Journal of Mathematical Modeling
Vol. 13, No. 1, 2025, pp. 153–167. Research Article

JMM

A novel class of exact penalty function approach for optimization problems with inequality constraints

Nurullah Yilmaz[†*], **Asuman Zeytinoglu**[†]

[†]*Department of Mathematics, Suleyman Demirel University, Isparta, Turkey*
Email(s): nurullahyilmaz@sdu.edu.tr, asumanzeytinoglu@sdu.edu.tr

Abstract. Nonlinear programming has always had an important place in the literature, from the past to the present. This study aims to solve the continuous constrained optimization problem, which is an important subclass of nonlinear programming problems. A new twice differentiable smoothing technique for exact penalty functions is presented. It has been demonstrated that any optimum solution of the smoothed exact penalty function coincides with an optimal solution of the original problem. Error analysis is carried out to demonstrate that the optimal solution of the smoothed exact penalty problem approximates to an optimal solution to the constrained optimization problem. The proposed smoothing technique is used to develop an algorithm that produces an optimal solution for the constrained optimization problem. The convergence of the method is demonstrated based on both theoretical and numerical considerations. Numerical examples are provided to illustrate the effectiveness of the proposed method.

Keywords: Constrained optimization, smoothing technique, exact penalty function.
AMS Subject Classification 2010: 90C30, 65K05,65D15.

1 Introduction

We consider the following problem

$$(\text{P}) \qquad \begin{cases} \min\limits_{x\in\mathbb{R}^n} f(x) \\ s.t.\;\; c_i(x) \le 0, \quad j = 1,2,\ldots,m, \end{cases}$$

where $f:\mathbb{R}^n \to \mathbb{R}$ and $c_i(x):\mathbb{R}^n \to \mathbb{R}$, $i \in I = \{1,2,...,m\}$ are continuously differentiable functions. The set of feasible solution is defined by $C_0 := \{x \in \mathbb{R}^n : c_i(x) \le 0,\ i = 1,2,\ldots,m\}$ and we assume that C_0 is non-empty [11, 12, 23, 33].

*Corresponding author

Received: 30 July 2024 / Revised: 8 October 2024 / Accepted: 23 October 2024
DOI: 10.22124/jmm.2024.28053.2472

The penalty function approach is a crucial method for solving the problem (P). It has been extensively utilized to address practical models, such as petrolium industries [14], economic load dispatching problems [9], and design problems [16]. The penalty function approach involves converting a constrained optimization problem into an unconstrained one. Applying the penalty function technique to (P) results in the transformation of the problem into the following unconstrained optimization problem:

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

where $F(x,\rho) = f(x) + \rho\sum_j H(c_i(x))$ and $\rho > 0$ parameter. The most common H functions are $H(t) = \max\{0,t\}^2$, $H(t) = \max\{0,t\}$, $H(t) = \max\{0,t\}^p$ $(0 < p \leq 1)$, $H(t) = \log(1+\max\{0,t\})$, etc. [13, 25]. Furthermore, when the parameter ρ grows, the solution of problem (1) approaches the solution of problem (P). Precision is considered to be one of the desirable characteristics of penalty functions. The function $F(x,\rho)$ is referred to be an exact penalty function for problem (P) if there exists a suitable choice of parameters such that the optimal solution to the penalty problem is also an optimal solution to the original problem [7, 20, 30]. For further information, we recommend consulting the studies by Dolgopolik and Antczak [1, 8].

One of the first known penalty functions is called the l_2 penalty function that is defined as

$$F_2(x,\rho) = f(x) + \rho\sum_i \max\{c_i(x),0\}^2. \tag{2}$$

It is differentiable but not necessarily exact when f and c_i $(i = 1,2,\ldots,m)$ are continuously differentiable [30]. The l_1 penalty function is well-known as one of the most common exact penalty functions. It is defined as

$$F_1(x,\rho) = f(x) + \rho\sum_i \max\{c_i(x),0\}, \tag{3}$$

by Eremin [10] and Zangwill [37]. It is exact but lacks differentiability. The primary drawback of the l_1 exact penalty function is that it restricts the employment of efficient algorithms such as Steepest Descent, Newton, Quasi-Newton, etc., in solving the penalty problem. Alternatively, to enhance the efficiency of the exact penalty function, lower-order exact penalty functions have gained prominence in the literature [24, 25]. The lower order l_p-exact penalty function is defined as

$$F_p(x,\rho) = f(x) + \rho\sum_i \max\{c_i(x),0\}^p, \tag{4}$$

where $0 < p < 1$ [5, 27]. Like the l_1 penalty function, the l_p penalty function is also exact but lacks differentiability and it is non-Lipschitz when $0 < p < 1$. Additionally, the presence of a non-smooth penalty function, particularly with a high penalty value, might lead to numerical instability in the solution process. In order to eliminate the deficiencies arising from the non-smooth penalty forms, smoothing techniques are proposed [6, 29, 38]. The smoothing approach involves representing a non-differentiable function using a set of smooth functions. Let $\mathbb{R}_+$ represent non-negative real numbers, a smoothing function is defined formally as follows.

Definition 1. *[3, 4] 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 $x \in R^n$.

