International Journal of Innovative Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012** ISSN 1349-4198
pp. 3717–3726

AN EVOLUTIONARY ALGORITHM FOR ZERO-ONE NONLINEAR OPTIMIZATION PROBLEMS BASED ON AN OBJECTIVE PENALTY FUNCTION METHOD

Zhiqing Meng[1], Chuangyin Dang[2], Min Jiang[1] and Rui Shen[1]

[1]College of Business and Administration
Zhejiang University of Technology
No. 18, Chaowang Road, Hangzhou 310023, P. R. China
{ mengzhiqing; jiangmin; shenrui }@zjut.edu.cn

[2]Department of System Engineering and Engineering Management
City University of Hong Kong
Tat Chee Avenue, Kowloon, Hong Kong SAR, P. R. China
mecdang@cityu.edu.hk

Received November 2010; revised March 2011

Abstract. *In many evolutionary algorithms, as fitness functions, penalty functions play an important role. In order to solve zero-one nonlinear optimization problems, a new objective penalty function is defined in this paper and some of its properties for solving integer nonlinear optimization problems are given. Based on the objective penalty function, an algorithm with global convergence for integer nonlinear optimization problems is proposed in theory. As a further application of the objective penalty function, a simple novel evolutionary algorithm is presented for solving zero-one nonlinear optimization problems. Numerical results on several examples show that the proposed evolutionary algorithm seems effective and efficient for some zero-one nonlinear optimization problems.*
Keywords: Evolutionary algorithm, Zero-one optimization problems, Objective penalty function, Fitness function

1. **Introduction.** It is well-known that evolutionary algorithms have been successfully applied to a variety of optimization problems, such as, constrained optimization problems [1], integrate linear programming [2] and mixed-integer bilevel programming problems [3]. Hu pointed out that evolutionary algorithms have many advantages [1]. Although the penalty function method is one of the most common approaches used in many evolutionary algorithms, the main drawback of the penalty function is that it is very difficult to control the penalty parameters which directly affect the efficiency and effectiveness of the algorithms [1]. 0-1 nonlinear programming is an NP-hard problem. To solve such a problem, a new simple evolutionary algorithm is proposed in this paper by introducing a new objective penalty function as a fitness function.

The problem to be considered in this paper is the following inequality constrained optimization problem:

$$\text{(COP)} \qquad \min \quad f_0(x) \qquad \text{s.t.} \quad f_i(x) \le 0, \quad i \in I = \{1, 2, \cdots, m\},$$

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 (COP). It is well-known that a penalty function for (COP) is defined as

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

with the corresponding penalty optimization problem for (COP) given by

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

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 (COP) for all $\rho \geq \rho^*$. In 1967, Zangwill [4] 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 (COP) given by

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

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

Exact penalty functions have attracted much attention from many researchers. In order to find out a better solution, the existing exact penalty function algorithms need to increase the penalty parameter to a very large value, and are also not differentiable. Hence, it is not easy for us to adopt an efficient algorithm, such as Newton Method, to solve constrained optimization problems via those exact penalty function methods. 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 methods with an objective penalty parameter have been discussed in [5, 6, 7, 8, 9], where the penalty function is defined as

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

with $p > 0$. 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 [5] considered the problem $\min\{f(x)|g(x) = 0\}$ and defined 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 [6, 7] discussed a similar type of $\phi(x, M)$. Furthermore, Burke [8] considered a more general type. Fiacco and McCormick [9] gave a general introduction of sequential unconstrained minimization techniques. Mauricio and Maculan [10] 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 et al. also studied an objective penalty function method as follows:

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

which is a smooth penalty function, while they did not consider integer nonlinear optimization problems in [12, 13]. Li et al. considered a pth power Lagrangian method for integer programming which differs from the objective penalty function method in [14, 15].

This paper will study an evolutionary algorithm with an objective penalty function for 0-1 optimization problems as follows. The remainder of the paper is organized as follows. In Section 2, a more general type of the penalty functions with some nice prospects to solve integer nonlinear optimization problems is presented. In Section 3, a new simple evolutionary algorithm with a new objective penalty function as a fitness function to solve 0-1 optimization problems is proposed and some numerical examples are given, which show that the number of iterations of the algorithm is small. In Section 4, the paper is concluded.

