Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2011, Article ID 402437, 13 pages
doi:10.1155/2011/402437

Research Article

An Auxiliary Function Method for Global Minimization in Integer Programming

Hongwei Lin, Yuping Wang, and Xiaoli Wang

School of Computer Science and Technology and Department of Applied Mathematics,
School of Science, Xidian University, Xi'an, Shaanxi 710071, China

Correspondence should be addressed to Hongwei Lin, linhongwei_jt@hotmail.com and Yuping Wang, ywang@xidian.edu.cn

Received 22 July 2011; Revised 10 October 2011; Accepted 10 October 2011

Academic Editor: Blas M. Vinagre

An auxiliary function method is proposed for finding the global minimizer of integer programming problem. Firstly, we propose a method to transform the original problem into an integer programming with box constraint, which does not change the properties of the original problem. For the transformed problem, we propose an auxiliary function to escape from the current local minimizer and to get a better one. Then, based on the proposed auxiliary function, a new algorithm to find the global minimizer of integer programming is proposed. At last, numerical results are given to demonstrate the effectiveness and efficiency of the proposed method.

1. Introduction

Consider the following integer programming problem:

$$\begin{aligned} \min \quad & f(x), \\ \text{s.t.} \quad & g_i(x) \leqslant 0, \quad i = 1, 2, \ldots, l, \\ & h_i(x) = 0, \quad i = l+1, l+2, \ldots, m, \\ & x \in Z^n, \end{aligned} \tag{P}$$

where Z^n is the n-dimensional integer set, and the set $S = \{x \in Z^n \mid g_i(x) \leqslant 0, i = 1, 2, \ldots, l, h_i(x) = 0, i = l+1, l+2, \ldots, m\}$ is bounded.

Many real-life applications can be modeled as problem (P), such as production planning, scheduling, and operations research problem. The objective functions of most of the problems are nonlinear and have more than one local optimal solutions over feasible region S. This requires the global optimization techniques to find the best solution amongst multiple local optima.

Since integer programming problems are generally NP-hard, there are no efficient algorithms with polynomial-time complexity for solving them. Thus, many approximate algorithms have been rapidly developed in recent years, such as greedy search (see [1–4]), simulated annealing (see [5, 6]), genetic algorithm (see [7–9]), tabu search (see [10, 11]), and discrete filled function techniques (see [12–17]). The discrete filled function method is one of the more recently developed global optimization methods to solve integer programming problems. Once a local minimizer has been found by a local search method, the discrete filled function method introduces an auxiliary function to escape from the current local minimizer and to get a better one.

At present, the discrete filled function methods mainly focus on the unconstrained integer programming problems, and most of the existing filled functions contain parameters which are needed to adjust. Thus, solving general integer programming problems is difficult, and solving the unconstrained integer programming problems needs much computation. In this paper, we propose an auxiliary function method to solve problem (P). The proposed auxiliary function has no parameters, and the computation of the proposed method is relatively small.

The remainder of this paper is organized as follows: Section 2 gives some useful notations and definitions. In Section 3, we propose a method to transform the original problem into an integer programming with box constraint. Based on the transformed problem, an auxiliary function is proposed in Section 4, and its properties are also analyzed. An algorithm for solving problem (P) is proposed in Section 5 with several numerical experiments. Some concluding remarks are given in Section 6.

2. Preliminaries

In this section, some useful notations and definitions are listed firstly. By estimating the bound of S, we can find a box region $\Omega = \prod_{i=1}^{n}[L_i, U_i]$ with $S \subset \Omega$. In most practical problems, Ω can be easily estimated or is usually given.

Let M be a constant which satisfies

$$M \geqslant \max_{z\in S} f(z) + 1, \tag{2.1}$$

and let

$$D = \{\pm e_i,\ i = 1, 2, \ldots, n\} \tag{2.2}$$

be a direction set, where e_i is an n-dimensional unit vector in which the ith component is 1, and the others are 0.

Definition 2.1. x is said to be a vertex of $Z^n \bigcap \Omega$, if and only if, for all $d \in D$, $x + d \in Z^n \bigcap \Omega$ and $x - d \notin Z^n \bigcap \Omega$, or $x + d \notin Z^n \bigcap \Omega$ and $x - d \in Z^n \bigcap \Omega$ hold.

The set of vertices of $\Omega \bigcap Z^n$ is denoted as V.

Definition 2.2 (see [18]). For all $x \in Z^n \bigcap \Omega$, the neighborhood of x is denoted as $N(x) = \{x\} \bigcup \{w \in Z^n \bigcap \Omega \mid w = x + d,\ d \in D\}$.

Definition 2.3 (see [18]). A point $x \in Z^n \bigcap \Omega$ is called a local minimizer (maximizer) of $f(x)$ over $Z^n \bigcap \Omega$ if for all $y \in N(x)$, $f(y) \geqslant f(x)(f(y) \leqslant f(x))$ holds; a point $x \in Z^n \bigcap \Omega$ is called a global minimizer (maximizer) of $f(x)$ over $Z^n \bigcap \Omega$ if for all $y \in Z^n \bigcap \Omega$, $f(y) \geqslant f(x)(f(y) \leqslant f(x))$ holds. In addition, if equality does not hold, x is called a strictly local (global) minimizer (maximizer) of $f(x)$.

