© August 2018 | IJIRT | Volume 5 Issue 3 | ISSN: 2349-6002

Comparative study of Smooth Penalty Function Algorithms for Solving Nonlinear Constrained Optimization Problems

Dharminder Singh
Post-Graduate Department of Mathematics, Guru Nanak College Budhlada, Mansa, Punjab (India)

***Abstract*—This paper undertakes a thorough comparative investigation of various smoothing techniques utilized in nonlinear optimization problems featuring inequality constraints. The objective of this comparative analysis is to elucidate the merits and limitations associated with each smoothing technique, offering insights into their applicability, convergence characteristics, and overall efficacy in the context of nonlinear inequality-constrained optimization problems. The findings of this study are intended to provide valuable guidance for researchers and practitioners engaged in the pursuit of optimal solutions, particularly in situations where conventional differentiability assumptions are not applicable.**

I. INTRODUCTION

In the realm of optimization, the Nonlinear Inequality Constrained Optimization Problem, denoted as (*P*), emerges as a focal point in addressing real-world challenges. Formally expressed as:

$$(P) \quad \text{minimize} \quad f(x)$$
$$\text{subject to} \quad g_i(x) \leq 0 \ \forall \ 1 \leq i \leq n.$$

Here, f and g_i represent real-valued functions defined across the continuum of real numbers. The feasible region, denoted as $X = \{x \in R^n \mid g_i(x) \leq 0\}$, encapsulates the constraints imposed on the optimization. Our pursuit is to minimize the objective function $f(x)$ while navigating through the intricacies of these constraints. In this pursuit, the functions f and g_i, assumed to be both second-order differentiable and continuous, underpin the foundation of numerous problems spanning engineering, management, and network domains.

This paper embarks on a journey to conduct a comprehensive comparative analysis of various smoothing tech- niques tailored for nonlinear inequality-constrained optimization. As we delve into these techniques, our objective is to unravel the nuanced strengths and limitations each method presents. By shedding light on their applicability, convergence properties, and overall performance, this study endeavors to provide valuable insights for researchers and practitioners grappling with scenarios where traditional differentiability assumptions fall short.

Various methods have been developed to address such non-linear inequality unconstrained optimization problems. One prevalent approach is the utilization of penalty function methods, which involves transforming the unconstrained optimization problem into a set of constrained optimization problems (COPs). This transformation enables the appli- cation of classical gradient methods. The seminal work by Zangwill [1] in 1967 introduced the classical l_1 penalty function, represented by

$$H_1(x, \mu) = f(x) + \mu \sum_{i=1}^{m} \max\{g_i(x), 0\}.$$

However, it's important to note that this l_1 penalty function, while precise, lacks smoothness, posing challenges for classical optimization methods such as Newton and gradient methods.

Another well-known penalty function, the l_2 penalty function, is presented as

$$H(x, \mu) = f(x) + \mu \sum_{i=1}^{m} [\max\{g_i(x), 0\}]^2.$$

Unlike the l_1 penalty, this function is smooth but not exact.

In 2003, Yang and Huang [2] introduced a novel penalty function known as the *k*-th power penalty function, repre- sented by $H_k(x, \rho) = [f(x) + \rho \sum_{i=1} \max\{g_i(x), 0\}]^k$. This function becomes the l_1 exact penalty function for

$k = 1$
and is smooth for $k > 1$ but non-differentiable for $0 < k \leq 1$.
A large number of scholars have come to the conclusion that the exact penalty function algorithms must have an increase in the exact penalty factors to locate a more optimal solution, and these functions cannot be differentiated [1, 3, 4, 5, 6].
The presence of smooth penalty functions is typically preferred in optimization problem solving due to the inherent lack of smoothness in exact penalty functions. Consequently, various innovative strategies have emerged in the field of exact penalty functions as discussed in [7, 8, 9, 10, 11]. The SPFM technique has been widely studied and introduced by Fiaccio and McCormick [7] as a general approach.
This paper focuses on the comparative analysis of these penalty functions and introduces a novel smoothing technique for the l_1 exact penalty function, rendering it second-order differentiable. The smooth penalty function $p_\varepsilon(t)$ is presented, and its application in obtaining a second-order differentiable approximation of the traditional l_1 penalty function is explored. The subsequent sections delve into the connection between the solutions of the smooth penalty function and the original inequality constrained optimization problem, present an algorithm based on the smooth penalty function for solving the constrained optimization problem, and conclude by assessing the practicality of the proposed technique through numerical examples.

