Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 782536, 16 pages
http://dx.doi.org/10.1155/2015/782536

Research Article

MEEF: A Minimum-Elimination-Escape Function Method for Multimodal Optimization Problems

Lei Fan,[1] Yuping Wang,[2] Xiyang Liu,[1] and Liping Jia[3]

[1] *Institute of Software Engineering, School of Software, Xidian University, Xi'an 710071, China*
[2] *School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[3] *College of Mathematics and Information Science, Leshan Normal University, Leshan 614000, China*

Correspondence should be addressed to Lei Fan; lfan@mail.xidian.edu.cn

Received 21 January 2015; Revised 2 March 2015; Accepted 5 March 2015

Academic Editor: Laura Gardini

Copyright © 2015 Lei Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Auxiliary function methods provide us effective and practical ideas to solve multimodal optimization problems. However, improper parameter settings often cause troublesome effects which might lead to the failure of finding global optimal solutions. In this paper, a minimum-elimination-escape function method is proposed for multimodal optimization problems, aiming at avoiding the troublesome "Mexican hat" effect and reducing the influence of local optimal solutions. In the proposed method, the minimum-elimination function is constructed to decrease the number of local optimum first. Then, a minimum-escape function is proposed based on the minimum-elimination function, in which the current minimal solution will be converted to the unique global maximal solution of the minimum-escape function. The minimum-escape function is insensitive to its unique but easy to adopt parameter. At last, an minimum-elimination-escape function method is designed based on these two functions. Experiments on 19 widely used benchmarks are made, in which influences of the parameter and different initial points are analyzed. Comparisons with 11 existing methods indicate that the performance of the proposed algorithm is positive and effective.

1. Introduction

Global optimization plays a significant role in many fields, for example, science, economics, and engineering. A global optimization problem can be formulated as follows:

$$\min_{x \in D} f(x), \tag{1}$$

where $x \in R^n$, $D = \{x \mid a_i \le x_i \le b_i, i = 1 \sim n\}$, and $f(x)$ is considered to be multimodal and continuously differentiable in this paper.

Early research on single modal global optimization problems had gained many results. However, these achievements can not solve multimodal global optimization problems (GOPs) effectively. Multimodal optimization problems are difficult to solve for the existing of many local optimal solutions, which often make the optimization algorithms trap into local optimum. More and more attention has been paid for multimodal optimization. As the main tasks of a solution algorithm, finding global optimal solutions of multimodal problems with small computational cost should avoid trapping into local optimal solutions. Deterministic algorithms are developed to deal with these difficulties and the literature review on this work can be referred to [1]. Auxiliary function method, as a kind of deterministic method, provides us an effective and practical idea to jump out from local optimal solutions, for example, filled function method (FFM) [2–7], tunneling method [8], basin-hopping method [9], sequential convexification method (SCM) [10], and cut-peak function method [11], and so forth. Auxiliary function is a transformed objective function that constructs a path from one of the local minimizers of the original objective function to another lower local minimizer [4]. Filled function method [2] is such an approach for multimodal problems to find the global minimal solutions, in which the filled function is constructed to help jump from one local optimal solution to another better one. However, this kind method is sensitive to its parameters. Improper settings often brings about "Mexican hat" effect [12], which might make the algorithm fail in finding global optimal solutions.

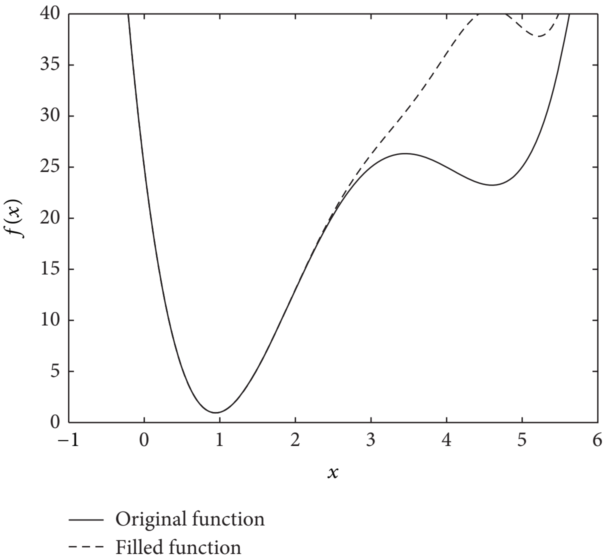


Figure 1: "Mexican hat" effect in filled functions.

