

Research Article

Smoothing Approximation to the Square-Order Exact Penalty Functions for Constrained Optimization

Shujun Lian and Jinli Han

College of Operations and Management, Qufu Normal University, Rizhao, Shandong 276826, China

Correspondence should be addressed to Shujun Lian; lsjsd2003@126.com

Received 13 May 2013; Accepted 18 September 2013

Academic Editor: Zhihua Zhang

Copyright © 2013 S. Lian and J. Han. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A method is proposed to smooth the square-order exact penalty function for inequality constrained optimization. It is shown that, under some conditions, an approximately optimal solution of the original problem can be obtained by searching an approximately optimal solution of the smoothed penalty problem. An algorithm based on the smoothed penalty functions is given. The algorithm is shown to be convergent under mild conditions. Two numerical examples show that the algorithm seems efficient.

1. Introduction

Consider the following nonlinear constrained optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ [P] \quad & \text{s.t. } g_i(x) \leq 0, \quad i = 1, 2, \dots, m, \\ & x \in R^n, \end{aligned} \quad (1)$$

where $f: R^n \rightarrow R$ and $g_i(x): R^n \rightarrow R, i \in I = \{1, 2, \dots, m\}$ are twice continuously differentiable functions. Let

$$G_0 = \{x \in R^n \mid g_i(x) \leq 0, i = 1, 2, \dots, m\}. \quad (2)$$

To solve [P], many penalty function methods have been proposed in numerous pieces of literature. One of the popular penalty functions is given by

$$F(x, q) = f(x) + q \sum_{i=1}^m (g_i^+(x))^2, \quad (3)$$

where $g_i^+(x) = \max\{0, g_i(x)\}, i = 1, 2, \dots, m$. Obviously, it is a continuously differentiable function, but it is not an exact penalty function. If each minimum of the penalty problem is a minimum of the original problem or each minimum of the original problem is a minimum of the penalty problem when the penalty parameter q is large enough, the corresponding penalty function is called exact penalty function.

In Zangwill [1], the classical l_1 exact penalty function is defined as follows:

$$f(x, q) = f(x) + q \sum_{i=1}^m g_i^+(x). \quad (4)$$

After Zangwill's development, exact penalty functions have attracted most of the attention (see, e.g., [2–6]). It is known from the theory of ordinary constrained optimization that the l_1 penalty function is a better candidate for penalization. However, it is not a smooth function and causes some numerical instability problems in its implementation when the value of the penalty parameter q becomes larger. Some methods for smoothing the exact penalty function are developed (see, e.g., [7–14]).

In [15, 16], the square-order penalty function

$$\varphi_q(x) = f(x) + q \sum_{i=1}^m \sqrt{g_i^+(x)} \quad (5)$$

has been introduced and investigated. The penalty function $\varphi_q(x)$ is exact but not smooth. Its smoothing has been investigated in [15, 16]. So, it can be applied to solve the problem [P] via a gradient-type or a Newton-type method.

In this paper, a new smoothing function to the square-order penalty function of the form (5) is investigated. The rest of this paper is organized as follows. In Section 2, a new smoothing function to the square-order penalty function

is introduced, and some fundamental properties of the smoothing function are discussed. In Section 3, an algorithm is presented to compute an approximate solution to $[P]$ based on the smooth penalty function and is shown to be convergent. In Section 4, two numerical examples are given to show the applicability of the algorithm. In Section 5, we conclude the paper with some remarks.

2. Smoothing Exact Lower Order Penalty Function

Consider the following lower order penalty problem:

$$[LOP] \quad \min_{x \in R^n} \varphi_q(x). \quad (6)$$

In this paper, we say that the pair (x^*, λ^*) satisfies the KKT condition if

$$\nabla f(x^*) = -\sum_{i \in I} \lambda_i^* \nabla g_i(x^*), \quad (7)$$

$$\lambda_i^* g_i(x^*) = 0, \quad \lambda_i^* \geq 0, \quad g_i(x^*) \leq 0, \quad i \in I$$

and that the pair (x^*, λ^*) satisfies the second-order sufficiency condition [17, page 169] if

$$\begin{aligned} \nabla_x L(x^*, \lambda^*) &= 0, \\ g_i(x^*) &\leq 0, \quad i \in I, \\ \lambda_i^* &\geq 0, \quad i \in I, \end{aligned} \quad (8)$$

$$\lambda_i^* g_i(x^*) = 0, \quad i \in I,$$

$$y^T \nabla^2 L(x^*, \lambda^*) y > 0, \quad \text{for any } y \in V(x^*),$$

where $L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i g_i(x)$ and

$$V(x^*) = \{y \in R^n \mid \nabla^T g_i(x^*) y = 0, \quad i \in A(x^*),$$

$$\nabla^T g_i(x^*) y \leq 0, \quad i \in B(x^*)\}, \quad (9)$$

$$A(x^*) = \{i \in I \mid g_i(x^*) = 0, \quad \lambda_i^* > 0\},$$

$$B(x^*) = \{i \in I \mid g_i(x^*) = 0, \quad \lambda_i^* = 0\}.$$

In order to establish the exact penalization, we need the following assumptions.