Smoothing functions are used to solve various non-smooth optimization problems [2, 15, 31, 34, 35]. Smoothing technique is used for non-smooth penalty problems for the first time by Pinar and Zenios in [22]. As a result, many smoothing techniques are developed for the exact penalty function in the literature [17, 18, 21, 26, 28, 32, 36].

As it is well-known that gradient based methods (e.g. Newtonian methods) which are powerful tools in nonlinear programming, usually need second-order continuously differentiability of the objective function. Hence, it is crucial to devise smoothing methods that render l_1 and l_p exact penalty functions continuously differentiable to the second order. While various smoothing techniques have been explored in the literature separately for l_1, l_p, and other penalty functions. There are very few studies that have a unified strategy that encompasses all of them.

The objective of this study is to introduce a twice continuously differentiable smoothing function to approximate the exact penalty function in a common form. By implementing the suggested smoothing technique for the exact penalty functions, we obtain a surrogate smoothed penalty problem. An investigation is conducted on the connections between the solutions produced for the original, exact penalty and the smoothed exact penalty problems. The objective is to develop an algorithm to solve problem (P) based on the smoothed penalty problem. This algorithm is applied to solve some test problems, and the results are reported in order to reveal the numerical performance of the proposed algorithm.

2 Main results

2.1 A new exact penalty function

In this part of the study, first we recall the definition a class of exact penalty functions as follows:

$$s(t) = \begin{cases} 0, & t < 0, \\ t^p, & t \geq 0, \end{cases}$$

where $0 < p \leq 1$. According to the new formula, the exact penalty function for problem (P) is defined by

$$F(x,\rho) = f(x) + \rho \sum_i s(c_i(x)), \tag{5}$$

and the obtained penalty form of the problem (P) is given by

$$\text{(PF)} \qquad \min_{x \in \mathbb{R}^n} F(x,\rho).$$

The following assumptions are needed to state the exactness of the above penalty problem.

Assumption 1. *$f(x)$ is a coercive function, i.e., $\lim_{||x|| \to \infty} f(x) = \infty$.*

Assumption 1 implies that there exists 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 minimizers 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 (PF) is a solution of (P).*

Proof. The proof is similar to the proof of [27, Corollary 2.3]. □

2.2 Smoothing techniques

The differentiability of penalty functions established with the function t is not always ensured. Particularly, at the instant $t = 0$, the function s lacks differentiability. Hence, we provide the subsequent smoothing functions for the function s.

The smoothing function of s is defined as

$$s_\varepsilon(t) = \begin{cases} 0, & t < 0, \\ \dfrac{t^{p+4}}{t^4+\varepsilon^4}, & t \geq 0, \end{cases} \tag{6}$$

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

Lemma 1. *For any $t \in \mathbb{R}$, the function $s_\varepsilon(t)$ satisfies the followings:*
(i) $s_\varepsilon(t)$ is twice continuously differentiable,
(ii) $\lim_{\varepsilon\to 0} s_\varepsilon(t) = s(t)$,
(iii) $0 \leq s(t) - s_\varepsilon(t) \leq K\varepsilon^p, \quad 0 < K < 1$.

Proof. (i) For any $\varepsilon > 0$, we have

$$s'_\varepsilon(t) = \begin{cases} 0, & t < 0, \\ \dfrac{pt^{p+7} + (p+4)\varepsilon^4 t^{p+3}}{(t^4+\varepsilon^4)^2}, & t \geq 0, \end{cases}$$

and

$$s''_\varepsilon(t) = \begin{cases} 0, & t < 0, \\ \dfrac{p(p+7)t^{p+6} + (p+4)(p+3)t^{p+2}\varepsilon^4}{(t^4+\varepsilon^4)^2} - \dfrac{8pt^{p+10} + 8(p+4)t^{p+6}\varepsilon^4}{(t^4+\varepsilon^4)^3}, & t \geq 0. \end{cases}$$

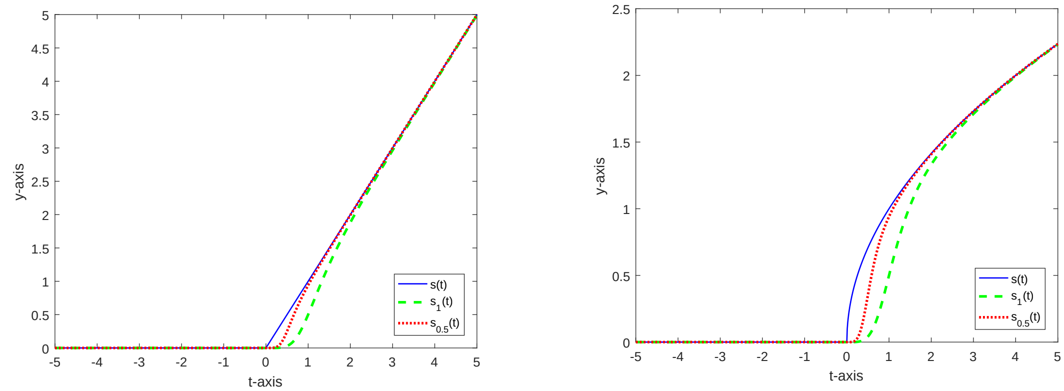
It is easy to see that functions $s'_\varepsilon(t)$ and $s''_\varepsilon(t)$ are continuous at the transition points $t = 0$.
(ii) We have the following clarification:

$$\lim_{\varepsilon\to 0} s_\varepsilon(t) = \begin{cases} \lim\limits_{\varepsilon\to 0} 0 = 0 = s(t), & t < 0, \\ \lim\limits_{\varepsilon\to 0}(\dfrac{t^{p+4}}{t^4+\varepsilon^4}) = t^p = s(t), & t \geq 0. \end{cases}$$

