

Article

A New Nonparametric Filled Function Method for Integer Programming Problems with Constraints

Suxia Ma ^{1,2}, Yuelin Gao ^{2,3,*} , Bo Zhang ⁴ and Wenlu Zuo ¹

¹ School of Mathematics and Information Sciences, North Minzu University, Yinchuan 750021, China; suxiama0220@163.com (S.M.); zuowenlu@163.com (W.Z.)

² Ningxia Province Cooperative Innovation Center of Scientific Computing and Intelligent Information Processing, North Minzu University, Yinchuan 750021, China

³ Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Yinchuan 750021, China

⁴ School of Mathematics and Statistics, Ningxia University, Yinchuan 750021, China; zbsd121@163.com

* Correspondence: gaoyuelin@263.net; Tel.: +86-139-9510-0900

Abstract: In this paper, we investigate and develop a new filled function method for solving integer programming problems with constraints. By adopting the appropriate equivalent transformation method, these problems are transformed into a class of box-constrained integer programming problems. Then, an effective nonparametric filled function is constructed, and a new global optimization algorithm is designed using the discrete steepest descent method. Numerical experiments illustrate that this algorithm has effectiveness, feasibility, and better global optimization ability.

Keywords: discrete global optimization; constrained integer program; nonparametric filled function method



Citation: Ma, S.; Gao, Y.; Zhang, B.; Zuo, W. A New Nonparametric Filled Function Method for Integer Programming Problems with Constraints. *Mathematics* **2022**, *10*, 734. <https://doi.org/10.3390/math10050734>

Academic Editors: Mihai Postolache, Jen-Chih Yao and Yonghong Yao

Received: 26 January 2022

Accepted: 23 February 2022

Published: 25 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Consider the following integer programming problem with constraints:

$$(P) \begin{cases} \min f(x) \\ s.t. \ g_i(x) \leq 0, \quad i = 1, 2, \dots, m, \\ x \in \mathbb{Z}^n, \end{cases}$$

where \mathbb{Z}^n denotes the set of integer points in \mathbb{R}^n , and $S = \{x \in \mathbb{Z}^n \mid g_i(x) \leq 0, i = 1, 2, \dots, m\}$ is bounded.

Some problems in economy, finance, engineering, and other fields can often be quantified as a global optimization problem (P). Most of these problems, however, are computationally and theoretically difficult to solve in polynomial time, so they are often referred to as NP-hard problems. So far, the existing global optimization algorithms for solving problem P can be classified into deterministic algorithms [1–5] and stochastic algorithms [6–9]. Among them, the filled function method, which was first proposed by Ge [1], is favored by many scholars as a deterministic method of global optimization. The main idea of this method is to continuously construct the filled function (FF) at the local minimizer (x_k^*) in the current iteration and then take the point near x_k^* as the initial point to minimize FF, so as to get out of the basin where x_k^* is located and find the local minimizer x_{k+1}^* better than x_k^* , which achieves the purpose of global optimization. Therefore, the filled function method avoids the “precocity” defect of general optimization algorithms. As defined in [1], if $\Psi(x, x_k^*)$ is called a FF of the objective function $f(x)$ at a local minimizer x_k^* , it needs to satisfy:

- (i) x_k^* is a strictly local maximizer of $\Psi(x, x_k^*)$, and the basin B_k^* of $f(x)$ at x_k^* becomes a part of a hill of $\Psi(x, x_k^*)$ at x_k^* ;

- (ii) in the basin where $f(x)$ is higher than B_k^* , $\Psi(x, x_k^*)$ has no local minimizer;
- (iii) if $f(x)$ has a basin \bar{B} lower than B_k^* , then minimizing $\Psi(x, x_k^*)$ must be able to find its local minimizer x_{k+1}^* in the line between x and x_k^* in \bar{B} .

In the late 1990s, Zhu [10] first proposed the discrete equivalent method of continuous FF, and the FF formula is as follows:

$$\Psi_{\gamma, \rho, x^*}(x) = \frac{1}{\gamma + f(x)} \exp\left(-\frac{\|x - x^*\|^2}{\rho^2}\right), \tag{1}$$

where γ and ρ denote parameters to ensure that the algorithm can be executed and the global minimizer can be obtained. For the exponential term $\exp\left(-\frac{\|x - x_k^*\|^2}{\rho^2}\right)$ with large $\|x - x_k^*\|^2$ or small ρ , this Formula (1) produces an almost flat image, which makes it difficult for the computer to distinguish between changes in function values. In addition, Zhu’s approach has been discussed in depth in [11–13]. In 2006, Wu et al. [14] proposed a FF (2) for solving integer programming problems with box constraints.

$$\Psi(x, x^*, r, q) = \frac{1}{1 + \|x - x^*\|^2} g_r(f(x) - \frac{f(x^*)}{2}) + q f_r(f(x) - \frac{f(x^*)}{2}), \tag{2}$$

where,

$$g_r(t) = \begin{cases} 0, & t \leq -r, \\ -\frac{2}{r^3}t^3 - \frac{3}{r^2}t^2 + 1, & -r < t \leq 0, \\ 1, & t \geq 0, \end{cases} \quad f_r(t) = \begin{cases} t + r, & t \leq -r, \\ \frac{r-2}{r^3}t^3 + \frac{r-3}{r^2}t^2 + 1, & -r < t \leq 0, \\ 1, & t > 0. \end{cases}$$

To address the nonlinear integer programming problem with inequality constraints, Yang et al. [15] proposed a FF in the following form in 2008:

$$\Psi(r, x, x^*) = \left(\frac{1}{1 + \|x - x^*\|^2} + 1\right) h(h_r(f(x) - f(x^*))) + \sum_{i=1}^m h_r(g_i(x) - r), \tag{3}$$

where

$$h_r(t) = \begin{cases} 0, & t \leq -r, \\ \frac{r-2}{r^3}t^3 + \frac{2r-3}{r^2}t^2 + t + 1, & -r < t \leq 0, \\ t + 1, & t > 0, \end{cases}$$

$$h(t) = \begin{cases} 0, & t \leq \frac{1}{2}, \\ -16t^3 + 36t^2 - 24t + 5, & \frac{1}{2} < t \leq 1, \\ 1, & t > 1. \end{cases}$$

To overcome the defect of FF in solving constrained global optimization problems, the Formula (3) combines the FF in unconstrained optimization with the penalty function in constrained optimization. For solving the nonlinear integer programming problem with box constraints, Lin et al. [16] also proposed a FF as follows.

$$\Psi(x, x^*, q) = \frac{1}{q + \|x - x^*\|} \phi_q\left(f(x) - \frac{f(x^*)}{4}\right), \tag{4}$$

where

$$\phi_q(t) \begin{cases} \arctan\left(-\frac{q^2}{t^2}\right) + \frac{\pi}{2}, & t \neq 0, \\ 0, & t = 0. \end{cases}$$

The above four FFs contain either parameters or exponential factors, which will reduce the computational efficiency of the FF algorithm. This is because such FFs are highly dependent on parameters, but the parameter selection is usually particularly difficult, and the exponential factors will lead to the overflow of numerical results. In addition to the above FFs, many scholars have also made innovations, such as [12,17–19]. More importantly, in [19], these two authors reviewed and summarized nine existing filled functions, and they believed that these filled functions had a broad development prospect. However, most of the FFs mentioned above are aimed at box constrained or unconstrained optimization problems, and some researches on constrained integer programming problems are still relatively few, so it is necessary to further propose a new filled function, which can solve constrained integer programming problems. At present, the research methods of constrained integer programming problems are mainly divided into two categories. One is to directly construct FF satisfying constraints; the other is to use penalty function method [20] to convert constrained optimization problems into unconstrained optimization problems or box-constrained optimization problems. It is satisfying that Lin [18] et al. cleverly used the idea of equivalent transformation for problem P, that is, the second type method; on this basis, it is worth us to further improve its transformation process more accurately.