2. **An Objective Penalty Function Method.** In this section, the following integer optimization problem is considered:

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

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

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

$$\begin{cases} Q(t) = 0 & \text{if and only if } \ t \le 0, \\ Q(t) > 0 & \text{if and only if } \ t > 0, \\ Q(t_2) > Q(t_1) & \text{if and only if } \ t_2 > t_1 > 0, \end{cases}$$

and

$$\begin{cases} P(t) = 0 & \text{if and only if } \ t \le 0, \\ P(t) > 0 & \text{if and only if } \ t > 0, \\ P(t_2) > P(t_1) & \text{if and only if } \ t_2 > t_1 > 0. \end{cases}$$

For example, $P(t) = Q(t) = \max\{0, t\}^2$. The definitions of Q and P differ from those in [13]. Given such Q and P, a penalty function with objective parameters could be defined as

$$F(x, M) = Q(f_0(x) - M) + \sum_{i \in I} P(f_i(x)), \tag{1}$$

where the objective parameter $M \in R$. Then, based on some theorems for the penalty function $F(x, M)$, an algorithm to solve the integer problem with global convergence without any convexity conditions is presented.

The objective penalty function is defined as (1), whose corresponding integer optimization problem given by:

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

Given an M', if an optimal solution x^* to (IOP) is also an optimal solution to $(\mathrm{P}(M))$ for $\forall M < M'$, then $F(x, M)$ is called an exact objective penalty function.

It is easy to prove the following theorems on the penalty function.

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

Theorem 2.2. *Let $M_* = \min_{x \in X} f_0(x)$. Suppose that for some M, x^*_M is an optimal solution to* $(\mathrm{P}(M))$*. Then the following three assertions hold:*

(i) If $F(x^*_M, M) = 0$, then x^*_M is a feasible solution to (IOP) and $M_* \le f_0(x^*_M) \le M$.

(ii) If $F(x^*_M, M) > 0$ and x^*_M is not a feasible solution to (IOP), then $M < M_*$ and $f_0(x^*_M) < M_*$.

(iii) If $F(x_M^*, M) > 0$ and x_M^* is a feasible solution to (IOP), then x_M^* is an optimal solution to (IOP).

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

(ii) Let x^* be an optimal solution to (IOP). Since $F(x_M^*, M) > 0$,

$$0 < F(x_M^*, M) \leq F(x^*, M) = Q(f_0(x^*) - M).$$

According to the definition of Q, $M < f_0(x^*) = M_*$. If $f_0(x_M^*) \leq M$, then $f_0(x_M^*) < M < M_*$. If $f_0(x_M^*) > M$, then

$$\begin{aligned} 0 &< Q(f_0(x_M^*) - M) \leq F(x_M^*, M) \\ &\leq F(x^*, M) = Q(f_0(x^*) - M). \end{aligned}$$

Hence, $f_0(x_M^*) < M_*$.

(iii) From the given conditions, and the fact that x_M^* is a feasible solution, we have

$$0 < Q(f_0(x_M^*) - M) \leq F(x, M) = Q(f_0(x) - M), \quad \forall x \in X. \tag{2}$$

(2) implies that

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

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

Theorem 2.2 points out a possible way to solve (IOP). The exact objective parameter M required in Theorem 2.2 may exist, as shown in the following example.

Example 2.1. *Consider the problem:*

$$\begin{array}{lll} (P2.1) & \min & x_1^2 + x_2^2 \\ & \text{s.t.} & -x_1 \leq 0,\ -x_2 \leq 0,\ (x_1, x_2) \in Z^2. \end{array}$$

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

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

It is clear that $(x_1, x_2) = (0, 0)$ is an optimal solution to $(P(M))$ (with $M < 0$). Since $F((0, 0), M) > 0$, $(x_1, x_2) = (0, 0)$ is of course an optimal solution to $(P2.1)$ by Theorem 2.2. And M is the exact objective penalty parameter.

Based on Theorem 2.2, we will develop an algorithm to compute a globally optimal solution to (IOP) which differs from the proposed objective penalty function in [13] in detail. As it solves the problem $(P(M))$ sequentially, we name it as Integer Objective Penalty Function Algorithm (IOPFA for short).