Assumption 1. $f(x)$ satisfies the following coercive condition:

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty. \quad (10)$$

Under Assumption 1, there exists a box X such that $G([P]) \subset \text{int}(X)$, where $G([P])$ is the set of global minima of problem $[P]$ and $\text{int}(X)$ denotes the interior of the set X . Consider the following problem:

$$\begin{aligned} \min \quad & f(x) \\ [P'] \quad \text{s.t.} \quad & g_i(x) \leq 0, \quad i = 1, \dots, m, \\ & x \in X. \end{aligned} \quad (11)$$

Let $G([P'])$ denote the set of global minima of problem $[P']$. Then $G([P']) = G([P])$.

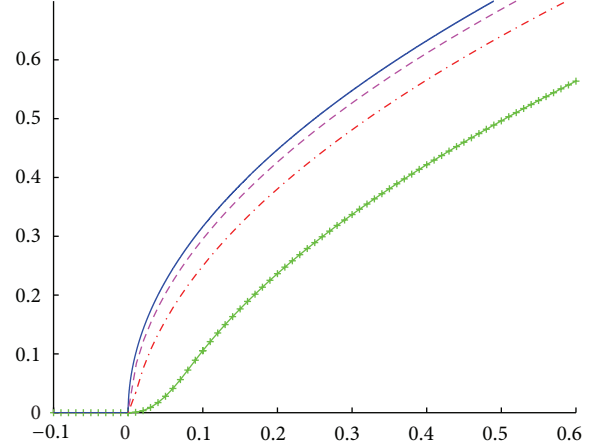


FIGURE 1: The behavior of $p_\epsilon(u)$ and $p(u)$.

Assumption 2. The set $G([P])$ is a finite set.

Then we consider the penalty problem of the form

$$[LOP'] \quad \min_{x \in X} \varphi_q(x). \quad (12)$$

Let $p(u) = (\max\{0, u\})^{1/2}$; that is,

$$p(u) = \begin{cases} u^{1/2} & \text{if } u > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (13)$$

then

$$\varphi_q(x) = f(x) + q \sum_{i=1}^m p(g_i(x)). \quad (14)$$

For any $\epsilon > 0$, let

$$p_\epsilon(u) = \begin{cases} 0 & \text{if } u \leq 0, \\ \frac{2}{3}\epsilon^{-2}u^{5/2} - \frac{1}{3}\epsilon^{-3}u^{7/2} & \text{if } 0 < u \leq \epsilon, \\ u^{1/2} - \frac{2}{3}\epsilon^{1/2} & \text{if } u > \epsilon. \end{cases} \quad (15)$$

It follows that

$$p'_\epsilon(u) = \begin{cases} 0 & \text{if } u \leq 0, \\ \frac{5}{3}\epsilon^{-2}u^{3/2} - \frac{7}{6}\epsilon^{-3}u^{5/2} & \text{if } 0 < u \leq \epsilon, \\ \frac{1}{2}u^{-1/2} & \text{if } u > \epsilon. \end{cases} \quad (16)$$

It is easy to see that $p_\epsilon(u)$ is continuously differentiable on R . Furthermore, we can obtain that $p_\epsilon(u) \rightarrow p(u)$ as $\epsilon \rightarrow 0$.

Figure 1 shows the behavior of $p(u)$ (represented by the real line), $p_{0.1}(u)$ (represented by the real line with plus sign), $p_{0.01}(u)$ (represented by the dash and dot line), and $p_{0.001}(u)$ (represented by broken line).

Let

$$\varphi_{q,\epsilon}(x) = f(x) + q \sum_{i=1}^m p_\epsilon(g_i(x)). \quad (17)$$

Then $\varphi_{q,\epsilon}(x)$ is continuously differentiable on R^n . Consider the following smoothed optimization problem:

$$[SP] \quad \min_{x \in X} \varphi_{q,\epsilon}(x). \quad (18)$$

Lemma 3. For any $x \in X$, $\epsilon > 0$,

$$-\frac{2}{3}mq\epsilon^{1/2} \leq \varphi_{q,\epsilon}(x) - \varphi_q(x) \leq \frac{4}{3}mq\epsilon^{1/2}. \quad (19)$$

