *For any $t \in \mathbb{R}$, the smoothing function $h_{2,\gamma}(t)$ satisfies that*

i. *$h_{2,\gamma}(t)$ is second-order continuously differentiable,*
ii. *$\lim_{\gamma\to 0} h_{2,\gamma}(t) = h(t)$.*

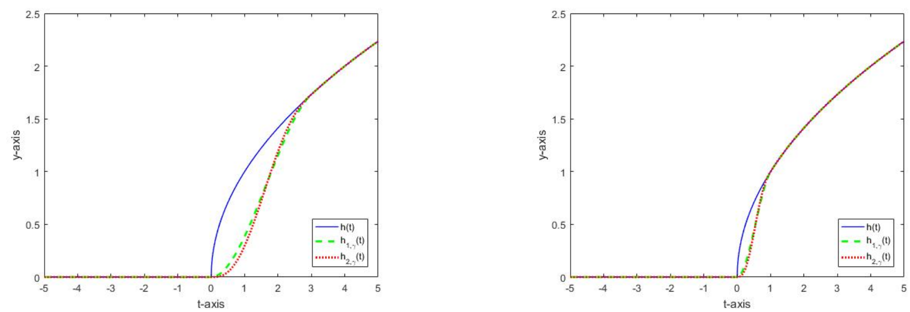
Proof. The proof is obtained similarly to the proof of Lemma 1. □

Example 1. *Let us consider the function $y = h(t)$. The graph of $h(t)$, $h_{1,\gamma}$ and $h_{2,\gamma}$ are illustrated in Figs. 1, 2 and 3, when $g(t) = t$, $g(t) = t^p$ with $p = \frac{1}{2}$ and $g(t) = \log(1 + t)$, respectively. It is observed that the smoothing functions approach the original function when $\gamma \to 0$.*



(A) For $\gamma = 3$. (B) For $\gamma = 1$.

FIGURE 1. The blue graph represents $h(t)$ for $g(t) = t$, the green graph is $h_{1,\gamma}(t)$ and the red graph is $h_{2,\gamma}(t)$.



(A) For $\gamma = 3$. (B) For $\gamma = 1$.

FIGURE 2. The blue graph represents $h(t)$ for $g(t) = t^p$, $p = 0.5$, the green graph is $h_{1,\gamma}(t)$ and the red graph is $h_{2,\gamma}(t)$.

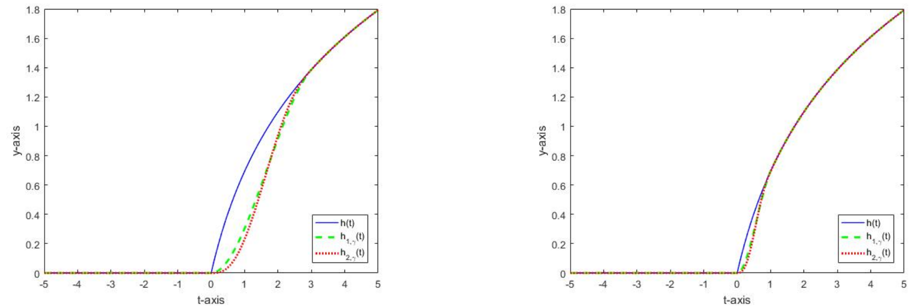
Remark 1. *It should be pointed out that the applied smoothing functions are non-convex.*

By using one of the smoothing functions given in (2) and (4), the smoothing exact penalty function is obtained as

$$\tilde{F}_g(x, \rho, \gamma) = f(x) + \rho \sum_{j \in J} h_{i,\gamma}\left(c_j(x)\right),$$

$i = 1, 2$. Therefore the smoothed penalty function problem is stated as

$$(PS_g) \quad \min_{x \in \mathbb{R}^n} \tilde{F}_g(x, \rho, \gamma).$$



(A) For $\gamma = 3$. (B) For $\gamma = 1$.

FIGURE 3. The blue graph represents $h(t)$ for $\log(1+t)$, the green graph is $h_{1,\gamma}(t)$ and the red graph is $h_{2,\gamma}(t)$.

Now let us investigate the relationship between the exact penalty problem and the smoothed exact penalty problem.