A SECOND ORDERED SMOOTH PENALTY FUNCTION

In first case, let us consider the real valued function $p(t)$ which is given below

$$p(t) = \begin{cases} 0 & t \leq 0 \\ t & t \geq 0 \end{cases} \quad (1)$$

It is straightforward to indicate that $p(t)$ is a continuous real-valued function, but it cannot be differentiated. Therefore, the optimal penalty problem P_1 is given by

$$P_1 \quad \min H_1(x,\rho) = f(x) + \rho \sum_{i=1}^{m} p(g_i(x)) \quad \text{s.t. } x \in \mathbb{R}^n$$

For $\rho > 0$, to smooth above function, we define:

$$p_\varepsilon(t) = \begin{cases} 0 & t < 0 \\ \frac{t^3}{3\varepsilon^2} & 0 \leq t < \varepsilon \\ t - \frac{2\varepsilon}{3} & t \geq \varepsilon \end{cases} \quad (2)$$

Here ε is the smoothing parameter. Let us consider $p : \mathbb{R} \to \mathbb{R}$ given as:

$$p(t) = \begin{cases} 0 & t \leq 0 \\ t^{\frac{2}{3}} & t \geq 0 \end{cases} \quad (3)$$

The function $p(t)$ is exact but not smooth. So to make it smooth write the optimization problem for it as:

$$PP_1 \quad \psi_\sigma(x) = f(x) + \sigma \sum_{i=1}^{m} p(g_i(x)) \quad (4)$$

the associated smooth penalty optimization problem reduces as:

$$\text{minimize } \psi_\sigma(x) \text{ s.t. } x \in \mathbb{R}^n \quad (5)$$

From the definition in (3), clearly, the function $p(t)$ on $\mathbb{R}^1$ does not fall into the class of continuous functions. We propose the introduction of a new function that possesses the ideal characteristics of continuity and differentiability in

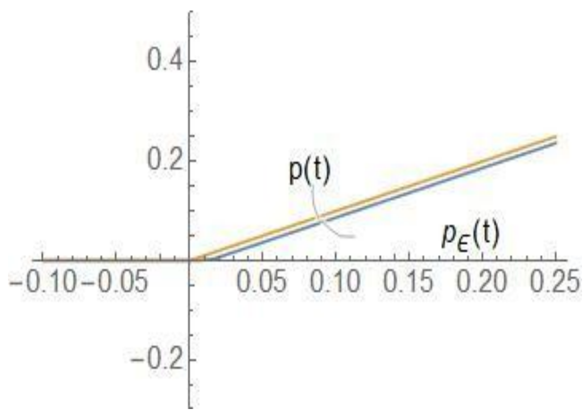

FIGURE 1. The behaviour of $p_\varepsilon(t)$ at $\varepsilon = 0.02$ and $p(t)$

order to avoid this limitation. Specifically, to find a function that possesses a continuous first-order derivative is our main objective. To fulfill these criteria, we define the smoothing function as follows:

$$p_\varepsilon(t) = \begin{cases} 0, & \text{if} \quad t \leq 0 \\ \frac{t^{4/3}}{2\varepsilon^{2/3}}, & \text{if} \quad t > 0 \text{ and } t \leq \varepsilon \\ t^{2/3} - \frac{\varepsilon^{2/3}}{2}, & \text{if} \quad t > \varepsilon \end{cases} \quad (6)$$

We prove that above $p_\varepsilon(t)$ is continuously differentiable and its derivative is given by:

$$p'_\varepsilon(t) = \begin{cases} 0 & t \leq 0 \\ \frac{2t^{1/3}}{3\varepsilon^{2/3}} & 0 \leq t \leq \varepsilon \\ \frac{2}{3t^{1/3}} & t \geq \varepsilon \end{cases} \quad (7)$$