In this paper, a new equivalent transformation method is used to reconstruct the problem P into a box-constrained optimization problem (OP). Based on the problem OP, a nonparametric FF with the same local minimizer as its objective function is constructed, and a new global optimization algorithm is designed. In the iterative process of the algorithm, only FF needs to be continuously minimized. A series of numerical experiments are carried out on a large number of test problems, and the numerical results show that the algorithm is effective.

The rest of this paper is organized as follows. Section 2 is a review of some assumptions and definitions. Section 3 is the detailed description of the equivalent transformation to the problem P. The nonparametric FF constructed and its properties are analyzed and proved in Section 4. In Section 5, a new global optimization algorithm is designed combined with the proposed new FF. Numerical experiments and conclusions are given in Sections 6 and 7, respectively.

2. Assumptions and Definitions

To make the problem P solvable, it is necessary to give some assumptions and definitions. In [21], the content of the improved definition of FF is extended as follows:

Definition 1. Suppose x_k^* is a discrete local minimizer of the objective function $f(x)$. If $\Psi(x, x_k^*)$ is called a FF of $f(x)$ at x_k^* , then it needs to satisfy:

- (i) x_k^* is a strictly discrete local maximizer of $\Psi(x, x_k^*)$; and the basin B_k^* of $f(x)$ at x_k^* becomes a part of a hill of $\Psi(x, x_k^*)$ at x_k^* ;
- (ii) $\Psi(x, x_k^*)$ has no discrete local minimizer in set $S_k^1 = \{x | f(x) \geq f(x_k^*), x \in S \setminus \{x_k^*\}\}$;
- (iii) If x_k^* is not a discrete global minimizer of $f(x)$, then $S_k^2 \neq \emptyset$, $\Psi(x, x_k^*)$, and $f(x)$ have the same discrete local minimizer in $S_k^2 = \{x | f(x) < f(x_k^*), x \in S\}$.

Assumption 1. Let τ be the set of all the local minimizers of problem P, and $L^* = \{f(x) | x \in \tau\}$.

Definition 2. For any integer point $x \in \mathbb{Z}^n$, the set $N(x) \subseteq \mathbb{Z}^n$ can be called a discrete neighborhood for x , where $N(x) = \{x, x \pm e_i : i = 1, 2, \dots, n\}$. The set of all axial directions $d \in N(x)$ can be defined as $D = \{d \in \mathbb{R}^n : d = \pm e_i, i = 1, 2, \dots, n\}$, where n denotes the number of variables of the $f(x)$, and e_i is a unit vector with the i th component equal to 1 and the remaining components equal to 0.

Definition 3. A vector $d \in D$ is called a discrete descent direction of $f(x)$ at $x \in S$ if d is such that $x + d \in S$ and $f(x + d) < f(x)$.

Definition 4. A vector $d^* \in D$ is called a discrete steepest descent direction of $f(x)$ at $x \in S$ if $f(x + d^*) \leq f(x + d)$ for any $d \in D^*$, where D^* denotes the set of all discrete descent directions of $f(x)$ at $x \in S$.

Definition 5. A point x_0 is called a (strictly) discrete local minimizer of problem P if an integer point $x_0 \in S$ is such that $f(x)(>) \geq f(x_0)$ for any $x \in N(x_0) \cap S$.

Definition 6. A point x_0 is called a (strictly) discrete global minimizer of problem P if an integer point $x_0 \in S$ is such that $f(x)(>) \geq f(x_0)$ for any $x \in S$.

At present, there are many local optimization algorithms to solve continuous optimization problems. However, there are few local search methods to solve discrete optimization problems. Different from continuous optimization problems, the discreteness of discrete optimization problems makes the gradient of function unusable, which leads to the birth of discrete descent algorithm. After that, many scholars also improved the algorithm for different problems, and proposed various discrete local minimization methods, such as variable neighborhood search algorithm [22], neighborhood search algorithm [22], and discrete steepest descent algorithms [18,23,24]. In this paper, we adopt the following discrete steepest descent algorithm in [24]:

3. Transformation of the Original Problem

To facilitate the construction of the FF, we first make an equivalent transformation to the problem P, so that the problem can be easily handled. The specific contents are as follows:

Theorem 1. For any $\bar{x} \in S$, let $\bar{u} = f(\bar{x})$, then problem

$$(\bar{P}) \begin{cases} \min f(x) \\ \text{s.t. } g_i(x) \leq 0, i = 1, 2, \dots, m, \\ x \in \mathbb{Z}^n, \\ f(x) \leq \bar{u}, \end{cases}$$

and P have the same global optimal solution.

Proof. Let x^* be a global optimal solution to \bar{P} . Suppose x^* is not global optimal for P, there must exist a $\hat{x} \in S = \{x \in \mathbb{R}^n | x \in \mathbb{Z}^n, g_i(x) \leq 0, i = 1, 2, \dots, m\}$ such that

$$f(\hat{x}) < f(x^*), \tag{5}$$

and

$$f(x^*) < \bar{u}, \tag{6}$$

then, it follows from Formulas (5) and (6) that $f(\hat{x}) < \bar{u}$, which indicates that \hat{x} is feasible for \bar{P} , so Formula (5) shows that \hat{x} contradicts the global optimal solution x^* of \bar{P} . \square

From the boundness of set S, it follows that there must be a feasible region of the problem \bar{P} contained in box Ω .

For convenience, let us define the following two univariate functions:

$$g(t) = \begin{cases} 0, & t > 0, \\ 1, & t \leq 0, \end{cases} \quad s(t) = \begin{cases} 0, & t = m + 1, \\ 1, & t < m + 1, \end{cases}$$

where m denotes the number of constraints of problem P.

Next, consider the following problem:

$$(OP) \begin{cases} \min F(x), \\ s.t. x \in \Omega \cap \mathbb{Z}^n, \end{cases}$$

where

$$F(x) = f(x) + (\bar{u} - f(x)) \times s \left(\sum_{i=1}^m g_i(g_i(x)) + g(f(x) - \bar{u}) \right).$$

Theorem 2. *If \bar{x} is a local minimizer of \bar{P} , then \bar{x} is also a local minimizer of OP; conversely, if \bar{x} is a local minimizer of OP, then \bar{x} is also a local minimizer of \bar{P} , or $F(\bar{x}) = \bar{u} + 1$.*

Proof. In Theorem 3.1 of [18], let $M = \bar{u} + 1$, then the proof of this theorem can be obtained, so it will not be repeated here. \square

Remark 1. *According to Theorem 2, we can know that problems \bar{P} and OP have the same minimizer except infeasible point. If the global minimizer of problem OP can be found, then the global minimizer of problem \bar{P} can also be found.*

4. Nonparametric Filled Function and Its Analytical Properties

Based on the previous three sections, we propose a new nonparametric FF:

$$\Psi(x, x_k^*) = \left(\pi - \arctan\left(\|x - x_k^*\|^2\right) \right) \times \phi(F(x) - F(x_k^*)) + \min\{0, (F(x) - F(x_k^*))^3\}, \quad (7)$$

where

$$\phi(r) = \begin{cases} 1, & r \geq 0, \\ 0, & r < 0, \end{cases}$$

$\|\cdot\|$ denotes Euclidean norm and x_k^* represents a discrete local minimizer. This FF avoids some of the disadvantages discussed in Section 1 and has the nice property of having the same local minimizer as the objective function.

