JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **5**, Number **3**, August **2009**

doi:10.3934/jimo.2009.5.585

pp. **585–601**

A PENALTY FUNCTION ALGORITHM WITH OBJECTIVE PARAMETERS FOR NONLINEAR MATHEMATICAL PROGRAMMING

Zhiqing Meng

College of Business and Administration
Zhejiang University of Technology, Zhejiang 310023, China

Qiying Hu

School of Management
Fudan University, Shanghai 201800, China

Chuangyin Dang

Department of Manufacturing Engineering & Engineering Management
City University of Hong Kong, Kowloon, Hong Kong, China

(Communicated by Jan Ruckmann)

Abstract. In this paper, we present a penalty function with objective parameters for inequality constrained optimization problems. We prove that this type of penalty functions has good properties for helping to solve inequality constrained optimization problems. Moreover, based on the penalty function, we develop an algorithm to solve the inequality constrained optimization problems and prove its convergence under some conditions. Numerical experiments show that we can obtain a satisfactorily approximate solution for some constrained optimization problems as the same as the exact penalty function.

1. **Introduction.** The problem we consider in this paper is the following inequality constrained optimization problem:

$$
\begin{array}{lll}
(P) & \min & f_0(x) \\
 & \text{s.t.} & f_i(x) \le 0,\ i \in I = \{1, 2, \cdots, m\},
\end{array}
$$

where $f_i : R^n \to R,\ i \in I_0 = \{0, 1, 2, \cdots, m\}$. Its feasible set is denoted by $X = \{x \in R^n \mid f_i(x) \le 0,\ i \in I\}$.

The penalty function method provides an important approach to solving (P). Its main idea is to transform (P) into a sequence of unconstrained optimization problems which are easier to solve. In recent years, many researchers have paid attention to it in both theoretical and practical aspects. It is well-known that a penalty function for (P) is defined as:

$$
F(x, \rho) = f_0(x) + \rho \sum_{i \in I} \max\{f_i(x), 0\}^2,
$$

2000 *Mathematics Subject Classification.* Primary: 65K05; Secondary: 90C30.

Key words and phrases. Nonlinear programming, penalty function, objective parameter, constraint.

The first author is supported by the National Natural Science Foundation of China under grunt 10671039 and SRG7002006.

and the corresponding penalty optimization problem for (P) is defined as:

$$(P_\rho) \qquad\qquad \min\ F(x,\rho) \text{ s.t. } x \in R^n.$$

The penalty function $F(x,\rho)$ is smooth if the constraints and objective function are differentiable, but it is not necessarily exact. Here, a penalty function $F(x,\rho)$ is exact if there is some ρ^* such that an optimal solution to (P_ρ) is also an optimal solution to (P) for all $\rho \geq \rho^*$. We call ρ a penalty parameter. In 1967, Zangwill [19] presented the following penalty function:

$$F_1(x,\rho) = f_0(x) + \rho \sum_{i \in I} \max\{f_i(x), 0\},$$

with the corresponding penalty optimization problem of (P)

$$(EP_\rho) \qquad\qquad \min F_1(x,\rho) \text{ s.t. } x \in R^n.$$

The penalty function $F_1(x,\rho)$ is exact under certain assumptions, but it is not smooth.

Exact penalty functions attract many researchers to study. For example, Han and Mangasrian [5] presented an exact penalty function for nonlinear programming. Rosenberg [14] gave a globally convergent algorithm for convex programming based on an exact penalty function. Rosenberg in [15] further studied stability of exact penalty functions in locally Lipschitz programming. Di Pillo and Grippo [12] proposed an exact penalty function method with global convergence for nonlinear programming problems. Zenios, et al. [20] discussed a smooth penalty function algorithm for network-structured problems. Pinar and Zenios [13] presented a smooth exact penalty function for convex constrained optimization problems, in which all the objective function and the constraint functions are convex, and the smoothing penalty function is first-order differentiable. Mongeau and Sartenaer [10] discussed automatic decrease of the penalty parameter in exact penalty function methods. However, in these papers [5], [10], [12], [13], [14], [15], [20], no new types of penalty functions are presented and the type of penalty functions studied in them is the same as that presented by [19]. On the other hand, the existing exact penalty function algorithms still need to increase the penalty parameter in order to find out a better solution and the penalty functions are not differentiable [5], [12], [14], [15], [19]. Hence, we cannot use any efficient algorithms, such as Newton Method, to solve constrained optimization problems via those exact penalty function methods.

Recently, Rubinov, Glover, Yang, and Huang [6], [16], [17], [18] presented a nonlinear Lagrangian penalty function, which is defined as

$$F_k(x,\rho) = [f_0(x)^k + \rho \sum_{i \in I} \max\{f_i(x), 0\}^k]^{\frac{1}{k}},$$

For some $k > 0$. The corresponding penalty optimization problem of (P) is

$$(EPk_\rho) \qquad\qquad \min F_k(x,\rho) \text{ s.t. } x \in R^n.$$

When $k = 1$, the problem (EPk_ρ) is exactly the same as (EP_ρ). The penalty function $F_k(x,\rho)$ is smooth for $k > 1$ if all the constraints and objective function are differentiable, but it is not necessarily smooth for $0 < k \leq 1$. Some conditions to ensure the exactness of the penalty function are required in [16], [17], however, they are not easy to check.

All the penalty function algorithms with the constrained penalty parameter need to increase the penalty parameter ρ gradually. So does the exact penalty function methods because we do not know exactly how big the penalty parameter ρ is needed.

In fact, from a computing point of view, it is impossible to take a very big value of the penalty parameter ρ due to the limited precision of a computer.

The penalty function method with an objective penalty parameter have been discussed in [1], [2], [3], [4], [11], where the penalty function is defined as

$$\phi(x, M) = (f_0(x) - M)^p + \sum_{i \in I} f_i(x)^p.$$

where $p > 0$. Suppose x^* is an optimal solution and f^* is the optimal value of the objective function, then a sequential penalty function method can be envisaged, in which a convergent sequence ($\{M^k\} \to f^*$) is generated so that the minimizers $x(M^k) \to x^*$. Morrison [11] considered the problem $\min\{f(x)|g(x) = 0\}$ and defines a penalty function problem: $\min\ (f(x) - M)^2 + |g(x)|^2$. Without convexity or continuity assumptions, a sequence of problems is constructed by choosing an appropriate convergent sequence M^k. Fletcher [3], [4] discussed a similar type of $\phi(x, M)$, furthermore Burke [1] considered a more general type. Fiacco and McCormick [2] gave a general introduction of sequential unconstrained minimization techniques. Mauricio and Maculan [8] discussed a Boolean penalty method for zero-one nonlinear programming and defined another type of penalty functions:

$$H(x, M) = \max\{f_0(x) - M, f_1(x), \cdots, f_m(x)\}.$$

Meng, Hu and Dang [9] also studied an objective penalty function method as follows

$$F(x, M) = (f_0(x) - M)^2 + \sum_{i \in I} \max\{f_i(x), 0\}^p,$$

which is a good smooth penalty function.