For finding a local minimizer of integer programming, the following local search algorithm is taken in most cases.

Local Search Algorithm (see [12])

Step 1. Choose an initial point $x \in Z^n \bigcap \Omega$.

Step 2. If x is a local minimizer of $f(x)$, then stop; otherwise, find another point $y \in N(x)$ which minimizes $f(x)$ over $N(x)$.

Step 3. Set $x := y$, and go to Step 2.

3. Transformation of the Original Problem

The local search algorithm usually requires that the search region should be a connected one. However, the feasible region of the original problem may not be connected, so we need to transform the original problem into a simple problem with a connected search region. In order to transform the original problem, we need to define three unary functions firstly:

$$g(t) = \begin{cases} 0, & t > 0, \\ 1, & t \leqslant 0, \end{cases}$$
$$h(t) = \begin{cases} 1, & t = 0, \\ 0, & t \neq 0, \end{cases} \tag{3.1}$$
$$s(t) = \begin{cases} 0, & t = m, \\ 1, & t < m, \end{cases}$$

where m is the number of constraints.

Consider the following function:

$$F(x) = f(x) + \left(M - f(x)\right) \times s\left(\sum_{i=1}^{l} g\left(g_i(x)\right) + \sum_{i=l+1}^{m} h(h_i(x)) \right) \tag{3.2}$$

and the problem

$$\begin{aligned} \min \quad & F(x), \\ \text{s.t.} \quad & x \in \Omega \bigcap Z^n, \end{aligned} \tag{TP}$$

where M is taken to satisfy (2.1). The following theorem will show that problem (TP) does not change the local minimizers of problem (P).

Theorem 3.1. *If x is a local minimizer of problem (P), then x is a local minimizer of problem (TP); in turn, if x is a local minimizer of the problem (TP), then x is a local minimizer of the problem (P), or $F(x) = M$.*

Proof. If x is a local minimizer of the problem (P), then for all $y \in N(x) \bigcap S$, one has $F(y) = f(y) \geqslant f(x) = F(x)$; for $y = N(x) \setminus S$ and by the definition of M, one has $F(y) = M > \max_{z \in S} f(z) \geqslant f(x)$. So $F(y) \geqslant F(x)$ for all $y \in N(x)$, namely, x is a local minimizer of the problem (TP).

If x is a local minimizer of the problem (TP), then for all $y \in N(x)$, one has $F(y) > F(x)$. If $F(x) < M$, namely, $x \in S$, then one has $f(y) = F(y) > F(x) = f(x)$ for all $y \in N(x) \bigcap S$. So x is a local minimizer of the problem (P). □

Theorem 3.1 shows that problem (P) and problem (TP) have the same minimizers except the infeasible points. If we can find a global minimizer of problem (TP), then problem (P) is solved.

In most cases, using the local search algorithm to solve problem (TP) will trap in a local minimizer of problem (TP). We hope to find a method to escape from a local minimizer and to get a better one. Section 4 will give an auxiliary function for solving this problem.

4. Auxiliary Function and Its Properties Proof

Let x_1^* be the current best local minimizer of problem (TP), and define the following two integer sets:

$$\begin{aligned} S_1 &= \left\{ x \in \Omega \bigcap Z^n \mid F(x) \geqslant F\left(x_1^*\right) \right\}, \\ S_2 &= \left\{ x \in \Omega \bigcap Z^n \mid F(x) < F\left(x_1^*\right) \right\}. \end{aligned} \tag{4.1}$$

To begin with, a unary function is given as follows:

$$w(t) = \begin{cases} 1, & t \geqslant 0, \\ t, & t < 0. \end{cases} \tag{4.2}$$

Consider the following auxiliary function:

$$AF\left(x, x_1^*\right) = \frac{1}{1 + \|x - x_1^*\|} \times w\left(\left(F(x) - F\left(x_1^*\right)\right)\left(1 + \|x - x_1^*\|\right)\right), \tag{4.3}$$

and its properties are analyzed as follows.

Theorem 4.1. *$AF(x, x_1^*)$ has no local minimizer in set $S_1 \setminus V$.*

By definition of S_1, for all $x \in S_1 \setminus V$, one has $F(x) \geqslant F(x_1^*)$, then

$$AF(x, x_1^*) = \frac{1}{1 + \|x - x_1^*\|} > 0. \tag{4.4}$$

In order to prove that x is not a local minimizer of $AF(x, x_1^*)$, we only need to find a point y_0 in $N(x)$ such that $AF(y_0, x_1^*) < AF(x, x_1^*)$.

If there exists $y_0 \in N(x)$ satisfies $F(y_0) < F(x_1^*)$, then

$$AF(y_0, x_1^*) = F(y_0) - F(x_1^*) < 0 < AF(x, x_1^*). \tag{4.5}$$

For all $y \in N(x)$ with $F(y) \geqslant F(x_1^*)$, since $x \notin V$, there exists $y_0 \in N(x)$ such that $\|y_0 - x_1^*\| > \|x - x_1^*\|$, then

$$AF(y_0, x_1^*) = \frac{1}{1 + \|y_0 - x_1^*\|} < \frac{1}{1 + \|x - x_1^*\|} = AF(x, x_1^*). \tag{4.6}$$

Therefore, $AF(x, x_1^*)$ has no minimizer in the set $S_1 \setminus V$.