Now, we define two sets:

$$S_k^1 = \{x \in \Omega \cap \mathbb{Z}^n \mid F(x) \geq F(x_k^*)\},$$

$$S_k^2 = \{x \in \Omega \cap \mathbb{Z}^n \mid F(x) < F(x_k^*)\},$$

the following is the verification that $\Psi(x, x_k^*)$ satisfies the FF defined in Section 2.

Theorem 3. *If $x_k^* \in \tau$, $x_1 \in S_k^1$, then x_k^* is a strictly discrete local maximizer of $\Psi(x, x_k^*)$.*

Proof. It follows from $x_1 \in S_k^1$ that $F(x_1) \geq F(x_k^*)$. From the definition of ϕ , there is

$$\Psi(x_1, x_k^*) = \pi - \arctan\left(\|x_1 - x_k^*\|^2\right) < \pi = \Psi(x_k^*, x_k^*).$$

Obviously, x_k^* is a strictly discrete local maximizer of $\Psi(x, x_k^*)$. \square

Theorem 4. *If $x_k^* \in \tau$ and $x_a, x_b \in S_k^1$ satisfy $\|x_a - x_k^*\| < \|x_b - x_k^*\|$, then*

$$\Psi(x_b, x_k^*) < \Psi(x_a, x_k^*) < \pi = \Psi(x_k^*, x_k^*).$$

Proof. Since $x_a, x_b \in S_k^1$, then $F(x_a) \geq F(x_k^*), F(x_b) \geq F(x_k^*)$. From the definition of function ϕ , it follows that

$$\Psi(x_a, x_k^*) = \pi - \arctan\left(\|x_a - x_k^*\|^2\right),$$

$$\Psi(x_b, x_k^*) = \pi - \arctan(\|x_b - x_k^*\|^2),$$

Upon the condition $\|x_a - x_k^*\| < \|x_b - x_k^*\|$, we have

$$\Psi(x_b, x_k^*) - \Psi(x_a, x_k^*) = -\arctan(\|x_b - x_k^*\|^2) - \left(-\arctan(\|x_a - x_k^*\|^2)\right) < 0,$$

thus $\Psi(x_b, x_k^*) < \Psi(x_a, x_k^*) < \pi = \Psi(x_k^*, x_k^*)$. \square

Theorem 4 shows that, on the high level set S_k^1 , the farther away from the current local minimizer x_k^* , the faster the FF value decreases. Thus, it ensures that the minimization process of $\Psi(x, x_k^*)$ can always be realized and a better point can be found faster.

Lemma 1. For any $x, x^* \in S$, if there is an $i \in \{1, 2, \dots, n\}$ such that $x + e_i \in S$, there must be a $d \in D$ that satisfies $\|x + d - x_k^*\| > \|x - x_k^*\|$.

Proof. If $x = x_k^*$, the conclusion is obvious. If $x \neq x_k^*$, there must be an $i \in \{1, 2, \dots, n\}$ such that $x_i \neq x_{ki}^*$. If $x_i > x_{ki}^*$, let $d = e_i$; otherwise, let $d = -e_i$. \square

Theorem 5. If x_k^* is a discrete local minimizer of $F(x)$, then $\Psi(x, x_k^*)$ has no discrete local minimizer over the set S_k^1 .

Proof. For any $x \in S_k^1$, we know $\Psi(x, x_k^*) = \pi - \arctan(\|x - x_k^*\|^2)$. The existence of $d \in D$ is such that $\|x + d - x_k^*\| > \|x - x_k^*\|$. Hence, we discuss the conclusion in the following two cases:

(1) If $F(x + d) \geq F(x_k^*)$, then

$$\Psi(x + d, x_k^*) - \Psi(x, x_k^*) = \arctan(\|x - x_k^*\|^2) - \arctan(\|x + d - x_k^*\|^2) < 0;$$

(2) If $F(x + d) < F(x_k^*) < F(x)$, then

$$\Psi(x + d, x_k^*) - \Psi(x, x_k^*) = (F(x + d) - F(x_k^*))^3 - \left(\pi - \arctan(\|x - x_k^*\|^2)\right) < 0.$$

Therefore, for all $x \in S_k^1$, there is always a $d \in D^*$ such that $\Psi(x + d, x_k^*) < \Psi(x, x_k^*)$, i.e., $\Psi(x, x_k^*)$ has no discrete local minimizers in the set S_k^1 . \square

Theorems 6 and 7 state that $F(x)$ and $\Psi(x, x_k^*)$ have the same local minimizer over S_k^2 .

Theorem 6. If x_1^* is a discrete local minimizer of $F(x)$ and satisfies $F(x_1^*) < F(x_k^*)$, then x_1^* is also a discrete local minimizer of $\Psi(x, x_k^*)$.

Proof. If x_1^* is a discrete local minimizer of $F(x)$ and satisfies $F(x_1^*) < F(x_k^*)$, then $x_1^* \in S_k^2$ and $\Psi(x_1, x_k^*) = (F(x_1) - F(x_k^*))^3 < 0$. For any $x \in N(x_1^*)$, there is $F(x) \geq F(x_1^*)$; let us consider the following two cases:

(1). If $x \in S_k^1 \cap N(x_1^*)$, then $\Psi(x, x_k^*) = \pi - \arctan(\|x - x_k^*\|^2) > 0 > \Psi(x_1, x_k^*)$.

(2). If $x \in S_k^2 \cap N(x_1^*)$, then $\Psi(x, x_k^*) = (F(x) - F(x_k^*))^3 \geq (F(x_1^*) - F(x_k^*))^3 = \Psi(x_1, x_k^*)$.

\square

Theorem 7. If x_2^* is a discrete local minimizer of $\Psi(x, x_k^*)$ and satisfies $\Psi(x_2^*) < 0$, then x_2^* is also a discrete local minimizer of $F(x)$, and $F(x_2^*) < F(x_k^*)$.

Proof. Let x_2^* be a discrete local minimizer of $\Psi(x, x_k^*)$ and satisfies $\Psi(x_2^*) < 0$, then $\Psi(x_2^*) < \Psi(x_k^*)$ and $x_2^* \in S_k^2$.