In this paper, we present a more general type of the penalty functions. It will give us good prospects to solve the constrained optimization problems.

Let functions $Q : R \to R \cup \{+\infty\}$ and $P : R \to R \cup \{+\infty\}$ satisfy respectively

$$\begin{cases} Q(t) > 0 & \text{for all } \ t \in R \setminus \{0\} \\ Q(0) = 0 & \\ Q(t_1) < Q(t_2) & \text{for } \ 0 \le t_1 < t_2 \end{cases}$$

and

$$\begin{cases} P(t) = 0 & \text{if and only if } \ t \le 0, \\ P(t) > 0 & \text{if and only if } \ t > 0. \end{cases}$$

We present the following penalty function with objective parameters:

$$F(x, M) = Q(f_0(x) - M) + \sum_{i \in I} P(f_i(x)),$$

where the objective parameter $M \in R$. If $Q(t),\ P(t),\ f_i(x)\ (i \in I_0)$ are all differentiable, then it is obvious that $F(x, M)$ is also differentiable. For example, letting $Q(t) = t^2$ and $P(t) = \max\{t, 0\}^4$, we have the following penalty function,

$$F(x, M) = (f_0(x) - M)^2 + \sum_{i \in I} f_i^+(x)^4,$$

where $f_i^+(x) = \max\{0, f_i(x)\},\ i \in I$. For $i \in I_0$, if f_i is first-order differentiable, it is clear that $f_i^+(x)^4$ is also first-order differentiable, and if f_i is second-order differentiable, it is clear that $f_i^+(x)^4$ is also second-order differentiable. And so is $F(x, M)$.

The remainder of this paper is organized as follows. In Section 2, we show some theorems for the penalty function $F(x, M)$ and present an algorithm to solve the original problem (P) with global convergence without any convex conditions. In Section 3, we give some numerical examples, which show that the number of iterations of the algorithm is few for obtaining a good approximate solution to (P).

2. A penalty function method with objective parameters.

Consider the following nonlinear optimization problem:

$$(\mathrm{P}(M)) \quad \min\ F(x, M), \quad \text{s.t. } x \in Y,$$

where $Y \subset R^n$ and the feasible set $X \subset Y$. Especially, when $Y = R^n$, $(\mathrm{P}(M))$ is an unconstrained optimization problem. If an optimal solution to $(P(M))$ for some M is also an optimal solution to (P), then M is called *an appropriate penalty parameter*. Next, we prove a theorem on the penalty function.

Theorem 2.1. *If x^* is an optimal solution to (P) and $M = f_0(x^*)$, then, x^* is also an optimal solution to $(P(M))$ with $F(x^*, M) = 0$.*

Proof. Since x^* is an optimal solution to (P) and $M = f_0(x^*)$, we have

$$F(x^*, M) = Q(f_0(x^*) - M) + \sum_{i \in I} P(f_i(x^*)) = 0.$$

It follows from $F(x, M) \geq 0$ for any $x \in R^n$ that x^* is an optimal solution to $(P(M))$. This completes the proof. □

For the reverse case, we first have the following result.

Theorem 2.2. *Suppose x^* is an optimal solution to (P), M is a constant and x_M^* is an optimal solution to $(P(M))$. If x_M^* further satisfies the following three conditions*

- *x_M^* and a feasible solution to (P),*
- *$F(x_M^*, M) \neq 0$,*
- *$M \leq f_0(x^*)$,*

then x_M^* is an optimal solution to (P).

Proof. From the given conditions, we have

$$Q(f_0(x_M^*) - M) = F(x_M^*, M) \leq F(x, M) = Q(f_0(x) - M), \ \forall x \in X \tag{1}$$

since x_M^* is a feasible solution and x^* is an optimal solution to (P), it follows from $M \leq f_0(x^*)$ that

$$f_0(x_M^*) - M \geq f_0(x_M^*) - f_0(x^*) \geq 0. \tag{2}$$

Note that $f_0(x) - M \geq 0$ for all $x \in X$. This together with (1) and (2) implies that

$$f_0(x_M^*) - M \leq f_0(x) - M, \qquad \forall x \in X.$$

So x_M^* is an optimal solution to (P). □

It is very important that Theorem 2.2 gives theoretically a good result for penalty function $F(x, M)$ with parameters M. Let $M_* = \min_{x \in X} f_0(x)$ and $M^* = \max_{x \in X} f_0(x)$. Now we prove the following theorem by Theorem 2.1 and Theorem 2.2.

Theorem 2.3. *Suppose that the feasible set X is connected and compact, f_0 is continuous on R^n, and for some M, x_M^* is an optimal solution to $(P(M))$. Then the following two assertions hold:*

(i) If $F(x_M^*, M) = 0$ then $M_* \leq M \leq M^*$.

(ii) If $F(x_M^*, M) \neq 0$ and $M \leq M^*$ then $M < M_*$. Furthermore, if x_M^* is a feasible solution to (P), then x_M^* is an optimal solution to (P).

Proof. (i) The conclusion is obvious from the conditions on P and Q.

(ii) This is proved by contradiction. If $M_* \leq M$, we have $M_* \leq M \leq M^*$. Since f_0 is continuous on the connected and compact set X, there is some $x_0 \in X$ such that $M = f_0(x_0)$. So, we get $F(x_0, M) = 0$. On the other hand, x_M^* is an optimal solution to $(P(M))$ and $F(x_M^*, M) > 0$. Hence, $0 < F(x_M^*, M) \leq F(x_0, M) = 0$.This is a contradiction. Therefore, $M_* > M$.

Furthermore, if x_M^* is a feasible solution to (P), then it follows from Theorem 2.2 that x_M^* is an optimal solution to (P). □

Theorem 2.2 points out a good way to solve (P). Moreover, Theorem 2.3 give us another way to solve (P) very well. The objective parameter M required in Theorem 2.2 may exist, as shown in the following example.