(iii) For any $\varepsilon > 0$,

$$s(t) - s_\varepsilon(t) = \begin{cases} 0, & t < 0, \\ t^p - \dfrac{t^{p+4}}{t^4+\varepsilon^4}, & t \geq 0, \end{cases}$$

For $t < 0$, the difference is 0, let us assume that $t \geq 0$. Now compute the maximum value of difference $d(t) = s(t) - s_\varepsilon(t)$. It is easy to see that $d(t) \geq 0$ and $d(0) = 0$. For any $t > 0$, the derivative of $d(t)$ is



(a) For $s(t) = \max\{0,t\}$. **(b)** For $s(t) = \max\{0,t\}^{\frac{1}{2}}$.

Figure 1: The blue graph represents $s(t)$, the green graph is $s_1(t)$ and the red graph is $s_{0.5}(t)$.

obtained as

$$\begin{aligned} d'(t) =& pt^{p-1} - \frac{(p+4)t^{p+3}(t^4+\varepsilon^4) - t^{p+4}4(t^3)}{(t^4+\varepsilon^4)^2} \\ =& \frac{pt^{p-1}\varepsilon^8 + p\varepsilon^4 t^{p+3} - 4\varepsilon^4 t^{p+3}}{(t^4+\varepsilon^4)^2} \\ =& \frac{\varepsilon^4 t^{p-1}\left(pt^4 + p\varepsilon^4 - 4t^4\right)}{(t^4+\varepsilon^4)^2}. \end{aligned}$$

Therefore the stationary point of $d(t)$ is obtained as $t = \mp\sqrt[4]{\frac{-p}{p-4}}\varepsilon$. Since $t > 0$, we have $t = \sqrt[4]{\frac{-p}{p-4}}\varepsilon$. Thus, the maximum difference is obtained as

$$d\left(\sqrt[4]{\frac{-p}{p-4}}\varepsilon\right) = K\varepsilon^p,$$

where $K = \left(\frac{p}{4-p}\right)^{\frac{p}{4}}\left(1 - \frac{p}{4}\right)$. □

Example 1. Let us consider the functions $s(t) = \max\{0,t\}$ and $s(t) = \max\{0,t\}^{\frac{1}{2}}$. The smoothing functions $s_\varepsilon(t)$ are given in the following Fig. 1. In Fig. 1 (a) and (b), the smoothing parameter ε is taken as 1 and 0.5. It is observed that the smoothing functions approach the original function when $\varepsilon \to 0$.

By utilizing the smoothing function given in (6), the smoothed exact penalty function is obtained as

$$\tilde{F}(x,\rho,\varepsilon) = f(x) + \rho\sum_{i\in I} s_\varepsilon\left(c_i(x)\right).$$

Therefore the smoothed penalty problem is obtained as

$$\text{(SPF)} \qquad \min_{x\in\mathbb{R}^n} \tilde{F}(x,\rho,\varepsilon).$$

Let us now analyze the error estimates.

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

$$0 \leq F(x,\rho) - \tilde{F}(x,\rho,\varepsilon) \leq K\varepsilon^p m\rho, \tag{7}$$

holds and

$$\lim_{\varepsilon \to 0} \tilde{F}(x,\rho,\varepsilon) = F(x,\rho), \tag{8}$$

is satisfied for $\varepsilon, \rho > 0$.

Proof. For any $\rho, \varepsilon > 0$,

$$\begin{aligned} F(x,\rho) - \tilde{F}(x,\rho,\varepsilon) &= f(x) + \rho \sum_{i \in I} s(c_i(x)) - \left[f(x) + \rho \sum_{i \in I} s_\varepsilon(c_i(x)) \right] \\ &= \rho \sum_{i \in I} \left[s(c_i(x)) - s_\varepsilon(c_i(x)) \right]. \end{aligned}$$

Thus, we have $0 \leq F(x,\rho) - \tilde{F}(x,\rho,\varepsilon) \leq K\varepsilon^p m\rho$, and $\lim_{\varepsilon \to 0} \tilde{F}(x,\rho,\varepsilon) = F(x,\rho)$. □

The following corollary shows that the distance between $F(x,\rho)$ and $\tilde{F}(x,\rho,\varepsilon)$ reduces as $\varepsilon \to 0$.

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

Definition 2 ([30])**.** *Let f^* be the optimal objective function value of the problem* (P) *and x be a feasible solution. If the condition $f(x) - f^* \leq \varepsilon$ holds, then x is called $\varepsilon-$approximate solution.*

Definition 3 ([30])**.** *If $c_i(x_\varepsilon) \leq \varepsilon$ for any $i \in I$ and for $\varepsilon > 0$, then the x_ε is called as $\varepsilon-$feasible solution of the problem* (P).