Now, let us prove that x_2^* is also a discrete local minimizer of $F(x)$. Suppose x_2^* is not a local minimizer of $F(x)$, there must be a $x_3^* \in N(x_2^*)$ such that $F(x_3^*) < F(x_2^*)$. Since $F(x_2^*) < F(x_k^*)$, then $F(x_3^*) < F(x_k^*)$. Thus,

$$\Psi(x_3^*, x_k^*) = (F(x_3^*) - F(x_k^*))^3 < (F(x_2^*) - F(x_k^*))^3 = \Psi(x_2^*, x_k^*),$$

which contradicts the fact that x_2^* is a discrete local minimizer of $\Psi(x, x_k^*)$. \square

In summary, it is proved by Theorems 3–7 that $\Psi(x, x_k^*)$ is a nonparametric FF.

5. Filled Function Algorithm

In this section, we develop a new nonparametric FF algorithm NPFPA based on the FF constructed in the previous section. The local descent method adopts the Algorithm 1. In this algorithm, the problem of boundary overflow is improved and some unnecessary computation is avoided. Moreover, the proposed FF has the same local minimizer as the objective function, so the constructed algorithm only needs to minimize the FF in the iterative process, which greatly reduces the amount of calculation.

Algorithm 1: (DSDA)

- **Step 1.** Select a starting point $x_0 \in S$.
- **Step 2.** If x_0 is a local minimizer for $f(x)$, then stop. Otherwise, let

$$d^* = \arg \min \{f(x_0 + d_i) : d_i \in D_{x_0}, f(x_0 + d_i) < f(x_0)\},$$

where $D_{x_0} = \{d \in D : x_0 + d \in S\}$ is a set of feasible directions at x_0 ;

- **Step 3.** Let $x_0 = x_0 + d^*$, and return to step 2.
-

6. Numerical Experiment

This section uses MATLAB (2016a) to encode and execute the Algorithm 2 and the algorithm in [18]. All calculations are performed on a desktop computer with Intel(R) Core(TM)i5-8500 3.00 GHz power processor 8.00 GB memory and Win10 operating system. The reliability and effectiveness of Algorithm 2 are verified using 13 test problems (a total of 84 test instances) to test and compare these two algorithms.

Algorithm 2: (NPFPA)

- **Step 0.** Give a starting point $x_0 \in S$; Let $\bar{u} = f(x_0), D = \{d_1, d_2, \dots, d_{2n}\}$, where $d_i = e_i, d_{n+i} = -e_i, i = 1, 2, \dots, n$.
- **Step 1.** Use Algorithm 1 to minimize $F(x)$ with x_0 and obtain a local minimizer x_1^* . Set $k = 1$.
- **Step 2.** Construct the following FF

$$\Psi(x, x_k^*) = (\pi - \arctan(\|x - x_k^*\|^2)) \times \phi(F(x) - F(x_k^*)) + \min\{0, (F(x) - F(x_k^*))^3\}.$$

- **Step 3.** Take $x_i = x_k^* + d_i$ as the initial point to minimize $\phi(x, x_k^*)$ and obtain the local minimizer x_k^f .
 - **Step 4.** If $F(x_k^f) < F(x_k^*)$, let $x_k^* = x_k^f, F(x_k^*) = F(x_k^f), k = k + 1$, and return to step 2.
 - **Step 5.** If $i \geq 2n$, go to step 6; otherwise, let $i = i + 1$ and return to step 3.
 - **Step 6.** Output x_k^* .
-

6.1. Test Function

This section mainly gives 13 test functions, where $f(x)$ denotes the objective function, x_0 is the initial point, and the global minimizer and the global minimum value are represented by x^* and $f(x^*)$, respectively.

Problem 1 ([25]).

$$\begin{aligned} \min \quad & f(x) = x_1^2 + x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 - 8x_1 - 2x_2 - 3x_3 - x_4 - 2x_5, \\ \text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 + x_5 \leq 400, \quad x_1 + 2x_2 + 2x_3 + x_4 + 6x_5 \leq 800, \\ & 2x_1 + x_2 + 6x_3 \leq 200, \quad x_3 + x_4 + 5x_5 \leq 200, \\ & x_1 + x_2 + x_3 + x_4 + x_5 \geq 55, \quad x_1 + x_2 + x_3 + x_4 \geq 48, \\ & x_2 + x_4 + x_5 \geq 34, \quad 6x_1 + 7x_5 \geq 104, \\ & 0 \leq x_i \leq 99, \quad x_i \in \mathbb{Z}^n, \quad i = 1, 2, 3, 4, 5. \end{aligned}$$

For Problem 1, the discrete global optimal solution is $x^* = (16, 22, 5, 5, 7)^T$, $f(x^*) = 807$. This problem was tested with five different initial points, and the numerical results are shown in Table 1.

Problem 2 ([26]).

$$\begin{aligned} \min \quad & f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \\ \text{s.t.} \quad & x_1^2 + x_2^2 \geq 0.25, \quad -\frac{1}{3}x_1 + x_2 \geq -0.1, \\ & x_i = y_i/10,000, \quad 0 \leq y_i \leq 10^5, \\ & y_i \in \mathbb{Z}^n, \quad i = 1, 2. \end{aligned}$$

Problem 2 has about 8.413×10^9 points in the feasible region. The discrete global optimal solution is $x^* = (1, 1)^T$, $f(x^*) = 0$. This problem was tested with nine different initial points, and the numerical results are shown in Table 1.

Problem 3 ([27]).

$$\begin{aligned} \min \quad & f(x) = 2x_1x_2^2 - 4x_1x_2 + x_1^2 + x_2^2 - 27x_1^2x_2^2, \\ \text{s.t.} \quad & x_1^2 + 2x_1x_2 + x_2^2 \leq 500, \quad x_1 + 2x_2^2 \leq 400, \\ & 0 \leq x_1 \leq 500, \quad 0 \leq x_2 \leq 100, \\ & x_i \in \mathbb{Z}^n, \quad i = 1, 2. \end{aligned}$$

Problem 3 has a lot of feasible points. The discrete global optimal solution is $x^* = (11, 11)^T$, $f(x^*) = -392,887$. This problem was tested with two different initial points, and the numerical results are shown in Table 1.

Problem 4 (Powell’s singular function [19,21]).

$$\begin{aligned} \min \quad & F(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, \\ \text{s.t.} \quad & x_i = \frac{y_i}{1000}, \quad -10,000 \leq y_i \leq 10,000, \quad i = 1, 2, 3, 4. \quad y_i \in \mathbb{Z}^n \end{aligned}$$

Problem 4 has 1.60032×10^{17} feasible points and multiple local minimizers. The discrete global minimizer is $x^* = (0, 0, 0, 0)^T$, $F(x^*) = 0$. This problem was tested with four different initial points, and the numerical results are shown in Table 1.

Problem 5 (Goldstein and Price’s function [19,21]).

$$\begin{aligned} \min \quad & F(x) = g(x)h(x), \\ \text{s.t.} \quad & x_i = \frac{y_i}{1000}, \quad -2000 \leq y_i \leq 2000, \quad i = 1, 2. \quad y_i \in \mathbb{Z}^n. \end{aligned}$$

$g(x) = 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2), h(x) = 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$ There are 1.60018001×10^7 feasible points and many discrete local minimizers in the feasible region of Problem 5. The discrete global optimal solution is $x^* = (0, -1)^T, F(x^*) = 3$. This problem was tested with six different initial points, and the numerical results are presented in Table 1.