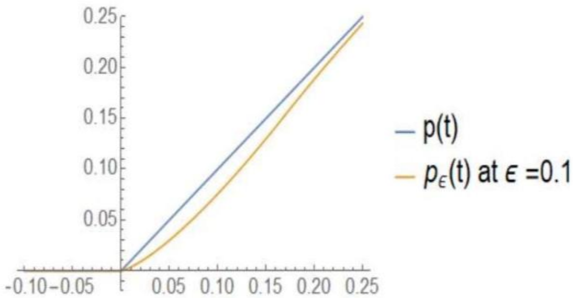

FIGURE 2. The behaviour of $p_\varepsilon(t)$ at $\varepsilon = 0.1$ and $p(t)$

Proposition 1. *For any* $\varepsilon > 0$, $p_\varepsilon(t)$ *is 2nd-order continuously differentiable function on* R, *where*

$$p'_\varepsilon(t) = \begin{cases} 0 & t < 0 \\ \frac{t^2}{\varepsilon^2} & 0 \le t < \varepsilon \\ 1 & t \ge \varepsilon \end{cases} \tag{8}$$

Now corresponding to the penalty function $p_\varepsilon(t)$, the penalty optimization problem is presented by following expression

$$H(x, \rho, \varepsilon) = f(x) + \rho \sum_{i=1}^{m} p_\varepsilon(g_i(x)).$$

where f and g_i's ($i = 1, 2, \ldots, n$) are assumed to be 2nd order continuously differentiable functions, so $H(x, \rho, \varepsilon)$ is 2nd order differentiable function which is also continuous on R^n. Thus, the original optimization problem reduces to the form:

(PP_1) min $H(x,\rho,\varepsilon)$ s.t. $x \in R^n$.

We examine the correlation in (P_1) and (PP_1).

Let $\psi_{\sigma,\varepsilon}(x) = f(x) + \sigma \sum_{i=1}^{m} p_\varepsilon(g_i(x))$

This smooth penalty optimization problem is written as:

min $\psi_{\sigma,\varepsilon}(x)$ so that $x \in R^n$ (9)

Proposition 2. *Let* $x \in H_0$, *and* $\varepsilon > 0$, *in this case we prove that*

$$0 \le \psi_\sigma(x) - \psi_{\sigma,\varepsilon}(x) \le \frac{1}{2} m\sigma\varepsilon^{\frac{2}{3}}$$

Proposition 3. *For* $\varepsilon > 0$ *and* $x \in R^n$,

$0 \le \psi_\sigma(x) - \psi_{\sigma,\varepsilon}(x) \le {}_2\, m\sigma\varepsilon\ 3$

$$0 \le H_1(x, \rho) - H(x, \rho, \varepsilon) \le \frac{2m\rho\varepsilon}{3}. \quad j$$

j

From above proposition, it is very much clear that ε plays an important role to control the gap between $H(x, \rho, \varepsilon)$ and $H_1(x, \rho)$ as well as between $\psi_\sigma(x)$ and $\psi_{\sigma,\varepsilon}(x)$. Moreover, it directly leads to the following result:

Proposition 4. *Let* ε_j *be the sequence of positive real numbers which converges to zero and* x^j *be the solution to the optimization problem* $\min_{x\in R^n} H(x, \rho, \varepsilon_j)$ *for given penalty parameter* $\rho > 0$. *Also let* x' *be an accumulation point of sequence* x_j. *Then* x' *is an optimal solution" for* P_2.

Similarly, we have the following result for second smooth penalty function.

Proposition 5. *Consider the sequence positive numbers* ε_j *such that it converges to zero as j tends to infinity. Also suppose that for minimization problem* $\min_{x\in H0} \psi_{\sigma,k}(x)$. *Then* $\min_{x\in H0} \psi_\sigma(x)$ *has the optimal solution* $\bar{x}$, *where* $\bar{x}$ *is the limit point of sequence* x_j .

Definition 1. *A point* $x \in X_0$ *is considered to be* ε-*approximate optimal solution to* (P) *if it meets the following conditions*

$|f^* - f(x)| \le \varepsilon$,

where f^* *denotes* (P)*'s optimal objective value.*

Definition 2. *A point* $x_\varepsilon \in R^n$ *is said to be* ε-*feasible to* (P) *when it meets the* $g_i(x_\varepsilon) \le \varepsilon$ *for all* $i = 1,2,\ldots,m$.