Example 2.1 Consider the problem:

$$(P2.1) \qquad \min \quad x_1^2 + x_2^2$$
$$\text{s.t.} \quad -x_1 \leq 0,\ -x_2 \leq 0.$$

It is clear that $(x_1^*, x_2^*) = (0, 0)$ is an optimal to (P2.1) and objective value is 0. Let's take $M = -1$. Define the penalty function:

$$F(x, -1) = (x_1^2 + x_2^2 + 1)^2 + (\max\{0, -x_1\}^4 + \max\{0, -x_2\}^4).$$

It is clear that $(x_1, x_2) = (0, 0)$ is an optimal solution to $(P(M))$ [with $M = -1$]. Since $F((0, 0), -1) > 0$, we know that $(x_1, x_2) = (0, 0)$ is an optimal solution to $(P2.1)$ by Theorem 2.2. □

Definition 2.1 A vector $x \in X$ is ϵ-feasible or ϵ-solution if

$$f_i(x) \leq \epsilon, \qquad \forall i \in I.$$

Based on Theorem 2.3, we develop an algorithm to compute a globally optimal solution to (P). It solves the problem (P(M)) sequentially and we call it as Objective Parameters Function Algorithm (OPFA for short).

OPFA Algorithm:

Step 1:: Choose $\epsilon \geq 0$, $x^0 \in X$, and $a_1 < \min_{x \in X} f_0(x)$. Let $k = 1$, $b_1 = f_0(x^0)$, and $M_1 = \frac{a_1 + b_1}{2}$.

Step 2:: Solve $\min_{x \in Y} F(x, M_k)$. Let x^k be a global minimizer.

Step 3:: If x^k is not feasible to (P), let $b_{k+1} = b_k$, $a_{k+1} = M_k$, $M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$, and go to Step 5. Otherwise, $x^k \in X$, and go to Step 4.

Step 4:: If $F(x^k, M_k) = 0$, let $a_{k+1} = a_k$, $b_{k+1} = M_k$, $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$, and go to Step 5. Otherwise, $F(x^k, M_k) > 0$, stop and x^k is an optimal solution to (P).

Step 5:: If $|b_{k+1} - a_{k+1}| \le \epsilon$ and $f_i(x^k) \le \epsilon, i = 1, 2, \cdots, m$, stop and x^k is an $\epsilon-$ solution to (P). Otherwise, let $k = k + 1$, and go to Step 2.

Remark 1. In the OPFA algorithm, it is assumed that we can always get $a_1 < \min_{x\in X} f_0(x)$.

The convergence of the OPFA algorithm is proved in the following theorem. Let

$$S(L, f_0) = \{x^k \mid L \ge Q(f_0(x^k) - y_k),\ k = 1, 2, \cdots\},$$

which is called a Q-level set. We say that $S(L, f_0)$ is bounded if, for any given $L > 0$ and a convergent sequence $y_k \to y^*$, $S(L, f_0)$ is bounded.

Theorem 2.4. *Let $Y = R^n$ or Y be an open set. Suppose that X is connected and compact, Q and f_i ($i \in I_0$) are continuous on R^n, and the Q-level set $S(L, f_0)$ is bounded. Let $\{x^k\}$ be the sequence generated by the OPFA algorithm.*

(i) If $\{x^k\}(k = 1, 2, \cdots, \bar{k})$ is a finite sequence (i.e., the OPFA algorithm stops at the $\bar{k}$-th iteration), then $x^{\bar{k}}$ is an optimal solution to (P) or is an ϵ- solution to (P).

(ii) If $\{x^k\}$ is an infinite sequence, then $\{x^k\}$ is bounded and any limit point of it is an optimal solution to (P).

Proof. We first show that the sequence $\{a_k\}$ increases and $\{b_k\}$ decreases with

$$a_k < M_k = \frac{a_k + b_k}{2} < b_k, \qquad k = 1, 2, \cdots \tag{3}$$

and

$$b_{k+1} - a_{k+1} = \frac{b_k - a_k}{2}, \qquad k = 1, 2, \cdots . \tag{4}$$

This is proved by the induction method as follows.

1. It is clear from the OPFA algorithm that $a_1 < M_1 = \frac{a_1+b_1}{2} < b_1$ and $b_2 - a_2 = \frac{b_1-a_1}{2}$.
2. Suppose that (3) and (4) hold for some $k \ge 1$. Consider $k + 1$. In Step 3 of OPFA Alrorithm, we have $b_{k+1} = b_k$, $a_{k+1} = M_k$ and $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$. Thus,

$$M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2} < \frac{b_k + b_k}{2} = b_k = b_{k+1},$$

and

$$M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2} > M_k = a_{k+1}.$$

Therefore, $b_k = b_{k+1}$, $a_k < a_{k+1}$, $a_{k+1} < M_{k+1} < b_{k+1}$, and

$$b_{k+1} - a_{k+1} = b_k - M_k = \frac{b_k - a_k}{2}.$$

In Step 4, we have $a_{k+1} = a_k$, $b_{k+1} = M_k$ and $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$. Thus,

$$M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2} < \frac{M_k + M_k}{2} = b_{k+1},$$

and

$$M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2} > \frac{a_k + a_k}{2} = a_k = a_{k+1}.$$

Therefore, $a_k = a_{k+1}$, $b_k > b_{k+1}$, $a_{k+1} < M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2} < b_{k+1}$, and

$$b_{k+1} - a_{k+1} = M_k - a_k = \frac{b_k - a_k}{2}.$$

By the induction method, (3) and (4) follows immediately.

From the algorithm, it is obvious that $\{a_k\}$ is increasing and $\{b_k\}$ is decreasing. Thus, both $\{a_k\}$ and $\{b_k\}$ converge. Let $a_k \to a^*$ and $b_k \to b^*$. By (4), we have $a^* = b^*$. Therefore, $\{M_k\}$ also converges to a^*.

(i) If the OPFA Algorithm terminates at the $\bar{k}$th iteration and the second situation of Step 4 occur, i.e., $x^{\bar{k}}$ is a feasible solution to (P) and $F(x^{\bar{k}}, M_{\bar{k}}) > 0$. By Theorem 2.3, $x^{\bar{k}}$ is an optimal solution to (P). Or, the stopping condition $|b_{\bar{k}+1} - a_{\bar{k}+1}| < \epsilon$ and $f_i(x^{\bar{k}}) \le \epsilon, i = 1, 2, \cdots, m$ will occur. By definition 2.1, $x^{\bar{k}}$ is an $\epsilon-$ solution to (P).

(ii) We first show that the sequence $\{x^k\}$ is bounded. Since x^k is an optimal solution to $\min\limits_{x\in Y} F(x, M_k)$,

$$F(x^k, M_k) \le Q(f_0(x^0) - M_k), \qquad k = 1, 2, \cdots.$$

Due to $M_k \to a^*$ as $k \to +\infty$, we conclude that there is some $L > 0$ such that

$$L > F(x^k, M_k) \ge Q(f_0(x^k) - M_k), \qquad k = 1, 2, \cdots.$$

Since the Q-level set $S(L, f_0)$ is bounded, the sequence $\{x^k\}$ is bounded.