Proof. Note that

$$p(g_i(x)) - p_\epsilon(g_i(x)) = \begin{cases} 0 & \text{if } g_i(x) \leq 0, \\ (g_i(x))^{1/2} - \frac{2}{3}\epsilon^{-2}(g_i(x))^{5/2} \\ \quad + \frac{1}{3}\epsilon^{-3}(g_i(x))^{7/2} & \text{if } 0 < g_i(x) \leq \epsilon, \\ \frac{2}{3}\epsilon^{1/2} & \text{if } g_i(x) > \epsilon. \end{cases} \quad (20)$$

When $g_i(x) \in (0, \epsilon]$, we have

$$-\frac{2}{3}\epsilon^{1/2} \leq p(g_i(x)) - p_\epsilon(g_i(x)) \leq \frac{4}{3}\epsilon^{1/2}. \quad (21)$$

Then

$$-\frac{2}{3}mq\epsilon^{1/2} \leq \varphi_{q,\epsilon}(x) - \varphi_q(x) \leq \frac{4}{3}mq\epsilon^{1/2}. \quad (22)$$

□

As a direct result of Lemma 3, we have the following result.

Theorem 4. Let $\{\epsilon_j\} \rightarrow 0$ be a sequence of positive numbers, and assume that x^j is a solution to $\min_{x \in X} \varphi_{q,\epsilon_j}(x)$ for some $q > 0$. Let \bar{x} be an accumulating point of the sequence $\{x^j\}$. Then \bar{x} is an optimal solution to $\min_{x \in X} \varphi_q(x)$.

Proof. Because x^j is a solution to $\min_{x \in X} \varphi_{q,\epsilon_j}(x)$, we have

$$\varphi_{q,\epsilon_j}(x^j) \leq \varphi_{q,\epsilon_j}(x). \quad (23)$$

By Lemma 3, we have

$$\begin{aligned} \varphi_q(x^j) &\leq \varphi_{q,\epsilon_j}(x^j) + \frac{2}{3}mq\epsilon_j^{1/2}, \\ \varphi_{q,\epsilon_j}(x) &\leq \varphi_q(x) + \frac{4}{3}mq\epsilon_j^{1/2}. \end{aligned} \quad (24)$$

It follows that

$$\begin{aligned} \varphi_q(x^j) &\leq \varphi_{q,\epsilon_j}(x^j) + \frac{2}{3}mq\epsilon_j^{1/2} \\ &\leq \varphi_{q,\epsilon_j}(x) + \frac{2}{3}mq\epsilon_j^{1/2} \\ &\leq \varphi_q(x) + \frac{4}{3}mq\epsilon_j^{1/2} + \frac{2}{3}mq\epsilon_j^{1/2} \\ &= \varphi_q(x) + 2mq\epsilon_j^{1/2}. \end{aligned} \quad (25)$$

Let $j \rightarrow 0$; we have

$$\varphi_q(\bar{x}) \leq \varphi_q(x). \quad (26)$$

We complete the proof. □

Theorem 5. Let $x_q^* \in X$ be an optimal solution of problem $[LOP']$ and $\bar{x}_{q,\epsilon} \in X$ an optimal solution of problem $[SP]$ for some $q > 0$ and $\epsilon > 0$. Then

$$-\frac{2}{3}mq\epsilon^{1/2} \leq \varphi_q(x_q^*) - \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) \leq \frac{4}{3}mq\epsilon^{1/2}. \quad (27)$$

If both x_q^* and $\bar{x}_{q,\epsilon}$ are feasible, then

$$0 \leq f(\bar{x}_{q,\epsilon}) - f(x_q^*) \leq \frac{2}{3}mq\epsilon^{1/2}. \quad (28)$$

Proof. By Lemma 3, we have

$$\begin{aligned} -\frac{2}{3}mq\epsilon^{1/2} &\leq \varphi_q(x_q^*) - \varphi_{q,\epsilon}(x_q^*) \\ &\leq \varphi_q(x_q^*) - \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) \\ &\leq \varphi_q(\bar{x}_{q,\epsilon}) - \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) \\ &\leq \frac{4}{3}mq\epsilon^{1/2}. \end{aligned} \quad (29)$$

Specially, if both x_q^* and $\bar{x}_{q,\epsilon}$ are feasible, we have

$$f(x_q^*) \leq f(\bar{x}_{q,\epsilon}) \quad (30)$$

by $\varphi_q(x_q^*) \leq \varphi_q(\bar{x}_{q,\epsilon})$.