IOPFA 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 $F(x^k, M_k) = 0$, let $a_{k+1} = a_k$, $b_{k+1} = f_0(x^k)$, $M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$, and go to Step 2. Otherwise, if $F(x^k, M_k) > 0$, go to Step 4.

Step 4: If x^k is not feasible to (IOP), let $b_{k+1} = b_k$, $a_{k+1} = \max\{f_0(x^k), M_k\}$, $M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$, and go to Step 2. Otherwise, stop and x^k is an optimal solution to (IOP).

In the IOPFA algorithm, it is assumed that we can always get $a_1 < \min_{x \in X} f_0(x)$. The convergence of the IOPFA algorithm is given in the following theorem. Let

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

which is called a Q-level set. We say that $S(L, f_0)$ is bounded if, for any given $L > 0$, $\{y^k\}$ is a convergent sequence with $y_k \to y^*$. In [12], the assumption that X is compact and connect is too strong. Therefore, the assumption is not adopted in this paper as we have better global convergence theorem as follows.

Theorem 2.3. *Let $M_* = \min_{x \in X} f_0(x)$. Suppose that 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 IOPFA algorithm.*

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

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

Proof: The sequence $\{a_k\}$ increases and $\{b_k\}$ decreases with

$$a_k \le M_* \le b_k, \quad k = 1, 2, \cdots \tag{3}$$

and

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

is proved according to the induction method as follows.

1. It is clear that from the IOPFA algorithm $a_1 \le M_* \le b_1$, $b_2 - a_2 \le \frac{b_1 - a_1}{2}$, $b_2 \le b_1$ and $a_2 \ge a_1$ for $k = 1$.
2. Suppose that (3) and (4) hold for some $k \ge 1$. Consider $k + 1$. In Step 3, we have $a_{k+1} = a_k$, $b_{k+1} = f_0(x^k)$, $M_{k+1} = \frac{a_{k+1} + b_{k+1}}{2}$ and x^k is feasible to (IOP). Thus,

$$a_{k+1} = a_k \le M_* \le f_0(x^k) = b_{k+1},$$

and

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

Therefore,

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

In Step 4, we have $b_{k+1} = b_k$, $a_{k+1} = \max\{f_0(x^k), M_k\}$. Thus,

$$a_{k+1} \ge M_k = \frac{a_k + b_k}{2} \ge \frac{a_k + a_k}{2} = a_k,$$

and

$$a_{k+1} \le M_* \le b_k = b_{k+1}.$$

Therefore,

$$b_{k+1} - a_{k+1} = b_k - \max\{f_0(x^k), M_k\} \le b_k - M_k = \frac{b_k - a_k}{2}.$$

With 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 (3), we have $a^* = b^* = M_*$. Therefore, $\{M_k\}$ also converges to M_*.

(i) If the IOPFA Algorithm terminates at the $\bar{k}$th iteration and the second situation of Step 4 occurs, by Theorem 2.2, $x^{\bar{k}}$ is an optimal solution to (IOP).

(ii) The sequence $\{x^k\}$ is bounded is proved first. Since x^k is an optimal solution to $\min_{x \in Y} F(x, M_k)$,

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

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

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

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 proved that

$$a_k \leq M_* \leq b_k, \quad k = 1, 2, \cdots$$

and all the sequences $\{a_k\}$, $\{b_k\}$ and $\{M_k\}$ converge to a^*. Let $k \to +\infty$, we obtain $a^* = M_*$. Let y^* be an optimal solution to (IOP). Then $M_* = f_0(y^*)$. Note that

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

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

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

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

Theorem 2.3 proves that the sequence $\{x^k\}$ generated by the IOPFA algorithm may converge to an optimal solution to (IOP) under some conditions. However, it is very difficult to find out an optimal solution to problem $\min\limits_{x \in Y} F(x, M_k)$ in Step 2 of the IOPFA algorithm. Hence, we will introduce a simple evolutionary algorithm to replace Step 2 of the IOPFA algorithm. This evolutional algorithm is efficient as shown in the numerical experiments in Section 3.

3. 0-1 Objective Penalty Function Algorithm.