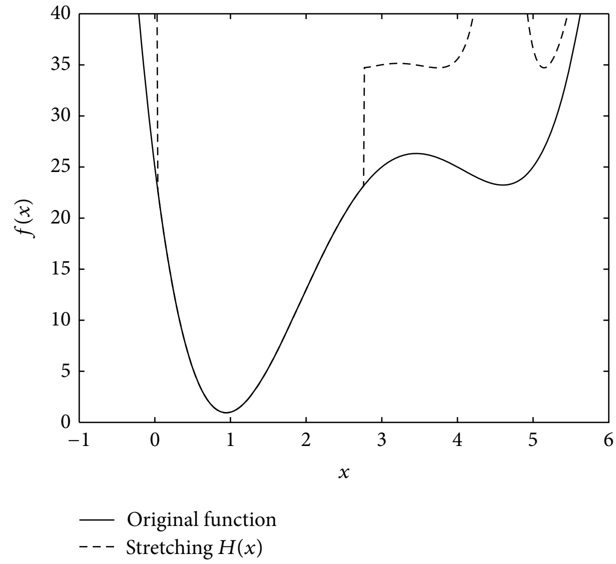


Figure 2: "Mexican hat" effect in stretching functions.

In this paper, a minimum-elimination function is constructed to eliminate the solutions worse than the best solution found so far. Using this function, the local optimal solutions can be decreased. Thus, the number of the local optimal solutions can be reduced significantly. However, flattened by the minimum-elimination function, much information of the original optimization problems might be eliminated. Hence, the optimization algorithms still have difficulties to solve the problems. Hence, we proposed a minimum-escape function based on the minimum-elimination function, which can help the algorithm find the better solutions. Converted by minimum-escape function, there will be only one global maximum. Therefore, this function can provide search algorithms directions away from the best solutions found so far and it can help the search algorithms avoid revisiting or trapping into the local minimal solutions that had been found. In this way, the troublesome "Mexican hat" effect can be avoided effectively.

The remainder of this paper is organized as follows: Section 2 introduces the motivation of this paper. Section 3 is dedicated to explaining the proposed minimum-elimination-escape function in detail. Section 4 gives the minimum-elimination-escape function method for multimodal problems. Experiments on the performance of the proposed algorithm are shown in Section 5. Finally, conclusions on this paper are drawn in Section 6.

2. Motivation

2.1. "Mexican Hat" Effect. When solving multimodal optimization problems, auxiliary function methods can jump out the local optimal solutions with the help of the tailor-made auxiliary functions. For many auxiliary function methods, for example, filled function method [2] and stretching function method [13], improper parameter settings often cause an unwilling phenomenon called "Mexican hat" effect [12]. As shown in Figures 1 and 2, this effect often introduces some troublesome local minimum for the auxiliary functions, which might make the search algorithm trap into local optimal solutions and leads to failure of finding global optimal solutions. This troublesome effect often introduces great additional complexity. It can be seen vividly that the local minimum of the auxiliary functions in Figures 1 and 2 cannot help the optimization methods jump out from the local minimum of the original functions. In this case, the endless loops will be brought about for the optimization methods, which makes the optimization process failure.

"Mexican hat" effect had been discussed in [12] in detail. Many efforts have been directed towards avoiding this unwilling phenomenon [11, 14]. The cut-peak function method [11] is such an auxiliary function method that can avoid "Mexican hat" effect. However, the cut-peak function method has its disadvantages on losing global optima during searching process.

2.2. Lost Global Optima. Although the cut-peak function method has advantage on keeping the best solutions found so far, it will lose the global optimal solutions especially at the end of the searching process. In [11], the cut-peak function is suggested as follows:

$$w\left(r, x^*, x\right) = f\left(x^*\right) - f_0\left(r, \left\|x - x^*\right\|\right), \tag{2}$$

where x^* is the best solution found so far, r is a positive parameter, and $f_0(r, \cdot)$ was given two concrete examples as follows:

$$f_0\left(r, t\right) = \frac{rt^2}{1 + t^2} \quad \text{or} \quad f_0\left(r, t\right) = r\left(1 - e^{-t^2}\right). \tag{3}$$

Then, a choice function is constructed to find better solutions as follows:

$$F\left(r, x^*, x\right) = \min\left\{f\left(x\right), w\left(r, x^*, x\right)\right\}. \tag{4}$$