II. THE ALGORITHMS

The algorithm on the basis of smoothed penalty problem given in (P_1) is given below.

Algorithm I

Step 1 Choose an initial point labeled as x_0. Set a stopping tolerance represented by $\varepsilon > 0$ which is a small positive value indicating the desired level of accuracy for the solution. Assign positive values for ε_0 and σ_0. Select two additional values: λ , which should be a decimal between 0 and 1 , and N which should be greater than 1. Start the iteration with initial value $j = 0$ and follow the next step.

Step 2 Utilize the current point x j (obtained from the previous step) as starting solution. Solve minx∈R∗ ψσj ,ε j (x)to get the next solution x∗. The algorithm's subsequent steps or iterations can be performed after x∗ has been achieved.

Step 3 If we get the desired ε−feasible solution as x∗, in that case, the solution is close to optimal. Otherwise, write x j+1 = x∗ with ε j+1 = λεj and σj+1 = Nσj, then follow

the second step with j = j + 1.
Now, on the basis of smoothed penalty problem that was shown in(PP1), we present a proposed technique for second smooth penalty function for solving the constrained optimization problem. The functionality of method is represented as follow:

Algorithm II
Step 1: First choose the initial guess of basic feasible solution x^0. Take $\varepsilon > 0$, $\varepsilon_0 > 0$, $\rho_0 > 0$, $0 < \delta < 1$. The multiplier value for the penalty parameter is taken to be greater than 1 denoted by N. Supoose $j = 0$ and then go to the next step.
Step 2: At next step, we arrive at x^j and by taking x^j as initial point again, evaluate $\min_{x \in R^n} F(x,\rho_j,\varepsilon_j)$. Let x^{j+1} considered to be the optimal solution attained.
Step 3: If x^{j+1} is supposed to be ε-feasible to (P), then stop the simulation. Otherwise, we can consider "$\rho_{j+1} = N\rho_j$, $\varepsilon_{j+1} = \delta\varepsilon_j$ and $j = j + 1$", then follow the procedure as given in step 2.

III. NUMERICAL EXAMPLES

Now, we will solve some constrained optimization problems with Algorithm I and II on Mathematica. In order to compare the efficiency of Algorithm I and II with those both of Algorithm III based on the l_1 exact penalty function and of Algorithm IV based on the l_2 penalty function, Algorithms III and IV are listed as follows:

Algorithm III
Step 1: Choose $x^0, \varepsilon > 0, \rho_0 > 0, N > 1$. Let $j = 0$ and go to Step 2 .
Step 2: Using x^j as the starting point to solve $\min_{x \in R^n} F_1(x, \rho_j) = f(x) + \rho_j \sum_{i \in I} \max\{g_i(x), 0\}$. Let x^{j+1} be the optimal solution obtained.
Step 3: If x^{j+1} is ε-feasible to (P), then stop. Otherwise, let $\rho_{j+1} = N\rho_j$ and $j = j+1$, then go to Step 2 .
Algorithm IV
Step 1: Choose $x^0, \varepsilon > 0, \rho_0 > 0, N > 1$. Let $j = 0$ and go to Step 2 .
Step 2: Using x^j as the starting point to solve $\min_{x \in R^n} F_2(x, \rho_j) = f(x) + \rho_j \sum_{i \in I} [\max\{g_i(x), 0\}]^2$. Let x^{j+1} be the optimal solution obtained.
Step 3: If x^{j+1} is ε-feasible to (P), then stop. Otherwise, let $\rho_{j+1} = N\rho_j$ and $j = j + 1$, then go to Step 2 . We resolve some numerical instances with the help of suggested algorithm using Mathematica software.
Example 1. *The Rosen-Suzen problem "in [12]*

$$\min f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$
$$s.t.\ g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_4 - 5 \le 0$$
$$g_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0$$
$$g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0"$$

We solve this equation using Mathematica software. Let $x^0 = (0, 0, 0, 0)$ We take initial value of penalty parameter $\rho_0 = 10$, $\varepsilon_0 = 0.3$, $\lambda = 0.1$ and $N = 3$ for Algorithm I and $x^0 = (0, 0, 0, 0)$, initial value of penalty parameter $\rho_0 = 3$, $\varepsilon_0 = 0.2$, $\lambda = 0.1$ and $N = 3$ for Algorithm II.