In this section, we will modify IOPFA to obtain an evolutionary method. In the following proposed algorithm, the objective penalty function is employed as a fitness function and generation population has only one individual in every step. Hence, the algorithm has two mutation operators, in which Step 2 as a mutation operator solves an unconstraint optimal problem $(P(M))$, Step 3 to Step 4 as another mutation operator modifies the bound of a_k and b_k to obtain a new unconstraint optimal problem $(P(M))$.

Now, with the IOPFA algorithm, a simple evolutionary algorithm can be built to solve the following class of 0-1 nonlinear programming problems:

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

Few exact penalty function methods can be used to solve this problem when $n > 32$ ([10]). Now, suppose that f, g_i $(i \in I)$ are differentiable in R^n. The nonlinear penalty function is defined as

$$F(x, M) = Q(f(x) - M) + \sum_{i=1}^{m} P(g_i(x)). \tag{5}$$

$$P(M) \ \min \ F(x, M) \ \text{ s.t. } \ x \in B^n = \{0, 1\}^n.$$

The first mutation operator is defined as that any second individual $x_{(i+1)} \in B^n$ is obtained by the descend direction $-\nabla_{x_{(i)}(j)} F(x_{(i)}, M_k)$ of the first individual such that its objective function may decrease. The second mutation operator is defined as that new bounds of a_k and b_k are obtained by objective function of individual $x_{(i+1)} \in B^n$ such that its interval of objective function may shrink. This evolutional strategy is successful and efficient as shown in the following numerical experiments. In order to solve $P(M)$, a 0-1 OPFF algorithm is shown as follows.

0-1 OPFF Algorithm:

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

Step 2: Solve $\min_{x \in B^n} F(x, M_k)$. A solution x^k is obtained in steps from Step 3 through to Step 7.

Step 3: (Initialization) Let $t_{\max} > 0$ be the maximum evolutionary generation with $t = 0$, $\bar{f}$ and $pop(t) = \{\}$. Let N be the number of mutation.

Step 4: (Generation) Generate only one individual as initial population $pop(t)$, where the individual point $x_{(t)} = (x_{(t)}(1), x_{(t)}(2), \cdots, x_{(t)}(n))$
$\in B^n$ defined by
$a = randperm(n)$;
$b = randperm(n)$;
for $j = 1:1:n$
$x_{(t)}(j) = \mathrm{mod}(a(j) + j * b(j), 2)$;
end
Let $\bar{x} = x_{(t)}$ be put in $pop(t)$.

Step 5: (Fitness) Evaluating the individual $x_{(t)}$ as per fitness function defined by $F(x_{(t)}, M_k)$.

Step 6: (Mutation) Execute the mutation operator defined by
let $x_{(0)} = x_{(t)}$;
for $i = 1:1:N$
$c = randperm(n)$;
for $j = 1:1:c(1)$
if $\nabla_{x_{(i)}(j)} F(x_{(i)}, M_k) < 0$
$x_{(i+1)}(j) = \mathrm{mod}(x_{(i)}(j) + 3, 2)$;
end
end
if $x_{(i+1)}$ is a feasible solution and $f_0(x_{(i+1)}) < \bar{f}$
let $\bar{x} = x_{(i+1)}$ replace $\bar{x}$ in $pop(t)$, and $\bar{f} = f_0(x_{(i+1)})$.
end
end.

Step 7: If $t < t_{\max}$, select the next generation population $pop(t+1)$. Let $t = t+1$, go to Step 4. Otherwise, the best solution $x^k = \bar{x}$ obtained from $p(t_{\max})$, go to Step 8.

Step 8: (Mutation) If $F(x^k, M_k) = 0$, let $a_{k+1} = a_k$, $b_{k+1} = f(x^k)$, $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$, and go to Step 2. If $F(x^k, M_k) > 0$ and x^k is not feasible to (IOP), let $b_{k+1} = b_k$, $a_{k+1} = \max\{f(x^k), M_k\}$, $M_{k+1} = \frac{a_{k+1}+b_{k+1}}{2}$, and go to Step 2; otherwise, go to Step 9.

Step 9: (Termination) If $|b_{k+1} - a_{k+1}| \leq \epsilon$, stop and x^k is a solution to (IOP). Otherwise, let $k = k+1$, and go to Step 2.