Theorem 2. *For any $x \in \mathbb{R}^n$, we have*

$$0 \leq F_g(x,\rho) - \tilde{F}_g(x,\rho,\gamma) \leq \rho m g(\gamma)$$

and

$$\lim_{\gamma \to 0} \tilde{F}_g(x,\rho,\gamma) = F_g(x,\rho),$$

for $\gamma > 0$.

Proof. For any $\rho, \gamma > 0$, we have

$$\begin{aligned} F_g(x,\rho) - \tilde{F}_g(x,\rho,\gamma) &= f(x) + \rho \sum_{j \in J} h(c_j(x)) - \left[f(x) + \rho \sum_{j \in J} h_{i,\gamma}(c_j(x)) \right] \\ &= \rho \sum_{j \in J} \left[h(c_j(x)) - h_{i,\gamma}(c_j(x)) \right], \end{aligned}$$

for $i = 1, 2$. Therefore, we obtain

$$\begin{aligned} F_g(x,\rho) - \tilde{F}_g(x,\rho,\gamma) &\leq \rho \sum_{j \in J} g(\gamma) \\ &\leq \rho m g(\gamma). \end{aligned}$$

□

It is easy to see that we have the following error estimates:

$$F_1(x,\rho) - \tilde{F}_1(x,\rho,\gamma) \leq \rho m \gamma,$$

for $g(t) = t$,

$$F_p(x,\rho) - \tilde{F}_p(x,\rho,\gamma) \le \rho m \gamma^p,$$

for $g(t) = t^p$, $0 < p < 1$ and

$$F_{\log}(x,\rho) - \tilde{F}_{\log}(x,\rho,\gamma) \le \rho m \log(1+\gamma),$$

for $g(t) = \log(1+t)$.

The following corollary indicates that the distance between $F_g(x,\rho)$ and $\tilde{F}_g(x,\rho,\gamma)$ decreases when the smoothing parameter decreases.

Corollary 1. *Let $\{\gamma_k\} \to 0$ and $\{x^k\}$ is an optimal solution of the problem $\min_{x\in\mathbb{R}^n} \tilde{F}_g(x,\rho_k,\gamma_k)$. If $\bar{x}$ is limit point of $\{x^k\}$, then $\bar{x}$ is the optimal solution to the problem (P_g).*

Definition 2. *[17] Let f^* be the optimal objective function value of the problem (P) and x be a feasible solution. If the condition*

$$f(x) - f^* \le \gamma$$

holds, then x is called $\gamma-$approximate solution.

Definition 3. *[17] If $c_j(x_\gamma) \le \gamma$ for any $j \in J$ and for $\gamma > 0$, then the x_γ is called as $\gamma-$feasible solution of the problem (P).*

Lemma 3. *[17, 24] Let x^* be the optimal solution to the problem (P_g). If x^* is a feasible solution to the problem (P), then x^* is the optimal solution for (P).*

Thus, we can give the following theorem on the relations of optimal solutions of the problems (P), (P_g) and (PS_g) .

Theorem 3. *Let $\rho > 0$, x^* be an optimal solution to the problem (P_g) and x_γ be and optimal solution to the problem (PS_g). Then the following holds:*

$$\lim_{\gamma\to 0} \tilde{F}_g(x_\gamma,\rho,\gamma) = F_g(x^*,\rho). \tag{5}$$

Moreover, if x^* is the optimal solution to the problem (P) and x_γ is the γ-feasible solution for the problem (P) , then x_γ is the approximate solution to the problem (P).

Proof. Let x^* be an optimal solution of (P_g) and x_γ be an optimal solution of (PS_g). By considering Theorem 2 and following inequalities

$$\begin{aligned} F_g(x^*,\rho) &\le F_g(x_\gamma,\rho), \\ \tilde{F}_g(x_\gamma,\rho,\gamma) &\le \tilde{F}_g(x^*,\rho,\gamma), \end{aligned} \tag{6}$$

we obtain

$$\begin{aligned} 0 &\le F_g(x^*,\rho) - \tilde{F}_g(x^*,\rho,\gamma) \\ &\le F_g(x^*,\rho) - \tilde{F}_g(x_\gamma,\rho,\gamma) \\ &\le F_g(x_\gamma,\rho) - \tilde{F}_g(x_\gamma,\rho,\gamma) \end{aligned}$$

$$\leq \quad m\rho g(\gamma).$$

Therefore, (5) is hold. Let x^* be an optimal solution of (P) and x_γ be $\gamma-$feasible solution (P). Since we have

$$0 \leq \left[f(x^*) + \rho \sum_j h(c_j(x^*))\right] - \left[f(x_\gamma) + \rho \sum_j h_{i,\gamma}(c_j(x_\gamma))\right] \leq m\rho g(\gamma),$$