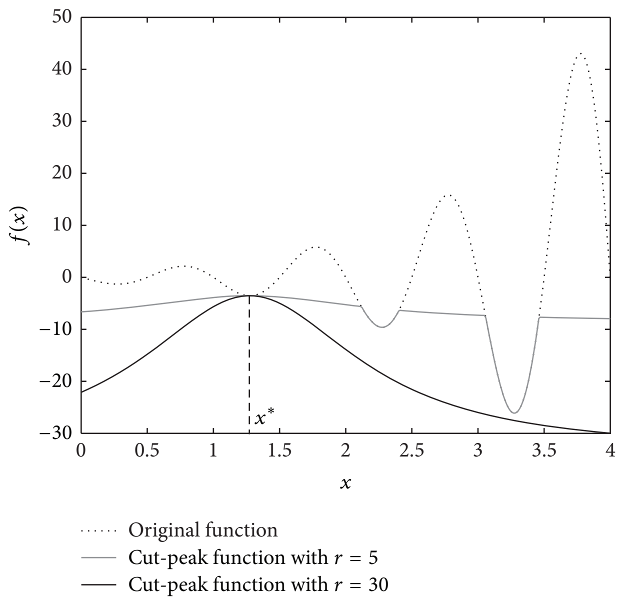


Figure 3: Cut-peak function. The dotted line indicates the original function, the solid lines in gray and black indicate the cut-peak function with $r = 5$ and $r = 30$, respectively.

When the parameter r is properly adjusted, the choice function (4) can keep the original function values of some better solutions unchanged and can find the better solutions easily. However, there are two cases that may lead to the failure of finding global optimal solutions. One case is that if the distance between the best solutions found so far and other local optimal solutions is too large, it will be of great difficulty to adopt the parameter r. The other case is that if the parameter r was larger than a certain value, all the solutions that are better than x^* will be ignored, which will lead to failure in finding global optimal solutions. In particular, if the function values of the local optimal solution and the global optimal solution have little difference, the global optimal solution will be lost with very high probability. It is very difficult to adopt r. So as to illustrate the influence of r obviously, we use the same example as [11], in which the objective function is $f(x) = -e^x \sin(2\pi x)$, where $x \in [0, 4]$. It is supposed that the best solution found so far is $x^* = 1.275$. The cut-peak function at x^* is constructed as $w(r, x^*, x) = f(x^*) - (r\|x - x^*\|^2/(1 + \|x - x^*\|^2))$ and the parameter r is set to be 5 and 30, respectively. From Figure 3, one can see vividly that the cut-peak function can keep the basins better than x^* unchanged when $r = 5$. However, when $r = 30$, there is no better basin that can be kept for the cut-peak function. It can be deduced easily that the smaller the parameter r is, the more better basins will be kept. Huang et al. proposed a revised cut-peak function in [15].

Although the cut-peak function can successfully avoid the “Mexican hat” effect, its performance is influenced significantly by its parameter r. What is more, it needs much computational cost to adopt the parameter. Therefore, we will design a new parameter-insensitive auxiliary function which can avoid the unwilling “Mexican hat” effect in this paper.

3. Minimum-Elimination-Escape Function

The proposed minimum-elimination-escape function in this section consists of minimum-elimination function and minimum-escape function. The minimum-elimination function is constructed to eliminate the solutions worse than the best solutions found so far, aiming to reduce the influence of local optimal solutions. Then, a minimum-escape function is proposed to transform the minimum found so far to the unique global maximum. Also, the minimum-escape function can lead optimization algorithms to explore the region far away from the best solutions found so far. The properties of the proposed minimum-elimination-escape function are also analyzed in this section.

3.1. Minimum-Elimination Function.

Local optimal solutions have great impact on the performance of an optimization algorithm, especially when there were a large number of local optimal solutions which often leads to the failure of finding global optimal solutions. To some degree, decreasing the number of local optimal solutions is a simple and effective way of reducing the complexity of the problem. Therefore, the minimum-elimination function is constructed here to eliminate the solutions worse than the best solutions found so far. In this way, the number of local optimal solutions can be reduced, which will be much helpful to the optimization algorithms.

The minimum-elimination function is constructed as follows:

$$S(x, x^*) = f(x^*) + \frac{1}{2}\left\{1 - (-1)^{\operatorname{sign}(f(x) - f(x^*))}\right\}\left[f(x^*) - f(x)\right], \tag{5}$$

where $f(x)$ is the original objective function, x^* is the best solution found so far and changing with generation, and $\operatorname{sign}(y)$ is defined as follows:

$$\operatorname{sign}(y) = \begin{cases} 1, & y > 0 \\ -1, & y = 0 \\ 0, & y < 0. \end{cases} \tag{6}$$

It is obvious that, for $\forall x \in D$, the minimum-elimination function has the following two properties:

(i) $S(x, x^*) = f(x^*)$ if $f(x) > f(x^*)$,

(ii) $S(x, x^*) = f(x)$ if $f(x) < f(x^*)$.