Lemma 2 ([13,30])**.** *Let x^* be the optimal solution to the problem* (PF). *If x^* is a feasible solution to the problem* (P)*, then x^* is the optimal solution for* (P).

Now, we present the following theorem regarding the connections between optimal solutions of problems (P), (PF), and (SPF).

Theorem 3. *Assume that $\rho > 0$, x^* is an optimal solution of the problem* (PF) *and x_ε be an optimal solution of the problem* (SPF). *Then,*

$$\lim_{\varepsilon \to 0} \tilde{F}(x_\varepsilon,\rho,\varepsilon) = F(x^*,\rho). \tag{9}$$

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

Proof. Assume that x^* is an optimal solution of (PF) and x_ε is an optimal solution of (SPF). By taking into account Theorem 2 and following inequalities

$$F(x^*,\rho) \leq F(x_\varepsilon,\rho), \tag{10}$$
$$\tilde{F}(x_\varepsilon,\rho,\varepsilon) \leq \tilde{F}(x^*,\rho,\varepsilon), \tag{11}$$

we obtain

$$\begin{aligned} 0 &\leq F(x^*,\rho) - \tilde{F}(x^*,\rho,\varepsilon) \leq F(x^*,\rho) - \tilde{F}(x_\varepsilon,\rho,\varepsilon) \\ &\leq F(x_\varepsilon,\rho) - \tilde{F}(x_\varepsilon,\rho,\varepsilon) \leq K\varepsilon^p m\rho. \end{aligned}$$

Therefore, (9) holds. Let x^* be an optimal solution of (P) and x_ε be $\varepsilon-$feasible solution (P). Since we have

$$0 \leq \left[f(x^*) + \rho \sum_i s(c_i(x^*)) \right] - \left[f(x_\varepsilon) + \rho \sum_i s_\varepsilon(c_i(x_\varepsilon)) \right] \leq K\varepsilon^p m\rho,$$

$c_i(x^*) \leq 0$ and $c_i(x_\varepsilon) \leq \varepsilon$, we have

$$\rho \sum_i s(c_i(x^*)) = 0, \quad 0 \leq \rho \sum_i s_\varepsilon(c_j(x_\varepsilon)) \leq K\varepsilon^p m\rho, \tag{12}$$

and we obtain $|f(x_\varepsilon) - f(x^*)| \leq 2K\varepsilon^p m\rho$. □

2.3 Algorithm

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

Smoothing Penalty Algorithm (SPA)

Step 1 Choose $x^0 \in \mathbb{R}^n$, $\varepsilon_0 > 0$, $\rho_0 > 0$ and the tolerance parameter is chosen as $\varepsilon = 10^{-4}$. Determine the parameters $N > 1$, $0 < \delta < 1$. Let $k = 0$ and go to Step 2.

Step 2 Consider x^k as an initial point and minimize $\tilde{F}(x,\rho_k,\varepsilon_k)$ by using any local search methods. Let x^{k+1} be an optimal solution.

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

Remark 1. *During Step 2 of SPA, the selection of gradient-based local search method (such as Steepest Descent, Newton, Quasi-Newton, etc.) depends on the level of smoothing approximation.*

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

We denote the following index sets

$$I_\varepsilon^-(x) = \{i | c_i(x) < \varepsilon, i \in I\}, \qquad I_\varepsilon^+(x) = \{i | c_i(x) \geq \varepsilon, i \in I\}.$$

For the convergence of SPA, the following theorem is presented.

Theorem 4. *Assume that Assumptions 1 and 2 are held. Then the sequence $\{x^k\}$ generated by SPA is bounded, and the limit point $\bar{x}$ is the optimal solution to the problem* (P).

Proof. The boundedness of $\{x^k\}$ is proved first. Since the sequence $\{F(x^k,\rho_k,\varepsilon_k)\}$ is bounded, then there exists a number L such that

$$\tilde{F}(x^k,\rho_k,\varepsilon_k) \leq L, \quad k=0,1,2,\ldots. \tag{13}$$

On the contrary, let $\{x^k\}$ be unbounded i.e., when $k\to\infty$, $\|x^k\|\to\infty$. Inequality (13) is re-stated as

$$L \geq \tilde{F}(x^k,\rho_k,\varepsilon_k) \geq f(x^k), \quad k=0,1,2,\ldots \tag{14}$$

and it contradicts with the Assumption 1. Therefore, $\{x^k\}$ must be bounded.

Now, we demonstrate that the limit point $\bar{x}$ of $\{x^k\}$ is the optimal solution of (P). We first prove that $\bar{x}$ is a feasible solution for 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 $i \in I$ for $c_i(\bar{x}) \geq \alpha > 0$ such that

$$\begin{aligned}\tilde{F}(x^k,\rho_k,\varepsilon_k) &= f(x^k) + \rho_k \sum_{i\in I} s_{\varepsilon_k}(c_i(x^k)) \\ &= f(x^k) + \rho_k \sum_{i\in I^+_{\varepsilon_k}(x^k)} s_{\varepsilon_k}(c_i(x^k)) + \rho_k \sum_{i\in I^-_{\varepsilon_k}(x^k)} s_{\varepsilon_k}(c_i(x^k)),\end{aligned} \tag{15}$$

where $c_i(x^k) \geq \alpha > 0$, the set $\{i : c_i(x^k) \geq \alpha\}$ is non-empty. There is $i_0 \in I$ with $c_{i_0}(x^k) \geq \alpha$. Since $\rho_k \to \infty$ as $k\to\infty$, from the equation (15) we obtain

$$\tilde{F}(x^k,\rho_k,\varepsilon_k) \to \infty. \tag{16}$$

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

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

$$\tilde{F}(x^k,\rho_k,\varepsilon_k) \leq \tilde{F}(x^*,\rho_k,\varepsilon_k), \quad k=1,2,\ldots. \tag{17}$$