$c_j(x^*) \leq 0$ and $c_j(x_\gamma) \leq \gamma$, then we have

$$\rho \sum_j h(c_j(x^*)) = 0, \qquad 0 \leq \rho \sum_j h_{i,\gamma}(c_j(x_\gamma)) \leq m\rho\gamma$$

and we obtain

$$|f(x_\gamma) - f(x^*)| < m\rho\left(\gamma + g(\gamma)\right).$$

□

2.3. **Algorithm.** In this section, the following algorithm is proposed to solve the penalty problem (P) by considering the surrogate problem (PS_g).

Algorithm A

Step 1 Select initial point x^0, and parameters $\gamma_0 > 0$, $\rho_0 > 0$. Determine the auxiliary parameters $\varepsilon > 0$, $N > 1$, $0 < \delta < 1$. Let $k = 0$ and go to Step 2.

Step 2 By using x^k as an initial point, solve the problem $\min_{x\in R^n} \tilde{F}_g(x, \rho_k, \gamma_k)$ with any local search methods. Let x^{k+1} be an optimal solution.

Step 3 If x^{k+1} is the ε-feasible solution to the problem (P), then stop. Otherwise, take $\rho_{k+1} = N\rho_k$, $\gamma_{k+1} = \delta\gamma_k$ and $k = k + 1$, and go back to Step 2 .

Remark 2. *In Step 2 of Algorithm A, any gradient based local search method (e.g. Steepest Descent, Newton, Quasi-Newton and etc.) can be used according to degree of smoothing approximation.*

Remark 3. *From the 3rd step of Algorithm A and Theorem 2, an approximate optimal solution of the problem P can be obtained.*

We denote the following index sets

$$J_\gamma^-(x) = \{j|c_j(x) < \gamma, j \in J\}, \qquad J_\gamma^+(x) = \{j|c_j(x) \geq \gamma, j \in J\}.$$

With these notations; the following theorem is given related to the convergence of Algorithm A.

Theorem 4. *Let the Assumption 1 is hold. Then the sequence $\{x^k\}$ generated by Algorithm A is bounded and the limit point $\bar{x}$ is the optimal solution to the problem (P).*

Proof. Let us first prove that $\{x^k\}$ is bounded. Since the sequence $\{F(x^k, \rho_k, \gamma_k)\}$ is a bounded sequence, then there exist a number L such that

$$\tilde{F}_g(x^k, \rho_k, \gamma_k) \leq L, \quad k = 0, 1, 2, \dots. \tag{7}$$

Assume to contrary that $\{x^k\}$ is unbounded. Without loss of generality, let $k \to \infty$, $\|x^k\| \to \infty$. The equation (7) is re-stated as

$$L \geq \tilde{F}_g(x^k, \rho_k, \gamma_k) \geq f(x^k), \quad k = 0, 1, 2, \dots$$

and it is a contradiction with the Assumption 1. The boundedness of $\{x^k\}$ is obtained.

Let us now show that the limit point $\bar{x}$ of $\{x^k\}$ is the optimal solution to the problem (P). Let us first show that the point $\bar{x}$ is a feasible solution to the problem (P). Let $\lim_{k\to\infty} x^k = \bar{x}$. On the contrary, suppose the point $\bar{x}$ is not a feasible solution to (P). Then there exists $j \in J$ for $c_j(\bar{x}) \geq \alpha > 0$ such that

$$\begin{aligned} \tilde{F}_g(x^k, \rho_k, \gamma_k) &= f(x^k) + \rho_k \sum_{j\in J} h_{i,\gamma_k}(c_j(x^k)) \\ &= f(x^k) + \rho_k \sum_{j\in J^+_{\gamma_k}(x^k)} h_{i,\gamma_k}(c_j(x^k)) \\ &\quad +\rho_k \sum_{j\in J^-_{\gamma_k}(x^k)} h_{i,\gamma_k}(c_j(x^k)). \end{aligned} \tag{8}$$