Theorem 4.2. *If there exists another local minimizer x_2^* of $F(x)$ such that $F(x_2^*) < F(x_1^*)$, then x_2^* is also a local minimizer of $AF(x, x_1^*)$.*

Proof. By $F(x_2^*) < F(x_1^*)$, one has

$$AF(x_2^*, x_1^*) = F(x_2^*) - F(x_1^*) < 0. \tag{4.7}$$

□

Since x_2^* is a local minimizer of $F(x)$, one has $F(y) \geqslant F(x_2^*)$ for all $y \in N(x_2^*)$.

When $F(y) \geqslant F(x_1^*)$, and by (4.3), one has

$$AF(y, x_1^*) = \frac{1}{1 + \|x - x_1^*\|} > 0 > AF(x_2^*, x_1^*). \tag{4.8}$$

When $F(y) < F(x_1^*)$, and by (4.3), one has

$$AF(y, x_1^*) = F(y) - F(x_1^*) \geqslant F(x_2^*) - F(x_1^*) = AF(x_2^*, x_1^*). \tag{4.9}$$

Namely, $AF(y, x_1^*) \geqslant AF(x_2^*, x_1^*)$ for all $y \in N(x_2^*)$. So, x_2^* is a local minimizer of $AF(x, x_1^*)$.

Theorem 4.3. *Any local minimizer $\overline{x}$ of $AF(x, x_1^*)$ is in set S_2 or a vertex of $\Omega \bigcap Z^n$. If $AF(\overline{x}, x_1^*) < 0$, then $\overline{x}$ is not only a local minimizer of $F(x)$ which is better than x_1^*, but also a local minimizer of $f(x)$ over S.*

Proof. Let $\overline{x}$ be a local minimizer of $AF(x,x_1^*)$, and suppose that $\overline{x}$ is not a vertex of $\Omega\bigcap Z^n$, then there exists an integer $i_0 \in [1,n]$ (n is the dimension of problem (P)), such that $\overline{x} \pm e_{i_0} \in \Omega\bigcap Z^n$. It can be proved that $F(\overline{x}) < F(x_1^*)$.

In fact, suppose that $F(\overline{x}) \geqslant F(x_1^*)$ holds.

If there exists $y_1 \in \{\overline{x} \pm e_{i_0}\}$ satisfies $F(y_1) < F(x_1^*)$, then

$$AF(\overline{x},x_1^*) = \frac{1}{1+\|\overline{x}-x_1^*\|} > 0 > F(y_1) - F(x_1^*) = AF(y_1,x_1^*). \tag{4.10}$$

Otherwise, there exists $y_2 \in \{\overline{x} \pm e_{i_0}\}$ satisfies $\|y_2 - x_1^*\| > \|\overline{x}-x_1^*\|$, then

$$AF(\overline{x},x_1^*) = \frac{1}{1+\|\overline{x}-x_1^*\|} > \frac{1}{1+\|y_2-x_1^*\|} = AF(y_2,x_1^*), \tag{4.11}$$

which contradicts with the fact that $\overline{x}$ is a local minimizer of $AF(x,x_1^*)$, so $\overline{x} \in S_2$.

If $AF(\overline{x},x_1^*) < 0$, we have $F(\overline{x}) < F(x_1^*)$. In fact, suppose that $F(\overline{x}) \geqslant F(x_1^*)$, then

$$AF(\overline{x},x_1^*) = \frac{1}{1+\|\overline{x}-x_1^*\|} > 0, \tag{4.12}$$

which contradicts with $AF(\overline{x},x_1^*) < 0$.

Suppose that $\overline{x}$ is not a local minimizer of $F(x)$, then there exists $y_3 \in N(\overline{x})$ such that $F(\overline{x}) > F(y_3)$ holds. By the definition of $AF(x,x_1^*)$, one has $AF(\overline{x},x_1^*) = F(\overline{x}) - F(x_1^*) > F(y_3) - F(x_1^*) = AF(y_3,x_1^*)$, which contradicts with the fact that $\overline{x}$ is a local minimizer of $AF(x,x_1^*)$.

By the definition of $F(x)$, we know that $F(\overline{x}) < F(x_1^*) \leqslant M$, then $\overline{x}$ is a feasible point. By Theorem 3.1, we know that $\overline{x}$ is a local minimizer of $f(x)$. The proof is completed. □

From Theorems 4.2 and 4.3, it can be seen that S_2 contains a minimizer of $AF(x,x_1^*)$, and when the minimizer $\overline{x}$ of $AF(x,x_1^*)$ satisfies $AF(\overline{x},x_1^*) < 0$, it is also a local minimizer of $F(x)$ and better than the current local minimizer. It can also be seen that even if x_1^* is an infeasible local minimizer, a feasible local minimizer of $f(x)$ will be found by an iteration, and there will be no infeasible local minimizer in the subsequent iterations.

5. Global Optimization Algorithm Based on New Auxiliary Function

5.1. Global Optimization Algorithm

Based on the theoretical analysis in Section 4, a global optimization algorithm to solve problem (P) is given as follows.