Similarly, we have

$$f(x^k) + \rho_k \sum_{i\in I} s_{\varepsilon_k}(c_i(x^k)) \leq f(x^*) + \rho_k \sum_{i\in I} s_{\varepsilon_k}(c_i(x^*)), \quad k=1,2,\ldots$$

and

$$f(x^k) \leq f(x^*). \tag{18}$$

So while $k\to\infty$,

$$f(\bar{x}) \leq f(x^*). \tag{19}$$

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

$$f(\bar{x}) \geq f(x^*). \tag{20}$$

From (19) and (20), we obtain $f(\bar{x}) = f(x^*)$. It means that $\bar{x}$ is the optimal solution for (P). □

Table 1: Numerical results for the Problem 1.

p	k	ρ_k	ε_k	x^{k+1}	$f(x^{k+1})$	$\max(c_i)$
	0	10	1	$(0.194848, 0.858616, 2.069137, -0.996066)$	-45.3617	0.4501
1	1	20	0.075	$(0.171191, 0.837043, 2.012521, -0.966813)$	-44.3066	0.0284
	2	40	0.0056	$(0.169725, 0.835985, 2.008664, -0.965255)$	-44.2384	0.0018
	3	80	$4.2187e-04$	$(0.169640, 0.835930, 2.008400, -0.965133)$	-44.2338	$8.3059e-05$
	0	10	1	$(0.191811, 0.855439, 2.063492, -0.995154)$	-45.2608	0.4119
0.5	1	100	0.1	$(0.170412, 0.836305, 2.010702, -0.965982)$	-44.2728	0.0153
	2	1000	0.01	$(0.169142, 0.835397, 2.009040, -0.964597)$	-44.2353	0.0006
	3	10000	0.001	$(0.169097, 0.835357, 2.008952, -0.964592)$	-44.2338	$2.2512e-05$
	0	10	1	$(2.160854, 2.211429, 4.992058, -3.197717)$	-79.4523	50.8584
0.1	1	70	0.1	$(0.170114, 0.836020, 2.010008, -0.965673)$	-44.2599	0.0103
	2	490	0.01	$(0.169577, 0.835659, 2.008613, -0.964972)$	-44.2345	0.0003
	3	3430	0.001	$(0.169562, 0.835649, 2.008573, -0.964957)$	-44.2338	$8.3706e-06$

3 Numerical results

To evaluate the numerical efficiency of SPA, we implement it on various benchmark problems found in the literature. The tables contain detailed results, and evaluations of these outcomes are provided. Firstly, the tables include a list of abbreviations used.

$$\begin{aligned}
k &: \text{ Number of iterations,}\\
x^k &: \text{ the result of } k\text{−th iteration,}\\
\rho_k &: \text{ penalty function parameter in the } k\text{−th iteration,}\\
\varepsilon_k &: \text{ smoothing parameter of the } k\text{−th iteration ,}\\
c_i(x^k) &: \text{ constraint function value at } x^k,\\
\tilde{F}(x^k, \rho_k, \varepsilon_k) &: \text{ value of function } \tilde{F} \text{ at point } x^k,\\
f(x^k) &: \text{ value of function } f \text{ at point } x^k.
\end{aligned}$$

Problem 1 ([17])**.** Consider the following problem which is called as Rosen-Suzuki problem:

$$\begin{aligned}
\min f(x) &= x_1^2 + x_2^2 + 2x^3 + x_4^2 - 5x_1 - 21x_3 + 7x_4\\
\text{s.t.} \quad c_1(x) &= 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \leq 0,\\
c_2(x) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0,\\
c_3(x) &= x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0.
\end{aligned}$$

We select the starting point as $x^0 = (0,0,0,0)$, $\rho_0 = 10$, $\varepsilon_0 = 1$. The obtained numerical results are illustrated in Table 1.

Applying SPA, the minimum value is found as -44.2338 for three different p values. In [17] and [30], the minimum values are found as -44.23040 and -44.2338, respectively. It is seen that proposed algorithm provides numerically better results than [17].