Problem 6 ([27]).

$$\begin{aligned} \min \quad & f(x) = 6x_1^2 + 18x_2^2 + 7x_3^2 - 2x_1 - 16x_2 - 31x_3 - 12x_1x_2x_3, \\ \text{s.t.} \quad & x_1 + x_2 + 2x_3 \leq 2000, \quad x_1 + 17x_2 \leq 8000, \\ & x_2 + 5x_3 \leq 4000, \quad x_1 + 7x_2 + x_3 \geq 200, \\ & x_1 + x_2 + x_3 \geq 200, \quad x_1^2 + x_2x_3 \geq 900, \\ & 0 \leq x_i \leq 999, \quad x_i \in \mathbb{Z}^n, \quad i = 1, 2, 3. \end{aligned}$$

This problem has a great number of feasible points. The discrete global optimal solution is $(720, 424, 428)^T, f(x^*) = -1,560,310,784$. This problem was tested with one initial point, and the numerical results are presented in Table 2.

Problem 7 ([18]).

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

The global optimal solution of Problem 5 is $x^* = (50, 50, 50, 50)^T, f^* = -1$. This problem was tested with three initial points, and the results are listed in Table 2.

Problem 8 ([9,18]).

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

The discrete global optimal solution of Problem 6 is $x^* = (15, 5)^T, f^* = -3250$. This problem was tested with three initial points, and the results are listed in Table 2.

Problem 9 ([9,18]).

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

The global optimal solution of Problem 7 is $x^* = (5, 1, 5, 0, 5, 10)^T$, $f^* = -310$. The problem was tested with three initial points and the results are listed in Table 2.

Problem 10 (Colville’s function [1,2,19]).

$$\begin{aligned} \min \quad & F(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ & + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1), \\ \text{s.t.} \quad & -10 \leq x_i \leq 10, \quad i = 1, 2, 3, 4. \quad x \in \mathbb{Z}^n. \end{aligned}$$

Problem 10 has 1.94481×10^5 feasible points and 41 discrete local minimizers. The discrete global minimizer is $x^* = (1, 1, 1, 1)^T$, $F(x^*) = 0$. This problem was tested with four different initial points, and the numerical results are shown in Table 2.

Problem 11 (Beale’s function [5]).

$$\begin{aligned} \min \quad & F(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2, \\ \text{s.t.} \quad & x_i = \frac{y_i}{1000}, \quad -10,000 \leq y_i \leq 10,000, \quad i = 1, 2. \quad x \in \mathbb{Z}^n. \end{aligned}$$

This problem has 4.00040001×10^8 feasible points in the feasible region. The discrete global optimal solution is $x^* = (3, 0.5)^T$, $F(x^*) = 0$. This problem was tested with six different initial points, and the numerical results are presented in Table 2.

Problem 12 (Rosenbrock’s function [1]).

$$\begin{aligned} \min \quad & f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \\ \text{s.t.} \quad & -5 \leq x_1 \leq 5, \quad i = 1, 2, \dots, n. \quad x \in \mathbb{Z}^n. \end{aligned}$$

This problem has about 11^n feasible points and many discrete local minimizers in the feasible region (it is known that this problem has 5, 6, 7, 9, and 11 discrete local minimizers when n is set to 2, 3, 4, 5, and 6, respectively, but for any n , it has only one discrete global minimizer). Besides, its discrete global optimal solution is $x^* = (1, 1, \dots, 1)^T$, $f(x^*) = 0$. The test results are presented in Table 3.

Problem 13 (Consider the following n -dimensional function [5]).

$$\begin{aligned} \min \quad & F(x) = (x_1 - 1)^2 + (x_n - 1)^2 + n \sum_{i=1}^{n-1} [(n - i)(x_i^2 - x_{i+1})^2], \\ \text{s.t.} \quad & -5 \leq x_1 \leq 5, \quad i = 1, 2, \dots, n. \quad x \in \mathbb{Z}^n. \end{aligned}$$

There are about 11^n feasible points and many discrete local minimizers in the feasible region of Problem 11. When $n = 2, 3, 4, 5, 6$, the problem has 4, 6, 7, 10, 12 local minimizers, respectively,

but it has only one discrete global minimizer $x^* = (1, 1, \dots, 1)^T$ for any n . The global minimum is $f(x^*) = 0$. The test results are presented in Table 3.

6.2. Numerical Results

This section is a numerical test of numerical examples, the results of which are given in Tables 1–4.

Table 1. Numerical comparison results of Problems 1–5.

PN	Alg	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
1	ours	5	(17, 18, 7, 7, 9)	(16, 22, 5, 5, 7)	807	0.9520	4	48,861
		5	(21, 34, 0, 0, 0)	(16, 22, 5, 5, 7)	807	2.9375	11	150,531
		5	(0, 0, 0, 48, 15)	(16, 22, 5, 5, 7)	807	3.4343	42	177,769
		5	(100, 0, 0, 0, 40)	(16, 22, 5, 5, 7)	807	4.9339	28	252,769
		5	(0, 8, 32, 8, 32)	(16, 22, 5, 5, 7)	807	1.7434	34	88,589
	[18]	5	(17, 18, 7, 7, 9)	(16, 22, 5, 5, 7)	807	2.5070	4	116,365
		5	(21, 34, 0, 0, 0)	(16, 22, 5, 5, 7)	807	5.0296	11	231,931
		5	(0, 0, 0, 48, 15)	(16, 22, 5, 5, 7)	807	14.1812	44	636,267
		5	(100, 0, 0, 0, 40)	(16, 22, 5, 5, 7)	807	10.8155	33	486,107
		5	(0, 8, 32, 8, 32)	(16, 22, 5, 5, 7)	807	11.1178	34	496,607
2	ours	2	(2, 2)	(1, 1)	0	0.0095	1	245
		2	(4, 4)	(1, 1)	0	0.0114	2	413
		2	(6, 6)	(1, 1)	0	0.0124	2	429
		2	(8, 8)	(1, 1)	0	0.0141	3	541
		2	(10, 10)	(1, 1)	0	0.0148	3	549
		2	(0, 0.5)	(1, 1)	0	0.0161	4	653
		2	(0, 10)	(1, 1)	0	0.0141	3	533
		2	(10, 3.2334)	(1, 1)	0	0.0172	4	645
	[18]	2	(0.3536, 0.3536)	(1, 1)	0	0.0185	5	753
		2	(2, 2)	(1, 1)	0	0.0162	2	418
		2	(4, 4)	(1, 1)	0	0.0129	2	433
		2	(6, 6)	(1, 1)	0	0.0149	2	500
		2	(8, 8)	(1, 1)	0	0.0160	3	614
		2	(10, 10)	(1, 1)	0	0.0164	3	608
		2	(0, 0.5)	(1, 1)	0	0.0175	4	710
		2	(0, 10)	(1, 1)	0	0.0154	3	582
		2	(10, 3.2334)	(1, 1)	0	0.0180	4	675
		2	(0.3536, 0.3536)	(1, 1)	0	0.0191	5	777
3	ours	2	(0, 13)	(11, 11)	−392,887	0.1396	3	7093
		2	(0, 0)	(11, 11)	−392,887	0.1378	2	7127
	[18]	2	(0, 13)	(11, 11)	−392,887	0.1808	3	10,205
		2	(0, 0)	(11, 11)	−392,887	0.1444	2	7127
4	ours	4	(10, 10, 10, 10)	(0, 0, 0, 0)	0	0.0725	2	4131
		4	(−10, −10, −10, −10)	(0, 0, 0, 0)	0	0.0538	2	2907
		4	(10, −10, −10, 10)	(0, 0, 0, 0)	0	0.0736	2	4131
		4	(1, −1, −1, 1)	(0, 0, 0, 0)	0	0.0575	2	3843
		4	(−10, 1, 0, 5)	(0, 0, 0, 0)	0	0.0456	2	2731
		4	(0, 0, 0, 0)	(0, 0, 0, 0)	0	0.0392	1	2585

Table 1. Continued.