In other words, only when the better solution was found, the value of $S(x, x^*)$ changed. These two properties can be intuitively illustrated by smoothing a function of one variable as an example in Figure 4, where the dashed line in Figure 4 represents the original function and the black solid line represents the minimum-elimination smoothing function. Here, the example function is taken as $f(x) = x + 10\sin(5x) - 7\sin(4x)$, where $x \in [0, 10]$. The minimum-elimination function is generated at the solution $x^* = 4.816$.

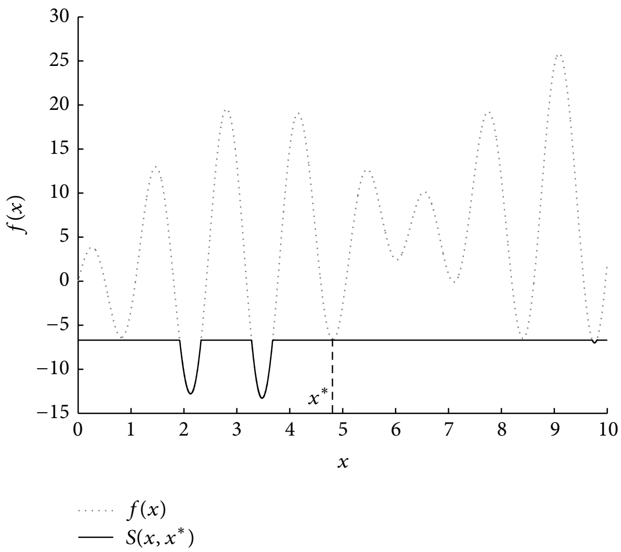


FIGURE 4: 2D example: minimum-elimination function at the current best solution x^*. The black solid line indicates the minimum-elimination function $S(x, x^*)$; the dotted line indicates the original function $f(x)$.

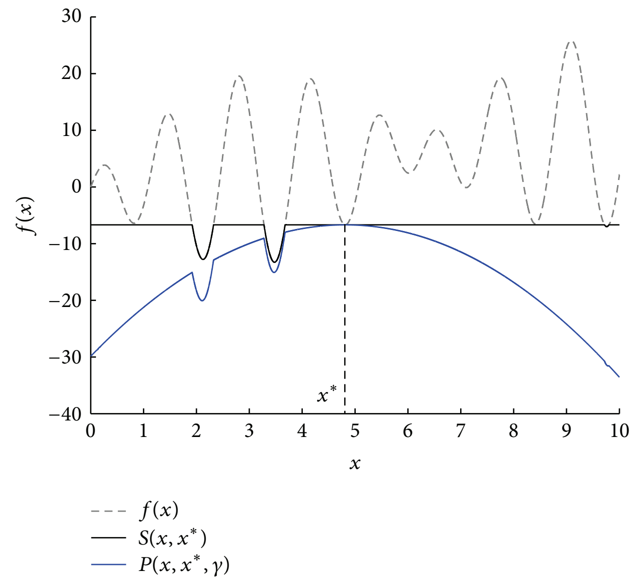


FIGURE 5: 2D example: minimum-escape function for the flattened problem (FGP) at the current best solution x^*.

Flattened by the minimum-elimination function, GP can be converted to the following problem:

$$\text{(FGP)} \quad \min_{x \in D} S\left(x, x^*\right), \tag{7}$$

where x^* is the best solution found so far. The flattened GP is denoted as FGP for convenience. It is obvious that GP and FGP share the same global optima. And FGP has fewer local minima than GP. However, after being flattened by the minimum-elimination smoothing function, there will exist much plane in FGP, so that much mathematical information will be lost.

3.2. Minimum-Escape Function. As mentioned above, FGP loses much information and contains many flatlands which are very difficult to be managed for an algorithm. Improper handling often brings about additional complexity for solving FGP. One important issue is to design suitable strategies of searching the field far away from the best solutions found so far, aiming to find better solutions or basins. Therefore, a minimum-escape function is proposed to deal with the flatlands, which is designed at the best solution x^* found so far as follows:

$$P\left(x, x^*, \gamma\right) = S\left(x, x^*\right) - \gamma \left\|x - x^*\right\|^2, \tag{8}$$

where the unique parameter $\gamma > 0$.