Algorithm 5.1.

Step 0. Take an initial point $x_0 \in \Omega \bigcap Z^n$, a sufficiently large number M.

Step 1. Use the local search algorithm in Section 2 to $F(x)$ at x_0, obtain a local minimizer x_1^* of $F(x)$, and set $k := 1$.

Step 2. Let $D = \{d_1, d_2, \ldots, d_{2n}\}$, where $d_i = e_i$, $d_{n+i} = -e_i$, $i = 1, 2, \ldots, n$.

Step 3. Construct the auxiliary function

$$AF(x, x_k^*) = \frac{1}{1 + \|x - x_k^*\|} \times w\left(\left(F(x) - F(x_k^*)\right)\left(1 + \|x - x_k^*\|\right)\right). \tag{5.1}$$

Let $x_{k,i} = x_k^* + d_i$, set $MS = \emptyset$, and $i := 1$.

Step 4. Use the local search algorithm in Section 2 to $AF(x, x_k^*)$ with the initial point $x_{k,i}$ and obtain a local minimizer $x_{k,i}^*$ of $AF(x, x_k^*)$ over $\Omega \bigcap Z^n$. If $F(x_{k,i}^*) < F(x_k^*)$, set $MS = MS \bigcup \{x_{k,i}^*\}$. Go to Step 5.

Step 5. If $i \geqslant 2n$, go to Step 6; otherwise, $i =: i + 1$; go to Step 4.

Step 6. If $MS = \emptyset$, go to Step 7; otherwise, find a point in $MS \neq \emptyset$ with the smallest function value of $F(x)$ and denote it as x_{k+1}^*. Set $k := k + 1$, and go back to Step 3.

Step 7. Output x_k^*.

Remarks on the Algorithm

(1) In Step 4, if a local minimizer x_{k+1}^* of $AF(x, x_k^*)$ is found, then x_{k+1}^* is a local minimizer of the problem (P). It is different from the discrete filled function method in which the local search algorithm should be employed to $F(x)$ and $AF(x, x_k^*)$ repeatedly. Therefore, the proposed algorithm can reduce the computational cost.

(2) From the above theorems, we can see that although x_1^* may be an infeasible point, $\{x_i^*, i \geqslant 2\}$ are all feasible local minimizers of $f(x)$ over S.

(3) Generally, M must be a sufficiently large number. In the implementation of the proposed algorithm, we first estimate an upper bound B_{upper} of $f(x)$ over S, then take $M = B_{\text{upper}} + 1$ or directly the upper bound B_{upper} if it is big enough.

5.2. Numerical Experiment

We apply the proposed algorithm to solve the following test problems.

Problem 1.

$$\begin{aligned}\min \quad & f(x) = (x_1-10)^3 + (x_2-20)^3, \\ \text{s.t.} \quad & -(x_1-5)^2 - (x_2-5)^2 + 100 \leqslant 0 \\ & -x_1 + 10 \leqslant 0, \\ & -x_2 + 5 \leqslant 0, \\ & x \in \Omega = \{x \mid 0 \leqslant x_i \leqslant 100,\ i = 1,\ 2\}.\end{aligned} \tag{5.2}$$

The box Ω has 10201 points. The global minimizer is $x^* = [15,5]^T$ with $f(x^*) = -3250$.

Problem 2.

$$\begin{aligned}\min \quad & f(x) = -2^4 \times \prod_{i=1}^{4} \frac{x_i}{100}, \\ \text{s.t.} \quad & \sum_{i=1}^{4} \left(\frac{x_i}{100}\right)^2 = 1, \\ & x \in \Omega = \{x \mid 0 \leqslant x_i \leqslant 100, i = 1,2,3,4\}.\end{aligned} \tag{5.3}$$

The box Ω has 104060401 points. The global minimizer is $x^* = [50,50,50,50]^T$ with $f(x^*) = -1$.

Problem 3.

$$\begin{aligned}\min \quad & f(x) = -25(x_1-2)^2 - (x_2-2)^2 - (x_3-1)^2 - (x_4-4)^2 - (x_5-1)^2 - (x_6-4)^2, \\ \text{s.t.} \quad & -(x_3-3)^2 - x_4 + 4 \leqslant 0, \\ & -(x_5-3)^2 - x_6 + 4 \leqslant 0, \\ & x_1 - 3x_2 - 2 \leqslant 0, \\ & -x_1 + x_2 - 2 \leqslant 0, \\ & x_1 + x_2 - 6 \leqslant 0, \\ & -x_1 - x_2 + 2 \leqslant 0, \\ & 0 \leqslant x_1 \leqslant 6, \quad 0 \leqslant x_2 \leqslant 8, \quad 0 \leqslant x_3 \leqslant 5, \\ & 0 \leqslant x_4 \leqslant 6, \quad 0 \leqslant x_5 \leqslant 10, \quad 0 \leqslant x_6 \leqslant 10.\end{aligned} \tag{5.4}$$

The box Ω has 320166 points. The global minimizer is $x^* = (5,1,5,0,5,10)^T$ with $f(x^*) = -310$.