Let $M_* = \min\limits_{x\in X} f_0(x)$. Without loss of generality, we assume $x^k \to x^*$. We have shown that

$$a_k < M_k < b_k, \qquad k = 1, 2, \cdots$$

and all the sequences $\{a_k\}$, $\{b_k\}$ and $\{M_k\}$ converge to a^*. By Step 3 and Theorem 2.3 (ii), we know $a_k \le M_*$, $k = 1, 2, \cdots$. Due to Step 4 and Theorem 2.3 (i), we know $M_* \le b_k$, $k = 1, 2, \cdots$. Thus, $a_k \le M_* \le b_k$. Letting $k \to +\infty$, we obtain $a^* = M_*$. Let y^* be an optimal solution to (P). Then $M_* = f_0(y^*)$. Note that

$$F(x^k, M_k) \le F(y^*, M_k) = Q(f_0(y^*) - M_k).$$

By letting $k \to +\infty$ in the above equation, we obtain

$$F(x^*, M_*) \le 0,$$

which implies $M_* = f_0(x^*)$. Therefore, x^* is an optimal solution to (P). □

In the OPFA Algorithm, we need to find an optimal solution of $\min\limits_{x\in Y} F(x, M_k)$, which is a difficult task. To avoid this difficulty, one may replace it with

$$\nabla F(x, M_k) = 0$$

if $Q(\cdot)$, $P(\cdot)$, $f_i(x)$ $(i \in I_0)$ are all differentiable. Then we obtain a modified OPFA algorithm as follows.

A Modified OPFA Algorithm:

Step 1:: Choose $\epsilon \ge 0$, $x^0 \in X$, and a_1 satisfying $a_1 < \min\limits_{x\in X} f_0(x)$. Let $k = 1$, $z^1 = x^0$, $b_1 = f_0(x^0)$, and $M_1 = \frac{a_1+b_1}{2}$. Choose a sequence $\delta_k > 0$ such that $\delta_k \to 0$, Go to Step 2.

Step 2:: Solve $\min\limits_{x\in Y} F(x, M_k)$. Let x^k be a point satisfying $\|\nabla F(x^k, M_k)\| \le \delta_k$.

Step 3:: If $F(x^k, M_k) > 0$, let $z^{k+1} = z^k$, $b_{k+1} = b_k$, $a_{k+1} = M_k$, $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$, and go to Step 5. Otherwise, $F(x^k, M_k) = 0$, and go to Step 4.

Step 4:: Let $a_{k+1} = a_k$, $b_{k+1} = M_k$, $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$, and go to Step 5.

Step 5:: If $|b_{k+1} - a_{k+1}| < \epsilon$ and $f_i(x^k) \le \epsilon, i = 1, 2, \cdots, m$, stop and x^k is an ϵ- solution to (P). Otherwise, let $k := k + 1$, and go to Step 2.

Remark: If $\{x^k\}$ is a finite sequence (i.e., the modified OPFA algorithm stops at the $\bar{k}$-th iteration), then $x^{\bar{k}}$ is an ϵ-solution to (P) in Step 5.

For the modified OPFA algorithm, we have the following theorem.

Theorem 2.5. *Let $Y = R^n$ or Y be an open set. Suppose that X is connected and compact, P and Q are continuously differentiable on R, and f_i $(i \in I_0)$ is continuously differentiable on R^n . Let $\{x^k\}$ be an infinite sequence generated by the modified OPFA algorithm. Suppose the Q-level set $S(L, f_0)$ and the sequence $\{F(x^k, M_k)\}$ are bounded.*

1. *The sequence $\{x^k\}$ is bounded, and for any limit point x^* of it, there exist $\lambda \in R$ and $\mu_i \in R$, $i = 1, 2, \cdots, m$, such that*

$$\lambda \nabla f_0(x^*) + \sum_{i \in I} \mu_i \nabla f_i(x^*) = 0.$$

2. *There exists some index $\overline{k}$ such that*
   (i) $a_k < M_*$ for all $k = 1, \cdots, \overline{k}$,
   (ii) x^k is (up to the perturbation given by δ_k) a global solution of the auxiliary problem (P(M))[with $M := M_k$] for all $k \ge \overline{k} + 1$, then $\{x^k\}$ is bounded and any limit point of it is an optimal solution to (P).

Proof. (i) Similarly to the proof of Theorem 2.4, we know that $\{a_k\}$ increases to a^* and $\{b_k\}$ decreases to a^* with

$$a_k < M_k < b_k, \qquad k = 1, 2, \cdots$$

and so $\{M_k\} \to a^*$. Since $\{F(x^k, M_k)\}$ is bounded and M_k converges to a^* as $k \to +\infty$, there is some $L > 0$ such that

$$L > F(x^k, M_k) \ge Q(f_0(x^k) - M_k), \qquad k = 1, 2, \cdots .$$

Applying the boundedness of the Q-level set $S(L, f_0)$, we conclude that $\{x^k\}$ is bounded.

Without loss of generality, suppose $x^k \to x^*$. Due to the assumption, we have

$$\nabla F(x^k, M_k) = Q'(f_0(x^k) - M_k) \nabla f_0(x^k) + \sum_{i \in I} P'(f_i(x^k)) \nabla f_i(x^k), \qquad k = 1, 2, \cdots .$$

Since $Q(\cdot)$, $P(\cdot)$, $f_i(x)$ $(i \in I_0)$ are all continuously differentiable, $Q'(f_0(x^k) - M_k) \to Q'(f_0(x^*) - a^*) = \lambda$ and $P'(f_i(x^k)) \to P'(f_i(x^*)) = \mu_i$ $(i \in I)$ as $k \to +\infty$. Let $k \to +\infty$, $\|\nabla F(x^k, M_k)\| \le \delta_k$ such that $\|\nabla F(x^*, a^*)\| = 0$. Thus, there exist $\lambda \in R$ and $\mu_i \in R$, $i = 1, 2, \cdots, m$, such that

$$\lambda \nabla f_0(x^*) + \sum_{i \in I} \mu_i \nabla f_i(x^*) = 0.$$

(ii) The proof is very similarly to that Theorem 2.4 (ii). □

Remark 2. The appropriate choice (P, Q) from the numerical iteration is very important. This differs from other penalty algorithms presented in [5], [6], [10], [12]- [20].