where $c_j(\bar{x}) \geq \alpha > 0$, the set $\{j : c_j(\bar{x}) \geq \alpha\}$ is non-empty. There is $j_0 \in J$ with $c_{j_0}(\bar{x}) \geq \alpha$. Since $\rho_k \to \infty$ as $k \to \infty$, from the equation (8) we obtain

$$\tilde{F}_g(x^k, \rho_k, \gamma_k) \to \infty.$$

This contradicts the boundedness of the sequence $\{\tilde{F}_g(x^k, \rho_k, \gamma_k)\}$. Thus $\bar{x}$ would be a feasible solution to the (P) problem.

Let us show that the $\bar{x}$ is an optimal solution for (P). Assume x^* is an optimal solution for (PS_g) and x^k is an optimal solution for the problem $\min_{x\in R^n} \tilde{F}_g(x^k, \rho_k, \gamma_k)$ then we have

$$\tilde{F}_g(x^k, \rho_k, \gamma_k) \leq \tilde{F}_g(x^*, \rho_k, \gamma_k), \quad k = 1, 2, \dots.$$

Similarly, we have

$$f(x^k) + \rho_k \sum_{j\in J} h_{i,\gamma_k}(c_j(x^k)) \leq f(x^*) + \rho_k \sum_{j\in J} h_{i,\gamma_k}(c_j(x^*)), \quad k = 1, 2, \dots$$

and

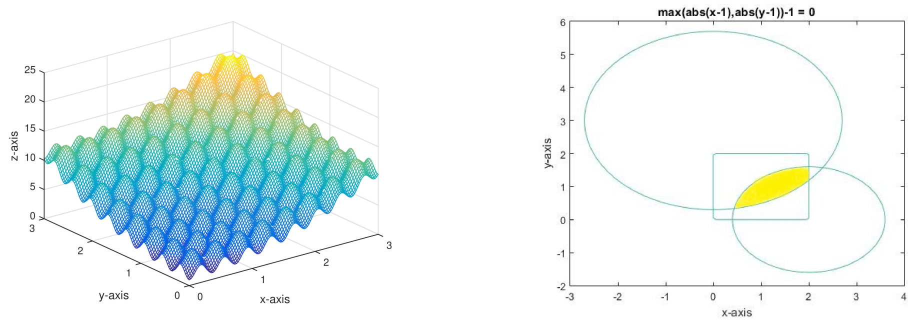
$$f(x^k) \leq f(x^*).$$

So while $k \to \infty$,

$$f(\bar{x}) \leq f(x^*). \tag{9}$$

Since x^* is the optimal solution for (P), we have

$$f(\bar{x}) \geq f(x^*). \tag{10}$$



(A) (B)

FIGURE 4. (A) The graph of f (B)The graph of feasible region.

From (9) and (10), we obtain $f(\bar{x}) = f(x^*)$. It means that $\bar{x}$ is the optimal solution for (P). □

3. Numerical Results

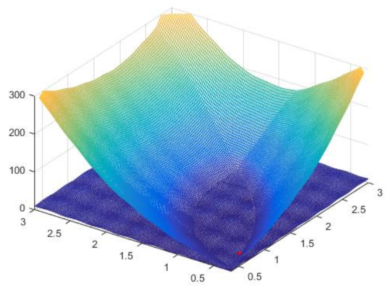
In order to analyze the numerical performance of Algorithm A, we apply it on some test problems in the literature. The results are presented in the tables with details and the evaluations on these results are given. Firstly, the abbreviations used in the tables are listed below.

k	:	Number of iterations
x^k	:	the result of $k-$th iteration
ρ_k	:	penalty function parameter in the $k-$th iteration
γ_k	:	smoothing parameter of the $k-$th iteration
$c_j(x^k)$	:	constraint function value at x^k
$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	:	value of function $\tilde{F}_g$ at point x^k.
$f(x^k)$	:	The value of the objective function at x^k