It follows that

$$0 \leq f(\bar{x}_{q,\epsilon}) - f(x_q^*). \quad (31)$$

On the other hand, by (14), (15), (17), and (19), we have

$$f(\bar{x}_{q,\epsilon}) - f(x_q^*) = \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) - \varphi_q(x_q^*) \leq \frac{2}{3}mq\epsilon^{1/2}. \quad (32)$$

We complete the proof. □

Theorem 6. Supposing that Assumptions 1 and 2 hold, and that, for any $x^* \in G([P])$, there exists a $\lambda \in R_+^m$ such that the pair (x^*, λ^*) satisfies the second-order sufficiency condition (8). Let $x^* \in X$ be a global solution of problem $[P]$ and $\bar{x}_{q,\epsilon} \in X$ a global solution of problem $[SP]$ for $\epsilon > 0$. Then there exists $q^* > 0$ such that for any $q > q^*$,

$$-\frac{2}{3}mq\epsilon^{1/2} \leq f(x^*) - \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) \leq \frac{4}{3}mq\epsilon^{1/2}, \quad (33)$$

where q^* is defined in Corollary 2.3 in [16].

Proof. By Corollary 2.3 in [16], we have that x^* is a global solution of problem $[LOP']$. Then, by Theorem 5, we have

$$-\frac{2}{3}mq\epsilon^{1/2} \leq \varphi_q(x_q^*) - \varphi_{q,\epsilon}(\bar{x}_{q,\epsilon}) \leq \frac{4}{3}mq\epsilon^{1/2}. \quad (34)$$

Since $\sum_{i=1}^m p(g_i(x^*)) = 0$, we have

$$\varphi_q(x^*) = f(x^*) + q \sum_{i=1}^m p(g_i(x^*)) = f(x^*). \quad (35)$$

We complete the proof. □

TABLE 1: Numerical results for Example 1 by Algorithm 7.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $f(x_j^*)$ |
|-----|--|-------|--------------|--------------|--------------|------------|
| 0 | $\begin{pmatrix} 0.7811047 \\ 1.057024 \end{pmatrix}$ | 5 | 0.1 | 0.0430043 | -2.904718 | 3.333603 |
| 1 | $\begin{pmatrix} 0.7260887 \\ 0.3992826 \end{pmatrix}$ | 50 | 0.05 | -0.7777236 | 0.000935 | 1.836004 |
| 2 | $\begin{pmatrix} 0.7244794 \\ 0.3991450 \end{pmatrix}$ | 500 | 0.025 | -0.7737306 | -0.000683 | 1.838842 |
| 3 | $\begin{pmatrix} 0.7245065 \\ 0.3990242 \end{pmatrix}$ | 5000 | 0.0175 | -0.7738962 | -0.000016 | 1.837684 |

Theorems 4 and 5 mean that an approximate solution to [SP] is also an approximate solution to [LOP]. Furthermore, by Theorem 6, an optimal solution to [SP] is an approximately optimal solution to [P]. Now we present a penalty function algorithm to solve [P].

3. A Smoothing Method

We propose the following algorithm to solve [P].

Algorithm 7. Consider the following.

Step 1. Choose a point x^0 . Given $\epsilon_0 > 0$, $q_0 > 0$, $0 < \eta < 1$, and $N > 1$, let $j = 0$, and go to Step 2.

Step 2. Use x^j as the starting point to solve $\min_{x \in R^n} \varphi_{q_j, \epsilon_j}(x)$. Let x_j^* be the optimal solution obtained (x_j^* is obtained by a quasi-Newton method and a finite difference gradient). Go to Step 3.

Step 3. Let $q_{j+1} = Nq_j$, $\epsilon_{j+1} = \eta\epsilon_j$, $x^{j+1} = x_j^*$ and $j = j + 1$; then go to Step 2.

Remark 8. From $0 < \eta < 1$ and $N > 1$, we can easily obtain that the sequence $\{\epsilon_j\}$ is decreasing to 0 and the sequence $\{q_j\}$ is increasing to $+\infty$ as $j \rightarrow +\infty$.

Now we prove the convergence of the algorithm under mild conditions.

Theorem 9. Suppose that, for any $q \in [q_0, +\infty)$, $\epsilon \in (0, \epsilon_0]$, the set

$$\arg \min_{x \in R^n} \varphi_{q, \epsilon}(x) \neq \emptyset. \quad (36)$$

Let $\{x_j^*\}$ be the sequence generated by Algorithm 7. If $\{x_j^*\}$ has limit point, then any limit point of $\{x_j^*\}$ is the solution of [P].

Proof. Let \bar{x} be any limit point of $\{x_j^*\}$. Then there exists a natural number set $J \in N$, such that $x_j^* \rightarrow \bar{x}$, $j \in J$. If we can prove that (i) $\bar{x} \in G_0$ and (ii) $f(\bar{x}) \leq \inf_{x \in G_0} f(x)$ hold, then \bar{x} is the optimal solution of [P].