3. **Numerical examples.** The feasible solution set X to the problem (P) is often unbounded. If an optimal solution to the problem (P′)

$$(\mathrm{P}') \qquad \min f(x) \qquad x \in X \cap Y$$

is an optimal solution to the problem (P), where $Y \subset R^n$ is bounded, then in the OPFA algorithm and the modified OPFA algorithm, we solve

$$(\mathrm{P}(M)) \quad \min \; F(x, M), \quad \text{s.t.} \;\; x \in Y.$$

The OPFA algorithm and the modified OPFA algorithm provide a method to solve (P). In the following examples, we simply choose

$$Q(t) = t^2 \qquad \text{or} \qquad Q(t) = a^{\alpha t^2},$$

and

$$P(t) = \beta \max\{t, 0\}^p,$$

with $a > 1, \beta > 0, p \geq 1$ and $\alpha \in (10^{-7}, 10^{-1})$. They are simple types of (P, Q). Let $\epsilon = 10^{-6}$, we want to get an ϵ- solution to (P) in the OPFA algorithm by Matlab6.5.

The OPFA algorithm and the modified OPFA algorithm is adapted to solve some problems where a starting point $x^0 \in X$ and a value $a_1 < \min_{x \in X} f_0(x)$ is known.

In order to compare convergence of the OPFA Algorithm for different penalty functions, we use the exact penalty function

$$F_1(x, \rho) = f_0(x) + \rho \sum_{i=1}^{m} \max\{f_i(x), 0\},$$

and the penalty function

$$F_2(x, \rho) = f_0(x) + \rho \sum_{i=1}^{m} \max\{f_i(x), 0\}^2,$$

to define the following Algorithm I and II.

Algorithm I:

Step 1: Given $x^0, \epsilon > 0, \rho_0 > 0$, and $N > 1$.
Let $j = 0$.

Step 2: Using the violation x^j as the starting point for solving the problem:
$\min_{x \in X} F_1(x, \rho_j) = f_0(x) + \rho_j \sum_{i=1}^{m} \max\{f_i(x), 0\}$
Let x^j be the optimal solution.

Step 3: If x^j is ϵ-feasible to (P),
then stop and get an approximate solution x^j of (P),
otherwise, let $\rho_{j+1} = N\rho_j$
and set $j := j + 1$ and go to Step 2.

Algorithm II:

Step 1: Given $x^0, \epsilon > 0, \rho_0 > 0$, and $N > 1$.
Let $j = 0$.

Step 2: Using the violation x^j as the starting point for solving the problem:
$\min_{x \in X} F_2(x, \rho_j) = f_0(x) + \rho_j \sum_{i=1}^{m} \max\{f_i(x), 0\}^2$.
Let x^j be the optimal solution.

Step 3: If x^j is ϵ-feasible to (P),
then stop and get an approximate solution x^j of (P),
otherwise, let $\rho_{j+1} = N\rho_j$
and set $j := j + 1$ and go to Step 2.

We define the constrain error $e(x^j)$, for the j'th step as

$$e(x^j) = \sum_{i=1}^{m} \max\{f_i(x^j), 0\}.$$

It is clear that x^j is ϵ-feasible to (P), when $e(x^j) < \epsilon$. For the next example we use both Algorithm I and II programmed in Matlab.

Example 3.1 Consider the following problem.

$$(P3.1) \qquad \min \quad x_1 + x_2$$
$$\text{s.t.} \quad x_1^2 - x_2 \le 0, \ -x_1 \le 0.$$

The optimal solution of (P3.1) is $(x_1^*, x_2^*) = (0, 0)$ and the optimal objective value is 0. The nonlinear penalty function is defined as

$$F(x, M) = (x_1 + x_2 - M)^2 + 100(\max\{0, x_1^2 - x_2\}^4 + \max\{0, -x_1\}^4).$$

Let $Y = \{(x_1, x_2) \mid 0 \le x_1 \le 100, 0 \le x_2 \le 100\}$. We use the OPFA algorithm to solve (P3.1) as follows.

(1) Let $x^0 = (2, 4) \in X,\ a_1 = -4,\ b_1 = 6,\ M_1 = 1$. We have an optimal solution $x^1 = (0.3333, 0.6667)$ to $\min_{x \in R^n} F(x, 1)$. Since $F(x^1, 1) = 0$, we have $a_2 = -4,\ b_2 = 1,\ M_2 = -1.5$.

(2) Solving $\min_{x \in R^n} F(x, -1.5)$, we obtain its optimal solution $x^2 = (0, 0)$. Since $F(x^2, -1.5) = 2.25 > 0,\ x^2 = (0, 0)$ is the optimal solution to (P3.1) by Theorem 2.2.

The following examples are solved with the modified OPFA algorithm.

Example 3.2 Consider the following problem:

$$
\begin{aligned}
(P3.2) \quad \min \quad & f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \\
\text{s.t.} \quad & g_1(x) = 2x_1^2 + x_2^2 + x_3^2 + 2x_1 + x_2 + x_4 - 5 \le 0 \\
& g_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \le 0 \\
& g_3(x) = x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \le 0
\end{aligned}
$$

The nonlinear penalty function of (P3.2) is defined as

$$F(x,M) = 10^{0.0001(f(x)-M)^2} + 1000(\max\{g_1(x),0\}^2 + \max\{g_2(x),0\}^2 + \max\{g_3(x),0\}^2). \quad (5)$$

Let $Y = \{(x_1,x_2,x_3,x_4) \mid -100 \le x_i \le 100, i = 1,2,3,4\}$. Let $x^0 = (0,0,0,0) \in X$, $a_1 = -200$, $b_1 = 0$, $M_1 = -100$. We have Table 3.1 by the modified OPFA algorithm.

Table 3.1

k	$g_1(x^k)$	$g_2(x^k)$	$g_3(x^k)$	$e(x^k)$	x^k	$f(x^k)$	M_k
1	-0.000212	-0.000170	-2.253092	0.000000	(0.285680, 0.773495, 1.969793, -0.986046)	-44.161631	-100.000000
2	0.000006	-0.000075	-2.103541	0.000006	(0.222761, 0.802096, 1.994979, -0.970156)	-44.215873	-50.000000
3	-0.000003	-0.000001	-2.103447	0.000000	(0.222758, 0.802092, 1.994986, -0.970173)	-44.216013	-47.080815

We get an approximately ϵ-feasible solution $x^1 = (0.285680, 0.773495, 1.969793, -0.986046)$ at the 1'th iteration, with its objective value $f(x^1) = -44.161631$. It is easily to check that both x^1 and x^3 are feasible solution to (P3.2).

If the nonlinear penalty function of (P3.2) is defined as

$$F(x,M) = 10^{0.0001(f(x)-M)^2} + 10(\max\{g_1(x),0\}^2 + \max\{g_2(x),0\}^2 + \max\{g_3(x),0\}^2), \quad (6)$$

then we have better results in Table 3.2 by the modified OPFA algorithm.