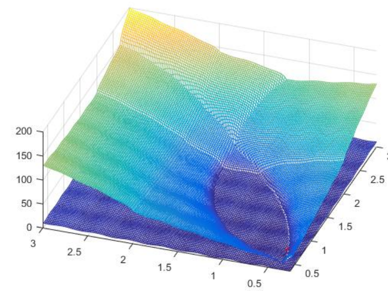
Problem 1. *[14] Consider the following problem*

$$\begin{aligned} \min f(x) \quad &= \quad x_1^2 + x_2^2 - \cos(17x_1) - \cos(17x_2) + 3 \\ s.t. \quad & \quad g_1(x) = (x_1 - 2)^2 + x_2^2 - 1.6^2 \leq 0, \\ & \quad g_2(x) = x_1^2 + (x_2 - 3)^2 - 2.7^2 \leq 0, \\ & \quad 0 \leq x_1 \leq 2, \quad 0 \leq x_2 \leq 2. \end{aligned}$$

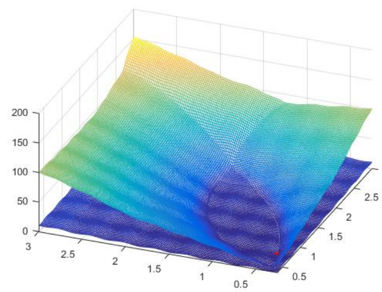
We select $x^0 = (1, 1)$ as starting point $\rho_0 = 10$, $\gamma_0 = 0.1$, $\eta_0 = 0.1$ and $N = 3$. The obtained numerical results are illustrated in Table 1 and 2.



(A) For $g(t) = t$.



(B) For $g(t) = t^p$, $p = \frac{1}{2}$.



(C) For $g(t) = \log(1 + t)$

FIGURE 5. The graph of $\tilde{F}_g(x, \rho, \gamma)$ with $\rho = 10$, $\gamma = 0.25$.

TABLE 1. Numerical results for the Problem 1

Penalty Function	k	x^{k+1}	ρ_k	γ_k	$(c_1(x^k), c_2(x^k))$	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	0	$(0.7256, 0.3985)$	10	0.1	$(-0.7770, 0.0044)$	1.8338	1.8301
	1	$(0.7254, 0.3992)$	30	0.01	$(-0.7759, 0.0001)$	1.8374	1.8373
	2	$(0.7254, 0.3993)$	90	0.001	$(-0.7759, 0.0000)$	1.8376	1.8376
$g(t) = t^p$	0	$(0.72540.3991)$	10	0.1	$(-0.7762, 0.0011)$	1.8366	1.8356
	1	$(0.7254, 0.3993)$	30	0.01	$(-0.7759, 0.0000)$	1.8376	1.8376
$g(t) = log(1 + t)$	0	$(0.72560.3985)$	10	0.1	$(-0.77700.0045)$	1.8337	1.8299
	1	$(0.7254, 0.3992)$	30	0.01	$(-0.7759, 0.0001)$	1.8374	1.8373
	2	$(0.7254, 0.3993)$	90	0.001	$(-0.7759, 0.0000)$	1.8376	1.8376

For different penalty types, the global minimizer is found as $x^* = (0.7254, 0.3993)$ with corresponding function value 1.8376. In [14,17], the resulting global minimizer is found as $x^* = (0.72540669, 0.3992805)$ and the corresponding function value 1.837623, and combining all three approaches our algorithm found the right point as in [14,17].

TABLE 2. Numerical results for the Problem 1

Penalty Function	iter	feval	Time	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	3	180	1.1094	1.8376	1.8376
$g(t) = t^p$	2	123	0.8125	1.8376	1.8376
$g(t) = \log(1+t)$	3	177	1.1875	1.8376	1.8376

Problem 2. *[14] Consider the following problem which is called as Rosen-Suzuki problem:*

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + 2x^3 + x_4^2 - 5x_1 - 21x_3 + 7x_4 \\ s.t. \quad & g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \leq 0, \\ & g_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0, \\ & g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0. \end{aligned}$$

We select the starting point as $x^0 = (0, 0, 0, 0)$, $\rho_0 = 10$, $\gamma_0 = 0.1$, $\eta_0 = 0.1$ and $N = 3$. The obtained numerical results are illustrated as in Table 3 and 4.