Problem 4 (Colville's function).

$$\begin{aligned} \min \quad & f(x) = 100\left(x_2 - x_1^2\right)^2 + (1-x_1)^2 + 90\left(x_4 - x_3^2\right)^2 + (1-x_3)^2 \\ & \quad + 10.1\left[(x_2-1)^2 + (x_4-1)^2\right] + 19.8(x_2-1)(x_4-1), \\ \text{s.t.} \quad & -10 \le x_i \le 10, x_i \text{ is integer}, \quad i = 1,2,3,4. \end{aligned} \tag{5.5}$$

This box-constrained problem has 1.94481 × 10^5 feasible points. The global minimizer is $x^* = [1,1,1,1]^T$ with $f(x^*) = 0$.

Problem 5 (Goldstein and Price's function).

$$\begin{aligned} \min \quad & f(x) = g(x)h(x), \\ \text{s.t.} \quad & x_i = \frac{y_i}{1000}, \quad -2000 \leqslant y_i \leqslant 2000, \quad y_i \text{ is integer}, \quad i = 1,2, \end{aligned} \tag{5.6}$$

where $g(x) = 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 16x_1x_2 + 3x_2^2)$ and $h(x) = 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$.

This box constrained problem has 1.6008001×10^7 feasible points. The global minimizer is $x^* = [0,-1]^T$ with $f(x^*) = 3$.

Problem 6 (Beale's function).

$$\begin{aligned} \min \quad & f(x) = [1.5 - x_1(1-x_2)]^2 + \left[2.25 - x_1\left(1 - x_2^2\right)\right]^2 + \left[2.65 - x_1\left(1 - x_2^3\right)\right]^2, \\ \text{s.t.} \quad & x_i = \frac{y_i}{1000}, \quad -10000 \leqslant y_i \leqslant 10000, \quad y_i \text{ is integer}, \quad i = 1,2. \end{aligned} \tag{5.7}$$

This box-constrained problem has 4.00040001 × 10^8 feasible points. The global minimizer is $x^* = [3, 0.5]^T$ with $f(x^*) = 0$.

Problem 7 (Rosenbrock's function).

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{24}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1-x_i)^2\right], \\ \text{s.t.} \quad & -5 \leqslant x_i \leqslant 5, \quad x_i \text{ is integer}, \quad i = 1,2,\ldots,25. \end{aligned} \tag{5.8}$$

This box-constrained problem has 1.08347 × 10^{26} feasible points. The global minimizer is $x^* = [1,1,\ldots,1]^T$ with $f(x^*) = 0$.

We apply the proposed algorithm to solve the above problems, and numerical results are shown in Tables 1 to 5. In these tables, x_0 denotes the initial point; x^* denotes the last local minimizer of the original objective function; E_{f+g} denotes the total number evaluations of original function and auxiliary (filled) function.

Table 1: Numerical results for Problems 1–3.

Problem	x_0	M	x^*	E_{f+g}
1	$[25, 25]$	10^8	$[15, 5]^T$	2632
	$[50, 50]$	10^8	$[15, 5]^T$	2832
	$[75, 75]$	10^8	$[15, 5]^T$	3032
2	$[25, 25, 25, 25]$	1	$[50, 50, 50, 50]^T$	12522
	$[50, 50, 50, 50]$	1	$[50, 50, 50, 50]^T$	9172
	$[75, 75, 75, 75]$	1	$[50, 50, 50, 50]^T$	13629
3	$[0, 0, 0, 0, 0, 0]$	1	$[5, 1, 5, 0, 5, 10]^T$	93307
	$[3, 4, 2, 3, 5, 5]$	1	$[5, 1, 5, 0, 5, 10]^T$	62671
	$[6, 8, 5, 6, 10, 10]$	1	$[5, 1, 5, 0, 5, 10]^T$	96443

Table 2: Numerical results of Problem 4 ($M = 10^8$).

Algorithm	x_0	x^*	E_{f+g}
AFM	$[1, 1, 0, 0]^T$	$[1, 1, 1, 1]^T$	3936
	$[1, 1, 1, 1]^T$	$[1, 1, 1, 1]^T$	3324
	$[-10, 10, -10, 10]^T$	$[1, 1, 1, 1]^T$	4977
	$[-10, -5, 0, 5]^T$	$[1, 1, 1, 1]^T$	3454
	$[-10, 0, 0, -10]^T$	$[1, 1, 1, 1]^T$	3571
	$[0, 0, 0, 0]^T$	$[1, 1, 1, 1]^T$	3539
DFMA	$[1, 1, 0, 0]^T$	$[1, 1, 1, 1]^T$	9153
	$[1, 1, 1, 1]^T$	$[1, 1, 1, 1]^T$	9123
	$[-10, 10, -10, 10]^T$	$[1, 1, 1, 1]^T$	14543
	$[-10, -5, 0, 5]^T$	$[1, 1, 1, 1]^T$	9248
	$[-10, 0, 0, -10]^T$	$[1, 1, 1, 1]^T$	9285
	$[0, 0, 0, 0]^T$	$[1, 1, 1, 1]^T$	9162
DFMB	$[1, 1, 0, 0]^T$	$[1, 1, 1, 1]^T$	6523
	$[1, 1, 1, 1]^T$	$[1, 1, 1, 1]^T$	6498
	$[-10, 10, -10, 10]^T$	$[1, 1, 1, 1]^T$	8553
	$[-10, -5, 0, 5]^T$	$[1, 1, 1, 1]^T$	6701
	$[-10, 0, 0, -10]^T$	$[1, 1, 1, 1]^T$	6655
	$[0, 0, 0, 0]^T$	$[1, 1, 1, 1]^T$	6530

Table 1 gives the results obtained by the proposed algorithm for Problems 1–3.