Table 3.2

k	$g_1(x^k)$	$g_2(x^k)$	$g_3(x^k)$	$e(x^k)$	x^k	$f(x^k)$	M_k
1	0.000152	0.000302	-1.856063	0.000454	(0.185978, 0.851199, 1.991509, -0.982807)	-44.230076	-100.000000
2	-0.001259	-0.000104	-1.856210	0.000000	(0.185817, 0.851023, 1.991418, -0.982952)	-44.228230	-50.000000

Table 3.3

Penalty function	Iter.	ρ_k	$e(x^k)$	$f(x^k)$	x^k
the modified OPFA Algorithm	1	1000	0.000000	-44.161631	(0.285680, 0.773495, 1.969793, -0.986046)
	2	1000	0.000006	-44.215873	(0.222761, 0.802096, 1.994979, -0.970156)
	3	1000	0.000000	-44.216013	(0.222758, 0.802092, 1.994986, -0.970173)
Algorithm I $F_1(x,\rho)$	1	10	0.001092	-43.294239	(0.236677, 1.174704, 1.752718, -1.210949)
	2	100	Inf	NaN	(-Inf, Inf, -Inf, -Inf)
	3	1000	0.000000	NaN	(NaN, NaN, NaN, NaN)
Algorithm II $F_2(x,\rho)$	1	10	0.135159	-44.372929	(0.248900, 0.716949, 2.031431, -0.943248)
	2	100	0.013654	-44.256309	(0.169707, 0.835556, 2.009784, -0.966218)
	3	1000	0.001368	-44.236091	(0.169574, 0.835536, 2.008748, -0.965013)

The testing results in Table 3.3 show that exact penalty function $F_1(x,\rho)$ is not stable convergent, when penalty parameter ρ becomes larger. If the penalty function $F_1(x,\rho)$ is exact, in general, it is not smooth. The computing problem needs to have a smaller penalty parameter. But many practical problems need a large penalty parameter. The convergence of Algorithm II is very slow.

Example 3.3 Consider the following problem:

$$
\begin{aligned}
(P3.3)\quad \min\quad & f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \\
\text{s.t.}\quad & g_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\
& g_2(x) = (x_1 - 5)^2 + x_2^2 + x_3^2 - 25 = 0 \\
& g_3(x) = (x_1 - 5)^2 + (x_2 - 5)^2 + (x_3 - 5)^2 - 25 \le 0
\end{aligned}
$$

Table 3.4

Penalty function	Iter.	ρ_k	$e(x^k)$	$f(x^k)$	x^k
the modified OPFM Algorithm	1	1000	0.050349	944.112402	(2.504463, 4.224005, 0.965885)
	2	1000	0.000019	944.215617	(2.500002, 4.221103, 0.965557)
	3	1000	0.000003	944.215648	(2.500000, 4.221105, 0.965542)
	4	1000	0.000000	944.215654	(2.500000, 4.221105, 0.965542)
Algorithm I $F_1(x, \rho)$	1	10	0.000076	944.215986	(2.499996, 4.224207, 0.951861)
	2	100	0.000000	944.215859	(2.500000, 4.224211, 0.951861)
	3	1000	0.000000	944.215859	(2.500000, 4.224211, 0.951861)
Algorithm II $F_2(x, \rho)$	1	10	0.113262	943.991108	(2.510094, 4.244666, 0.888449)
	2	100	0.011481	944.192099	(2.501018, 4.221972, 0.964724)
	3	1000	0.001152	944.213291	(2.500102, 4.221391, 0.964592)
	4	10000	0.000115	944.215416	(2.500010, 4.221332, 0.964578)

Let $Y = \{(x_1, x_2, x_3) \mid 0 \le x_i \le 100, i = 1, 2, 3\}$. For the nonlinear penalty function defined as

$$
\begin{aligned}
F(x, M) = 10^{0.0000001(f(x)-M)^2} - 1 + 1000(\max\{g_1(x), 0\}^2 + \max\{-g_1(x), 0\}^2 \\
+ \max\{g_2(x), 0\}^2 + \max\{-g_2(x), 0\}^2 + \max\{g_3(x), 0\}^2). \quad (7)
\end{aligned}
$$

Let $x^0 = (0, 0, 5) \in X$, $a_1 = -1200$, $b_1 = 975$, $M_1 = -5512.5$. We have results in Table 3.4. The testing results in Table 3.4 show that the OPFA algorithm gets an approximate solution within fewer iterations than Algorithm I. Also, the modified OPFA algorithm has a better approximate solution than Algorithm II in Table 3.4.

Example 3.4 Consider the following problem (Example 1 in [20]):

$$
\begin{aligned}
(P3.4)\quad \min\quad & f(x) = 100x_1 + 120x_2 + 90x_3 + 80x_4 + 70x_5 + 140x_6 \\
& +40x_7 + 20x_8 + 30x_9 + 20x_{10} + 40x_{11} + 10x_{12} \\
\text{s.t.}\quad & g_1(x) = x_1 + x_2 + x_3 - 25 = 0 \\
& g_2(x) = x_4 + x_5 + x_6 - 15 = 0 \\
& g_3(x) = x_1 + x_4 - 20 = 0 \\
& g_4(x) = x_2 + x_5 - 10 = 0 \\
& g_5(x) = x_3 + x_6 - 10 = 0 \\
& g_6(x) = x_7 + x_8 + x_9 - 50 = 0 \\
& g_7(x) = x_{10} + x_{11} + x_{12} - 30 = 0 \\
& g_8(x) = x_7 + x_{10} - 20 = 0 \\
& g_9(x) = x_9 + x_{11} - 40 = 0 \\
& g_{10}(x) = x_9 + x_{12} - 20 = 0 \\
& g_{11}(x) = x_1 + x_7 - 30 \le 0 \\
& g_{12}(x) = x_3 + x_9 - 30 \le 0 \\
& 0 \le x_i \le 75, \qquad i = 1, \cdots, 12.
\end{aligned}
$$

For the nonlinear penalty function defined as

$$\begin{aligned} F(x,\rho,M) &= 10^{0.0000001(f(x)-M)^2}-1 \\ &\quad +3000(\sum_{i=1}^{10} g_i(x)^2+\max\{g_{11}(x),0\}^2+\max\{g_{12}(x),0\}^2), \quad (8) \end{aligned}$$

let $x^0=(15,5,5,5,5,5,10,30,10,10,10,10)\in X,\ a_1=-30000,\ b_1=-6000,\ M_1=-18000$. After 3 iterations, we obtain ϵ- solution $x^3=(15,10,0,5,0,10,$ $15,15,20,5,25,0)$ with $f(x^3)=7100.006841$, which is almost the same as that in [20].

Example 3.5 Consider the following problem (Example 2 in [20]):

$$\begin{aligned} (P3.5)\quad \min\quad & f(x)=10x_2+2x_3+x_4+3x_5+4x_6 \\ \text{s.t.}\quad & g_1(x)=x_1+x_2-10=0 \\ & g_2(x)=-x_1+x_3+x_4+x_5=0 \\ & g_3(x)=-x_2-x_3+x_5+x_6=0 \\ & g_4(x)=10x_1-2x_3+3x_4-2x_5-16\le 0 \\ & g_5(x)=x_1+4x_3+x_5-10\le 0 \\ & 0\le x_1\le 12 \\ & 0\le x_2\le 18 \\ & 0\le x_3\le 5 \\ & 0\le x_4\le 12 \\ & 0\le x_5\le 1 \\ & 0\le x_6\le 16. \end{aligned}$$

For the nonlinear penalty function defined as

$$\begin{aligned} F(x,\rho,M) &= 10^{0.000001(f(x)-M)^2}-1 \\ &\quad +3000(\sum_{i=1}^{3} g_i(x)^2+\max\{g_4(x),0\}^2+\max\{g_5(x),0\}^2), \quad (9) \end{aligned}$$

let $\rho=3000,\ x^0=(0,10,0,0,0,10)\in X,\ a_1=-2000,\ b_1=140,\ M_1=-930$. After 2 iterations, the algorithm yields an ϵ-solution $x^2=(1.3568,86432,0.3276,$ $1.0292,0,8.9708)$ with $f(x^2)=124.000081$, which is almost the same as that in [20].

Now, by using the modified OPFA algorithm, we can solve the following class of 0-1 constrained nonlinear programming problems:

$$\begin{aligned} (PNL-01)\quad \min\quad & f_0(x) \\ \text{s.t.}\quad & g_i(x)\le 0,\ i=1,2,\cdots,m \\ & x\in B^n=\{0,1\}^n. \end{aligned}$$

Few exact methods can be used to solve this problem when $n>32$ ([8]). In order to use the modified OPFA algorithm, the 0-1 constraint $x\in B^n=\{0,1\}^n$ is replaced with an equivalent constraint

$$x_i^2-x_i=0,\ 0\le x_i\le 1,\ i=1,2,\cdots,n.$$

Then, the nonlinear penalty function is defined as

$$F(x,M)=(f_0(x)-M)^2+\beta\sum_{i=1}^{m}\max\{g_i(x),0\}^p+\beta\sum_{i=1}^{n}(x_i^2-x_i)^p. \qquad (10)$$

Let $Y=\{(x_1,x_2,\cdots,x_n)\mid 0\le x_i\le 1,\ i=1,2,\cdots,n\}$ in the following examples.

Example 3.6 Consider the following problem (Example 1 in [7]):

$$\begin{array}{lll}(P3.6) & \min & f(x)=x_1+x_2x_3-x_3\\ & \text{st.} & g_1(x)=-2x_1+3x_2+x_3-3\le 0\\ & & x_1,x_2,x_3=0 \text{ or } 1.\end{array}$$

The nonlinear penalty function is defined as

$$F(x,M)=(f_0(x)-M)^2+\beta\max\{g_1(x),0\}^4+\beta\sum_{i=1}^{3}(x_i^2-x_i)^4.$$

Let $\beta=10000,\ x^0=(0,0,0)\in X,\ a_1=-200,\ b_1=0,\ M_1=-100.$ For $k=1$ in the modified OPFA algorithm, we get an optimal solution $x^*=(0,0,1)$ and $f(x^*)=-1$, which is the same as that in [7].

Example 3.7 Consider the following problem (Example 6 in [7]):

$$\begin{array}{lll}(P3.7) & \min & f(x)=4x_1x_3x_4+6x_3x_4x_5+12x_1x_5-2x_1x_2-8x_1x_3\\ & \text{s.t.} & g_1(x)=8x_1x_4+4x_1x_3x_5+x_2x_3x_4+x_1x_5-5x_2x_5-5\le 0\\ & & g_2(x)=6x_3x_4+3x_1x_2x_3+2x_1x_2x_4-x_3x_5-4\le 0\\ & & g_3(x)=-2x_2x_3-9x_2x_3x_5+8\le 0\\ & & x_1,x_2,x_3,x_4,x_5=0 \text{ or } 1.\end{array}$$

The nonlinear penalty function is defined as

$$F(x,M)=(f_0(x)-M)^2+\beta\sum_{i=1}^{3}\max\{g_1(x),0\}^2+\beta\sum_{i=1}^{2}(x_i^2-x_i)^2.$$

Let $\beta=10000,\ a_1=-200,\ b_1=0,\ M_1=-100.$ We choose the initial point $x^0=(0.5,0.5,0.5,0.5,0.5)$. With one iteration, we get an optimal solution $x^*=(0,1,1,0,1)$ with $f(x^*)=0$.

Example 3.8 Consider the following problem (Problem 1 in [8]):

$$\begin{array}{lll}(P3.8) & \min & f(x)=\sum_{i=1}^{n}(x_i^2-1.8x_i)+0.81n\\ & \text{s.t.} & g_1(x)=\sum_{i=1}^{n}x_i-n+1\le 0\\ & & x_i=0 \text{ or } 1, i=1,2,\cdots,n.\end{array}$$

The nonlinear penalty function is defined as

$$F(x,M)=(f_0(x)-M)^2+\beta\max\{g_1(x),0\}^2+\beta\sum_{i=1}^{n}(x_i^2-x_i)^2.$$

Let $\beta = 10^8$, $a_1 = -2000$, $b_1 = 0.81n$, $M_1 = (a_1 + b_1)/2$. We choose the initial point $x^0 = (0.5, \cdots, 0.5)$. The numerical results are given in Table 3.5. We easily know that the optimal solution $x^* = (0, 1, 1, \cdots, 1)^T$ and the optimal objective value $f(x^*) = 0.01n - 0.8$.

Table 3.5

	n	4	8	16	32	48	64	128	256	380
the modified	Iter	1	1	1	1	1	1	1	1	1
OPFA algorithm	f^*	0.840	0.880	0.960	1.120	1.280	1.440	2.080	3.360	4.60
	Iter	1	1	1	1	2	2	2	2	2
Algorithm I	f^*	0.840	0.880	0.960	1.120	1.280	1.440	2.080	3.360	4.746
	Iter	5	4	4	4	4	4	-	-	-
Algorithm II	f^*	1.640	0.880	0.960	1.120	1.280	1.440	-	-	-

n: number of variables, Iter: number of iterations, f^*: the objective value.

In Table 3.5, f^* is the objective value of a feasible solution to (P3.8). The testing results in Table 3.5 show that the OPFA algorithm may get an optimal solution within fewer iterations than Algorithms I and II. For $n = 128, 256, 380$, Algorithm II can not obtain any feasible solution to (P3.8).

Example 3.9 Consider the following problem (Problem 2 in [8]):

$$(P3.9) \quad \min \quad f(x) = \sin(\pi + (\pi/n)\sum_{i=1}^{n} x_i)$$

$$\text{s.t.} \quad g_1(x) = \sum_{i=1}^{n} x_i - n/2 + 1 \leq 0$$

$$x_i = 0 \text{ or } 1, i = 1, 2, \cdots, n.$$

The nonlinear penalty function is defined as

$$F(x, M) = (f_0(x) - M)^2 + \beta \max\{g_1(x), 0\}^2 + \beta \sum_{i=1}^{n} (x_i^2 - x_i)^2.$$

Let $\beta = 1000000; b_1 = 0$, $M_1 = (a_1 + b_1)/2$, $x^0 = (0.5, \cdots, 0.5)$. Numerical results are given in Table 3.6.

Table 3.6

	n	8	16	32	48	64	80	100	128
the modified	$-a_1$	200	20000	20000	20000	60000	150000	150000	160000
OPFA	Iter	7	13	16	9	11	11	8	10
algorithm	f^*	-0.9239	-0.9808	-0.9952	-0.9979	-0.9988	-0.9969	-0.9980	-0.9997
	Iter	-	-	-	-	-	-	-	-
Algorithm I	f^*	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	Iter	-	-	-	-	-	-	-	-
Algorithm II	f^*	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

n: number of variables, Iter: number of iterations, f^*: the objective value.

We can obtain a good feasible solution by the modified OPFA algorithm in Table 3.6, when we try to choose several a_1 from -200 to -160000 for different n. But, we can not obtain a good solution by Algorithms I and II, even we try to choose many different penalty parameters ρ to F_1 and F_2. When the penalty parameter ρ changes, Algorithms I and II always keep the same solution and the same objective value 0 when it gets a solution after some iterations.

For $n > 30$, Mauricio and Maculan ([8]) pointed out that it is hard to solve (PNL-01). It is well-known that (PNL-01) is an NP-hard problem. However, the numerical results in Table 3.5 and Table 3.6 show that the modified OPFA algorithm is capable of solving larger scale (PNL-01) with $n > 30$. The above numerical experiments show that the number of iterations of the modified OPFA algorithm is few.

The previous numerical experiments show that the results obtained by the modified OPFA algorithm is better than or the same as Algorithm I and Algorithm II. Also the penalty functions with objective parameters may obtain better ϵ-solutions within much fewer iterations. Therefore, there exists a function $Q(t)$ such that the modified OPFA algorithm converges faster. This means that the OPFA algorithm can be efficient for solving constrained nonlinear programming.

4. **Conclusions.** This paper has presented a penalty function with objective parameters. We have proved that the penalty function is good under some mild conditions. Based on the penalty function, we have developed an OPFA algorithm to solve constrained nonlinear programming and proved its global convergence without differentiability. The OPFA algorithm differs from other penalty function algorithms that we need only to give a fixed penalty parameter in it. Numerical experiments show that the OPFA algorithm has good convergence.

Acknowledgments. We would like to thank the referees very much for their valuable comments and suggestions. Zhiqing Meng specially thanks to Professor Xiaoqi Yang for his help on researching penalty function.

REFERENCES

[1] J. V. Burke, *An exact penalization viewpoint of constrained optimization*, SIAM Journal on Controland Optimization, **29** (1991), 968–998.

[2] A. V. Fiacco and G. P. McCormick, "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," New York: Wiley, 1968.

[3] R. Fletcher, "Practical Method of Optimization," A Wiley-Interscience Publication, New York, 1981.

[4] R. Fletcher, *Penalty functions*, in "Mathematical Programming: The State of the Art" (Ed. A. Bachem, M. Grotschel, B. Korte), Springer, Berlin, (1983), 87–114.

[5] S. P. Han and O. L. Mangasrian, *Exact penalty function in nonlinear programming*, Mathematical Programming, **17** (1979), 251–269.

[6] X. X. Huang and X. Q. Yang, *Duality and exact penalization for vector optimization via augumentd Lagrangian*, Journal of Optimization Theory and Applications, **111** (2001), 615–640.

[7] H. Li, *An appproximate method for local optmal for nonlinear mixed integer programming problems*, Computers & Operation Reaserch, **19** (1992), 435–444.

[8] D. Mauricio and N. Maculan, *A boolean penalty method for zero-One nonlinear programming*, Jouranl of Global Optimaization, **16** (2000), 343–354.

[9] Z. Meng, Q. Hu and C. Dang, *An Objective Penalty Function Method for Nonlinear Programming*, Applied Mathematics Letter, **17** (2004), 683–689.

[10] M. Mongeau and A. Sartenaer, *Automatic decrease of the penalty parameter in exact penalty functions methods*, European Journal of Operational Research, **83** (1995), 686–699.

[11] D. D. Morrison, *Optimization by least squares*, SIAM Journal on Numerical Analysis, **5** (1968), 83–88.

[12] G. Di Pillo and L. Grippo, *An exact penalty function method with global conergence properties for nonlinear programming problems*, Mathemathical Programming, **36** (1986), 1–18.

[13] M. C. Pinar and S. A. Zenios, *On smoothing exact penalty functions for convex constraints optimization*, SIAM Journal on Optimization, **4** (1994), 486–511.

[14] E. Rosenberg, *Globally convergent algorithms for convex programming*, Mathematics of Operations Rresearch, **6** (1981), 437–443.

[15] E. Rosenberg, *Exact penalty functions and stability in locally Lipschitz programming*, Math. Programming, **30** (1984), 340–356.

[16] A. M. Rubinov, B. M. Glover and X. Q. Yang, *Extended lagrange and penalty functions in continuous optimization*, Optimization, **46** (1999), 327–351.

[17] A. M. Rubinov, B. M. Glover and X. Q. Yang, *Decreasing functions with applications to penalization*, SIAM Journal on Optimization, **10** (1999), 289–313.

[18] X. Q. Yang and X. X. Huang, *A nonlinear Lagrangian approach to constrained optimization problems*, SIAM Journal on Optimization, **11** (2001), 1119–1144.

[19] W. I. Zangwill, *Nonlinear programming via penalty function*, Manangement Science, **13** (1967), 334–358.

[20] S. A. Zenios, M. C. Pinar and R. S. Dembo, *A smooth penalty function algorithm for network-structured problems*, European Journal of Operational Research, **64** (1993), 258–277.

Received June 2008; 1st revision October 2008; final revision November 2008.

E-mail address: mengzhiqing@zjut.edu.cn
E-mail address: qyhu@fudan.edu.cn
E-mail address: mecdang@cityu.edu.hk