TABLE I. Numerical results using Algorithm I

No. iter. k	x^{k+1}	ρ_k	ε_k	$f(x^{k+1})$
1	(0.175210, 0.839826, 2.025449, -0.977656)	10	0.3	-44.4496
2	(0.169887, 0.835779, 2.009614, -0.965623)	30	0.03	-44.2340
3	(0.169578, 0.835545, 2.008691, -0.964919)	90	0.003	-44.2338
4	(0.169561, 0.835533, 2.008630, -0.964878)	270	0.0003	-44.2338

TABLE II. Numerical results using Algorithm II

No. iter. k	x^{k+1}	σ_k	ε_k	$f(x^{k+1})$
1	(0.169255,0.834042,2.012210,-0.972317)	3	0.2	-44.2534
2	(0.169480,0.835149,2.00954,-0.966767)	9	0.02	-44.2339
3	(0.169559,0.835532,2.00863,-0.964877)	27	0.002	-44.2338

TABLE III. Numerical results using Algorithm III

No. iter. k	x^{k+1}	σ_k	$f(x^{k+1})$
1	(0.339654,0.677748,2.240736,-1.231420)	1	-48.629509
3	(0.171993,0.831486,2.009344,-0.963467)	4	-44.233741

TABLE IV. Numerical results using Algorithm IV

No. iter. k	x^{k+1}	σ_k	$f(x^{k+1})$
1	((0.339654,0.677748,2.240736,-1.231420))	1	-48.629509
23	(0.169555,0.835503,2.008651,-0.964856)	4194304	-44.233837

Example 2. *Consider the following example given in [13]*

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

We take initial point as $x^0 = (0, 0, 0)$ with $\rho_0 = 10$, the value of $N = 10$, $\varepsilon_0 = 0.1$, $\delta = 0.01$ and $\varepsilon = 10^{-6}$. The calculations using Mathematica are given in table 5, 6, 7 and 8.

TABLE V. Numerical results using Algorithm I

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.510168, 4.227381, 0.967761)	10	944.097939
2	(2.501018, 4.221964, 0.964756)	100	944.203874
3	(2.500101, 4.221383, 0.964623)	1000	944.214474
4	(2.500010, 4.221324, 0.964610)	10000	944.215534

TABLE VI. Numerical results using Algorithm II

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.504965,4.225815,0.966155)	10	944.070969
2	(2.500013,4.221979,0.961797)	100	944.215173
3	(2.500000,4.221953,0.961829)	1000	944.215662
4	(2.500000,4.221953,0.961829)	10000	944.215661

TABLE VII. Numerical results using Algorithm III

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.506435,3.672177,2.301453)	10	946.478819
2	(2.500000,3.685055,2.273845)	100	946.523123

TABLE VIII. Numerical results using Algorithm IV

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.510169,4.227378,0.967778)	10	943.980275
7	(2.500000,4.221318,0.964609)	10000000	944.215652

Example 3. *Consider the floowing probelm given in [14]*

$$\begin{aligned} &\min f(x,y) = -x - y, \\ &s.t.\ g_1(x,y) = y - 2x^4 + 8x^3 - 8x^2 - 2 \le 0 \\ &\quad g_2(x,y) = y - 4x^4 + 32x^3 - 88x^2 + 96x - 36 \\ &\qquad 0 \le x \le 3 \\ &\qquad 0 \le y \le 4. \end{aligned}$$

TABLE IX. Numerical results using Algorithm I

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.072463, 4.018562)	5	-6.091026
2	(2.003826, 4.002522)	15	-6.006348
3	(2.000211, 4.000148)	45	-6.000360

TABLE X. Numerical results using Algorithm II

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.084564, 4.198454)	5	-6.28302
2	(2.033748, 4.194743)	15	-6.22849
3	(2.028747, 4.003435)	45	-6.03218
4	(2.000132,4.000153)	135	-6.00029

In above exmple, we take the intial point $x^0 = (0, 0)$ with value of initial parameter $\rho_0 = 5$ and multiplier $N = 3$. The value of ε_0 is taken as 0.1 and the value of multiplier $\delta = 0.1$.