Problems 1–3 are with nonlinear objective function and nonlinear constraint functions. It is hard for an algorithm to solve. However, it can be seen from Table 1 that for Problems 1–3, the proposed algorithm can find the global optimal solutions for all these three problems using relatively small number of function evaluations.

In order to demonstrate the efficiency of the proposed algorithm, we compare the proposed algorithm with other two discrete filled function methods DFMA and DFMB ([14, 19]) for Problems 4–7. The results are summarized in Tables 2–5, where DFMA and DFMB denote the discrete filled function methods proposed in [14, 19], respectively, and AFM denotes the proposed auxiliary function algorithm. In implementation of the algorithm, $F(x) = f(x)$ for Problems 4–7.

For Problem 4, it can be seen from Table 2 that all three algorithms can find global optimal solution, but the number of the function evaluations used by proposed algorithm AFM is only about 1/3 of that used by DFMA and only about half of that used by DFMB. For

Table 3: Numerical results of Problem 5 ($M = 10^8$).

Algorithm	x_0	x^*	E_{f+g}
AFM	$[2,-2]^T$	$[0,-1]^T$	118743
	$[0,-1]^T$	$[0,-1]^T$	83181
	$[-2,-2]^T$	$[0,-1]^T$	91850
	$[-0.5,-1]^T$	$[0,-1]^T$	86024
	$[1,-1.5]^T$	$[0,-1]^T$	113741
	$[1,-1]^T$	$[0,-1]^T$	91698
DFMA	$[2,-2]^T$	$[0,-1]^T$	268489
	$[0,-1]^T$	$[0,-1]^T$	254444
	$[-2,-2]^T$	$[0,-1]^T$	270930
	$[-0.5,-1]^T$	$[0,-1]^T$	254444
	$[1,-1.5]^T$	$[0,-1]^T$	267978
	$[1,-1]^T$	$[0,-1]^T$	254444
DFMB	$[2,-2]^T$	$[0,-1]^T$	176397
	$[0,-1]^T$	$[0,-1]^T$	170351
	$[-2,-2]^T$	$[0,-1]^T$	175828
	$[-0.5,-1]^T$	$[0,-1]^T$	171831
	$[1,-1.5]^T$	$[0,-1]^T$	173889
	$[1,-1]^T$	$[0,-1]^T$	173334

Table 4: Numerical results of Problem 6 ($M = 10^8$).

Algorithm	x_0	x^*	E_{f+g}
AFM	$[10,-10]^T$	$[3,0.5]^T$	815212
	$[0.997,-6.867]^T$	$[3,0.5]^T$	813561
	$[0,-1]^T$	$[3,0.5]^T$	414793
	$[1,1]^T$	$[3,0.5]^T$	438751
	$[-2,2]^T$	$[3,0.5]^T$	537546
	$[0,0]^T$	$[3,0.5]^T$	410793
DFMA	$[10,-10]^T$	$[3,0.5]^T$	2189978
	$[0.997,-6.867]^T$	$[3,0.5]^T$	2192230
	$[0,-1]^T$	$[3,0.5]^T$	2197097
	$[1,1]^T$	$[3,0.5]^T$	1356906
	$[-2,2]^T$	$[3,0.5]^T$	1356906
	$[0,0]^T$	$[3,0.5]^T$	2527711
DFMB	$[10,-10]^T$	$[3,0.5]^T$	1430248
	$[0.997,-6.867]^T$	$[3,0.5]^T$	1431740
	$[0,-1]^T$	$[3,0.5]^T$	1439584
	$[1,1]^T$	$[3,0.5]^T$	830822
	$[-2,2]^T$	$[3,0.5]^T$	829445
	$[0,0]^T$	$[3,0.5]^T$	911090

Problems 5 and 6, we can see from Tables 3-4 that all three algorithms can also find global optimal solutions, but the number of function evaluations used by DFMA is more than twice of that used by AFM, and the number of function evaluations used by DFMB is more than 1.5 times of that used by AFM. For Problem 7, it can be seen from Table 5 that the number of function evaluations used by AFM is also much smaller than those used by DFMA and

Table 5: Numerical results of Problem 7 ($M = 10^6$).