(i) Suppose, to the contrary, that $\bar{x} \notin G_0$; then there exist $\delta_0 > 0$, $i_0 \in I$, and the subset $J_1 \subset J$ such that

$$g_{i_0}(x_j^*) \geq \delta_0 \quad (37)$$

for any $j \in J_1$.

If $\epsilon_j \geq g_{i_0}(x_j^*) \geq \delta_0$, it follows from Step 2 in Algorithm 7 and (15) that

$$\begin{aligned} f(x_j^*) + \frac{2}{3}q_j\epsilon_j^{-2}\delta_0^{5/2} - \frac{1}{3}q_j\epsilon_j^{-3}\delta_0^{7/2} \\ \leq \varphi_{q_j, \epsilon_j}(x_j^*) \leq \varphi_{q_j, \epsilon_j}(x) = f(x) \end{aligned} \quad (38)$$

for any $x \in G_0$, which contradicts with $\epsilon_j \rightarrow 0$ and $q_j \rightarrow +\infty$.

If $g_{i_0}(x_j^*) \geq \delta_0 > \epsilon_j$ or $g_{i_0}(x_j^*) > \epsilon_j \geq \delta_0$, it follows from Step 2 in Algorithm 7 and (15) that

$$f(x_j^*) + q_j \left(\delta_0^{1/2} - \frac{2}{3}\epsilon_j^{1/2} \right) \leq \varphi_{q_j, \epsilon_j}(x_j^*) \leq \varphi_{q_j, \epsilon_j}(x) = f(x) \quad (39)$$

for any $x \in G_0$, which contradicts with $\epsilon_j \rightarrow 0$ and $q_j \rightarrow +\infty$.

Then we have $\bar{x} \in G_0$.

(ii) For any $x \in G_0$, it holds that

$$f(x_j^*) \leq \varphi_{q_j, \epsilon_j}(x_j^*) \leq \varphi_{q_j, \epsilon_j}(x) = f(x); \quad (40)$$

then $f(\bar{x}) \leq \inf_{x \in G_0} f(x)$ holds.

This completes the proof. \square

4. Numerical Examples

In this section, we solve two numerical examples to show the applicability of Algorithm 7 on Fortran.

Example 1 (see [18, Example 4.1]). We can see the following:

$$\begin{aligned} \min \quad & f(x) = x_1^2 + x_2^2 - \cos(17x_1) - \cos(17x_2) + 3 \\ \text{s.t.} \quad & g_1(x) = (x_1 - 2)^2 + x_2^2 - 1.6^2 \leq 0 \\ & g_2(x) = x_1^2 + (x_2 - 3)^2 - 2.7^2 \leq 0 \\ & 0 \leq x_1 \leq 2 \\ & 0 \leq x_2 \leq 2. \end{aligned} \quad (41)$$

Starting point $x^0 = (0, 0)$, $q_0 = 5.0$, $\epsilon_0 = 0.1$, $\eta = 0.5$, and $N = 10$, we obtain the results by Algorithm 7 shown in Table 1.

Furthermore, the algorithms based on the penalty function (3) or the exact penalty function (4) are described as follows.

TABLE 2: Numerical results for Example 1 by Algorithm 10.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $f(x_j^*)$ |
|-----|--|------------------|--------------|--------------|--------------|------------|
| 0 | $\begin{pmatrix} 0.003902 \\ 0.006869 \end{pmatrix}$ | 0.1 | 10^{-1} | 1.424453 | 1.668849 | 1.009072 |
| 1 | $\begin{pmatrix} 0.3739730 \\ 0.7309830 \end{pmatrix}$ | 0.1×5 | 10^{-2} | 0.6182998 | -2.001706 | 1.686692 |
| 2 | $\begin{pmatrix} 0.3971605 \\ 0.7207292 \end{pmatrix}$ | 0.1×5^2 | 10^{-3} | 0.5285448 | -1.937189 | 1.833846 |
| 3 | $\begin{pmatrix} 0.7277128 \\ 0.3901970 \end{pmatrix}$ | 0.1×5^3 | 10^{-4} | -0.7890318 | 0.050637 | 1.761504 |
| 4 | $\begin{pmatrix} 0.7259367 \\ 0.3969422 \end{pmatrix}$ | 0.1×5^4 | 10^{-5} | -0.7791997 | 0.012894 | 1.815956 |
| 5 | $\begin{pmatrix} 0.7254778 \\ 0.3987666 \end{pmatrix}$ | 0.1×5^5 | 10^{-6} | -0.7765785 | 0.002733 | 1.832846 |
| 6 | $\begin{pmatrix} 0.7271451 \\ 0.3997456 \end{pmatrix}$ | 0.1×5^6 | 10^{-7} | -0.7800440 | 0.000062 | 1.837930 |
| 7 | $\begin{pmatrix} 0.7271337 \\ 0.3997256 \end{pmatrix}$ | 0.1×5^7 | 10^{-8} | -0.7800310 | 0.000150 | 1.837770 |
| 8 | $\begin{pmatrix} 0.7260696 \\ 0.3994698 \end{pmatrix}$ | 0.1×5^8 | 10^{-9} | -0.7775255 | -0.000066 | 1.837740 |

TABLE 3: Numerical results for Example 1 by Algorithm 11.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $f(x_j^*)$ |
|-----|--|-------|--------------|--------------|--------------|------------|
| 0 | $\begin{pmatrix} 0.001377 \\ 0.002065 \end{pmatrix}$ | 0.1 | 10^{-1} | 1.434500 | 1.697615 | 1.000896 |
| 1 | $\begin{pmatrix} 0.7513676 \\ 1.085665 \end{pmatrix}$ | 2 | 10^{-2} | 0.1777513 | -3.060769 | 2.840876 |
| 2 | $\begin{pmatrix} 0.7188042 \\ 0.3974435 \end{pmatrix}$ | 40 | 10^{-3} | -0.7605761 | -0.000021 | 1.844097 |

Algorithm 10. Consider the following.

Step 1. Choose a point x^0 , and a stopping tolerance $\epsilon > 0$. Given $\epsilon_0 > 0$, $q_0 > 0$, $0 < \eta < 1$, and $N > 1$, let $j = 0$, and go to Step 2.

Step 2. Use x^j as the starting point to solve $\min_{x \in R^n} F(x, q_j)$. Let x_j^* be the optimal solution obtained (x_j^* is obtained by a quasi-Newton method and a finite difference gradient). Go to Step 3.

Step 3. Let $q_{j+1} = Nq_j$, $\epsilon_{j+1} = \eta\epsilon_j$, $x^{j+1} = x_j^*$, and $j = j + 1$; then go to Step 2.

Algorithm 11. Consider the following.

Step 1. Choose a point x^0 and a stopping tolerance $\epsilon > 0$. Given $\epsilon_0 > 0$, $q_0 > 0$, $0 < \eta < 1$, and $N > 1$, let $j = 0$, and go to Step 2.

Step 2. Use x^j as the starting point to solve $\min_{x \in R^n} f(x, q_j)$. Let x_j^* be the optimal solution obtained (x_j^* is obtained by

a quasi-Newton method and a finite difference gradient). Go to Step 3.

Step 3. Let $q_{j+1} = Nq_j$, $\epsilon_{j+1} = \eta\epsilon_j$, $x^{j+1} = x_j^*$, and $j = j + 1$; then go to Step 2.

Let $x^0 = (0, 0)$, $q_0 = 0.1$, $\epsilon_0 = 0.1$, $\eta = 0.1$, and $N = 5$; numerical results by Algorithm 10 are shown in Table 2.

Let $x^0 = (0, 0)$, $q_0 = 0.1$, $\epsilon_0 = 0.1$, $\eta = 0.1$, and $N = 20$; numerical results by Algorithm 11 are shown in Table 3.

This example is a nonconvex problem with 22 local optimal solutions in the interior of the feasible region. By Sun and Li [18], we know that $x^* = (0.7255, 0.3993)$ is a global minimum with global optimal value $f^* = 1.8376$. It is clear from Table 1 that the obtained approximately optimal solution is $x^* = (0.7245065, 0.3990242)$ with corresponding objective function value 1.837684.

From Tables 1–3, one can see that Algorithm 11 converges faster than Algorithms 7 and 10, but the solution generated by Algorithm 11 is the worst. Algorithm 10 is the slowest one, and the solution generated by Algorithm 10 is worse than the solution generated by Algorithm 7.

TABLE 4: Numerical results for Example 2 by Algorithm 7.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $g_3(x_j^*)$ | $f(x_j^*)$ |
|-----|---|-------|--------------|--------------|--------------|--------------|------------|
| 0 | (1.139751, 1.272704, 3.819159, 1.997231) | 2 | 1.0 | 15.35879 | 19.17715 | 17.95990 | -70.16554 |
| 1 | (0.1705428, 0.8361722, 2.011177, -0.9678533) | 4 | 10^{-1} | 0.0115902 | 0.0232417 | -1.856925 | -44.28859 |
| 2 | (0.1335792, 0.8091212, 1.995527, -1.008688) | 8 | 10^{-2} | -0.2599155 | 0.0007742 | -1.780662 | -44.02616 |
| 3 | (0.1585001, 0.8339736, 2.014753, -0.959688) | 16 | 10^{-3} | -0.00367924 | -0.0003119 | -1.881673 | -44.22965 |

TABLE 5: Numerical results for Example 2 by Algorithm 10.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $g_3(x_j^*)$ | $f(x_j^*)$ |
|-----|---|------------------|--------------|--------------|--------------|--------------|------------|
| 0 | (0.4528408, 0.8017877, 2.580122, -1.214387) | 0.1 | 1.0 | 3.203103 | 4.425251 | 1.858834 | -53.31968 |
| 1 | (0.2167079, 0.8595678, 2.192063, -1.159403) | 0.1×5 | 10^{-1} | 0.7715020 | 1.643783 | -0.039055 | -47.79021 |
| 2 | (0.1782670, 0.8383819, 2.051085, -1.015255) | 0.1×5^2 | 10^{-2} | 0.1530515 | 0.3785772 | -1.457033 | -45.08350 |
| 3 | (0.1712600, 0.8356738, 2.017540, -0.9755791) | 0.1×5^3 | 10^{-3} | 0.030094 | 0.078609 | -1.795672 | -44.41169 |
| 4 | (0.1696707, 0.8354126, 2.010606, -0.9668932) | 0.1×5^4 | 10^{-4} | 0.005887 | 0.015878 | -1.865860 | -44.26973 |
| 5 | (0.1836584, 0.8497245, 1.993892, -0.9809219) | 0.1×5^5 | 10^{-5} | 0.001217 | 0.002323 | -1.824922 | -44.23592 |
| 6 | (0.1835818, 0.8496645, 1.993588, -0.9808556) | 0.1×5^6 | 10^{-6} | -0.000298 | 0.000466 | -1.826614 | -44.23108 |
| 7 | (0.1843284, 0.8502323, 1.992886, -0.9814753) | 0.1×5^7 | 10^{-7} | -0.000141 | 0.000219 | -1.824903 | -44.23038 |
| 8 | (0.1843219, 0.8502275, 1.992824, -0.9814662) | 0.1×5^8 | 10^{-8} | -0.000412 | -0.000131 | -1.825209 | -44.22948 |

TABLE 6: Numerical results for Example 2 by Algorithm 11.

| j | x_j^* | q_j | ϵ_j | $g_1(x_j^*)$ | $g_2(x_j^*)$ | $g_3(x_j^*)$ | $f(x_j^*)$ |
|-----|---|-------|--------------|--------------|--------------|--------------|------------|
| 0 | (0.1670927, 0.8365505, 2.011684, -0.9752350) | 2 | 1.0 | -0.001971 | 0.043152 | -1.815266 | -44.31766 |
| 1 | (0.1692354, 0.8394703, 2.006255, -0.9683554) | 4 | 0.1 | -0.003364 | 0.000497 | -1.862336 | -44.23219 |
| 2 | (0.1691869, 0.8394533, 2.006037, -0.9682074) | 8 | 0.01 | -0.004265 | -0.001105 | -1.863955 | -44.22833 |

Example 2 (see the Rosen-Suzki problem in [15]). We can see the following:

$$\begin{aligned}
 \min \quad & f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \\
 \text{s.t.} \quad & g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \leq 0 \\
 & g_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0 \\
 & g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0.
 \end{aligned} \tag{42}$$

Let $x^0 = (1, 1, 1, 1)$, $q_0 = 2.0$, $\epsilon_0 = 1.0$, $\eta = 0.1$, and $N = 2$; the results by Algorithm 7 are shown in Table 4.

Let $x^0 = (1, 1, 1, 1)$, $q_0 = 0.1$, $\epsilon_0 = 1.0$, $\eta = 0.1$, and $N = 5$; numerical results by Algorithm 10 are shown in Table 5.

Let $x^0 = (1, 1, 1, 1)$, $q_0 = 2.0$, $\epsilon_0 = 1.0$, $\eta = 0.1$, and $N = 2$; the results by Algorithm 11 are shown in Table 6.

It is clear from Table 4 that the obtained approximately optimal solution is $x^* = (0.1585001, 0.8339736, 2.014753, -0.959688)$ with corresponding objective function value -44.22965. From [15], the obtained approximately optimal solution is $x^* = (0.169234, 0.835656, 2.008690, -0.964901)$ with corresponding objective function value -44.233582.

From Tables 4–6, one can see that Algorithm 11 converges faster than Algorithms 7 and 10, but the solution generated by Algorithm 11 is the worst. Algorithm 10 is the slowest one, and the solution generated by Algorithm 10 is worse than the solution generated by Algorithm 7.

From Tables 1–6, one can see that Algorithm 7 yields some approximate solutions to $[P]$ that have a better objective function value in comparison with Algorithms 10 and 11.

5. Conclusion

In this paper, we propose a method for smoothing the non-smooth square-order exact penalty function for inequality constrained optimization. Error estimations are obtained among the optimal objective function values of the smoothed penalty problem, of the nonsmooth penalty problem, and of the original optimization problem. The algorithm based on the smoothed penalty functions is shown to be convergent under mild conditions.

According to the numerical results given in Section 4, one may draw that the smoothing penalty function $\varphi_q(x)$ yields some better convergence results for computing an approximate solution to $[P]$ than $F(x, q)$ and $f(x, q)$.

Finally, we give some advices on how to choose a parameter in the algorithm. According to our experience, initially, q_0 may be 0.1, 1, 5, 10, 100, 1000, or 10000, $N = 2, 5, 10$, or 100, and the iteration formula $q = Nq$. The initial value of ϵ_0 may be 10, 5, 1, 0.5, or 0.1, $\eta = 0.5, 0.1, 0.05$, or 0.01, and the iteration formula $\epsilon = \eta\epsilon$.

Acknowledgments

This work is supported by National Natural Science Foundation of China (10971118 and 71371107) and the Foundation of Shandong Province (J10LG04 and ZR2012AL07).

References

- [1] W. I. Zangwill, "Non-linear programming via penalty functions," *Management Science*, vol. 13, pp. 344–358, 1967.
- [2] M. S. Bazaraa and J. J. Goode, "Sufficient conditions for a globally exact penalty function without convexity," *Mathematical Programming Studies*, vol. 19, pp. 1–15, 1982.
- [3] S. P. Han and O. L. Mangasarian, "Exact penalty functions in nonlinear programming," *Mathematical Programming*, vol. 17, no. 1, pp. 251–269, 1979.
- [4] O. L. Mangasarian, "Sufficiency of exact penalty minimization," *SIAM Journal on Control and Optimization*, vol. 23, no. 1, pp. 30–37, 1985.
- [5] G. D. Pillo, "Exact penalty methods," in *Algorithms for Continuous Optimization*, E. Spedicato, Ed., pp. 209–253, Kluwer Academic, New York, NY, USA, 1994.
- [6] C. Yu, K. L. Teo, L. Zhang, and Y. Bai, "A new exact penalty function method for continuous inequality constrained optimization problems," *Journal of Industrial and Management Optimization*, vol. 6, no. 4, pp. 895–910, 2010.
- [7] F. S. Bai and X. Y. Luo, "Modified lower order penalty functions based on quadratic smoothing approximation," *Operations Research Transactions*, vol. 16, no. 2, pp. 9–22, 2012.
- [8] C. Chen and O. L. Mangasarian, "Smoothing methods for convex inequalities and linear complementarity problems," *Mathematical Programming B*, vol. 71, no. 1, pp. 51–69, 1995.
- [9] S. J. Lian, "Smoothing approximation to l1 exact penalty function for inequality constrained optimization," *Applied Mathematics and Computation*, vol. 219, no. 6, pp. 3113–3121, 2012.
- [10] Z. Q. Meng and S. Gao, "Smoothed square-root penalty function for nonlinear constrained optimization," *Operations Research Transactions*, vol. 17, no. 2, pp. 70–80, 2013.
- [11] M. Pinar and S. Zenios, "On smoothing exact penalty functions for convex constrained optimization," *SIAM Journal on Optimization*, vol. 4, pp. 468–511, 1994.
- [12] C. Y. Wang, W. L. Zhao, J. C. Zhou, and S. J. Lian, "Global convergence and finite termination of a class of smooth penalty function algorithms," *Optimization Methods and Software*, vol. 28, no. 1, pp. 1–25, 2013.
- [13] X. S. Xu, Z. Q. Meng, J. W. Sun, L. G. Huang, and R. Shen, "A second-order smooth penalty function algorithm for constrained optimization problems," *Computational Optimization and Applications*, vol. 55, no. 1, pp. 155–172, 2013.
- [14] X. Q. Yang, Z. Q. Meng, X. X. Huang, and G. T. Y. Pong, "Smoothing nonlinear penalty functions for constrained optimization problems," *Numerical Functional Analysis and Optimization*, vol. 24, no. 3–4, pp. 351–364, 2003.
- [15] Z. Meng, C. Dang, and X. Yang, "On the smoothing of the square-root exact penalty function for inequality constrained optimization," *Computational Optimization and Applications*, vol. 35, no. 3, pp. 375–398, 2006.
- [16] Z. Y. Wu, F. S. Bai, X. Q. Yang, and L. S. Zhang, "An exact lower order penalty function and its smoothing in nonlinear programming," *Optimization*, vol. 53, no. 1, pp. 51–68, 2004.
- [17] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John & Wiley Sons, New York, NY, USA, 2nd edition, 1993.
- [18] X. L. Sun and D. Li, "Value-estimation function method for constrained global optimization," *Journal of Optimization Theory and Applications*, vol. 102, no. 2, pp. 385–409, 1999.