Algorithm I seems to converge relatively quickly, reaching a small ε_k. Algorithm II shows a similar trend with decreasing ε_k and $f(x^{k+1})$ values with increasing iterations. Algorithm III and IV have comparably large iterations but reach comparable ε_k and $f(x^{k+1})$ values.

If prioritizing rapid convergence is imperative, Algorithms I and II emerge as more favorable options. In the event that stability is of paramount importance, it is noteworthy that Algorithm III demonstrates a rapid stabilization;

TABLE XI. Numerical results using Algorithm III

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.348483, 4.199243)	5	-6.54773
2	(2.002342,4.010239)	500	-6.01258

TABLE XII. Numerical results using Algorithm IV

k	x^{k+1}	ρ_k	$f(x^{k+1})$
1	(2.647483, 4.485243)	5	-7.13273
7	(2.000394,4.002021)	500000	-6.00242

however, it may not attain the minimal objective function value. Algorithm IV could be deemed suitable in cases where an acceptable compromise between convergence speed and the attainment of the ultimate objective function value is permissible.

IV. CONCLUSION

In summary, the comparative analysis of the four algorithms elucidates distinct convergence behaviors and perfor- mance attributes. Algorithms I and II demonstrate an accelerated convergence rate, rendering them advantageous in scenarios prioritizing expeditious optimization. Algorithm III, while exhibiting prompt stabilization, may not attain the minimal objective function value. Algorithm IV, offering a compromise between convergence speed and the final objective function value, is a viable option when a balanced approach is deemed acceptable. The selection of the most appropriate algorithm hinges upon the specific requirements and priorities inherent to the optimization problem. Ad- ditional considerations, including computational efficiency and robustness, play a pivotal role in making an informed decision tailored to the unique characteristics of the optimization task.

REFERENCES

[1] W. I. Zangwill, "Non-linear programming via penalty functions," Management science 13, 344–358 (1967).

[2] X. Yang, Z. Meng, X. Huang, and G. Pong, "Smoothing nonlinear penalty functions for constrained optimization problems," (2003).

[3] S.-P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," Mathematical programming 17, 251–269 (1979).

[4] M. Ç. Pinar and S. A. Zenios, "On smoothing

exact penalty functions for convex constrained optimization," SIAM Journal on Optimization 4, 486–511 (1994).

[5] E. Rosenberg, "Globally convergent algorithms for convex programming," Mathematics of Operations Research 6, 437–444 (1981).

[6] S. A. Zenios, M. C. Pinar, and R. S. Dembo, "A smooth penalty function algorithm for network-structured problems," European Journal of Operational Research 83, 220–236 (1995).

[7] A. V. Fiacco and G. P. McCormick, Nonlinear programming: sequential unconstrained minimization techniques (SIAM, 1990).

[8] R. Fletcher, "A model algorithm for composite nondifferentiable optimization problems," in Nondifferential and Variational Techniques in Optimization (Springer, 1982) pp. 67–76.

[9] R. Fletcher, "An exact penalty function for nonlinear programming with inequalities," Mathematical Programming 5, 129–150 (1973).

[10] D. Mauricio and N. Maculan, "A boolean penalty method for zero-one nonlinear programming," Journal of Global Optimization 16, 343–354 (2000).

[11] Z. Meng, Q. Hu, C. Dang, and X. Yang, "An objective penalty function method for nonlinear programming," Applied mathematics letters 17, 683–689 (2004).

[12] Z. Meng, C. Dang, M. Jiang, and R. Shen, "A smoothing objective penalty function algorithm for inequality constrained optimization problems," Numerical functional analysis and optimization 32, 806–820 (2011).

[13] J. Lasserre, "A globally convergent algorithm for exact penalty functions," European Journal of Operational Research 7, 389–395 (1981).

[14] S. Lian and Y. Duan, "Smoothing of the lower-order exact penalty function for inequality constrained optimization," Journal of Inequalities and Applications 2016, 1–12 (2016).