Algorithm	x_0	x^*	E_{f+g}
AFM	$[0,\ldots,0]^T$	$[1,\ldots,1]^T$	873553
	$[3,\ldots,3]^T$	$[1,\ldots,1]^T$	557111
	$[-5,\ldots,-5]^T$	$[1,\ldots,1]^T$	463176
	$[2,-2,\ldots,2,-2,2]^T$	$[1,\ldots,1]^T$	576770
	$[3,-3,\ldots,3,-3,3]^T$	$[1,\ldots,1]^T$	794156
	$[5,-5,\ldots,5,-5,5]^T$	$[1,\ldots,1]^T$	581242
DFMA	$[0,\ldots,0]^T$	$[1,\ldots,1]^T$	893881
	$[3,\ldots,3]^T$	$[1,\ldots,1]^T$	1317062
	$[-5,\ldots,-5]^T$	$[1,\ldots,1]^T$	899485
	$[2,-2,\ldots,2,-2,2]^T$	$[1,\ldots,1]^T$	896281
	$[3,-3,\ldots,3,-3,3]^T$	$[1,\ldots,1]^T$	1516925
	$[5,-5,\ldots,5,-5,5]^T$	$[1,\ldots,1]^T$	1518820
DFMB	$[0,\ldots,0]^T$	$[1,\ldots,1]^T$	615173
	$[3,\ldots,3]^T$	$[1,\ldots,1]^T$	956979
	$[-5,\ldots,-5]^T$	$[1,\ldots,1]^T$	620725
	$[2,-2,\ldots,2,-2,2]^T$	$[1,\ldots,1]^T$	617573
	$[3,-3,\ldots,3,-3,3]^T$	$[1,\ldots,1]^T$	655048
	$[5,-5,\ldots,5,-5,5]^T$	$[1,\ldots,1]^T$	756943

DFMB. These results demonstrate that the proposed algorithm is effective and more efficient than the compared algorithms.

6. Concluding Remarks

In this paper, we propose an auxiliary function method for finding the global minimizer of integer programming problems. The auxiliary function can help to escape from a local minimizer and to get a better one. The minimizer of the auxiliary function also is a feasible minimizer of the original problem and better than the current best one obtained. The numerical results show that the proposed algorithm is more effective and efficient than the compared ones.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 60873099), the Ph.D. Programs Foundation of Education Ministry of China (no. 20090203110005), National Program on Key Basic Research Project of China (973 Program) Grant no. 2010CB328102, and the Fundamental Research Funds for the Central Universities (no. k50510030014).

References

[1] J. Bang-Jensen, G. Gutin, and A. Yeo, “When the greedy algorithm fails,” *Discrete Optimization*, vol. 1, no. 2, pp. 121–127, 2004.

[2] V. Chvátal, “A greedy heuristic for the set-covering problem,” *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[3] G. Dobson, “Worst-case analysis of greedy heuristics for integer programming with nonnegative data,” *Mathematics of Operations Research*, vol. 7, no. 4, pp. 515–531, 1982.

[4] A. Federgruen and H. Groenevelt, “The greedy procedure for resource allocation problems: necessary and sufficient conditions for optimality,” *Operations Research*, vol. 34, no. 6, p. 909–918 (1987), 1986.

[5] C. Mohan and H. T. Nguyen, “A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems,” *Computational Optimization and Applications*, vol. 14, no. 1, pp. 103–132, 1999.

[6] S. L. Rosen and C. M. Harmonosky, “An improved simulated annealing simulation optimization method for discrete parameter stochastic systems,” *Computers & Operations Research*, vol. 32, no. 2, pp. 343–358, 2005.

[7] J. Cai and G. Thierauf, “Evolution strategies for solving discrete optimization problems,” *Advances in Engineering Software*, vol. 25, no. 2-3, pp. 177–183, 1996.

[8] N. Turkkan, “Discrete optimization of structures using a floating-point genetic algorithm,” in *Proceedings of the Annual Conference of the Canadian Society for Civil Engineering*, Moncton, Canada, 2003.

[9] S. J. Wu and P. T. Chow, “Steady-state genetic algorithms for discrete optimization of trusses,” *Computers and Structures*, vol. 56, no. 6, pp. 979–991, 1995.

[10] F. Glover, “Tabu search and adaptive memory programming C advances applications, and challenges,” in *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies*, R. S. Barr, R. V. Helgason, and J. L. Kennington, Eds., pp. 1–75, Kluwer Academic Publishers, 1996.

[11] L. Michel and P. Van Hentenryck, “A simple tabu search for warehouse location,” *European Journal of Operational Research*, vol. 157, no. 3, pp. 576–591, 2004.

[12] S. F. Woon and V. Rehbock, “A critical review of discrete filled function methods in solving nonlinear discrete optimization problems,” *Applied Mathematics and Computation*, vol. 217, no. 1, pp. 25–41, 2010.

[13] Y. Shang and L. Zhang, “A filled function method for finding a global minimizer on global integer optimization,” *Journal of Computational and Applied Mathematics*, vol. 181, no. 1, pp. 200–210, 2005.

[14] C. K. Ng, D. Li, and L. S. Zhang, “Discrete global descent method for discrete global optimization and nonlinear integer programming,” *Journal of Global Optimization*, vol. 37, no. 3, pp. 357–379, 2007.

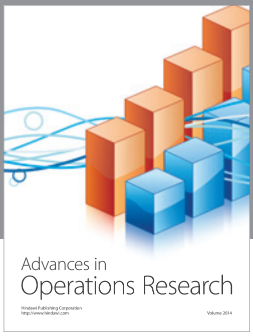
[15] Y. l. Shang and L. Zhang, “Finding discrete global minima with a filled function for integer programming,” *European Journal of Operational Research*, vol. 189, no. 1, pp. 31–40, 2008.