PN	Alg	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
	[18]	2	(10, 10, 10, 10)	(0, 0, 0, 0)	0	0.0777	2	4439
		4	(−10, −10, −10, −10)	(0, 0, 0, 0)	0	0.0725	2	4599
		4	(10, −10, −10, 10)	(0, 0, 0, 0)	0	0.0715	2	4439
		4	(1, −1, −1, 1)	(0, 0, 0, 0)	0	0.0560	2	4151
		4	(−10, 1, 0, 5)	(0, 0, 0, 0)	0	0.0653	2	4423
		4	(0, 0, 0, 0)	(0, 0, 0, 0)	0	0.0343	1	2585
5	ours	2	(2, −2)	(0, −1)	3	1.3741	2	84,013
		2	(0, −1)	(0, −1)	3	0.8289	2	72,013
		2	(−2, −2)	(0, −1)	3	1.3745	1	84,013
		2	(−0.5, −1)	(0, −1)	3	0.8986	2	74,013
		2	(1, −1.5)	(0, −1)	3	1.0622	1	78,013
		2	(1, −1)	(0, −1)	3	0.9923	1	76,013
	[18]	2	(2, −2)	(0, −1)	3	1.5708	2	84,013
		2	(0, −1)	(0, −1)	3	0.9917	2	71,922
		2	(−2, −2)	(0, −1)	3	1.5720	1	84,077
		2	(−0.5, −1)	(0, −1)	3	1.1065	2	74,103
		2	(1, −1.5)	(0, −1)	3	1.2851	1	78,031
		2	(1, −1)	(0, −1)	3	1.1790	1	76,101

Table 2. Numerical comparison results of Problems 6–11.

PN	Alg	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
6	ours	3	(700, 400, 450)	(720, 424, 428)	−1,560,310,784	0.7027	1	35,971
	[18]	3	(700, 400, 450)	(720, 424, 428)	−1,560,310,784	0.9201	1	46,178
7	ours	4	(25, 25, 25, 25)	(50, 50, 50, 50)	−1	0.3793	1	5869
		4	(50, 50, 50, 50)	(50, 50, 50, 50)	−1	0.1189	1	5509
		4	(75, 75, 75, 75)	(50, 50, 50, 50)	−1	0.1888	2	8949
	[18]	4	(25, 25, 25, 25)	(50, 50, 50, 50)	−1	0.1429	1	5968
		4	(50, 50, 50, 50)	(50, 50, 50, 50)	−1	0.1228	1	5670
		4	(75, 75, 75, 75)	(50, 50, 50, 50)	−1	0.2517	3	11,930
8	ours	2	(25, 25)	(15, 5)	−3250	0.0576	5	2421
		2	(50, 50)	(15, 5)	−3250	0.0655	5	2620
		2	(75, 75)	(15, 5)	−3250	0.0755	5	2819
	[18]	2	(25, 25)	(15, 5)	−3250	0.0621	5	2730
		2	(50, 50)	−	−	−	−	−
		2	(75, 75)	(15, 5)	−3250	0.0840	5	3132
9	ours	6	(0, 0, 0, 0, 0, 0)	(5, 1, 5, 0, 5, 10)	−310	0.2280	5	11,263
		6	(3, 4, 2, 3, 5, 5)	(5, 1, 5, 0, 5, 10)	−310	0.1689	3	8639
		6	(6, 8, 5, 6, 10, 10)	(5, 1, 5, 0, 5, 10)	−310	0.1304	1	6589
	[18]	6	(0, 0, 0, 0, 0, 0)	(5, 1, 5, 0, 5, 10)	−310	0.3489	5	16,037
		6	(3, 4, 2, 3, 5, 5)	(5, 1, 5, 0, 5, 10)	−310	0.2197	3	10,017
		6	(6, 8, 5, 6, 10, 10)	(5, 1, 5, 0, 5, 10)	−310	0.2649	3	12,153
10	ours	4	(5, ⋯, 5)	(1, 1, 1, 1)	0	0.1000	5	5761
		4	(−5, ⋯, −5)	(1, 1, 1, 1)	0	0.0448	2	2667
		4	(10, ⋯, 10)	(1, 1, 1, 1)	0	0.1169	6	7969
		4	(−10, ⋯, −10)	(1, 1, 1, 1)	0	0.0542	2	2827
	[18]	4	(5, ⋯, 5)	(1, 1, 1, 1)	0	0.1075	5	6761
		4	(−5, ⋯, −5)	(1, 1, 1, 1)	0	0.0707	2	4615
		4	(10, ⋯, 10)	(1, 1, 1, 1)	0	0.1245	6	7969
		4	(−10, ⋯, −10)	(1, 1, 1, 1)	0	0.0756	2	4775

Table 2. Continued.

PN	Alg	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
11	ours	2	(10, -10)	(3, 0.5)	0	1.0960	5	742,921
		2	(9.997, -6.867)	(3, 0.5)	0	12.5392	5	730,377
		2	(0, -1)	(3, 0.5)	0	4.0511	4	244,019
		2	(1, 1)	(3, 0.5)	0	3.8061	4	237,587
		2	(2, 2)	(3, 0.5)	0	3.7028	4	237,579
		2	(0, 0)	(3, 0.5)	0	3.8508	4	240,019
	[18]	2	(10, -10)	(3, 0.5)	0	15.7839	5	892,789
		2	(9.997, -6.867)	(3, 0.5)	0	15.1406	5	880,245
		2	(0, -1)	(3, 0.5)	0	6.0974	4	470,075
		2	(1, 1)	(3, 0.5)	0	5.8505	4	463,643
		2	(2, 2)	(3, 0.5)	0	5.8602	4	463,635
		2	(0, 0)	(3, 0.5)	0	5.9400	4	466,075

Table 3. Numerical comparison results of Problems 12 and 13.

PN	Alg	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
12	ours	25	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	7.1418	1	258,951
		25	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	13.4385	2	496,651
		25	(5, ..., 5)	(1, ..., 1)	0	14.1907	2	495,451
		25	(-5, ..., -5)	(1, ..., 1)	0	8.8529	1	318,901
	[18]	25	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	7.4281	1	269,332
		25	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	15.7155	2	580,802
		25	(5, ..., 5)	(1, ..., 1)	0	14.0939	2	495,451
		25	(-5, ..., -5)	(1, ..., 1)	0	9.1232	1	328,637
	ours	50	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	59.5855	1	2,035,301
		50	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	113.7989	2	3,985,901
		50	(5, ..., 5)	(1, ..., 1)	0	114.7591	2	3,980,901
		50	(-5, ..., -5)	(1, ..., 1)	0	73.8136	1	2,525,301
	[18]	50	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	62.4965	1	2,134,733
		50	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	116.8327	2	4,092,162
		50	(5, ..., 5)	(1, ..., 1)	0	116.1587	2	4,029,451
		50	(-5, ..., -5)	(1, ..., 1)	0	75.7253	1	2,590,704
	ours	100	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	515.5150	1	16,140,600
		100	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	996.9730	2	31,941,801
		100	(5, ..., 5)	(1, ..., 1)	0	983.3028	2	31,921,800
		100	(-5, ..., -5)	(1, ..., 1)	0	637.1618	1	20,100,602
	[18]	100	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	523.2555	1	16,382,953
		100	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	1026.1101	2	32,874,946
		100	(5, ..., 5)	(1, ..., 1)	0	996.5619	2	32,352,242
		100	(-5, ..., -5)	-	-	-	-	-
13	ours	25	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	7.1793	1	240,873
		25	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	14.0434	2	443,610
		25	(5, ..., 5)	(1, ..., 1)	0	12.7954	2	495,453
		25	(-5, ..., -5)	(1, ..., 1)	0	8.5232	1	318,902
	[18]	25	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	7.7181	1	258,951
		25	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	15.7225	2	496,652
		25	(5, ..., 5)	(1, ..., 1)	0	13.8766	2	537,318
		25	(-5, ..., -5)	(1, ..., 1)	0	9.3565	1	350,080
	ours	50	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	58.2227	1	2,035,301
		50	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	134.6884	2	4,931,800
		50	(5, ..., 5)	(1, ..., 1)	0	108.2429	2	3,980,891
		50	(-5, ..., -5)	(1, ..., 1)	0	70.6625	1	2,525,306
	[18]	50	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	62.0151	1	2,185,716
		50	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	117.6758	2	3,985,901
		50	(5, ..., 5)	(1, ..., 1)	0	117.5395	2	4,322,674
		50	(-5, ..., -5)	(1, ..., 1)	0	76.3166	1	2,727,364
	ours	100	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	483.2232	1	16,140,601
		100	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	994.5747	2	39,723,601
		100	(5, ..., 5)	(1, ..., 1)	0	927.8696	2	31,921,801
		100	(-5, ..., -5)	(1, ..., 1)	0	594.7694	1	20,100,601
	[18]	100	(5, ..., 5, -5, ..., -5)	(1, ..., 1)	0	516.1914	1	17,241,803
		100	(-5, ..., -5, 5, ..., 5)	(1, ..., 1)	0	1555.5000	2	62,127,119
		100	(5, ..., 5)	(1, ..., 1)	0	995.4259	2	34,245,962
		100	(-5, ..., -5)	-	-	-	-	-

Table 4. Numerical results of Problems 10 and 12.

PN	DN	x_0	x^*	$f(x^*)$	T	Iter	$N_{F+\Psi}$
10	4	(1, 1, 0, 0)	(1, 1, 1, 1)	0	0.0500	3	3023
	4	(1, 1, 1, 1)	(1, 1, 1, 1)	0	0.0362	1	2393
	4	(−10, 10, −10, 10)	(1, 1, 1, 1)	0	0.4176	7	8947
	4	(−10, −5, 0, 5)	(1, 1, 1, 1)	0	0.0730	3	4539
	4	(−10, 0, 0, −10)	(1, 1, 1, 1)	0	0.0548	3	3201
	4	(0, 0, 0, 0)	(1, 1, 1, 1)	0	0.0462	3	3041
12	25	(0, ⋯, 0)	(1, ⋯, 1)	0	8.5956	2	312,651
	25	(3, ⋯, 3)	(1, ⋯, 1)	0	9.3240	1	493,051
	25	(−5, ⋯, −5)	(1, ⋯, 1)	0	8.6237	3	318,901
	25	(2, −2, ⋯, 2, −2, 2)	(1, ⋯, 1)	0	7.0355	1	315,151
	25	(3, −3, ⋯, 3, −3, 3)	(1, ⋯, 1)	0	11.7225	2	611,549
	25	(5, −5, ⋯, 5, −5, 5)	(1, ⋯, 1)	0	12.7954	2	613,949

To better reveal the effectiveness of our algorithm, we compared it with the algorithm in [18]. For fair, these two algorithms all use the same initial point and the same local descent Algorithm 1. Then, these two algorithms are adopted to test 13 kinds of test problems (72 test examples in total) in the same computing environment. Finally, the T, Iter, and $N_{F+\Psi}$ of these two algorithms are compared, respectively. Furthermore, all boldface data in these tables indicate that our algorithm is superior to that in [18].

The numerical results in Tables 1–3 show that the proposed algorithm can accurately find the global minimizers of these 72 test instances and solve the test problem of up to 100 variables, which indicates that our algorithm is feasible. Among all the test instances, the algorithm in [18] fails to solve three of them. The comparison results of numerical calculation of the 72 test examples in Tables 1–3 show that these two algorithms have almost the same number of iterations, which indicates that their performance may be similar in a sense. However, the proposed algorithm can always find the global minimizer faster than the algorithm in [18] for 66 test instances, and our algorithm had 64 test instances with less $N_{F+\Psi}$ than the latter, which shows that our algorithm can save more CPU time than the latter.

To further test the effectiveness of our algorithm, we calculated Problems 4–6, 8, and 9 using our algorithm and algorithms A, B, C, and D in [19] and compared the total evaluation times of functions (include filled function and objective function) solved by the five algorithms. The comparison results are shown in Table 5. Note that for problems 4, 5, and 11, the initial points (in Tables 1 and 2) adopted by our algorithm are the same as those in [19]; however, for problems 10 and 12, the initial points used in Tables 2 and 3 are different from those in [19]. Therefore, by adopting the same initial point as in [19], we have carried out additional numerical experiments on problems 10 and 12, which ensures the fairness of the numerical results (see Table 4).

Table 5. Numerical comparison results of Problems 4, 5, and 10–12.

PN	DN	x^*	$f(x^*)$	$N_{F+\Psi}$				
				Ours	A	B	C	D
4	4	(0, 0, 0, 0)	0	3388	9065	6472	37,906	18,936
5	2	(0, −1)	3	78,041	266,455	173,605	1,218,432	663,573
10	4	(1, 1, 1, 1)	0	4024	9735	6927	39,650	19,252
11	2	(3, 0.5)	0	405,417	1,970,575	1,145,488	245,235	2,826,691
12	25	(1, ⋯, 1)	0	396,936	1,173,742	646,390	3,558,639	1,732,339

The numerical comparison results of $N_{F+\Psi}$ in Table 5 indicate that our algorithm is superior to A, B, and D in solving these five problems and is inferior to C in solving Problem 11. This means that algorithm C may be better than the other four algorithms at

solving certain problems. Besides, the above results also show that most numerical results of algorithm B are better than algorithms A, C, and D, except for Problem 11.

To sum up, on the premise of ensuring the effectiveness, the proposed algorithm can solve the difficult constrained integer programming problem with few function evaluations in a short time, and the computational performance is better than the algorithms in [18,19], which shows that our algorithm is efficient. The disadvantage of our algorithm is that it is difficult to solve higher-dimensional problems.

7. Conclusions

In this paper, a new nonparametric filled function is constructed, which not only has good discreteness but also has no exponential and logarithmic terms. On the basis of theoretical analysis, a new global algorithm for solving constrained integer programming problems is designed, and its feasibility and effectiveness are verified by numerical comparison experiments. In future studies, we will try to apply the proposed algorithm to other problems in the optimization field, such as fractional programming problems with constraints [28]. The equivalent transformation method in this paper is promising, so further work could propose the filled function with better properties.

Author Contributions: S.M. carried out the methodology, investigation, and writing the draft. Y.G. supervised the research and edited and reviewed the final draft. B.Z. and W.Z. performed the experiments and reviewed the final draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant (11961001), the Construction Project of first-class subjects in Ningxia higher Education (NXYLXK2017B09), and the Major proprietary funded project of North Minzu University (ZDZX201901).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: We declare there is no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PN	The number of the problem
Alg	Algorithm type
DN	Number of decision variables
T	CPU running time
Iter	The number of iterations when the algorithm terminates
$N_{F+\Psi}$	The number of function evaluations of $f(x)$ and $\Psi(x, x_k^*)$ before termination
–	Global optimization failed

References

1. Ge, R.P. A filled function method for finding a global minimizer of a function of several variables. *Math. Program.* **1990**, *46*, 191–204.
2. Ge, R.P.; Huang, C.B. A continuous approach to nonlinear integer programming. *Appl. Math. Comput.* **1989**, *34*, 39–60. [[CrossRef](#)]
3. Juan, D.M.; Hugo, D.S. An augmented filled function for global nonlinear integer optimization. *TOP* **2020**, *28*, 689–704.
4. Levy, A.V.; Montalvo, A. The Tunneling Algorithm for the Global Minimization of Functions. *J. Korean Soc. Ind. Appl.* **1985**, *6*, 15–29. [[CrossRef](#)]
5. Lee, W.J.; Cabot, A.V.; Venkataramanan, M.A. A branch and bound algorithm for solving separable convex integer programming problems. *Comput. Oper. Res.* **1994**, *21*, 1011–1024. [[CrossRef](#)]
6. Borodin, L.P.; Konyagin, S.V. Projection greedy algorithm. *Math. Notes* **2021**, *110*, 16–25. [[CrossRef](#)]
7. Rosen, S.L.; Harmonosky, C.M. An improved simulated annealing simulation optimization method for discrete parameter stochastic systems. *Comput. Oper. Res.* **2005**, *32*, 343–358. [[CrossRef](#)]

8. Turkkan, N. Discrete optimization of structures using a floating-point genetic algorithm. Fourth International Conference on Modelling. In Proceedings of the Annual Conference of the Canadian Society for Civil Engineering, Moncton, NB, Canada, 4–7 June 2003.
9. Laurent, M.; Pascal, V.H. A simple tabu search for warehouse location. *Eur. J. Oper. Res.* **2004**, *157*, 576–591.
10. Zhu, W.X.; Zhang, L.S. An approximate algorithm for nonlinear integer programming. *Eur. J. Oper. Res.* **1994**, *74*, 170–178. [[CrossRef](#)]
11. Gu, Y.H.; Wu, Z.Y. A new filled function method for nonlinear integer programming problem. *Appl. Math. Comput.* **2006**, *173*, 938–950. [[CrossRef](#)]
12. Ng, C.K.; Zhang, L.S.; Li, D.; Tian, W.W. Discrete filled function method for discrete global optimization. *Comput. Optim. Appl.* **2005**, *31*, 87–115. [[CrossRef](#)]
13. Yang, Y.J.; Liang Y.M. A new discrete filled function algorithm for discrete global optimization. *J. Comput. Optim.* **2007**, *202*, 280–291.
14. Wu, Z.Y.; Mammadov, M.; Bai, F.S. A Filled Function Method for Box-constrained System of Nonlinear Equations. In Proceedings of the APCCAS 2006—2006 IEEE Asia Pacific Conference on Circuits and Systems, Singapore, 4–7 December 2006; pp. 622–625.
15. Yang, Y.J.; Wu, Z.Y.; Bai, F. A Filled Function Method For Constrained Nonlinear Integer Programming. *J. Ind. Manag. Optim.* **2008**, *4*, 353–362. [[CrossRef](#)]
16. Lin, Y.J.; Yang, Y.J. A new filled function method for constrained nonlinear equations. *Appl. Math. Comput.* **2012**, *219*, 3100–3112. [[CrossRef](#)]
17. Yuan, L.Y.; Wan, Z.P.; Tang, Q.H.; Zheng, Y. A class of parameter-free filled functions for box-constrained system of nonlinear equations. *Acta Math. Appl. Sin.* **2016**, *32*, 355–364. [[CrossRef](#)]
18. Lin, H.W.; Wang, Y.P.; Wang X.L. An auxiliary function method for global minimization in integer programming. *Math. Probl. Eng.* **2011**, *2*, 402437. [[CrossRef](#)]
19. Woon, S.F.; Rehbock, V. A critical review of discrete filled function methods in solving nonlinear discrete optimization problems. *Appl. Math. Comput.* **2010**, *217*, 25–41. [[CrossRef](#)]
20. Liu, Q.; Xu, Y.Q.; Zhou, Y. A class of exact penalty functions and penalty algorithms for nonsmooth constrained optimization problems. *J. Glob. Optim.* **2020**, *76*, 745–768. [[CrossRef](#)]
21. Ng, C.K.; Li, D.; Zhang, L.S. Discrete global descent method for discrete global optimization and nonlinear integer programming. *J. Glob. Optim.* **2007**, *37*, 357–379. [[CrossRef](#)]
22. Pirlot, M. General local search methods. *Eur. J. Oper. Res.* **1996**, *92*, 493–511. [[CrossRef](#)]
23. Shang, Y.L.; Zhang, L.S. Finding discrete global minima with a filled function for integer programming. *Eur. J. Oper. Res.* **2008**, *189*, 31–40. [[CrossRef](#)]
24. Yang, Y.J.; He, M.L.; Gao, Y.L. Discrete Global Optimization Problems with a Modified Discrete Filled Function. *J. Oper. Res. Soc. China* **2015**, *3*, 297–315. [[CrossRef](#)]
25. Mohan, C.; Nguyen, H.T. A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems. *Comput. Optim. Appl.* **1999**, *14*, 103–132. [[CrossRef](#)]
26. Schittkowski, K. *More Test Examples for Nonlinear Programming Codes*; Springer: Berlin/Heidelberg, Germany, 1987.
27. Conley, W. *Computer Optimization Techniques*; Petrocelli Books Inc.: New York, NY, USA, 1980.
28. Moustaid, M.B.; Rikouane, A.; Dali, I.; Laghdir, M. Sequential approximate weak optimality conditions for multiobjective fractional programming problems via sequential calculus rules for the Brøndsted-Rockafellar approximate subdifferential. *Rend. Circ. Mat. Palerm.* **2021**, 1–18. [[CrossRef](#)]