Applying Algorithm A, the minimizer is found as $x^* = (0.1697, 0.8358, 2.0084, -0.9651)$ with the corresponding function value -44.2338. In [14], the resulting global minimizer is found as $x^* = (0.1684621, 0.8539065, 2.000167, -0.9755604)$ with the corresponding function value -44.23040. In [17], the global minimizer is obtained as $x^* = (0.170189, 0.835628, 2.008242, -0.95245)$ with corresponding function value -44.2338. It can be observed that our algorithms provide numerically better results than [14] and find approximate solutions with lower iteration numbers compared to [17].

TABLE 3. Numerical results for Problem 2.

Penalty Function	k	x^{k+1}	ρ_k	γ_k	$(c_1(x^k), c_2(x^k), c_3(x^k))$	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	0	$(0.1697, 0.8355, 2.0092, -0.9656)$	10	0.1	$(0.0019, 0.0052, -1.8773)$	-44.2396	-44.2455
	1	$(0.1696, 0.8356, 2.0086, -0.9650)$	30	0.01	$(0.0001, 0.0002, -1.8826)$	-44.2340	-44.2342
	2	$(0.1696, 0.8356, 2.0086, -0.9650)$	90	0.001	$(-0.0001, 0.0000, -1.8827)$	-44.2338	-44.2338
$g(t) = t^p$	0	$(0.1696, 0.8356, 2.0088, -0.9651)$	10	0.1	$(0.0005, 0.0013, -1.8815)$	-44.2353	-44.2367
	1	$(0.1695, 0.8355, 2.0086, -0.9650)$	30	0.01	$(-0.0007, 0.0000, -1.8827)$	-44.2338	-44.2338
$g(t) = log(1+t)$	0	$(0.1697, 0.8355, 2.0092, -0.9656)$	10	0.1	$(0.0019, 0.0053, -1.8772)$	-44.2398	-44.2458
	1	$(0.1696, 0.8356, 2.0086, -0.9650)$	30	0.01	$(0.0001, 0.0002, -1.8826)$	-44.2340	-44.2342
	2	$(0.1696, 0.8356, 2.0086, -0.9650)$	90	0.001	$(-0.0001, 0.0000, -1.8827)$	-44.2338	-44.2338

Problem 3. *[17] Consider the following problem*

$$\begin{aligned} \min f(x) &= 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\ s.t. \quad & g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0, \\ & g_2(x) = (x_1 - 5)^2 + x_2^2 + x_3^2 - 25 = 0 \end{aligned}$$

TABLE 4. Numerical results for Problem 2.

Penalty Function	iter	feval	Time	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	3	510	1.1406	-44.2338	-44.2338
$g(t) = t^p$	2	475	0.79688	-44.2338	-44.2338
$g(t) = \log(1 + t)$	3	460	0.98438	-44.2338	-44.2338

$$g_3(x) = (x_1 - 5)^2 + (x_2 - 5)^2 + (x_3 - 5)^2 - 25 \leq 0.$$

We select $x^0 = (2, 2, 1)$ *as a starting point* $\rho_0 = 100$, $\gamma_0 = 0.1$, $\eta_0 = 0.1$ *and* $N = 3$. *The obtained numerical results are illustrated as in Table* 5 *and* 6.

By considering Algorithm A the global minimizer is found as $x^* = (2.5001, 4.1754, 1.1474)$ with corresponding function value 944.2157 by using $g(t) = t$, and $x^* = (2.5000, 4.2213, 0.9647)$ and corresponding value as 944.2157 by using $g(t) = t^p$ and $g(t) = \log(1+t)$. In [17], the obtained global minimizer is obtained as $x^* = (2.5000, 4.2213, 0.9647)$ with the corresponding function value 944.2157. According to these results, we deduce that by using Algorithm A the correct solutions is obtained with a lower number of iterations than [17].

TABLE 5. Numerical results for Problem 3.

Penalty Function	k	x^{k+1}	ρ_k	γ_k	$(c_1(x^k), c_2(x^k), c_3(x^k))$	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	0	$(2.5001, 4.1754, 1.1474)$	100	0.1	$(0.0012 - 0.0001 - 3.2283)$	944.3897	944.2571
	1	$(2.5000, 4.1753, 1.1474)$	300	0.01	$(0.0000, 0.0000, -3.2274)$	944.2652	944.2653
$g(t) = t^p$	0	$(2.5012, 4.2220, 0.9649)$	100	0.1	$(0.0123, 0.0007, -1.8682)$	945.4946	944.1889
	1	$(2.5000, 4.2213, 0.9647)$	300	0.01	$(0.0000, -0.0000, -1.8599)$	944.2156	944.2156
$g(t) = log(1 + t)$	0	$(2.5000, 4.2213, 0.9648)$	100	0.1	$(0.0004, 0.0000, -1.8607)$	944.3356	944.2148
	1	$(2.5000, 4.2213, 0.9648)$	300	0.01	$(0.0000, 0.0000, -1.8604)$	944.2156	944.2156

TABLE 6. Numerical results for Problem 3.

Penalty Function	iter	feval	Time	$\tilde{F}_g(x^k, \rho_k, \gamma_k)$	$f(x^k)$
$g(t) = t$	2	328	0.8125	944.2652	944.2653
$g(t) = t^p$	2	300	0.70313	944.2156	944.2156
$g(t) = \log(1 + t)$	2	300	0.71875	944.2156	944.2156

4. Conclusion

In this study, a new class of exact penalty function is given and smoothing penalty function is proposed for this new exact penalty function. A new minimization algorithm is developed in order to solve the problem (P) by the help of surrogate problem (PS_g). The algorithm is applied to the test problems and satisfactory results are obtained.

The proposed smoothing technique for the non-smooth exact penalty functions has a flexible structure. It is available for both Lipschitz and non-Lipschitz penalty functions. This is the most important feature of our smoothing technique and that distinguishes our smoothing technique from other techniques.

Algorithm A is in all cases highly effective for small and medium scale optimization problems. By applying this algorithm, the optimum value is found quickly and the algorithm offers high accuracy in finding the optimal point.

Author Contribution Statements The authors contributed equally to this work. All authors read and approved the final copy of this paper.

Declaration of Competing Interests The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements This paper was presented in 4th International Conference on Pure and Applied Mathematics (ICPAM - VAN 2022), Van-Turkey, June 22-23, 2022.

References

[1] Eremin, I. I., The penalty method in convex programming, *Cybernetics*, 3(4) (1967), 53-56. https://doi.org/10.1007/BF01071708

[2] Zangwill, W. I., Nonlinear programing via penalty functions, *Management Science*, 13 (1967), 344-358. http://www.jstor.org/stable/2627851.

[3] Rubinov, A. M., Glover, B. M., Yang, X. Q., Decreasing functions with applications to penalization, *SIAM J. Optim.*, 10 (1999), 289-313. https://doi.org/10.1137/S105262349732609

[4] Rubinov, A. M., Yang, X. Q., Bagirov, A. M., Penalty functions with a small penalty parameter, *Optim. Methods Softw.*, 17(5) (2002), 931-964. https://doi.org/10.1080/1055678021000066058

[5] Wu, Z. Y., Bai, F. S., Yang, X. Q., Zhang, L. S., An exact lower-order penalty function and its smoothing in nonlinear programming, *Optimization*, 53(1) (2004), 51-68. https://doi.org/10.1080/02331930410001662199

[6] Bai, F. S., Wu, Z. Y., Zhu, D. L., Lower order calmness and exact penalty function, *Optim. Methods Softw.*, 21(4) (2006), 515-525. https://doi.org/10.1080/10556780600627693

[7] Pinar, M. C., Zenios, S., On smoothing exact penalty functions for convex constrained optimization, *SIAM J. Optim.*, 4(3) (1994), 468-511. https://doi.org/10.1137/0804027

[8] Chen, C., Mangasarian, O. L., A class of smoothing functions for nonlinear and mixed complementarity problem, *Comput. Optim. Appl.*, 5 (1996), 97-138. https://doi.org/10.1007/BF00249052

[9] Bertsekas, D. P., Nondifferentiable optimization via approximation. *In: Balinski, M. L., Wolfe, P. (eds) Nondifferentiable Optimization. Mathematical Programming Studies*, 3 (1975), 1-25. https://doi.org/10.1007/BFb0120696

[10] Zang, I., A smoothing out technique for min-max optimization,*Math. Programm.*, (19) 1980, 61-77. https://doi.org/10.1007/BF01581628

[11] Bagirov, A. M., Al Nuaimat, A., Sultanova, N. Hyperbolic smoothing function method for minimax problems, *Optimization*, 62(6) (2013), 759-782. https://doi.org/10.1080/02331934.2012.675335

[12] Yilmaz, N. and Sahiner, A., On a new smoothing technique for non-smooth, non-convex optimization, *Numer. Algebra Control Optim.*, 10(3) (2020), 317-330. https://doi.org/10.3934/naco.2020004

[13] Wu, Z. Y., Lee, H. W. J., Bai, F. S., Zhang, L. S., Quadratic smoothing approximation to l_1 exact penalty function in global optimization, *J. Ind. Manag. Optim.*, 1(4) 2005, 533-547. https://doi.org/10.3934/jimo.2005.1.533

[14] Lian, S. J., Smoothing approximation to l_1 exact penalty for inequality constrained optimization, *Appl. Math. Comput.*, 219(6) (2012), 3113-3121. https://doi.org/10.1016/j.amc.2012.09.042

[15] Xavier, A. E., Hyperbolic penalty: a new method for nonlinear programming with inequalities, *Int. Trans. Op. Res.*, 8(6) (2001), 659-671. https://doi.org/10.1111/1475-3995.t01-1-00330

[16] Liu, B., On smoothing exact penalty function for nonlinear constrained optimization problem,*J. Appl. Math. Comput.*, 30 (2009), 259-270. https://doi.org/10.1007/s12190-008-0171-z

[17] Xu, X., Meng, Z., Sun, J., Shen, R., A penalty function method based on smoothing lower order penalty function, *J. Comput. Appl. Math.*, 235(14) (2011), 4047-4058. https://doi.org/10.1016/j.cam.2011.02.031

[18] Meng, Z., Dang, C., Jiang, M., Shen, R., A smoothing objective penalty function algorithm for inequality constrained optimization problems, *Numer. Funct. Anal. Optim.*, 32(7) (2011), 806-820. https://doi.org/10.1080/01630563.2011.577262

[19] Sahiner, A., Kapusuz, G., Yilmaz, N., A new smoothing approach to exact penalty functions for inequality constrained optimization problems, *Numer. Algebra Control Optim.*, 6(2) (2016), 161-173. https://doi.org/10.3934/naco.2016006

[20] Xu, X., Dang, C., Chan, F., Wang, Y., On smoothing ℓ_1 exact penalty function for constrained optimization problems, *Numer. Funct. Anal. Optim.*, 40(1) (2019), 1-18. https://doi.org/10.1080/01630563.2018.1483948

[21] Lee, J. and Skipper, D., Virtuous smoothing for global optimization, *J. Glob. Optim.*, 69 (2017), 677-699. https://doi.org/10.1007/s10898-017-0533-x

[22] Xu, L., Lee, J. and Skipper, D., More virtuous smoothing, *SIAM J. Optim.*, 29(2) 2019, 1240-1259. https://doi.org/10.1137/18M11728

[23] Grossmann, C., Smoothing techniques for exact penalty function methods, *In: Panaroma of Mathematics: Pure and Applied Contemporary Mathematics,*, 658 (2016), 249-265.

[24] Hassan M., Baharum, A., Generalized logarithmic penalty function method for solving smooth nonlinear programming involving invex functions, *Arab. J. Basic Appl. Sci.*, 26(1) (2019), 202-214. https://doi.org/10.1080/25765299.2019.1600317

[25] Dolgopolik, M. V., Smooth exact penalty functions: a general approach, *Optim. Lett.*, 10 (2016), 635-648. https://doi.org/10.1007/s11590-015-0886-3

[26] Lucidi S., Rinaldi, F., Exact penalty functions for nonlinear integer programming problems, *J. Optim. Theory Appl.*, 145 (2010), 479-488. https://doi.org/10.1007/s10957-010-9700-7

[27] Di Pillo, G., Lucidi, S., Rinaldi, F., An approach to constrained global optimization based on exact penalty functions, *J. Glob Optim.*, 54 (2012), 251-260. https://doi.org/10.1007/s10898-010-9582-0

[28] Antczak, T., A new exact exponential penalty function method and nonconvex mathematical programming, *Appl. Math. Comput.*, 217(15) (2011), 6652-6662. https://doi.org/10.1016/j.amc.2011.01.051