If we can obtain an optimal solution in every Step 2 of the 0-1 OPFF algorithm, the 0-1 OPFF algorithm is convergent under some conditions by Theorem 2.3. Therefore, in order to show the efficiency of the 0-1 OPFF algorithm, the following examples of 0-1 nonlinear programming are solved with the proposed algorithm in Matlab.

Example 3.1. *Consider the following problem (Example 1 in* [11]*):*

$$\begin{aligned} (P3.1) \quad \min \quad & f(x) = x_1 + x_2 x_3 - x_3 \\ \text{s.t.} \quad & g_1(x) = -2x_1 + 3x_2 + x_3 - 3 \leq 0 \\ & x_1, x_2, x_3 = 0 \text{ or } 1. \end{aligned}$$

The nonlinear penalty function is defined as

$$F(x, M) = (f(x) - M)^2 + \beta \max\{g_1(x), 0\}^2.$$

Let $\beta = 10000$*,* $a_1 = -200$*,* $b_1 = 0$*,* $M_1 = -100$*. Choose* $t_{\max} = 5$ *as the maximum evolutionary generation and* $N = 3$ *as the number of mutation. For* $k = 1$ *in 0-1 OPFF algorithm, we get an optimal solution* $x^* = (0, 0, 1)$ *and* $f(x^*) = -1$*, which is the same as those in* [11, 13].

Example 3.2. *Consider the following problem (Example 6 in* [11]*):*

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

The nonlinear penalty function is defined as

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

Let $\beta = 10000$*,* $a_1 = -200$*,* $b_1 = 0$*,* $M_1 = -100$*. Choose* $t_{\max} = 5$ *as the maximum evolutionary generation and* $N = 5$ *as the number of mutation. With one iteration, an optimal solution* $x^* = (0, 1, 1, 0, 1)$ *with* $f(x^*) = 0$ *is obtained.*

Example 3.3. *Consider the following problem (Problem 1 in* [10]*):*

$$\begin{aligned} (P3.3) \quad \min \quad & f(x) = \sum_{i=1}^{n} (x_i^2 - 1.8x_i) + 0.81n \\ \text{s.t.} \quad & g_1(x) = \sum_{i=1}^{n} x_i - n + 1 \le 0 \\ & x_i = 0 \text{ or } 1, \quad i = 1, 2, \cdots, n. \end{aligned}$$

The nonlinear penalty function is defined as

$$F(x, M) = (f(x) - M)^2 + \beta \max\{g_1(x), 0\}^2.$$

Let $\beta = 10^8$*,* $a_1 = -2000$*,* $b_1 = 0.81n$*,* $M_1 = (a_1 + b_1)/2$*. Choose* $t_{\max} = 5$ *as the maximum evolutionary generation and* $N = n$ *as the number of mutation. The numerical results are given in Table 1. It is easily known that the optimal solution* $x^* = (0, 1, 1, \cdots, 1)^T$ *and the optimal objective value* $f(x^*) = 0.01n + 0.8$*.*

In Table 1, f^* *is the objective value of a feasible solution to (P3.3). The testing results in Table 1 show that the 0-1 OPFF algorithm may get an optimal solution within fewer*

Table 1. Numerical results of Example 3.3

	n	*256*	*380*	*500*	*1000*	*2000*	*4000*	*6000*	*10000*
The OPFF algorithm	*Iter*	*1*	*1*	*1*	*1*	*1*	*1*	*1*	*1*
	f^*	*3.36*	*4.6*	*5.8*	*10.8*	*20.8*	*40.8*	*60.8*	*100.8*

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

iterations than the OPFA Algorithms in [13]. *For $n > 380$, it is very difficult for the OPFA Algorithms in* [13] *to obtain optimal solution to (P3.3).*

Example 3.4. *Consider the following problem (Problem 2 in* [10]*):*

$$(P3.4) \quad \min \quad f(x) = \sin\left(\pi + (\pi/n)\sum_{i=1}^{n} x_i\right)$$

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

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

The nonlinear penalty function is defined as

$$F(x, M) = (f(x) - M)^2 + \beta \max\{g_1(x), 0\}^2.$$

Let $\beta = 10^8$, $a_1 = -2000$, $b_1 = 0$, $M_1 = (a_1 + b_1)/2$. Choose $t_{\max} = 5$ as the maximum evolutionary generation and $N = n$ as the number of mutation. Numerical results are given in Table 2.

Table 2. Numerical results of Example 3.4

	n	*100*	*300*	*600*	*1000*	*3000*
The OPFF algorithm	*Iter*	*1*	*1*	*4*	*8*	*3*
	f^*	*–0.999507*	*–0.999945*	*–0.999986*	*–0.999995*	*–0.999999*

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

As shown in Table 2, we can always obtain a good feasible solution with the 0-1 OPFF algorithm, even when different maximum evolutionary generation $t_{\max}$ and number of mutation N are chosen. Even when the penalty parameters M, β, a_1, b_1 change, the 0-1 OPFF algorithm can always keep and obtain the same solution which is good after several iterations.

Mauricio and Maculan ([10]) pointed out that for $n > 30$ 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 1 and Table 2 show that the 0-1 OPFF algorithm is capable of solving larger scale (PNL-01) with $n > 1000$. The above numerical experiments show that the number of iterations of the 0-1 OPFF algorithm is very small. We can also say that it is easy to control the value of penalty parameter in the 0-1 OPFF algorithm by numerical experiments.

4. **Conclusion.** Based on an objective penalty fitness function, the paper has presented a novel algorithm – IOPFA algorithm, to solve integer nonlinear optimization problems. Its global convergence without differentiability and convexity has been proved. Then, a 0-1 OPFF algorithm based on the objective penalty fitness function is proposed to solve zero-one nonlinear programming. Numerical experiments show that the 0-1 OPFF algorithm may be efficient for some 0-1 nonlinear optimization problems.

Acknowledgment. This research was supported by the National Natural Science Foundation of China under grant 10971193 and the Natural Science Foundation of Zhejiang Province with grant Y6090063. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] Y. Hu, Hybid-fitness function evolutionary algorithm based on simplex crossover and PSO mutation for constrained optimization problems, *International Journal of Pattern Recognition and Artificial Intelligence*, vol.23, no.1, pp.1-13, 2009.

[2] M.-J. Wang and M.-F. Yang, Application of signed distances method to integrate linear programming for fuzzy multi-objective project management, *International Journal of Innovative Computing, Information and Control*, vol.7, no.1, pp.237-252, 2011.

[3] N. I. Kalashnykova, V. V. Kalashnikov, S. Dempe and A. A. Franco, Application of a heuristic algorithm to mixed-integer bi-level programming problems, *International Journal of Innovative Computing, Information and Control*, vol.7, no.4, pp.1819-1829, 2011.

[4] W. I. Zangwill, Nonlinear programming via penalty function, *Manangement Science*, vol.13, no.5, pp.334-358, 1967.

[5] D. D. Morrison, Optimization by least squares, *SIAM Journal on Numerical Analysis*, vol.5, no.1, pp.83-88, 1968.

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

[7] R. Fletcher, Penalty functions, in *Mathematical Programming: The State of the Art*, A. Bachem, M. Grotschel and B. Korte (eds.), Berlin, Springer, 1983.

[8] J. V. Burke, An exact penalization viewpoint of constrained optimization, *SIAM Journal on Controland Optimization*, vol.29, no.4, pp.968-998, 1991.

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

[10] D. Mauricio and N. Maculan, A boolean penalty method for zero-one nonlinear programming, *Jouranl of Global Optimaization*, vol.16, no.4, pp.343-354, 2000.

[11] H. Li, An appproximate method for local optimal for nonlinear mixed integer programming problems, *Computers & Operation Research*, vol.19, no.3, pp.435-444, 1992.

[12] Z. Meng, Q. Hu, C. Dang and X. Yang, An objective penalty function method for nonlinear programming, *Applied Mathematics Letters*, vol.17, no.4, pp.683-689, 2004.

[13] Z. Meng, Q. Hu and C. Dang, A penalty function algorithm with objective parameters for nonlinear mathematical programming, *Journal of Industrial and Management Optimization*, vol.5, no.3, pp.585-601, 2009.

[14] D. Li and D. J. White, pth power lagrangian method for integer programming, *Annals of Operations Research*, vol.98, no.1-4, pp.151-170, 2000.

[15] D. Li and X. L. Sun, Success guarantee of dual search in integer programming: p-th power lagrangian method, *Journal of Global Optimization*, vol.18, no.3, pp.235-253, 2000.