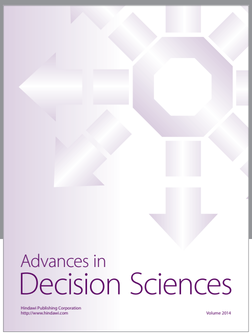
[16] Y. Yongjian and L. Yumei, “A new discrete filled function algorithm for discrete global optimization,” *Journal of Computational and Applied Mathematics*, vol. 202, no. 2, pp. 280–291, 2007.

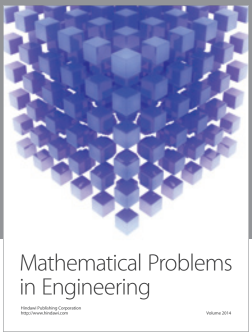
[17] Y. Yang, Z. Wu, and F. Bai, “A filled function method for constrained nonlinear integer programming,” *Journal of Industrial and Management Optimization*, vol. 4, no. 2, pp. 353–362, 2008.

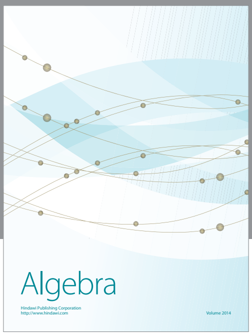
[18] Y. H. Gu and Z. Y. Wu, “A new filled function method for nonlinear integer programming problem,” *Applied Mathematics and Computation*, vol. 173, no. 2, pp. 938–950, 2006.

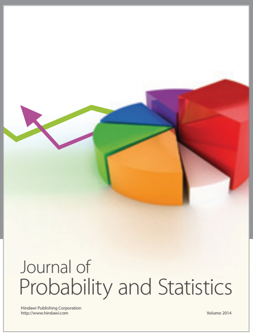
[19] C. K. Ng, L. S. Zhang, D. Li, and W. W. Tian, “Discrete filled function method for discrete global optimization,” *Computational Optimization and Applications*, vol. 31, no. 1, pp. 87–115, 2005.


Advances in
Operations Research

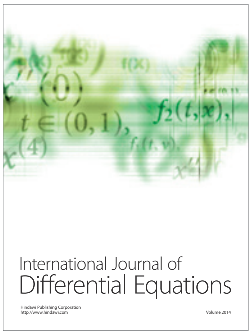

Advances in
Decision Sciences

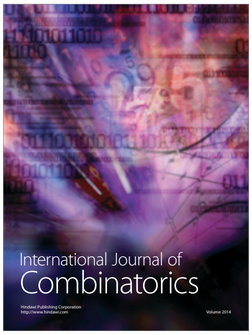

Mathematical Problems
in Engineering

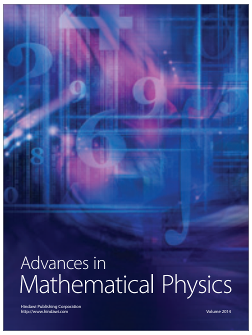

Algebra

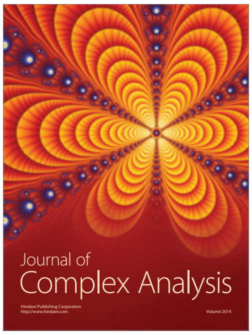

Journal of
Probability and Statistics

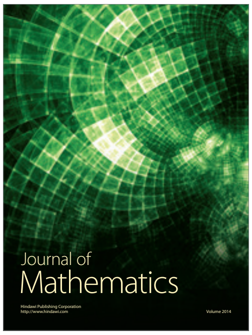
The Scientific
World Journal

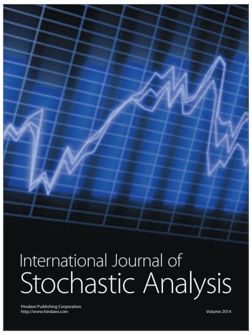

International Journal of
Differential Equations

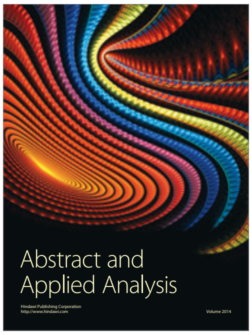

International Journal of
Combinatorics

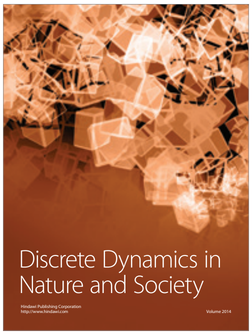

Advances in
Mathematical Physics

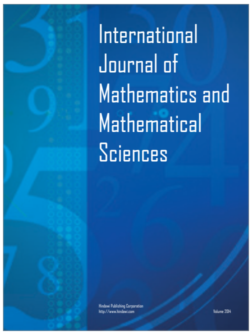

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

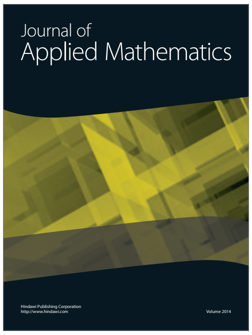

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization