



---

*Research article*

## A new filled function method based on global search for solving unconstrained optimization problems

Jia Li<sup>1</sup>, Yuelin Gao<sup>2,\*</sup>, Tiantian Chen<sup>1</sup> and Xiaohua Ma<sup>2</sup>

<sup>1</sup> School of Mathematics and Information Sciences, North Minzu University, Yinchuan 750021, China

<sup>2</sup> Ningxia province cooperative innovation center of scientific computing and intelligent information processing, North Minzu University, Yinchuan 750021, China

\* **Correspondence:** Email: gaoyuelin@263.net.

**Abstract:** The filled function method is a deterministic algorithm for finding a global minimizer of global optimization problems, and its effectiveness is closely related to the form of the constructed filled function. Currently, the filled functions mainly have three drawbacks in form, namely, parameter adjustment and control (if any), inclusion of exponential or logarithmic functions, and properties that are discontinuous and non-differentiable. In order to overcome these limitations, this paper proposed a parameter-free filled function that does not include exponential or logarithmic functions and is continuous and differentiable. Based on the new filled function, a filled function method for solving unconstrained global optimization problems was designed. The algorithm selected points in the feasible domain that were far from the global minimum point as initial points, and improved the setting of the step size in the stage of minimizing the filled function to enhance the algorithm's global optimization capability. In addition, tests were conducted on 14 benchmark functions and compared with existing filled function algorithms. The numerical experimental results showed that the new algorithm proposed in this paper was feasible and effective.

**Keywords:** unconstrained global optimization; filled function method; global minimizer; parameter-free; step size setting

**Mathematics Subject Classification:** 90C26, 90C30

---

### 1. Introduction

Global optimization methods have been widely used in many fields, such as economics, computer applications, artificial intelligence, bioengineering, etc., and have developed into an important research area. However, due to the existence of exponentially many local minima in global optimization

problems, the functions of these multi-minima points pose the following difficulties in finding the global optimal solution: How to search for a better local minimum point from the domain where the current local minimum point is located, and how to determine whether the current minimum point is the global minimum point. With the wide application of global optimization problems, both the theory and algorithms of global optimization have been developed accordingly. Many methods have been proposed for solving global optimization problems, which can be broadly classified into two categories, i.e., deterministic algorithms and stochastic algorithms. Stochastic algorithms mainly include: cuckoo search algorithm [1], grey wolf optimization algorithm [2], pigeon-inspired optimization algorithm [3, 4], Harris hawks optimization algorithm [5], snow geese algorithm [6], and other bionic algorithms. This type of method utilizes probabilistic mechanisms to search for satisfactory solutions within the feasible domain of an optimization problem. However, due to a lack of theoretical guarantees, these methods often get stuck in local minima and exhibit slower convergence rates. In contrast, deterministic algorithms can leverage the mathematical properties of the problem to ensure convergence to the global minimum within a specified error tolerance after a finite number of iterations. Deterministic algorithms mainly include: the filled function method [7, 8], the branch and bound method [9], the tunneling method [10], and other algorithms; Among them, the filled function method is a type of deterministic algorithm capable of globally optimizing optimization problems, which has a strong ability to jump out of the current non-global minima, and it is one of the better ways to study and solve the first difficult problem, and it has a very high efficiency and practicability in practical applications. The method was originally proposed by Ge in [7], and its core idea is to construct an auxiliary function called the filled function at the current local minimum point of the objective function, and use it as a bridge to jump out of the current non-global minima, so that the objective function can keep searching for better local minima to avoid falling into the local optimum.

In general, the filled function method is an optimization algorithm that involves a two-stage alternating loop. The first stage is the minimization phase, which starts from a given initial point and uses a local search algorithm to obtain a local minima  $x_1^*$  of the objective problem. Then, transitioning to the second stage, i.e., the filling stage, where a filled function is constructed at  $x_1^*$ , and the filled function is minimized by using the points near the current minima as the initial points. If the minima  $x_1^*$  of the objective problem obtained in the minimization stage is a non-global minima, then the minima  $x'$  of the filled function must be in a basin which is lower than  $B_1^*$ , and  $x'$  serves as a better feasible point for the objective problem. Subsequently, the objective problem is resolved with  $x'$  as the new initial point, thus jumping out of the current local minima and finding another better point. The above two phases alternate until no better minima  $x'$  can be found in the filling phase, then the current local minima of the objective problem is considered to be the globally optimal solution. The original definition of the filled function was proposed by Ge in [7], and it is defined as follows:

**Definition 1.** A function  $P(x)$  is called the filled function of  $f(x)$  at the local minimum point  $x_1^*$  if  $P(x)$  has the following properties:

- (i)  $x_1^*$  is a maximizer of  $P(x)$  and the whole basin  $B_1^*$  of  $f(x)$  at  $x_1^*$  becomes a part of a hill of  $P(x)$ ;
- (ii)  $P(x)$  has no minimizers or saddle points in any higher basin of  $f(x)$  than  $B_1^*$  ;
- (iii) If  $f(x)$  has a lower basin than  $B_1^*$ , then there is a point  $x'$  in such a basin that minimizes  $P(x)$  on the line through  $x'$  and  $x_1^*$ .

In reference [7], the specific form of the first-generation filled function is also provided:

$$P(x, x_1^*, \gamma, \rho) = \frac{1}{\gamma + f(x)} \exp\left(-\frac{\|x - x_1^*\|^2}{\rho^2}\right), \quad (1.1)$$

where  $\gamma$  and  $\rho$  are two parameters that need to be scientifically adjusted to ensure that the filled function method can be executed and that global minima can be obtained. Tuning these parameters poses a significant challenge, stemming from the unpredictable nature of their ideal values. Only via numerous trials can we home in on the right settings, a process that inevitably adds to the algorithm's computational intricacy. Additionally, the filled function contains an exponential term  $\exp(-\frac{\|x-x_1^*\|^2}{\rho^2})$ . When the parameter  $\rho$  is too small or the value of  $\|x - x_1^*\|$  is too large, the graph of the filled function tends to be flat. This makes it difficult for the algorithm to discern changes in the filled function values, thereby causing the objective function to fall into local optima. Since there is still room for improvement in the form of the first-generation filled function, scholars have successively proposed some bi-parameter filled functions [11–16]. However, this type of filled function still faces issues such as strong dependence on parameters or exponential terms. In order to reduce the number of parameters, scholars also proposed some single-parameter filled functions. For example, the Q-function introduced by Ge and Qin [8] has the following form:

$$Q(x, x_1^*, \gamma) = -[f(x) - f(x_1^*)] \exp(\gamma \|x - x_1^*\|). \quad (1.2)$$

However, the filled function still contains exponential terms. In order to overcome the drawbacks caused by exponential terms, Liu proposed a new class of filled functions [17, 18]:

$$H(x, x_1^*, \mu) = \frac{1}{\ln(1 + f(x) - f(x_1^*))} - \mu \|x - x_1^*\|^2, \quad (1.3)$$

$$H(x, x_k^*, \mu) = \frac{1}{\arctan(f(x) - f(x_k^*))} - \mu \|x - x_k^*\|^p. \quad (1.4)$$

The introduction of  $H(x, \mu)$  reduces the number of parameters of the filled function from two to one and eliminates exponential terms. However, it is discontinuous at points  $x \in S = \{x \mid f(x) = f(x_1^*)\}$ , where gradient information is not available. This limitation restricts the choice of local minimization methods during the minimization phase.

In response to the drawbacks of the Q-function, Lin proposed a one-parameter filled function in 2014 that does not contain exponential terms and is continuously differentiable, but still suffers from the disadvantage of needing to adjust the parameters, which is of the following form [19]:

$$FF(x, x_1^*, p) = \frac{p}{1 + \|x - x_1^*\|} h_{\frac{1}{p}}(f(x) - f(x_1^*)), \quad (1.5)$$

where  $p$  is a parameter, and

$$h_c(t) = \begin{cases} c & t \geq 0 \\ t^3 + c & t < 0 \end{cases}.$$

In further research on filled functions, we have discovered that regardless of the number of parameters incorporated, they inevitably exert a negative impact on the algorithm's performance.

Consequently, several scholars have proposed various types of parameter-free filled functions [20–24]. However, their numerical experimental performance is often less than ideal. For instance, while the filled function in [22] is devoid of parameters and exponential terms, it incorporates a sign function within its formulation. Unfortunately, this sign function is characterized by its discontinuity and non-differentiability, rendering gradient-based local minimization methods ineffective during the minimization phase. The filled function presented in [23] successfully circumvents this significant limitation, yet it incorporates exponential terms that, as we have previously discussed, have the potential to hinder the effectiveness of the filled function algorithm. This integration of exponential terms, while possibly beneficial in certain contexts, may adversely affect the algorithm's performance in terms of stability, computational efficiency, and its ability to accurately converge to a global optimum. In [24], a novel continuous and differentiable filled function is introduced, distinguished by its exclusion of both parametric and exponential terms. Moreover, an algorithm is presented that is based on this filled function and incorporates coordinate directions during the minimization process. During the minimization process of the filled function in this algorithm, coordinate directions are utilized, but the determination of the perturbation step size  $\alpha_0$  is not thoroughly deliberated. Notably, selecting an unsuitable initial point could gravely impact the efficacy of the filled function, potentially impeding its ability to guide the objective function toward a superior local minimum. Therefore, an inappropriate choice of the initial point may deteriorate the overall performance of the algorithm. To address the aforementioned issue, this paper introduces a significant refinement to the filled function originally presented in reference [22]. Specifically, we substitute the sign function within the filled function with a continuously differentiable univariate function. As a result, we propose a novel continuously differentiable filled function that is devoid of parameters, exponential terms, and logarithmic terms. This refined function is designed to effectively tackle the unconstrained global optimization problem. Based on the proposed filled function, we have devised a groundbreaking filled function method (hereinafter referred to as NFFM). During the minimization phase of this NFFM, we deliberately choose coordinate directions as our primary search directions. When perturbing the current local minimum point, we implement an incremental sequence for adjusting the perturbation step size. Precisely, if the local minimum discovered along a specific coordinate direction under the initial perturbation step size fails to yield an improved local minimum for the objective function, we enlarge the perturbation step size and persist in our search for a superior local minimum along that same coordinate direction. This iterative process continues until we reach the boundary of the objective function's feasible region, thus guaranteeing that our algorithm conducts an exhaustive global search without overlooking any potentially advantageous local minima of the objective function.

The organization of the subsequent sections in this paper is outlined as follows: In Section 2, we commence with the introduction of pivotal prerequisite knowledge, establishing a robust theoretical framework for subsequent deliberations. Proceeding from this, we shall construct an unprecedented parameter-free filled function and conduct in-depth research and analysis of its theoretical properties. Moving on to Section 3, we will design an innovative NFFM based on the aforementioned parameter-free filled function. In Section 4, we undertake a meticulous series of numerical experiments. Through a selection of 14 emblematic multimodal test functions, we comprehensively validate the viability and efficacy of the proposed NFFM. These outcomes not only underscore the superiority of NFFM but also furnish pivotal reinforcement for subsequent practical implementations. Finally, in the concluding section of this paper, we will provide a comprehensive summary of the research content and explicitly

highlight the achievements and major contributions of this study.

## 2. A new filled function for unconstrained global optimization problems

### 2.1. Preliminaries

This paper considers the following unconstrained global optimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in R^n, \end{aligned} \tag{P}$$

where  $f(x) : R^n \rightarrow R$  is the objective function.

To ensure the existence of the global optimal solution for problem (P) and the feasibility of the proposed filled function method, the objective function should satisfy the following basic assumptions. Moreover, a related definitions are presented in this section.

**Assumption 1.** *The objective function in problem (P) is continuously differentiable.*

**Assumption 2.**  $\xi$  is a set of all local minimizers of problem (P), and  $\mathcal{L}^* = \{f(x) \mid x \in \xi\}$  is finite.

**Assumption 3.**  $f(x)$  is a coercive function, i.e.,  $\|x\| \rightarrow \infty, f(x) \rightarrow \infty$ .

Assumption 3 indicates that there exists a compact set  $\Omega \subset R^n$ , which contains all the local minima of  $f(x)$  in its interior, and the function values of  $f(x)$  at the boundary of  $\Omega$  are greater than the function values at any point inside  $\Omega$ . When the global optimization problem (P) satisfies the above assumptions, all the global minima of problem (P) will be contained within a bounded closed interval  $\Omega$ . Therefore, the problem (P) considered in this paper can be converted into:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{P^*}$$

where  $\Omega = \{x_i \in R \mid l_i \leq x_i \leq u_i, i = 1, 2, \dots, n\}$  is referred to as the box constraint of problem  $P^*$ .

**Lemma 1.** *If  $x_k^*$  is a local minimum point of  $f(x)$ , then:*

$$\begin{aligned} \Omega_1 &= \{x \in \Omega \mid f(x) \geq f(x_k^*), x \neq x_k^*\}, \\ \Omega_2 &= \{x \in \Omega \mid f(x) < f(x_k^*)\}. \end{aligned}$$

The original definition of the filled function was proposed by Ge in [7], which provides a reference criterion for researchers to construct filled functions that can escape from local non-global minima. However, the concepts of basins and peaks involved in the definition are more difficult to understand, and the third condition in the definition is not conducive to constructing and minimizing the filled function. In order to address these limitations, researchers have continuously improved the original definition of the filled function [11, 12, 25], among which the widely used definition [12] is as follows:

**Definition 2.** *A function  $P(x)$  is called the filled function of  $f(x)$  at the local minimum point  $x_1^*$  if  $P(x)$  has the following properties:*

- (i)  $x_k^*$  is a strictly local maximum of  $P(x, x_k^*)$ ;
- (ii) For  $P(x, x_k^*)$ , none of the points in  $\Omega_1$  is a stationary point;
- (iii) If  $x_k^*$  is not a global minimum of  $f(x)$ , then  $P(x, x_k^*)$  in  $\Omega_2$  has a local minimum.

The improved properties of the filled function ensure that when the local search method is used to minimize the constructed filled function, the sequence of iteration points does not terminate at any point where the objective function value is greater than  $f(x_k^*)$ . If  $x_k^*$  is not a global minima, then there must exist a local minimum  $x'$  of the filled function such that the function value at that point is less than the current local minimum value  $f(x_k^*)$  of the objective function. That is, any local minimum point of  $P(x, x_k^*)$  belongs to the set  $\Omega_2$ , enabling the objective function  $f(x)$  to escape from the current local minimum point and find a better minimum point by a local search algorithm starting from the minimum point of the filled function.

## 2.2. A new filled function and its properties

We assume that  $x_k^*$  represents the current local minimum point of  $f(x)$ . Based on Definition 2, we propose a novel continuous, differentiable, and parameter-free filled function:

$$\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|) \cdot (1 + \varphi(f(x) - f(x_k^*))), \quad (2.1)$$

$$\varphi(t) = \begin{cases} 0 & t \geq 0 \\ -t^2 & t < 0. \end{cases}$$

First of all, we establish through the subsequent lemma and theorem that  $x_k^*$  is a strictly local maximum for the function  $\Psi(x, x_k^*)$ .

**Lemma 2.** *If  $x_k^* \in \xi$  and  $x \in \Omega_1$ , then  $\Psi(x, x_k^*) < 0 = \Psi(x_k^*, x_k^*)$ .*

*Proof.* Since for  $x \in \Omega_1$ , we have  $f(x) \geq f(x_k^*)$ , then  $\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|^2)$ .

From the properties of function  $h(\omega) = -\arctan(\omega)$ , we know that: (1)  $h(\omega) > 0$ , if  $\omega < 0$ , (2)  $h(\omega) = 0$ , if  $\omega = 0$ , (3)  $h(\omega) < 0$ , if  $\omega > 0$ . Because  $\|x - x_k^*\|^2 > 0$ , and  $\|x_k^* - x_k^*\|^2 = 0$ , then  $\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|^2) < 0 = -\arctan(\|x_k^* - x_k^*\|^2) = \Psi(x_k^*, x_k^*)$ .  $\square$

**Theorem 1.** *If  $x_k^* \in \xi$ , then  $x_k^*$  is a strictly local maximum of  $\Psi(x, x_k^*)$ .*

*Proof.* Since  $x_k^*$  is a local minimum point of  $f(x)$ , then there is a very small positive number  $\delta > 0$ , for any  $\forall x \in B(x_k^*, \delta) \cap \Omega_1$ ,  $f(x) \geq f(x_k^*)$ . From Lemma 2, We know  $\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|^2) < 0 = \Psi(x_k^*, x_k^*)$ . Thus,  $x_k^*$  is a strictly local maximum of  $\Psi(x, x_k^*)$ .  $\square$

The above Lemma 2 and Theorem 1 indicate that the function  $\Psi(x, x_k^*)$  satisfies the first condition of Definition 2. Next, we prove that the function  $\Psi(x, x_k^*)$  also satisfies the second and third conditions of Definition 2. First, we define  $d^* = x - x_k^*$  as a feasible direction of  $f(x)$ .

**Theorem 2.** *For  $\forall x \in \Omega_1$ ,  $\nabla^T \Psi(x, x_k^*) d^* < 0$ .*

*Proof.* Since  $x_k^*$  is a local minimum point of  $f(x)$ , then for  $\forall x \in \Omega_1$ , we have  $f(x) \geq f(x_k^*)$ , and  $\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|^2)$ ; thus,

$$\nabla\Psi(x, x_k^*) = \frac{-2(x - x_k^*)}{1 + \|x - x_k^*\|^4},$$

$$\nabla^T\Psi(x, x_k^*)d^* = \frac{-2\|x - x_k^*\|^2}{1 + \|x - x_k^*\|^4} < 0.$$

□

**Remark 1.** By Theorem 2, the following can be concluded:

- (1) For each  $x \in \Omega_1$ , the previously defined feasible direction  $d^*$  is a descent direction for the function  $\Psi(x, x_k^*)$ .
- (2) For each  $x \in \Omega_1$ ,  $\nabla\Psi(x, x_k^*) \neq 0$ , thus none of the points in  $\Omega_1$  are stationary points or saddle points. This conclusion ensures that the numerical experimentation process will disregard regions where the objective function value exceeds the current local minimum value.
- (3) The minimization process of  $\Psi(x, x_k^*)$  will not stop within  $\Omega_1$ .
- (4) Theorem 2 proves that the function  $\Psi(x, x_k^*)$  satisfies the second condition in Definition 2.

**Theorem 3.** Assume that  $x_k^*$  is a local minimum point of  $f(x)$ . If it is not a global minimum point, then  $\Psi(x, x_k^*)$  has a local minimum point in  $\Omega_2$ . Furthermore, let  $d^* = x_q - x_k^*$ , if there exists  $x_q \in \Omega_2$  such that  $\nabla^T f(x_q)d^* > \eta > 0$ , then  $\nabla^T\Psi(x_q, x_k^*)d^* > 0$ , where

$$\eta = \frac{\|x_q - x_k^*\|^2 \cdot (1 - (f(x_q) - f(x_k^*))^2)}{(1 + \|x_q - x_k^*\|^4) \cdot \arctan(\|x_q - x_k^*\|^2) \cdot (f(x_q) - f(x_k^*))}. \quad (2.2)$$

*Proof.* Assume  $x_j^* \notin \Omega_2$  and is a local minimum of  $\Psi(x, x_k^*)$ , then  $x_j^* \in \Omega_1$ , and  $\nabla\Psi(x_j^*, x_k^*) = \frac{-2\|x_j^* - x_k^*\|^2}{1 + \|x_j^* - x_k^*\|^4} = 0$ , so  $x_j^* = x_k^*$ . From Theorem 1, we know  $x_k^*$  is a strictly local maximum of  $\Psi(x, x_k^*)$ , so  $x_j^* \neq x_k^*$ , it contradicts Theorem 1. If  $x_j^*$  is a saddle point, and it contradicts Theorem 2. Thus,  $x_j^*$  must be in  $\Omega_2$ .

Next, we will demonstrate that the feasible direction  $d^*$  is an ascent direction for some  $x_q \in \Omega_2$ . If there exists  $x_q \in \Omega_2$  such that  $\nabla^T f(x_q)d^* > \eta > 0$ , then, according to Eq (2.2), we know that

$$\frac{\|x_q - x_k^*\|^2 \cdot (1 - (f(x_q) - f(x_k^*))^2)}{1 + \|x_q - x_k^*\|^4} = \arctan(\|x_q - x_k^*\|^2) \cdot (f(x_q) - f(x_k^*)) \cdot \eta.$$

Equation (2.1) implies:

$$\Psi(x_q, x_k^*) = -\arctan(\|x_q - x_k^*\|^2) \cdot (1 - (f(x_q) - f(x_k^*))^2).$$

At this point, the gradient of  $\Psi(x_q, x_k^*)$  at  $x_q$  is:

$$\nabla\Psi(x_q, x_k^*) = \frac{-2(x_q - x_k^*)}{1 + \|x_q - x_k^*\|^4} \cdot (1 - (f(x_q) - f(x_k^*))^2) + [-\arctan(\|x_q - x_k^*\|^2) \cdot 2(f(x_q) - f(x_k^*)) \cdot \nabla f(x_q)].$$

Hence,

$$\begin{aligned}\nabla^T \Psi(x_q, x_k^*) d^* &= \frac{-2\|x_q - x_k^*\|^2}{1 + \|x_q - x_k^*\|^4} \cdot (1 - (f(x_k^*) - f(x_q))^2) \\ &\quad + 2\arctan(\|x_q - x_k^*\|^2) \cdot (f(x_k^*) - f(x_q)) \cdot \nabla^T f(x_q) \cdot (x_q - x_k^*) \\ &> \frac{-2\|x_q - x_k^*\|^2}{1 + \|x_q - x_k^*\|^4} \cdot (1 - (f(x_k^*) - f(x_q))^2) + 2\arctan(\|x_q - x_k^*\|^2) \cdot (f(x_k^*) - f(x_q)) \cdot \eta \\ &= 0.\end{aligned}$$

That is,

$$\nabla^T \Psi(x_q, x_k^*) d^* > 0.$$

□

Theorem 3 demonstrates that within the interval  $\Omega_2$ , the direction  $d^* = x_q - x_k^*$  is an ascent direction for  $\Psi(x, x_k^*)$ . By Assumption 1,  $f(x)$  is continuously differentiable, thus there exists a stationary point along the direction  $d^*$  for  $\Psi(x_q, x_k^*)$  within  $\Omega_2$ . Furthermore, Theorem 3 also confirms that Eq (2.1) satisfies the third condition of Definition 2.

**Theorem 4.** *If  $x_k^* \in \xi$ , and  $x_a, x_b \in \Omega_1$  such that  $\|x_a - x_k^*\| < \|x_b - x_k^*\|$ , then  $\Psi(x_b, x_k^*) < \Psi(x_a, x_k^*)$ .*

*Proof.* Since  $x_k^*$  is a local minimum of  $f(x)$ , and  $x_a, x_b \in \Omega_1$ , according to the filled function Eq (2.1), we have

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

because

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

and  $\|x_a - x_k^*\| < \|x_b - x_k^*\|$ , then  $\|x_a - x_k^*\|^2 < \|x_b - x_k^*\|^2$ . From the properties of the arctan function  $h(\omega) = -\arctan(\omega)$ , it can be inferred that  $\Psi(x_b, x_k^*) - \Psi(x_a, x_k^*) < 0$ . That is  $\Psi(x_b, x_k^*) < \Psi(x_a, x_k^*)$ . □

**Remark 2.** *Theorem 4 proves that for any two points in  $\Omega_1$ , the further away from  $x_k^*$ , the smaller the value of the filled function, which shows that in the interval  $\Omega_1$ ,  $\Psi(x, x_k^*)$  will always be a peak, implying that the minimization process of  $\Psi(x, x_k^*)$  will always be realized.*

**Theorem 5.** *If  $x_k^*$  is the global optimal solution of  $f(x)$ , then the filled function  $\Psi(x_q, x_k^*)$  is monotonically decreasing for all  $x \in \Omega$ .*

*Proof.* Because  $x_k^*$  is the global optimal solution of  $\varphi(x)$ , for  $\forall x \in \Omega$ , we have  $\varphi(x) > \varphi(x_k^*)$ . Thus, from Eq (2.1), we obtain  $\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|^2)$ . According to the property of the function  $h(\omega) = -\arctan(\omega)$ , we can conclude that  $\Psi(x, x_k^*)$  is monotonically decreasing for  $\forall x \in \Omega$ . □

Theorem 5 is an additional property of the filled function  $\Psi(x_q, x_k^*)$ , which shows that if  $x_k^*$  is a global optimal solution of  $f(x)$ , then  $\Psi(x, x_k^*)$  has no stationary points in the global search domain, and for any feasible direction  $d^* = x - x_k^*$ , there will be  $\nabla^T \Psi(x, x_k^*) d^* < 0$ .



### 3. A new filled function algorithm

#### 3.1. A new filled function algorithm without parameter

The previous section has rigorously demonstrated through in-depth theoretical analysis that the constructed filled function  $\Psi(x, x_k^*)$  in this paper fully satisfies the properties of a filling function outlined in Definition 2. Based on this foundation, the current section will further elaborate on the development of a novel parameter-free filled function algorithm—the NFFM, derived from the formulated filled function as indicated in Eq (2.1). Subsequently, we shall present the pseudocode of the innovative filled function algorithm introduced in this study and provide a thorough textual elaboration of its entire operational procedure. Furthermore, we will offer precise explanations for each individual step.

**Step 0:** Choose an initial point  $x_0 \in \Omega$ ;  $e_i, i = 1, 2, \dots, 2n$ , where  $e_i$  is a unit coordinate vector;  $n$  is the number of variables of the objective function  $f(x)$ ;  $\varepsilon > 0$  (let  $\varepsilon = 10^{-6}$ ); set  $k = 1, i = 1, a_0 = 0$ , where  $k$  represents the number of iterations; let  $\delta = \delta_0 + a_0 \cdot H$ , where  $\delta > 0$  is the step length,  $\delta_0$  is the given initial step length,  $H = D/G$  is the unit segment step length,  $D = \frac{l_1 - u_1}{2}$ , and  $G$  is the maximum number of segments, then go to Step 1.

**Step 1:** Take  $x_0$  as the initial point and use the local search method to minimize the objective function  $f(x)$  to obtain its local minima  $x_k^*$ , then go to Step 2;

**Step 2:** Construct the non-parameter filled function at  $x_k^*$

$$\Psi(x, x_k^*) = -\arctan(\|x - x_k^*\|) \cdot (1 + \varphi(f(x) - f(x_k^*)))$$

where,

$$\varphi(t) = \begin{cases} 0 & t \geq 0 \\ -t^2 & t < 0 \end{cases}$$

and go to Step 3.

**Step 3:** If  $a_0 \leq G$ , go to Step 4; otherwise,  $x_k^*$  serves as a global optimum solution for the objective function  $f(x)$ , and the algorithm terminates.

**Step 4:** If  $i \leq 2n$ , then let  $x_i = x_k^* + \delta e_i$  and go to Step 5; otherwise, let  $a_0 = a_0 + 1, \delta = \delta_0 + a_0 \cdot H, i = 1$ , and go to Step 3.

**Step 5:** If  $x_i \in \Omega$ , minimize  $\Psi(x, x_k^*)$  starting from  $x_i$  to obtain a local minimum point  $x_k^f$ , then go to Step 6; otherwise, let  $i = i + 1$  and go to Step 4.

**Step 6:** Using  $x_k^f$  as the initial point, minimize the objective function  $f(x)$  using a local minimization method to obtain a local minimum point  $x_k^f$ . If  $f(x_k^f) - f(x_k^*) < -\varepsilon$ , then set  $x_k^* = x_k^f, k = k + 1, i = 1, a_0 = 1, \delta = \delta_0 + a_0 \cdot H$ , and go to Step 2; otherwise, let  $i = i + 1$ , then go to Step 4.

Next, some detailed explanations are provided regarding the specific steps of NFFM:

(1) In the iteration process of NFFM, we commence by minimizing the objective function  $f(x)$  from a predefined initial point  $x_0$  within the feasible domain  $\Omega$  of the specified test function. To validate the escape capabilities of our proposed filled function algorithm, during experimentation, we select initial points that are situated relatively distant from the global minimum of the objective function, yet still within the feasible domain of the test function. Following this, we deploy local optimization algorithms to minimize the objective function, ultimately yielding a local minimum point denoted as  $x_k^*$ .

(2) Construct a filled function  $\Psi(x, x_k^*)$  at the local minimizer  $x_k^*$  of the objective function. Since  $x_k^*$  is a strict local maximizer of the filled function, it cannot be directly used as the initial point for minimizing the filled function. Therefore, it is necessary to perturb this point. In existing filled function algorithms, the initial point for minimizing the filled function is usually generated as  $x_i = x_k^* + \delta e_i$ , but the value of the perturbation step size  $\delta$  is rarely discussed. The choice of  $\delta$  is generally related to the number of local minimizers of the objective function or the size of the feasible domain. Therefore, in the proposed *NFFM*, during the stage of minimizing the filled function, an increasing arithmetic sequence with  $H$  as the common difference is constructed, and  $\delta = \delta_0 + a_0 \cdot H$  is set as the perturbation step size to generate the initial point for minimizing the filled function.

(3) The next step is to minimize  $\Psi(x, x_k^*)$  with the initial point  $x_i = x_k^* + \delta e_i$ . Since  $i \leq 2n$ , this process will be executed. If a local minimizer  $x_k^f$  of  $\Psi(x, x_k^*)$  is found, then  $x_k^f$  will be used as the initial point to minimize the objective function  $f(x)$  in order to obtain a local minimizer  $x_k^f$ . If the function value of  $f(x)$  at  $x_k^f$  is smaller than the function value at  $x_k^*$ , then  $x_k^*$  is replaced by  $x_k^f$  and the process is repeated. If not, then  $i = i + 1$ , change the search direction, and continue minimizing the filled function. If after iterating until  $i > 2n$ , a better local minimum of the objective function is still not obtained, then  $a_0$  is incremented by one, increasing the perturbation step size. With the updated perturbation step size, a new initial point is generated for minimizing the filled function  $\Psi(x, x_k^*)$  iteratively. This process of minimizing the filled function is repeated. If this iteration continues until reaching the threshold  $G$  of  $a_0$ , and a superior local minimum of the objective function  $f(x)$  is still not achieved, then  $x_k^*$  is regarded as the global minimum point and the algorithm terminates.

(4) Since both the objective function and the filled function are continuously differentiable, gradient-based local optimization algorithms can be utilized for the minimization process. In this paper, we employ the quasi-Newton method (short for *BFGS*) as the local optimization method to minimize the objective function and the filled function.

---

**Algorithm 1** A new filled function algorithm (NFFM).
 

---

```

1: Initialization:
2: Set parameters  $a_0, D, G, \varepsilon$ , etc
3:  $x \leftarrow x_0; k \leftarrow 1; i \leftarrow 1$ 
4: Main step:
5:  $x_k^* \leftarrow$  Optimization  $f(x)$  from  $x$  ▷  $x$  as the initial point minimizes the objective function
6: while  $a_0 \leq G$  do
7:   if  $i \leq 2 * n$  then
8:      $x_i = x_k^* + \delta e_i$ 
9:   else
10:     $a_0 \leftarrow a_0 + 1$ 
11:   end if
12:    $xk \leftarrow$  Optimization  $\Psi(x, x_k^*)$  from  $x_i$  ▷  $x_i$  as the initial point minimizes the filled function
13:    $xkf \leftarrow$  Optimization  $f(x)$  from  $xk$  ▷  $xk$  as the initial point minimizes the objective function
14:   if  $f(xkf) - f(x_k^*) < -\varepsilon$  then ▷ Update local optima
15:      $x \leftarrow xkf$ 
16:      $fbest \leftarrow f(xkf)$ 
17:      $iter \leftarrow iter + 1; i \leftarrow 1$ 
18:   else
19:      $i \leftarrow i + 1$ 
20:   end if
21: end while
22: return  $x, fbest, iter$ 

```

---

### 3.2. The time complexity analysis of NFFM

In this section, we will analyze the time complexity of NFFM jumping out one local minimum. NFFM is compatible with various local search algorithms, and during the experimental process, both NFFM and the other comparative algorithms adopt the same local search method. We assume that the time complexity of the local search algorithm for finding local solutions from an initial point is  $T(local)$ , a supposition that applies universally to all filled function methods. Subsequently, we utilize  $T(local)$  to optimize the objective function from the initial point  $x_0$ , aiming to obtain the local minimum  $x_k^*$ . Since  $x_k^*$  is a local maximum point of the filled function  $\Psi(x, x_k^*)$ , it is necessary to perturb  $x_k^*$  in order to minimize  $\Psi(x, x_k^*)$ . In step four of the NFFM algorithm process, the worst-case scenario is that each of the generated initial points begins to perform a local search on the filled function. The maximum number of the generated initial points for the filled function is  $(G + 1) * 2n$ , where  $n$  represents the dimension of the optimization problem and  $G$  is the maximum threshold value of  $a_0$ . Then, it will take  $(G + 1) * 2n * T(local) = O(n * T(local))$  for NFFM to optimize the filled function from all the generated initial points. Therefore, the time complexity of NFFM is  $T(local) + O(n * T(local)) + T(local) = O(n * T(local))$ .

## 4. Numerical experiment

To thoroughly validate the feasibility and effectiveness of NFFM, this section applies the algorithm to 14 classic test functions, which originate from a selection of global optimization problems listed in [26]. The results are then compared with four filled function algorithms that have emerged in the past four years. All numerical experiments were conducted using the Matlab(R2023a) software on a desktop computer equipped with a 64-bit Windows 10 operating system, an Intel(R) Core(TM) i5-8500 CPU @ 3.00 GHz processor, and 8.00 GB of RAM.

### 4.1. Test functions

**Problem 1.** Treccani function.

$$\begin{aligned} \min f(x) &= x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2, \\ \text{s.t. } & -3 \leq x_i \leq 3, i = 1, 2. \end{aligned}$$

The function is multimodal, with two global minima at  $x^* = (0, 0)$  or  $(-2, 0)$ , and the global optimum value is  $f(x^*) = 0$ . The NFFM successfully found the global minima and optimal values using initial points of  $(-1, -2)$  and  $(-3, 3)$ .

**Problem 2.** Six-hump back camel function.

$$\begin{aligned} \min f(x) &= 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 - x_1x_2 - 4x_2^2 + 4x_2^4, \\ \text{s.t. } & -3 \leq x_i \leq 3, i = 1, 2. \end{aligned}$$

The function is a commonly used multimodal, non-convex function, featuring six local minima and two global minima at either  $x^* = (-0.0898, -0.7127)$  or  $(0.0898, 0.7127)$ , with a global optimum value of  $f(x^*) = -1.0316$ . The NFFM successfully located the global minimum points and global optimal values using  $(2, -1)$  and  $(-3, 3)$  as initial points.

**Problem 3.** Rastrigin function.

$$\begin{aligned} \min f(x) &= x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \\ \text{s.t. } & -3 \leq x_i \leq 3, i = 1, 2. \end{aligned}$$

The function is a commonly used multimodal, non-convex function often employed as a test problem for assessing the performance of optimization algorithms. It features multiple local minima, presenting a challenging problem for optimization algorithms. It has a single global minimum at  $x^* = (0, 0)$ , with a global optimum value of  $f(x^*) = -2$ . The NFFM successfully found the global minima and optimal values using initial points of  $(1, 1)$  and  $(-2, -2)$ .

**Problem 4.** Three-hump back camel function.

$$\begin{aligned} \min f(x) &= 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2, \\ \text{s.t. } & -3 \leq x_i \leq 3, i = 1, 2. \end{aligned}$$

The function is a commonly used multimodal function with three local minima within the search domain and one global minimum. The global minimum is located at  $x^* = (0, 0)$ , with a global optimal

value of  $f(x^*) = 0$ . The NFFM successfully found the global minimum and its optimal value using initial points of  $(-2, 2)$  and  $(-3, 3)$ .

**Problem 5.** Two-dimensional function.

$$\begin{aligned} \min f(x) &= [1 - 2x_2 + c\sin(4\pi x_2) - x_1]^2 + [x_2 - 0.5\sin(2\pi x_1)]^2, \\ \text{s.t. } &0 \leq x_1 \leq 10, -10 \leq x_2 \leq 0, \end{aligned}$$

where  $c = 0.5, 0.2, 0.05$ .

The function is a multimodal, non-convex function with multiple local minima, but it has only one global optimum  $f(x^*) = 0$  across different coefficients  $c$ . When  $c = 0.5$ , the NFFM successfully found the global optimum using initial points  $(0, 0)$  and  $(5, -5)$ . Similarly, for  $c = 0.2$ , it utilized initial points  $(6, -2)$  and  $(0, -10)$ , and for  $c = 0.05$ , it employed initial points  $(10, -10)$  and  $(5, -5)$  to achieve the global optimum.

**Problem 6.** Goldstein and Price function.

$$\begin{aligned} \min f(x) &= g(x)h(x), \\ \text{s.t. } &-3 \leq x_i \leq 3, i = 1, 2, \end{aligned}$$

where,

$$\begin{aligned} g(x) &= 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2), \\ h(x) &= 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2). \end{aligned}$$

The function is a classic optimization test function commonly employed for evaluating the performance of optimization algorithms. It features multiple local minima and a complex curve shape, exhibiting non-convexity and nonlinearity in the two-dimensional space. Thus, it serves as a robustness and global search capability test for algorithms. Within the search domain, the function has only one global minimum at  $x^* = (0, -1)$ , with a global optimal value of  $f(x^*) = 3$ . The NFFM successfully found the global minimum and its optimal value using an initial point of  $(-3, -3)$ .

**Problem 7.** Two-dimensional Shubert function.

$$\begin{aligned} \min f(x) &= \left\{ \sum_{i=1}^5 \text{icos}[(i+1)x_1 + i] \right\} \left\{ \sum_{i=1}^5 \text{icos}[(i+1)x_2 + i] \right\}, \\ \text{s.t. } &-10 \leq x_i \leq 10, i = 1, 2. \end{aligned}$$

The function is a complex multimodal, non-convex function, containing approximately 760 local minima within the search domain. These local minima are distributed around multiple peaks of the function, posing a greater challenge for algorithms to locate the global minimum. The intricate structure and numerous local minima of this function make it more difficult to assess the robustness and global search capabilities of algorithms, hence it is utilized to evaluate the performance of the algorithm proposed in this paper. Using  $(5, 5)$  as the initial starting point, NFFM successfully found the global optimal value of the problem, which is  $f(x^*) = -186.7309$ .

**Problem 8.** Generalized penalized function.

$$\begin{aligned} \min f(x) &= \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}, \\ \text{s.t. } &-65.536 \leq x_i \leq 65.536, i = 1, 2, \end{aligned}$$

where,  $(a_{ij}) = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \dots & 32 & 32 & 32 \end{pmatrix}$ .

The function is one of the classic functions used to test the performance of optimization algorithms. It contains multiple local minima within its search domain, along with numerous steep valleys and peaks, presenting a challenge for optimization algorithms to find the global minimum. Therefore, testing the algorithm proposed in this paper on this function provides an effective evaluation of the algorithm's robustness and global search capabilities. The NFFM successfully found the global minimum point at the point  $x^* = (-32, -32)$  and the global optimal value of  $f(x^*) = 0.99800383779445$  starting from the initial points  $(-40, 20)$  and  $(50, 50)$ .

**Problem 9.** Shekel's function

$$\begin{aligned} \min f(x) &= - \sum_{i=1}^m \left[ \sum_{j=1}^4 (x_j - a_{i,j})^2 + c_i \right]^{-1}, \\ \text{s.t. } & 0 \leq x_j \leq 10, j = 1, 2, 3, 4, \end{aligned}$$

where  $m = 5, 7, 10$ ,  $a_{i,j}$  and  $c_i$  with  $i = 1, 2, 3, 4, 5$ ,  $j = 1, 2, 3, 4$  are provided in Table 1.

This function is one of the classic multimodal optimization test functions, featuring multiple local minima, which poses a certain challenge for optimization algorithms to search for the global minimum. It can be used to evaluate the global search ability and convergence speed of the algorithm. The global minima of this function is  $x^* = (4, 4, 4, 4)$ , and the global optimum is  $f(x^*) = -10.1532$  when  $m = 5$ ,  $f(x^*) = -10.4029$  when  $m = 7$ , and  $f(x^*) = -10.5364$  when  $m = 10$ . The NFFM proposed in this paper, with  $(0, 0, 0, 0)$  as the initial point, both successfully found its global minima and global optimum.

**Table 1.** The coefficients for Problem 9.

$i$	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	$c_i$
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.3
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.5
6	2.0	9.0	2.0	9.0	0.6
7	5.0	5.0	3.0	3.0	0.3
8	8.0	1.0	8.0	1.0	0.7
9	6.0	2.0	6.0	2.0	0.5
10	7.0	3.6	7.0	3.6	0.5

**Problem 10.** n-Dimensional Sine-square function.

$$\begin{aligned} \min f(x) &= \frac{\pi}{n} \left[ 10 \sin^2(\pi x_1) + \sum_{i=1}^{n-1} \left[ (x_i - 1)^2 (1 + 10 \sin^2(\pi x_{i+1})) \right] + (x_n - 1)^2 \right], \\ \text{s.t. } & -10 \leq x_i \leq 10, i = 1, 2, \dots, n. \end{aligned}$$

The function is characterized by its periodicity, nonlinearity, multimodality, and non-convexity, with multiple local minima. The number of local minima within the feasible domain is approximately  $10^n$ . This problem is a high-dimensional optimization problem, and NFFM focuses on solving problems

ranging from 2 to 200 dimensions. Starting from the initial point of  $(-7.5, -7.5 \dots, -7.5)$ , the algorithm successfully finds the global minimum  $x^* = (1, 1, \dots, 1)$ , and its global optimum  $f(x^*) = 0$ .

**Problem 11.** n-Dimensional Rastrigin function.

$$\begin{aligned} \min f(x) &= 10n + \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i)), \\ \text{s.t. } & -5.12 \leq x_i \leq 5.12, i = 1, 2, \dots, n. \end{aligned}$$

The function exhibits multimodal and non-convex characteristics, encompassing multiple local minima within the feasible region, as well as a global minimum located at the origin of the parameter space, with a global minimum value of 0. This function is commonly used to evaluate the performance of optimization algorithms when dealing with challenging functions. To address this high-dimensional optimization problem, the NFFM specifically targets optimization tasks ranging from 2 to 200 dimensions. Starting from the initial point  $(-2.56, -2.56 \dots, -2.56)$ , the algorithm successfully locates the global minimum point  $x^* = (0, 0, \dots, 0)$  and achieves the global optimal value of  $f(x^*) = 0$ .

**Problem 12.** Griewank function.

$$\begin{aligned} \min f(x) &= \sum_{i=1}^n \frac{x_i^2}{4000} - \sum_{i=1}^n \log \left[ 2 + \cos \frac{x_i}{\sqrt{i}} \right] + n \log 3, \\ \text{s.t. } & -200 \leq x_i \leq 400, i = 1, 2, \dots, n. \end{aligned}$$

The Griewank function is renowned for its high degree of nonlinearity and complex behavior in high-dimensional spaces, featuring multiple local minima and a single global minimum. This characteristic poses a significant challenge for many optimization algorithms in finding the global optimal solution. Therefore, it is commonly used as a benchmark test function to evaluate the performance of optimization algorithms. The NFFM proposed in this paper, focusing on the optimization search for 2–200 dimensional problems, takes  $(300, 300, \dots, 300)$  as the initial point and successfully finds its global minima  $x^* = (0, 0, \dots, 0)$  and the global optimum value of  $f(x^*) = 0$ .

**Problem 13.** Schewefel function.

$$\begin{aligned} \min f(x) &= 418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}), \\ \text{s.t. } & -500 \leq x_i \leq 500, i = 1, 2, \dots, n. \end{aligned}$$

The function represents a multidimensional nonlinear optimization problem. It has a wide range of values, typically spanning from  $[-500, 500]$ . The function exhibits multiple local optima, among which there is only one global optimum, occurring at  $x_i = 420.9687$ , with a global minimum value of  $f(x^*) = 0$ . Additionally, this function is a typical deceptive problem, as the global minimum is located far from another local optimum, and the function exhibits steep curves near the optimal solution, making it prone to falling into local optima. Escaping from a local optimum is challenging, posing high demands on the algorithm. Simple algorithms are unlikely to find the optimal solution, thus making it a suitable benchmark for verifying the global search capability and ability to escape local optima of the proposed algorithm in this paper. The NFFM algorithm introduced in this paper focuses on optimizing problems spanning 2 to 200 dimensions. Starting from the initial point  $(0, 0, \dots, 0)$ , it successfully finds the global minimum and the corresponding global optimal value.

**Problem 14.** Rosenbrock function.

$$\min f(x) = \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right],$$

$$s.t. \quad -30 \leq x_i \leq 30, i = 1, 2, \dots, n.$$

The function serves as a classic non-convex optimization problem for testing optimization algorithms. It possesses a global minimum point  $x^* = (1, 1, \dots, 1)$ , yet around this global minimum, there exists a long and tortuous valley. The valley is narrow, requiring optimization algorithms to overcome the function's tortuosity and avoid getting trapped in local optima in order to effectively find the global optimum. Therefore, it is used to evaluate the effectiveness and robustness of optimization algorithms in handling complex, multimodal, and long-valley optimization problems. The NFFM proposed in this paper, focusing on the optimization search for 2–200 dimensional problems, takes  $(-15, -15, \dots, -15)$  as the initial point and successfully finds its global minima  $x^* = (1, 1, \dots, 1)$  and the global optimum value of  $f(x^*) = 0$ .

#### 4.2. Numerical results

This section compares the NFFM with five other filled function algorithm on 14 multimodal test functions, including high-dimensional complex unconstrained optimization problems (Problems 10–14). In Table 2, we offer comprehensive clarifications and explanations for the symbols utilized in Tables 3–8.

**Table 2.** Interpretation of relevant symbols.

symbol	Explanation
$No.$	The number of the problem
$n$	Dimension of the objective function
$x_0$	The initial point of the objective function
$x^*$	The global minimum of the objective function
$f(x^*)$	The global optimum value of the objective function
$Iter$	The iteration number
$T(s)$	The total running time required until the algorithm stops (measured in seconds)
$NFf$	The total number of function evaluations of the objective and filled functions
$PFF1$	Two-parameter filled function algorithm in [16]
$PFF2$	Single-parameter filled function algorithm in [15]
$PFF3$	The parameter-free filled function algorithm in [21]
$PFF4$	The parameter-free filled function algorithm in [24]
$PFF5$	The parameter-free filled function algorithm in [22]

Table 3 details the specific numerical experimental results of NFFM on the 14 test functions. To verify the feasibility of NFFM, we employed the following evaluation metrics: the number of iterations ( $Iter$ ), the global optimum of the objective problem ( $f(x^*)$ ), the total number of function evaluations ( $NFf$ ), and the total running time ( $T(s)$ ). As shown in Table 3, NFFM successfully finds the global



optimal values for all the test problems, using points that are relatively far from the global minimum in the feasible region as the initial points. This demonstrates the effectiveness of the proposed non-parametric filled function algorithm in global optimization.

To comprehensively evaluate the performance of NFFM, we conducted a comparison with five other filled function algorithms (PFFF1–PFFF5) on 14 multimodal test functions. Tables 4 to 8 present detailed numerical results of these algorithms on the test functions, where “-” indicates a failure to locate the global optimum. These results were obtained by faithfully reproducing the procedures of the algorithms as described in the original literature. In our experiments, all algorithms used the same initial points and a local search method (Newton’s method), while the parameter settings for the comparison algorithms adhered to the values provided in the original literature. One of the algorithms, PFFF1, designed based on a two-parameter filled function, has been proven to outperform the filled function algorithms in [7, 13] as stated in [16]. PFFF2, while also based on a two-parameter filling function, was treated as a single-parameter algorithm in the experiments conducted in [15], where the parameter  $\epsilon$  was fixed at  $10^{-5}$ . When compared with algorithms in [7, 27], PFFF2 exhibited superior performance. Both PFFF3 and PFFF4 have shown better performance when compared to algorithms in [13, 22, 23], respectively. Additionally, the filled function proposed in this paper is an improvement of the one introduced in [13], and PFFF5 is a filled function algorithm designed based on the filled function from reference [13]. Comparing NFFM with these five algorithms provides strong evidence that the proposed NFFM algorithm possesses significant competitiveness.

**Table 3.** Numerical Experimental Results of NFFM for Problems 1–14.

<i>NO.</i>	<i>n</i>	$x_0$	$f(x^*)$	<i>Iter</i>	<i>NFf</i>	<i>T(s)</i>	<i>D/G</i>
1	2	(-1,2)	2.3314e-16	2	11466	0.3768	3/10
	2	(3,3)	1.6583e-13	1	2208	0.0689	3/10
2	2	(2,-1)	-1.0316	1	5403	0.1521	3/10
	2	(-3,3)	-1.0316	2	5250	0.1177	3/10
3	2	(1,1)	-2.0000	3	11187	0.3188	3/20
	2	(-2,-2)	-2.0000	3	11028	0.3202	3/20
4	2	(-2,2)	1.2063e-11	1	5361	0.2048	3/10
	2	(-3,3)	6.0581e-10	1	5370	0.1505	3/10
$(c = 0.5)$	2	(0,0)	1.0071e-11	2	5850	0.2304	10/20
	2	(5,-5)	1.3798e-10	4	6396	0.3038	10/20
5 $(c = 0.2)$	2	(6,-2)	2.5576e-10	2	5871	0.1729	10/20
	2	(0,-10)	8.3350e-11	2	11130	0.3122	10/20
$(c = 0.05)$	2	(10,-10)	5.7877e-11	3	13356	0.3738	10/50
	2	(5,-5)	4.3862e-8	2	5478	0.1576	10/20
6	2	(-3,-3)	3.0000	1	5496	0.1463	3/10
7	2	(5,5)	-186.7309	3	5286	0.1182	20/30
8	2	(-40,20)	0.998003938	3	12075	0.6224	65.536/65
	2	(50,50)	0.998003940	3	12147	0.6320	65.536/65
9 $(m = 5)$	4	(0,0,0,0)	-10.1532	2	45265	0.9075	10/20

*Continued on next page*

<i>NO.</i>		<i>n</i>	$x_0$	$f(x^*)$	<i>Iter</i>	<i>NFf</i>	<i>T(s)</i>	<i>D/G</i>
9	( <i>m</i> = 7)	4	(0,0,0,0)	-10.4029	2	51205	1.1519	10/20
	( <i>m</i> = 10)	4	(0,0,0,0)	-10.5364	2	47080	1.1174	10/20
10		2	(-7.5,-7.5)	4.4998e-14	3	31164	0.6559	10/20
		5	(-7.5, ..., -7.5)	3.6297e-15	9	91410	0.9884	10/20
		10	(-7.5, ..., -7.5)	5.0081e-09	7	399927	1.8787	10/20
		30	(-7.5, ..., -7.5)	1.2918e-11	12	5502810	15.6206	10/20
		50	(-7.5, ..., -7.5)	2.4807e-11	16	11562873	33.9494	10/20
		100	(-7.5, ..., -7.5)	1.4958e-8	51	183671429	721.4294	10/20
		200	(-7.5, ..., -7.5)	1.0357e-9	4	205588629	847.5251	10/20
11		2	(-2.56,-2.56)	3.4824e-11	5	34005	0.8710	5.12/50
		5	(-2.56, ..., -2.56)	4.3485e-12	11	214302	2.2427	5.12/50
		10	(-2.56, ..., -2.56)	1.2790e-13	14	767118	5.3170	5.12/50
		30	(-2.56, ..., -2.56)	1.6768e-9	23	8639917	35.6461	5.12/50
		50	(-2.56, ..., -2.56)	3.5811e-12	5	17655639	75.4816	5.12/50
		100	(-2.56, ..., -2.56)	4.5475e-13	75	87546396	450.2820	5.12/50
		200	(-2.56, ..., -2.56)	2.0600e-10	63	352329282	2655.6060	5.12/50
12		2	(300,300)	0	79	151752	4.9212	300/600
		5	(300, ..., 300)	0	148	1192452	14.8535	300/600
		10	(300, ..., 300)	3.0873e-12	102	5495303	30.7575	300/600
		30	(300, ..., 300)	5.0235e-12	132	107425695	176.2816	300/600
		50	(300, ..., 300)	2.0286e-11	117	348382122	520.6993	300/600
		100	(300, ..., 300)	9.7856e-11	130	607111303	1558.6716	300/600
		200	(300, ..., 300)	1.4511e-9	203	2019818247	10605.0670	300/600
13		2	(0,0)	2.5455e-5	7	10482	0.5411	500/100
		5	(0, ..., 0)	6.3638e-5	16	51006	1.4617	500/100
		10	(0, ..., 0)	1.2728e-4	31	219516	3.3319	500/100
		30	(0, ..., 0)	3.8183e-4	91	2481116	21.3445	500/100
		50	(0, ..., 0)	6.3638e-4	152	8590134	77.8044	500/100
		100	(0, ..., 0)	1.2728e-3	301	55840981	709.5823	500/200
		200	(0, ..., 0)	2.5130e-3	601	357280113	7149.2103	500/200
14		2	(-15,-15)	0.0566	1	7743	0.7729	30/10
		5	(-15, ..., -15)	1.4644e-8	3	44862	0.6354	30/10
		10	(-15, ..., -15)	2.5650e-4	2	133001	0.9756	30/10
		30	(-15, ..., -15)	2.0058e-7	2	1074181	4.5460	30/10
		50	(-15, ..., -15)	2.8443e-7	2	2922300	11.1526	30/10
		100	(-15, ..., -15)	4.3167e-7	2	11470065	39.3521	30/10
		200	(-15, ..., -15)	2.5979e-6	2	45435648	166.6213	30/10

**Table 4.** Comparison of numerical experiment results.

NO.	$n$	NFFM				PFFF1				
		Iter	$f(x^*)$	NFf	$T(s)$	Iter	$f(x^*)$	NFf	$T(s)$	
1	2	2	2.3314e-16	11466	0.3768	-	-	-	-	
	2	1	1.0583e-13	2208	0.0689	1	1.0583e-13	3200	0.2108	
2	2	1	-1.0316	5403	0.1521	1	-1.0316	3203	0.1430	
	2	2	-1.0316	5250	0.1177	3	-1.0316	4446	0.2326	
3	2	3	-2.0000	11187	0.3188	3	-2.0000	4304	0.2365	
	2	3	-2.0000	11028	0.3203	5	-1.7578	3722	0.1320	
4	2	1	1.2063e-11	5361	0.2048	1	1.2063e-11	3221	0.1647	
	2	1	6.0581e-10	5370	0.1505	1	6.0581e-10	3230	0.1433	
c=0.5	2	2	1.0071e-11	5850	0.2304	2	1.6582e-14	3511	0.1602	
	2	4	1.3798e-10	6396	0.3038	-	-	-	-	
5	c=0.2	2	2.5576e-10	5871	0.1729	1	0.6165	3257	0.1858	
	2	2	8.3350e-11	11130	0.3122	2	1.6212	3273	0.1320	
c=0.05	2	3	5.7877e-11	13356	0.3738	5	0.6184	4647	0.2486	
c=0.05	2	2	4.3862e-8	5478	0.1576	1	0.6184	3227	0.1473	
6	2	1	3.0000	5496	0.1463	1	3.0000	3287	0.1228	
7	2	3	-186.7309	5286	0.1182	2	-123.5768	3678	0.2178	
8	2	3	0.9980	12075	0.6224	3	0.9980	3442	0.2802	
	2	3	0.9980	12147	0.6320	649	0.9980	6414	0.6793	
9	m=5	4	2	-10.1532	45265	0.9075	1	-5.0552	10648	0.2654
	m=7	4	2	-10.4029	51205	1.1519	1	-5.0877	10663	0.5475
m=10	4	2	-10.5364	47080	1.1174	1	-5.1285	10663	0.5501	
10	2	3	4.4998e-14	31164	0.6559	5	3.7060e-14	4100	0.2099	
	5	9	3.6297e-15	91410	0.9884	5	6.9600e-4	17574	0.4629	
	10	7	5.0081e-9	399927	1.8787	5	3.5669e-12	75565	1.2692	
	30	12	1.2918e-11	5502810	15.6206	6	1.6043e-13	647069	8.3721	
	50	16	2.4807e-11	11562873	33.9494	29	1.4360e-15	3194755	45.2871	
	100	51	1.4958e-8	183671429	721.4294	1122.3393e-12	18688418	346.6963		
11	2	5	3.4824e-11	34005	0.8710	5	0	3767	0.2041	
	5	11	4.3485e-12	214302	2.2427	11	0	20876	0.5491	
	10	14	1.2790e-13	767118	5.3170	21	0	89474	1.4570	
	30	23	1.6768e-9	8639917	35.6461	61	0	1202189	15.7959	
	50	5	3.5811e-12	17655639	75.4816	101	0	4527401	66.7309	
	100	75	4.5475e-13	87546396	450.2820	201	1.1369e-13	30059598	602.1092	
12	2	79	0	151752	4.9212	10	1.3323e-14	7003	0.3397	
	5	148	0	1192452	14.8535	84	2.1156e-12	120565	2.7870	

*Continued on next page*

NO.	n	NFFM			PFFF1				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
12	10	102	3.0873e-12	5495303	30.7575	-	-	-	-
	30	132	5.0235e-12	107425695	176.2816	-	-	-	-
	50	117	2.0286e-11	348382122	520.6993	-	-	-	-
	100	130	9.7856e-11	607111303	1558.6716	-	-	-	-
	200	203	1.4511e-9	2019818247	10605.0670	-	-	-	-
13	2	7	2.5455e-5	10482	0.5411	-	-	-	-
	5	16	6.3638e-5	51006	1.4617	-	-	-	-
	10	31	1.2728e-4	219516	3.3319	-	-	-	-
	30	91	3.8183e-4	2481116	21.3445	-	-	-	-
	50	152	6.3638e-4	8590134	77.8044	-	-	-	-
	100	301	1.2728e-3	55840981	709.5823	-	-	-	-
14	200	601	2.5130e-3	357280113	7149.2103	-	-	-	-
	2	1	0.0566	7743	0.7729	1	0.0566	3332	0.1317
	5	3	1.4644e-8	44862	0.6354	1	0.0173	16376	0.4277
	10	2	2.5650e-4	133001	0.9756	-	-	-	-
	30	2	2.0058e-7	1074181	4.5460	-	-	-	-
	50	2	2.8443e-7	2922300	11.1526	-	-	-	-
	100	2	4.3167e-7	11470065	39.3521	-	-	-	-
200	2	2.5979e-6	45435648	166.6213	-	-	-	-	

**Table 5.** Comparison of numerical experimental results.

NO.	n	NFFM			PFFF2					
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)	
1	2	2	2.3314e-16	11466	0.3768	2	1.0118e-17	429	0.0439	
	2	1	1.0583e-13	2208	0.0689	1	1.0583e-13	330	0.1212	
2	2	1	-1.0316	5403	0.1521	1	-1.0316	267	0.0261	
	2	2	-1.0316	5250	0.1177	2	-1.0316	444	0.0500	
3	2	3	-2.0000	11187	0.3188	3	-2.0000	876	0.0518	
	2	3	-2.0000	11028	0.3202	5	-1.5156	1047	0.0672	
4	2	1	1.2063e-11	5361	0.2048	1	1.2063e-11	405	0.0591	
	2	1	6.0581e-10	5370	0.1505	1	6.0581e-10	294	0.1092	
5	c=0.5	2	2	1.0071e-11	5850	0.2304	2	2.4581e-15	1080	0.1082
		2	4	1.3798e-10	6396	0.3038	3	1.0276	1311	0.0696
	c=0.2	2	2	2.5576e-10	5871	0.1729	1	0.6165	750	0.0432
		2	2	8.3350e-11	11130	0.3122	10	8.8414	2814	0.1848

*Continued on next page*

NO.	n	NFFM				PFFF2				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)	
5	c=0.05	2	3	5.7877e-11	13356	0.3738	9	0.6184	1581	0.0908
		2	2	4.3862e-8	5478	0.1576	1	0.6184	468	0.0732
6		2	1	3.0000	5496	0.1463	1	3.0000	525	0.0749
7		2	3	-186.7309	5286	0.1182	6	-185.9494	1953	0.1603
8		2	3	0.9980	12075	0.6224	3	0.9980	1287	0.1539
		2	3	0.9980	12147	0.6320	-	-	-	-
9	m=5	4	2	-10.1532	45265	0.9075	3	-5.0552	3190	0.1984
	m=7	4	2	-10.4029	51205	1.1519	2	-5.0877	2020	0.2010
	m=10	4	2	-10.5364	47080	1.1174	2	-5.1285	2034	0.1529
10		2	3	4.4998e-14	31164	0.6559	5	6.4387e-13	1419	0.1248
		5	9	3.6297e-15	91410	0.9884	23	1.0258e-7	40356	1.0217
		10	7	5.0081e-9	399927	1.8787	9	1.3036e-5	70884	1.2359
		30	12	1.2918e-11	5502810	15.6206	24	1.7083e-15	997735	13.4804
		50	16	2.4807e-11	11562873	33.9494	204	1.3464e-15	20862621	298.6366
		100	51	1.4958e-8	183671429	721.4294	-	-	-	-
		200	4	1.0357e-9	205588629	847.5251	-	-	-	-
11		2	5	3.4824e-11	34005	0.8710	7	0	1446	0.1291
		5	11	4.3485e-12	214302	2.2427	16	0	15366	0.4982
		10	14	1.2790e-13	767118	5.3170	31	0	105963	1.9649
		30	23	1.6768e-9	8639917	35.6461	81	5.6843e-14	2404639	38.0193
		50	5	3.5811e-12	17655639	75.4816	151	0	12165642	226.8942
		100	75	4.5475e-13	87546396	450.2820	301	2.2737e-13	101637310	2677.3067
		200	63	2.0600e-10	352329282	2655.6060	-	-	-	-
12		2	79	0	151752	4.9212	79	0	151752	4.9212
		5	148	0	1192452	14.8535	148	0	1192452	14.8535
		10	102	3.0873e-12	5495303	30.7575	124	6.2421e-12	422004	4.9854
		30	132	5.0235e-12	107425695	176.2816	-	-	-	-
		50	117	2.0286e-11	348382122	520.6993	-	-	-	-
		100	130	9.7856e-11	607111303	1558.6716	-	-	-	-
		200	203	1.4511e-9	2019818247	10605.0670	-	-	-	-
13		2	7	2.5455e-5	10482	0.5411	-	-	-	-
		5	16	6.3638e-5	51006	1.4617	-	-	-	-
		10	31	1.2728e-4	219516	3.3319	-	-	-	-
		30	91	3.8183e-4	2481116	21.3445	-	-	-	-
		50	152	6.3638e-4	8590134	77.8044	-	-	-	-
		100	301	1.2728e-3	55840981	709.5823	-	-	-	-
		200	601	2.5130e-3	357280113	7149.2103	-	-	-	-
14		2	1	0.0566	7743	0.7729	1	0.0566	576	0.1136

Continued on next page

NO.	n	NFFM			PFFF2				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
14	5	3	1.4644e-8	44862	0.6354	1	0.0173	1740	0.1217
	10	2	2.5650e-4	133001	0.9756	-	-	-	-
	30	2	2.0058e-7	1074181	4.5460	-	-	-	-
	50	2	2.8443e-7	2922300	11.1526	-	-	-	-
	100	2	4.3167e-7	11470065	39.3521	-	-	-	-
	200	2	2.5979e-6	45435648	166.6213	-	-	-	-

**Table 6.** Comparison of numerical experimental results.

NO.	n	NFFM			PFFF3					
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)	
1	2	2	2.3314e-16	11466	0.3768	3	0	612	0.1133	
	2	1	1.0583e-13	2208	0.0689	1	1.0583e-13	333	0.0245	
2	2	1	-1.0316	5403	0.1521	1	-1.0316	399	0.0531	
	2	2	-1.0316	5250	0.1177	2	-1.0316	486	0.0411	
3	2	3	-2.0000	11187	0.3188	-	-	-	-	
	2	3	-2.0000	11028	0.3202	-	-	-	-	
4	2	1	1.2063e-11	5361	0.2048	1	1.2063e-11	363	0.0496	
	2	1	6.0581e-10	5370	0.1505	1	6.0581e-10	366	0.0265	
5	c=0.5	2	2	1.0071e-11	5850	0.2304	1	0.5175	279	0.0156
		2	4	1.3798e-10	6396	0.3038	9	0.5996	1710	0.0809
	c=0.2	2	2	2.5576e-10	5871	0.1729	1	0.6165	444	0.0656
		2	2	8.3350e-11	11130	0.3122	2	0.3558	711	0.0304
c=0.05	2	3	5.7877e-11	13356	0.3738	3	0.6184	1035	0.0856	
	2	2	4.3862e-8	5478	0.1576	1	0.6184	411	0.0319	
6	2	1	3.0000	5496	0.1463	1	3.0000	456	0.0292	
7	2	3	-186.7309	5286	0.1182	4	-186.7309	996	0.0396	
8	2	3	0.9980	12075	0.6224	2	0.9980	702	0.0389	
	2	3	0.9980	12147	0.6320	2	0.9980	819	0.0582	
9	m=5	4	2	-10.1532	45265	0.9075	2	-10.1529	2070	0.1036
	m=7	4	2	-10.4029	51205	1.1519	2	-10.4027	2095	0.0948
	m=10	4	2	-10.5364	47080	1.1174	2	-10.5364	2095	0.1276
10	2	3	4.4998e-14	31164	0.6559	-	-	-	-	
	5	9	3.6297e-15	91410	0.9884	-	-	-	-	
	10	7	5.0081e-9	399927	1.8787	-	-	-	-	
	30	12	1.2918e-11	5502810	15.6206	-	-	-	-	
	50	16	2.4807e-11	11562873	33.9494	4	5.8993e-13	442425	3.4794	

*Continued on next page*

NO.	n	NFFM			PFFF3				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
10	100	51	1.4958e-8	183671429	721.4294	7	1.1811e-11	2602871	27.8996
	200	4	1.0357e-9	205588629	847.5251	2	2.1345e-11	3424839	44.0043
11	2	5	3.4824e-11	34005	0.8710	4	0	849	0.0924
	5	11	4.3485e-12	214302	2.2427	-	-	-	-
	10	14	1.2790e-13	767118	5.3170	-	-	-	-
	30	23	1.6768e-9	8639917	35.6461	34	5.6843e-14	501456	4.1769
	50	5	3.5811e-12	17655639	75.4816	53	0	1876902	17.3860
	100	75	4.5475e-13	87546396	450.2820	103	4.5475e-13	13499054	168.2512
12	200	63	2.0600e-10	352329282	2655.6060	205	2.2737e-13	105396561	2099.2134
	2	79	0	151752	4.9212	33	0.3247	7281	0.3332
	5	148	0	1192452	14.8535	83	0.7564	83454	1.6411
	10	102	3.0873e-12	5495303	30.7575	124	6.2421e-12	422004	4.9854
	30	132	5.0235e-12	107425695	176.2816	-	-	-	-
	50	117	2.0286e-11	348382122	520.6993	-	-	-	-
13	100	130	9.7856e-11	607111303	1558.6716	-	-	-	-
	200	203	1.4511e-9	2019818247	10605.0670	-	-	-	-
	2	7	2.5455e-5	10482	0.5411	-	-	-	-
	5	16	6.3638e-5	51006	1.4617	-	-	-	-
	10	31	1.2728e-4	219516	3.3319	-	-	-	-
	30	91	3.8183e-4	2481116	21.3445	-	-	-	-
14	50	152	6.3638e-4	8590134	77.8044	-	-	-	-
	100	301	1.2728e-3	55840981	709.5823	-	-	-	-
	200	601	2.5130e-3	357280113	7149.2103	-	-	-	-
	2	1	0.0566	7743	0.7729	1	0.0566	528	0.0614
	5	3	1.4644e-8	44862	0.6354	1	0.0173	1944	0.0755
	10	2	2.5650e-4	133001	0.9756	-	-	-	-
14	30	2	2.0058e-7	1074181	4.5460	-	-	-	-
	50	2	2.8443e-7	2922300	11.1526	-	-	-	-
	100	2	4.3167e-7	11470065	39.3521	-	-	-	-
	200	2	2.5979e-6	45435648	166.6213	-	-	-	-

**Table 7.** Comparison of numerical experiment results.

NO.	n	NFFM			PFFF4				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
1	2	2	2.3314e-16	11466	0.3768	2	4.3211e-14	1038	0.9396
	2	1	1.0583e-13	2208	0.0689	1	1.0583e-13	1092	0.0492

*Continued on next page*

NO.	n	NFFM			PFFF4					
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)	
2	2	1	-1.0316	5403	0.1521	1	-1.0316	531	0.0401	
	2	2	-1.0316	5250	0.1177	2	-1.0316	618	0.0796	
3	2	3	-2.0000	11187	0.3188	3	-2.0000	1929	0.1434	
	2	3	-2.0000	11028	0.3202	3	-1.7578	1674	0.0673	
4	2	1	1.2063e-11	5361	0.2048	1	1.2063e-11	2661	0.0587	
	2	1	6.0581e-10	5370	0.1505	1	6.0581e-10	2670	0.1149	
c=0.5	2	2	1.0071e-11	5850	0.2304	3	1.1035e-7	49275	1.6369	
	2	4	1.3798e-10	6396	0.3038	7	1.1204e-9	50250	1.4413	
5	c=0.2	2	2.5576e-10	5871	0.1729	3	8.7391e-10	51357	1.5487	
	2	2	8.3350e-11	11130	0.3122	9	6.1308e-7	52491	1.4257	
c=0.05	2	3	5.7877e-11	13356	0.3738	5	0.1022	49710	1.5305	
	2	2	4.3862e-8	5478	0.1576	3	5.6362e-11	43698	1.1451	
6	2	1	3.0000	5496	0.1463	1	3.0000	2454	0.1439	
7	2	3	-186.7309	5286	0.1182	4	-186.7309	3411	0.1825	
8	2	3	0.9980	12075	0.6224	3	0.9980	3024	0.1564	
	2	3	0.9980	12147	0.6320	7	0.9980	3903	0.1949	
9	m=5	4	2	-10.1532	45265	0.9075	2	-10.1532	17835	0.8182
	m=7	4	2	-10.4029	51205	1.1519	2	-10.4029	20520	0.7257
	m=10	4	2	-10.5364	47080	1.1174	2	-10.5364	19065	0.8135
10	2	3	4.4998e-14	31164	0.6559	5	1.8586e-12	4929	0.1414	
	5	9	3.6297e-15	91410	0.9884	7	4.7871e-14	29904	0.5543	
	10	7	5.0081e-9	399927	1.8787	6	2.2716e-8	145090	1.1568	
	30	12	1.2918e-11	5502810	15.6206	11	1.4568e-9	1301504	6.3026	
	50	16	2.4807e-11	11562873	33.9494	20	5.3092e-12	5046909	23.6914	
	100	51	1.4958e-8	183671429	721.4294	47	3.7590e-8	22979621	138.2573	
11	200	4	1.0357e-9	205588629	847.5251	12	2.1157e-9	48143520	451.1563	
	2	5	3.4824e-11	34005	0.8710	3	2.6645e-12	12903	0.4066	
	5	11	4.3485e-12	214302	2.2427	6	7.1054e-12	65448	1.1110	
	10	14	1.2790e-13	767118	5.3170	-	-	-	-	
	30	23	1.6768e-9	8639917	35.6461	31	7.7853e-10	1104840	8.6058	
	50	5	3.5811e-12	17655639	75.4816	-	-	-	-	
12	100	75	4.5475e-13	87546396	450.2820	201	7.5033e-12	26778837	273.1368	
	200	63	2.0600e-10	352329282	2655.6060	401	8.4128e-12	195042762	3062.2324	
12	2	79	0	151752	4.9212	15	0.0295	6678	0.2673	
	5	148	0	1192452	14.8535	28	0.0295	50982	1.1381	
	10	102	3.0873e-12	5495303	30.7575	33	0.0295	167277	2.0386	
	30	132	5.0235e-12	107425695	176.2816	53	1.0672e-10	2829680	20.6502	
	50	117	2.0286e-11	348382122	520.6993	80	2.2460e-10	9895530	75.8761	

Continued on next page



NO.	n	NFFM			PFFF4			
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf
12	100	1309.7856e-11	607111303	1558.6716	1202.3546e-10	49981870	477.1591	
	200	203 1.4511e-9	2019818247	10605.0670	1697.3183e-10	261965511	3468.8519	
13	2	7 2.5455e-5	10482	0.5411	-	-	-	-
	5	16 6.3638e-5	51006	1.4617	-	-	-	-
	10	31 1.2728e-4	219516	3.3319	-	-	-	-
	30	91 3.8183e-4	2481116	21.3445	-	-	-	-
	50	152 6.3638e-4	8590134	77.8044	-	-	-	-
	100	301 1.2728e-3	55840981	709.5823	-	-	-	-
	200	601 2.5130e-3	357280113	7149.2103	-	-	-	-
14	2	1 0.0566	7743	0.7729	1	0.0566	19362	0.5234
	5	3 1.4644e-8	44832	0.6354	3	1.4644e-8	100416	1.3234
	10	2 2.5650e-4	133001	0.9756	2	2.5650e-04	331254	2.5687
	30	2 2.0058e-7	1074181	4.5460	2	0.0495	2670247	11.1110
	50	2 2.8443e-7	2922300	11.1526	2	0.0896	7241490	27.4103
	100	2 4.3167e-7	11470065	39.3521	2	0.7187	28412815	99.1462
	200	2 2.5979e-6	45435648	166.6213	2	0.0157	112403421	428.3822

**Table 8.** Comparison of numerical experiment results.

NO.	n	NFFM			PFFF5				
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
1	2	2	2.3314e-16	11466	0.3768	2	5.6913e-14	530	0.0156
	2	1	1.0583e-13	2208	0.0689	1	1.0583e-13	433	0.0192
2	2	1	-1.0316	5403	0.1521	1	-1.0316	427	0.0355
	2	2	-1.0316	5250	0.1177	2	-	-	-
3	2	3	-2.0000	11187	0.3188	3	-	-	-
	2	3	-2.0000	11028	0.3202	3	-2.0000	811	0.0121
4	2	1	1.2063e-11	5361	0.2048	1	1.2063e-11	463	0.0293
	2	1	6.0581e-10	5370	0.1505	1	6.0581e-10	442	0.0175
c=0.5	2	2	1.0071e-11	5850	0.2304	2	1.2736e-14	563	0.0216
	2	4	1.3798e-10	6396	0.3038	4	3.9226e-3	1138	0.0173
5 c=0.2	2	2	2.5576e-10	5871	0.1729	3	-	-	-
	2	2	8.3350e-11	11130	0.3122	5	-	-	-
c=0.05	2	3	5.7877e-11	13356	0.3738	5	-	-	-
	2	2	4.3862e-8	5478	0.1576	3	-	-	-
6	2	1	3.0000	5496	0.1463	1	3.0000	478	0.0204

*Continued on next page*

NO.	n	NFFM				PFFF5			
		Iter	$f(x^*)$	NFf	T(s)	Iter	$f(x^*)$	NFf	T(s)
7	2	3	-186.7309	5286	0.1182	4	-186.7309	1193	0.1175
8	2	3	0.9980	12075	0.6224	2	0.9980	898	0.1181
	2	3	0.9980	12147	0.6320	62	-	-	-
m=5	4	2	-10.1532	45265	0.9075	1	-5.0552	1880	0.1560
9 m=7	4	2	-10.4029	51205	1.1519	1	-5.0877	1895	0.1181
m=10	4	2	-10.5364	47080	1.1174	1	-5.1285	1895	0.1058

Tables 4–8 show the numerical computation results of the NFFM compared with the other five filled function algorithms, respectively. The comparison of numerical results in Table 4 shows that the NFFM's accuracy of the global optimum is better than that of PFFF1 in 13 out of the 14 algorithms tested, and the number of iterations in 12 of these examples is also better or not worse than that of PFFF1. In this paper, the NFFM's performance advantage is more obvious in terms of the performance of the algorithms for solving the high-dimensional test problems. The NFFM successfully optimizes them all, and the accuracy and number of iterations of the obtained global optimums are better than PFFF1. However, for the successfully solved examples, the NFFM outperforms PFFF1 only for a few of the successfully solved cases, in terms of the total amount of function computation and time consumed for the objective function and the filled function.

Comparison of the numerical experimental results in Table 5 shows that the accuracy of the global optimum of PFFF2 is better than that of the NFFM only in solving the 30-dimensional and 50-dimensional problems of Problem 10 and the 2–200-dimensional problems of Problem 11, but the number of iterations used is higher than that of the NFFM. PFFF1 fails to find an optimal solution for the 200-dimensional problem of Problem 11. In addition, for Problem 13 and Problem 14, the NFFM in this paper is able to successfully find their global optimal solutions, while PFFF2 fails to find an optimal solution for them. Therefore, in summary, the NFFM performs better than PFFF2.

In order to prove the effectiveness of the algorithm more strongly, the NFFM was experimentally compared with the parameter-free filled function algorithms PFFF3 and PFFF4, which have been proposed in the last three years. The comparison of the specific experimental computational results is shown in Tables 6 and 7. It can be seen from Table 6 that PFFF3 fails in finding the optimum for some examples. Neglecting the problems with the same number of iterations and global optimum and the problems where PFFF3 fails to find the optimal value, and comparing the number of iterations and global optimum of the NFFM with those of PFFF3, the NFFM in this paper has a slightly higher number of iterations than that of PFFF3 only in the solving of Problem 8, and the NFFM's accuracy of the number of iterations and global optimum is superior to that of PFFF3 for the rest of the problems. Similarly, ignoring the problem with the same number of iterations and global optimum and the problem where PFFF4 fails to find the optimum, the NFFM outperforms PFFF4 in terms of the accuracy of the global optimum on the solution of the rest of the problems, and the number of iterations at the termination of the algorithm is slightly higher than that of PFFF4 in only one of the algorithms. In addition, for Problem 13, which is a typical spoofing problem, it is very easy to make the algorithm fall into a local optimal solution. In this paper, the NFFM can successfully find the global optimal

solution of this problem, but the algorithms PFFF3 and PFFF4 cannot achieve it. Therefore, evaluated from the two perspectives of iteration number and global optimum, the NFFM is efficient and feasible in solving the above 14 unconstrained optimization problems. Furthermore, to highlight the superior performance of the improved filled function in this paper, we have conducted a comparative analysis of the numerical results of NFFM and algorithm PFFF5 on Problems 1 to 9. These problems utilize the same test functions as those used to validate PFFF5 in reference [22]. The detailed comparison results are presented in Table 8, showing that NFFM successfully finds the global optimal solutions for all nine problems with higher accuracy.

Tables 4–8 present the numerical results of the filled function algorithm in solving 14 test functions from a given initial point. To validate the independence of our proposed algorithm from the choice of the initial point, we have conducted additional experiments on Problems 11 and 12. Specifically, we randomly generated 10 points within the feasible region of the target problems and independently applied these points to these two problems. We employed three distinct metrics to compare the capabilities of the NFFM against other filled function algorithms. These metrics are: the average number of iterations( $I_i$ ); the average value of the global optimum achieved by the algorithm( $f_m$ ); and the success rate in successfully finding the global optimum over 10 independent runs of the algorithm are denoted as  $SR$ . The specific numerical experimental results are presented in Tables 9 and 10. In a comprehensive comparison, NFFM has exhibited comprehensive advantages in the numerical computation phase, with all its indicators surpassing those of the other four comparative algorithms. It is worth noting that the average global optimal value achieved by the NFFM possesses remarkably high precision, and its success rate nearly reaches 100%, thus proving the superior performance and reliability of NFFM in addressing related problems.

However, considering the total running time and the total number of function evaluations, the numerical results in Tables 4 to 7 indicate that, as the dimension of the test functions increases during the optimization of problems 10 to 14, the total running time and the total number of function evaluations also increase accordingly. This is primarily due to the fact that traditional filled function algorithms often employ a fixed or limited number of perturbation step sizes when generating initial points for minimizing filled functions. This approach can lead to the algorithms getting stuck in local minima and failing to find the global optimal solution when dealing with high-dimensional and complex unconstrained optimization problems. To address this issue, the proposed algorithm NFFM in this paper perturbs the current local minimum and uses it as an initial point during the minimization of the filled function. It continuously searches for a better local minimum. If a better solution is not found, the algorithm increases the perturbation of the current local minimum of the objective function until it reaches the boundary of the feasible region, at which point the algorithm terminates. This approach may require more time and computational resources when dealing with some high-dimensional problems. However, this design ensures a more comprehensive exploration of the solution space, increasing the likelihood of finding the global optimal solution. Although the proposed NFFM in this paper may require more time when solving high-dimensional unconstrained optimization problems, the time complexity analysis reveals that the time complexity for other algorithms to escape a local minimum is  $O(n * T(local))$ . Similarly, the time complexity of NFFM to escape a local minimum is also  $O(n * T(local))$ . Therefore, in terms of time complexity, the proposed algorithm is comparable to other contrasting algorithms.

In summary, the proposed NFFM in this paper significantly reduces the number of iterations,

improves the accuracy of the global optimal solution, and effectively tackles high-dimensional multimodal unconstrained optimization problems, exhibiting strong ability to escape from local minima. Through extensive numerical comparative experiments and validation on high-dimensional problems, the NFFM has been proven feasible and efficient, outperforming algorithms PFFF1–PFFF5 in terms of computational capability.

**Table 9.** Comparison of numerical experimental results at random initial points.

NO.	$n$	NFFM			PFFF1			PFFF2		
		$I_t$	$f_m$	SR	$I_t$	$f_m$	SR	$I_t$	$f_m$	SR
11	2	3.3	1.4214e-12	10/10	3.3	9.8055e-14	10/10	2.8	5.0775e-1	4/10
	5	8.4	4.8127e-11	10/10	7.7	8.1784e-13	10/10	-	-	0/10
	10	11	3.2427e-10	10/10	11.8	8.3063e-12	10/10	-	-	0/10
	30	21.3	5.8421e-10	9/10	38	4.8772e-12	10/10	-	-	0/10
	50	30.8	1.1750e-9	10/10	60.1	3.8301e-11	10/10	-	-	0/10
	100	41.6	2.6335e-9	10/10	117.7	4.0907e-10	10/10	-	-	0/10
12	2	47.9	1.4622e-13	9/10	20.4	4.1687e-13	9/10	-	-	0/10
	5	72.1	2.0000e-12	10/10	45.6	1.7502e-12	10/10	-	-	0/10
	10	70.8	2.5637e-12	9/10	-	-	0/10	-	-	0/10
	30	119.3	2.2261e-11	10/10	-	-	0/10	-	-	0/10
	50	131.3	5.5394e-11	10/10	-	-	0/10	-	-	0/10
	100	121.3	6.7660e-11	10/10	-	-	0/10	-	-	0/10

**Table 10.** Comparison of numerical experimental results at random initial points.

NO.	$n$	NFFM			PFFF3			PFFF4		
		$I_t$	$f_m$	SR	$I_t$	$f_m$	SR	$I_t$	$f_m$	SR
11	2	3.3	1.4214e-12	10/10	3.5	1.1842e-15	6/10	3.5	2.7807e-12	10/10
	5	8.4	4.8127e-11	10/10	7.2	8.5266e-15	5/10	7.9	5.9607e-11	10/10
	10	11	3.2427e-10	10/10	9.2	9.4740e-15	6/10	13.6	4.9224e-11	10/10
	30	21.3	5.8421e-10	9/10	35	1.1369e-13	3/10	26.8	1.2427e-10	10/10
	50	30.8	1.1750e-9	10/10	47.5	2.8422e-14	2/10	34	3.9248e-9	10/10
	100	41.6	2.6335e-9	10/10	91.3	1.2790e-13	7/10	33.7	1.9168e-10	10/10
12	2	47.9	1.4622e-13	9/10	15.2	2.1978e-1	9/10	15.8	2.1651e-2	10/10
	5	72.1	2.0000e-12	10/10	33.8	6.0327e-1	10/10	27.1	2.3622e-2	10/10
	10	70.8	2.5637e-12	9/10	44.3	3.2493e-2	9/10	35.1	2.0699e-2	10/10
	30	119.3	2.2261e-11	10/10	-	-	0/10	62.1	2.0699e-2	10/10
	50	131.3	5.5394e-11	10/10	-	-	0/10	91.5	3.5085e-10	10/10
	100	121.3	6.7660e-11	10/10	-	-	0/10	88.7	6.8476e-10	10/10

## 5. Conclusions

This paper introduces a novel, parameter-free, and continuously differentiable filled function that eliminates the traditional exponential and logarithmic terms, while rigorously proving its filling property. By leveraging this innovative function, we have developed a new filled function algorithm that significantly enhances its performance in addressing multimodal global optimization problems by optimizing the perturbation step size. Through extensive testing and comparative analysis on 14 benchmark functions, we have convincingly demonstrated the efficacy and precision of our algorithm in addressing high-dimensional multimodal global optimization problems. Our proposed algorithm surpasses existing methods, exhibiting superior solution accuracy, stability, and reliability. This accomplishment introduces a robust new instrument for tackling global optimization challenges, thereby emphasizing the significant potential and widespread applicability of the filled function approach in contemporary optimization research.

### Author contributions

J. Li, Y. Gao, T. Chen and X. Ma: writing, review and editing. All authors have read and approved the final version of the manuscript for publication.

### Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgments

This research is supported by the Natural Science Foundation of NingXia Hui Autonomous Region (2022AAC02043); the Construction Project of first-class subjects in Ningxia higher Education (NXYLXK2017B09); the Major Research Funding Project of North Minzu University (ZDZX201901); the Graduate Innovation Project of North Minzu University (2024YCX24037); the basic discipline research projects supported by Nanjing Securities (NJZQJCXK202201).

### Conflict of interest

The authors declare no conflicts of interest.

### References

1. R. Soto, B. Crawford, R. Olivares, J. Barraza, I. Figueroa, F. Johnson, et al., Solving the non-unicost set covering problem by using cuckoo search and black hole optimization, *Nat. Comput.*, **16** (2017), 213–229. <http://doi.org/10.1007/s11047-016-9609-7>
2. X. Liu, Y. Wang, M. Zhou, Dimensional learning strategy-based grey wolf optimizer for solving the global optimization problem, *Comput. Intel. Neurosci.*, **2022** (2022). <http://doi.org/10.1155/2022/3603607>

3. J. S. Pan, A. Q. Tian, V. Snášel, L. Kong, S. C. Chu, Maximum power point tracking and parameter estimation for multiple-photovoltaic arrays based on enhanced pigeon-inspired optimization with taguchi method, *Energy*, **251** (2022), 123863. <https://doi.org/10.1016/j.energy.2022.123863>
4. A. Q. Tian, X. Y. Wang, H. Xu, J. S. Pan, V. Snášel, H. X. Lv, Multi-objective optimization model for railway heavy-haul traffic: addressing carbon emissions reduction and transport efficiency improvement, *Energy*, **294** (2024), 130927. <https://doi.org/10.1016/j.energy.2024.130927>
5. İ. Çetinbaşı, B. Tamyürek, M. Demirtaş, The hybrid harris hawks optimizer-arithmetic optimization algorithm: a new hybrid algorithm for sizing optimization and design of microgrids, *IEEE Access*, **10** (2022), 19254–19283. <http://doi.org/10.1109/ACCESS.2022.3151119>
6. A. Q. Tian, F. F. Liu, H. X. Lv, Snow geese algorithm: a novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems, *Appl. Math. Model.*, **126** (2024), 327–347. <https://doi.org/10.1016/j.apm.2023.10.045>
7. G. Renpu, A filled function method for finding a global minimizer of a function of several variables, *Math. Program.*, **46** (1990), 191–204. <http://doi.org/10.1007/bf01585737>
8. R. P. Ge, Y. F. Qin, A class of filled functions for finding global minimizers of a function of several variables, *J. Optim. Theory Appl.*, **54** (1987), 241–252. <http://doi.org/10.1007/bf00939433>
9. H. S. Ryoo, N. V. Sahinidis, A branch-and-reduce approach to global optimization, *J. Global Optim.*, **8** (1996), 107–138. <http://doi.org/10.1007/bf00138689>
10. A. V. Levy, A. Montalvo, The tunneling algorithm for the global minimization of functions, *SIAM J. Sci. Stat. Comput.*, **6** (1985), 15–29. <http://doi.org/10.1137/0906002>
11. L. S. Zhang, C. K. Ng, D. Li, W. W. Tian, A new filled function method for global optimization, *J. Global Optim.*, **28** (2004), 17–43. <http://doi.org/10.1023/b:jogo.0000006653.60256.f6>
12. Y. Yang, Y. Shang, A new filled function method for unconstrained global optimization. *Appl. Math. Comput.*, **173** (2006), 501–512. <http://doi.org/10.1016/j.amc.2005.04.046>
13. C. Wang, Y. Yang, J. Li, A new filled function method for unconstrained global optimization, *J. Comput. Appl. Math.*, **225** (2009), 68–79. <http://doi.org/10.1016/j.cam.2008.07.001>
14. F. Wei, Y. Wang, H. Lin, A new filled function method with two parameters for global optimization, *J. Optim. Theory Appl.*, **163** (2014), 510–527. <http://doi.org/10.1007/s10957-013-0515-1>
15. Y. Gao, Y. Yang, M. You, A new filled function method for global optimization, *Appl. Math. Comput.*, **268** (2015), 685–695. <http://doi.org/10.1016/j.amc.2015.06.090>
16. T. M. El-Gindy, M. S. Salim, A. I. Ahmed, A new filled function method applied to unconstrained global optimization, *Appl. Math. Comput.*, **273** (2016), 1246–1256. <http://doi.org/10.1016/j.amc.2015.08.091>
17. X. Liu, Finding global minima with a computable filled function, *J. Global Optim.*, **19** (2001), 151–161. <http://doi.org/10.1023/a:1008330632677>
18. X. Liu, A class of generalized filled functions with improved computability, *J. Comput. Appl. Math.*, **137** (2001), 61–69. [http://doi.org/10.1016/S0377-0427\(00\)00697-X](http://doi.org/10.1016/S0377-0427(00)00697-X)
19. H. Lin, Y. Gao, Y. Wang, A continuously differentiable filled function method for global optimization, *Numer. Algor.*, **66** (2014), 511–523. <http://doi.org/10.1007/s11075-013-9746-3>

20. X. Wu, Y. Wang, N. Fan, A new filled function method based on adaptive search direction and valley widening for global optimization, *Appl. Intell.*, **51** (2021), 6234–6254. <http://doi.org/10.1007/s10489-020-02179-0>
21. A. I. Ahmed, A new parameter free filled function for solving unconstrained global optimization problems, *Int. J. Comput. Math.*, **98** (2021), 106–119. <http://doi.org/10.1080/00207160.2020.1731484>
22. S. Ma, Y. Yang, H. Liu, A parameter free filled function for unconstrained global optimization, *Appl. Math. Comput.*, **215** (2010), 3610–3619. <https://doi.org/10.1016/j.amc.2009.10.057>
23. H. Liu, Y. Wang, S. Guan, X. Liu, A new filled function method for unconstrained global optimization, *Int. J. Comput. Math.*, **94** (2017), 2283–2296. <http://doi.org/10.1080/00207160.2017.1283021>
24. R. Pandiya, W. Widodo, Salmah, I. Endrayanto, Non parameter-filled function for global optimization, *Appl. Math. Comput.*, **391** (2021), 125642. <https://doi.org/10.1016/j.amc.2020.125642>
25. R. Pandiya, S. Salmah, W. Widodo, I. Endrayanto, Finding global minima with an inflection point-based filled function algorithm, *Numer. Algor.*, **92** (2023), 1403–1424. <https://doi.org/10.1007/s11075-022-01346-3>
26. A. Hedar, *Test functions for unconstrained global optimization*, 2013.
27. W. X. Zhu, Dynamic globally concavized filled function method for continuous global optimization, *J. Optim. Theory Appl.*, **139** (2008), 635–648. <https://doi.org/10.1007/s10957-008-9405-3>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)