Table 2: Numerical results for the Problem 2.

p	k	ρ_k	ε_k	x^{k+1}	$f(x^{k+1})$	$\max(c_i)$
	0	10	1	$(2.523920, 4.260011, 0.973563)$	943.1775	0.4657
1	1	60	0.01	$(2.500148, 4.221540, 0.964771)$	944.2091	0.0029
	2	360	0.0001	$(2.500003, 4.221295, 0.964715)$	944.2156	$1.8428e-05$
	0	1000	1	$(2.506229, 4.230478, 0.966764)$	943.9648	0.1128
0.5	1	8000	0.1	$(2.500255, 4.221665, 0.964750)$	944.2057	0.0045
	2	64000	0.01	$(2.500009, 4.221320, 0.964671)$	944.2153	0.0002
	3	512000	0.001	$(2.500000, 4.221302, 0.964667)$	944.2157	$-2.3762e-05$
	0	1000	1	$(2.505230, 4.228200, 0.966348)$	944.0211	0.0877
0.1	1	8000	0.01	$(2.500013, 4.221286, 0.964775)$	944.2154	0.0001
	2	64000	0.0001	$(2.500006, 4.221281, 0.964774)$	944.2157	$-2.6880e-07$

Problem 2 ([30])**.** Consider the following problem

$$\begin{aligned} \min f(x) &= 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3, \\ \text{s.t.} \quad & c_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0, \\ & c_2(x) = (x_1-5)^2 + x_2^2 + x_3^2 - 25 = 0, \\ & c_3(x) = (x_1-5)^2 + (x_2-5)^2 + (x_3-5)^2 - 25 \le 0. \end{aligned}$$

We select $x^0 = (2,2,1)$ as a starting point $\rho_0 = 10$ and $\rho_0 = 1000$, $\varepsilon_0 = 1$. The obtained numerical results are illustrated in Table 2.

By considering SPA, the obtained minimum values using three different p values are lower than the value 944.2157 which is computed in [30]. As seen from Table 2, the solutions are obtained with a lower number of iterations than [30].

Problem 3 ([19])**.** Consider the following problem

$$\begin{aligned} \min f(x) &= -x_1x_2x_3, \\ \text{s.t.} \quad & c_1(x) = x_1^2 + 2x_2^2 + 4x_3^2 - 48 \le 0. \end{aligned}$$

We select $x^0 = (3,3,3)$ as a starting point $\rho_0 = 10$ and $\rho_0 = 100$, $\varepsilon_0 = 1$. The obtained numerical results are illustrated in Table 3.

By considering SPA, the obtained minimum values using three different p values are similar to the result in [19]. As seen from Table 3, the solutions are obtained with a lower number of iterations than [19].

Table 3: Numerical results for the Problem 3.

p	k	ρ_k	ε_k	x^{k+1}	$f(x^{k+1})$	$\max(c_i)$
	0	10	1	$(4.014449, 2.838644, 2.007225)$	-22.8235	0.3474
1	1	30	0.001	$(3.994331, 2.819267, 2.009295)$	-22.6268	0.0003
	2	90	0.00001	$(3.994328, 2.819263, 2.009282)$	-22.6268	$-6.7974e-09$
	0	10	1	$(4.012771, 2.837457, 2.006385)$	-22.8448	0.3070
0.5	1	30	0.01	$(3.992859, 2.818700, 2.010478)$	-22.6273	0.0012
	2	90	0.0001	$(3.992851, 2.818700, 2.010411)$	-22.6265	$4.2335e-06$
	0	100	1	$(4.005350, 2.832210, 2.002675)$	-22.7183	0.1285
0.1	1	400	0.1	$(3.988135, 2.817613, 2.013771)$	-22.6288	0.0042
	2	1600	0.01	$(3.988070, 2.817522, 2.013615)$	-22.6259	0.0001
	3	6400	0.001	$(3.988067, 2.817519, 2.013610)$	-22.6258	$4.5717e-06$

Table 4: Numerical results for the Problem 4.

p	k	ρ_k	ε_k	x^{k+1}	$f(x^{k+1})$	$\max(c_i)$
	0	100	1	$(2.340485, 1.951866, -0.480279, 4.368116, -0.624371, 1.039865, 1.592081)$	680.3210	0.2187
1	1	500	0.01	$(2.333113, 1.950701, -0.479270, 4.366800, -0.624250, 1.037373, 1.597314)$	680.6287	0.0014
	2	2500	0.0001	$(2.332967, 1.950705, -0.479249, 4.366787, -0.624252, 1.037318, 1.597434)$	680.6317	$9.7652e-06$
	0	10	1	$(2.345867, 1.952200, -0.481684, 4.369682, -0.624298, 1.040767, 1.590960)$	680.1369	0.3526
0.5	1	80	0.01	$(2.329879, 1.950641, -0.479399, 4.367403, -0.622420, 1.037868, 1.593847)$	680.6344	0.0007
	2	640	0.0001	$(2.329628, 1.950710, -0.479361, 4.367406, -0.622420, 1.037769, 1.594064)$	680.6323	$1.3692e-06$
	0	10	1	$(9.665132, 11.880905, -0.001194, 10.999269, -0.083889, 1.120468, 1.468908)$	15.9126	60318.3191
0.1	1	100	0.63	$(4.621250, 9.962333, -0.000532, 10.986290, -0.150844, 0.806047, 1.672461)$	35.2315	29948.2913
	2	1000	0.3969	$(2.249945, 1.874762, -0.323445, 4.553344, -0.552284, 1.064130, 1.509449)$	681.6144	0.0354
	3	10000	0.2501	$(2.323612, 1.951682, -0.362875, 4.363414, -0.622020, 1.053920, 1.575226)$	680.6504	0.0098
	4	100000	0.1575	$(2.321946, 1.951802, -0.362695, 4.363307, -0.622036, 1.053274, 1.576647)$	680.6741	0.0014
	5	1000000	0.0992	$(2.321950, 1.951792, -0.362696, 4.363304, -0.622037, 1.053275, 1.576643)$	680.6752	0.0004
	6	10000000	0.0625	$(2.321950, 1.951789, -0.362696, 4.363303, -0.622037, 1.053276, 1.576643)$	680.6755	0.0001
	7	100000000	0.0394	$(2.321950, 1.951788, -0.362696, 4.363303, -0.622037, 1.053276, 1.576643)$	680.6756	$2.6725e-05$

Problem 4 ([19])**.** Consider the following problem

$$
\begin{aligned}
\min f(x) = {} & (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7, \\
\text{s.t.} \quad & c_1(x) = 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127 \le 0, \\
& c_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \le 0, \\
& c_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \le 0, \\
& c_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0.
\end{aligned}
$$

We select $x^0 = (1, 2, 0, 4, 0, 1, 1)$ as a starting point $\rho_0 = 10$ and $\rho_0 = 100$, $\varepsilon_0 = 1$. The obtained numerical results are illustrated in Table 4.

Table 5: Numerical results for the Problem 5.

p	k	ρ_k	ε_k	x^{k+1}	$f(x^{k+1})$	$\max(c_i)$
	0	10	1	$(0.056466, 1.029278, 2.075206, -0.974306)$	-45.2034	0.4302
1	1	60	0.01	$(0.000436, 1.000235, 2.000505, -0.999736)$	-44.0081	0.0029
	2	360	0.0001	$(0.000817, 1.000188, 1.999865, -0.998740)$	-43.9966	$2.2158e-05$
	0	10	1	$(0.052510, 1.027057, 2.069015, -0.973659)$	-45.0955	0.3928
0.5	1	80	0.01	$(0.000237, 1.000145, 2.000155, -0.999936)$	-44.0033	0.0012
	2	640	0.0001	$(0.000834, 1.000182, 1.999806, -0.998928)$	-43.9968	$3.8382e-06$
	0	100	1	$(0.024624, 1.012648, 2.031673, -0.983253)$	-44.4860	0.1742
0.1	1	300	0.01	$(-0.002643, 0.999726, 2.002024, -0.997763)$	-44.0007	0.0003
	2	900	0.0001	$(-0.002916, 0.999595, 2.001745, -0.997971)$	-43.9982	$6.3476e-07$

Problem 5 ([19]). Consider the following problem

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4, \\ \text{s.t.} \quad & c_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0, \\ & c_2(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0, \\ & c_3(x) = 2x_1^2 + x_2^2 + +x_3^2 + 2x_1 - x_2 - x_4 - 5 \le 0. \end{aligned}$$

We select $x^0 = (0,0,0,0)$ as a starting point $\rho_0 = 10$ and $\rho_0 = 100$, $\varepsilon_0 = 1$. The obtained numerical results are illustrated in Table 5.

By considering SPA, the obtained minimum values for problems 3–5 using three different p values are similar to the result in [19]. As seen from Tables 3–5, the solutions are obtained with a lower number of iterations than the algorithm given by Liu in [19].

4 Conclusion

This paper introduces a new twice continuously differentiable smoothing exact penalty function that includes both l_1 and l_p exact penalty forms. A novel minimization approach is devised to address problem (P) using the surrogate problem (SPF). The algorithm is implemented on the test tasks and achieves good outcomes.

The suggested method for smoothing non-smooth exact penalty functions possesses a versatile structure. The option is accessible for both Lipschitz and non-Lipschitz penalty functions. This feature is the most crucial aspect of our smoothing process, setting it apart from other procedures.

SPA is consistently highly effective for optimization problems of small and medium scale. By implementing this technique, the optimal value is efficiently determined, and the algorithm provides a high level of precision in identifying the optimal value.

References

[1] T. Antczak, *A new exact exponential penalty function method and nonconvex mathematical programming*, Appl. Math. Comput. **217** (2011) 6652–6662.

[2] A.M. Bagirov, A. Al Nuaimat, N. Sultanova, *Hyperbolic smoothing function method for minimax problems*, Optimization **62** (2013) 759–782.

[3] C. Chen, O.L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problem*, Comput. Optim. Appl. **5** (1996) 97–138.

[4] X. Chen, *Smoothing methods for nonsmooth, nonconvex minimization*, Math. Program. Ser. (B) **134** (2012) 71–99.

[5] F.S. Bai, Z.Y. Wu, D.L. Zhu, *Lower order calmness and exact penalty function*, Optim. Methods Softw. **21** (2006) 515–525.

[6] D.P. Bertsekas, *Nondifferentiable optimization via approximation*, Math. Prog. Stud. **3** (1975) 1–25.

[7] G.D. Pillo, S. Lucidi, F. Rinaldi, *An approach to constrained global optimization based on exact penalty functions*, J. Global Optim. **54** (2012) 251–260.

[8] M.V. Dolgopolik, *Smooth exact penalty functions: a general approach*, Optim. Lett. **10** (2016) 635–648.

[9] I. Eke, *Combined heat and power economic dispatch by Taguchi-based filled function*, Eng. Optim. **55** (2023) 791-805.

[10] I.I. Eremin,*The penalty method in convex programming*, Cybernet. Syst. Anal. **3** (1967) 53–56.

[11] A.V. Fiacco, G.P. McCormick, *Nonlinear Programming*, SIAM Philadelphia, 1990.

[12] C. Grossmann, M. Zadlo, *A general class of penalty/ barrier path-following Newton methods for nonlinear programming*, Optimization **54** (2005) 161-190.

[13] M. Hassan, A. Baharum, *Generalized logarithmic penalty function method for solving smooth nonlinear programming involving invex functions*, Arab. J. Basic Appl. Sci. **26** (2019) 202–214.

[14] M. Hassan, S.A. Sulaiman, *Logarithmic penalty function approach for enhancing the alkylation process in Kaduna and Warri Refineries*, Matematika **40** (2024) 17-25.

[15] J. Lee, D. Skipper, *Virtuous smoothing for global optimization*, J. Global Optim. **69** (2017) 677–699.

[16] H. Liao, X. Yuan, R. Gao, *An exact penalty function optimization method and its application in stress constrained topology optimization and scenario based reliability design problems*. Appl. Math. Model. **125** (2024) 260-292.

[17] S.J. Lian, *Smoothing approximation to l_1 exact penalty for inequality constrained optimization*, Appl. Math. Comput. **219** (2012) 3113–3121.

[18] B. Liu, *On smoothing exact penalty function for nonlinear constrained optimization problem*, J. Appl. Math. Comput. **30** (2009) 259–270.

[19] B. Liu, *A smoothing penalty function method for the constrained optimization problem*, Open J. Optim. **8** (2019) 113–126.

[20] S. Lucidi, F. Rinaldi, *Exact penalty functions for nonlinear integer programming problems*, J. Optim. Theory Appl. **145** (2010) 479–488.

[21] Z. Meng, C. Dang, M. Jiang, R. Shen, *A smoothing objective penalty function algorithm for inequality constrained optimization problems*, Numer. Funct. Anal. Optim. **32** (2011) 806–820.

[22] M.C. Pinar, S. Zenios, *On smoothing exact penalty functions for convex constrained optimization*, SIAM J. Optim. **4** (1994) 468–511.

[23] L.M. Ramirez, N. Maculan, A.E. Xavier, V.L. Xavier, *A constraint qualification for the dislocation hyperbolic augmented Lagrangian algorithm*, Results Control Optim. **15** (2024) 100429.

[24] A.M. Rubinov, B.M. Glover, X.Q. Yang, Decreasing functions with applications to penalization, SIAM J. Optim. **10** (1999) 289–313.

[25] A.M. Rubinov, X.Q. Yang, A.M. Bagirov, *Penalty functions with a small penalty parameter*, Optim. Methods Softw. **17** (2002) 931–964.

[26] A. Sahiner, G. Kapusuz, N. Yilmaz, *A new smoothing approach to exact penalty functions for inequality constrained optimization problems*, Numer. Algebra Control. Optim. **6** (2016) 161–173.

[27] Z.Y. Wu, F.S. Bai, X.Q. Yang, L.S. Zhang, *An exact lower-order penalty function and its smoothing in nonlinear programming*, Optimization **53** (2004) 51–68.

[28] Z.Y. Wu, H.W. J. Lee, F.S. Bai, L.S. Zhang, *Quadratic smoothing approximation to l_1 exact penalty function in global optimization*, J. Ind. Manag. Optim. **1** (2005) 533–547.

[29] A.E. Xavier, *Hyperbolic penalty: a new method for nonlinear programming with inequalities*, Int. Trans. Oper. Res. **8** (2001) 659–671.

[30] X. Xu, Z. Meng, J. Sun, R. Shen, *A penalty function method based on smoothing lower order penalty function*, J. Comput. Appl. Math. **235** (2011) 4047–4058.

[31] L. Xu, J. Lee, D. Skipper, *More virtuous smoothing*, SIAM J. Optim. **29** (2019) 1240–1259.

[32] X. Xu, C. Dang, F. Chan, Y. Wang, *On smoothing ℓ_1 exact penalty function for constrained optimization problems*, Numer. Funct. Anal. Optim. **40** (2019) 1–18.

[33] X.Q. Yang, Z.Q. Meng, X.X. Huang, G.T.Y. Pong, *Smoothing nonlinear penalty functions for constrained optimization problems*, Numer. Funct. Anal. Optim. **24** (2003) 351–364.

[34] N. Yilmaz, A. Sahiner, *On a new smoothing technique for non-smooth, non-convex optimization*, Numer. Algebra Control Optim. **10** (2020) 317–330.

[35] N. Yilmaz, *Introducing three new smoothing functions: Analysis on smoothing-Newton algorithms*, J. Math. Model. **12** (2024) 463-479.

[36] N. Yilmaz, H. Ogut, *An exact penalty function approach for inequality constrained optimization problems based on a new smoothing technique*. Commun. Fac. Sci. Univ. Ank. Ser. A1 Math. Stat. **72** (2023) 761-777.

[37] W. I. Zangwill, *Nonlinear programing via penalty functions*, Manag. Sci. **13** (1967) 344–358.

[38] I. Zang, *A smoothing out technique for min-max optimization*, Math. Program. **19** (1980) 61–77.