Figures 5–8 show the properties of the minimum-escape function vividly. Among these figures, one can have intuitive impression of the proposed functions. In Figures 4 and 5, a 2-dimensional example was employed. Figures 6 to 8 illustrate the performance of the minimum-elimination smoothing function on the multimodal function and the performance of the proposed minimum-escape function on the smoothed function. It can be seen vividly from these 3D figures that flattened by the minimum-elimination function, much plane and holes are left. By the constructed function (8), the flattened function will be transformed to an arch-like function with x^* as its unique global maximal point. From Figures 5 and 8, we can see obviously that the holes still exist in the slope of the minimum-escape functions. These holes may be the same holes of the flattened functions, which are the basins (A basin [16] of $f(x)$ at an isolated minimizer x_1^* is a connected to domain B_1 which contains x_1^* and in which starting from any point the steepest descent trajectory of $f(x)$ converges to x_1^* but outside which the steepest descent trajectory of $f(x)$ does not converge to x_1^*) of $S(x, x^*)$, which are also the basins of $f(x)$ where the better local optimal solutions locate. Therefore, the aim of the algorithm is to find these holes. We can imagine that a water-drop on this vault will slide down along any direction due to the influence of gravity. Its track might go through some of these holes. Motivated by this phenomenon, we considered that a method can be employed as gravity to guide the search process to find the holes, and another method can be used to find the local optimum. In the following subsection, the properties of the proposed function will be analyzed in detail.

3.3. Properties of Minimum-Escape Function. From Figures 4 to 8, it can be seen that the minimum-escape function has the following properties:

(1) x^* is the unique global maximum of $P(x, x^*, \gamma)$, and $P(x^*, x^*, \gamma) = f(x^*)$;

(2) $P(x, x^*, \gamma)$ has no stationary points in the region $S_1 = \{x \mid f(x) \geq f(x^*),\ x \in D/\{x^*\}\}$;

(3) if x^* is not a global minimizer of $f(x)$, then $P(x, x^*, \gamma)$ can distinguish the better solution than x^*.

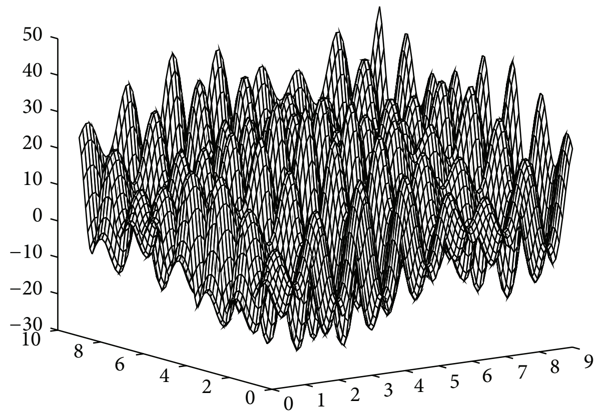


Figure 6: 3D example: original function.

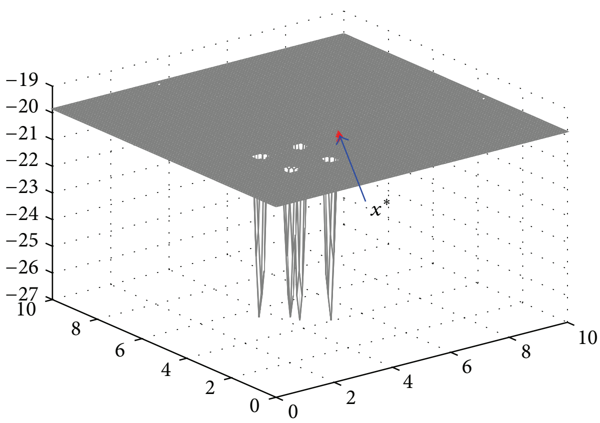


Figure 7: 3D example: flattened by the minimum-elimination function at the current best solution x^*.

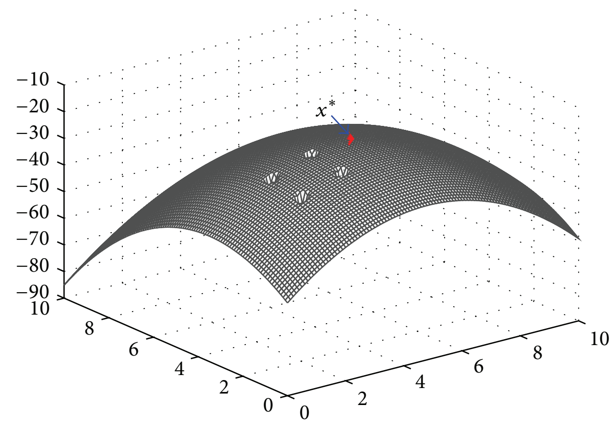


Figure 8: 3D example: minimum-escape function for the flattened problem (FGP) at the current best solution x^*.

To illustrate the properties of the proposed minimum-escape function, we choose x^* as the reference point and generate the following function as a discrimination function:

$$J(x, x^*, \gamma) = \begin{cases} \dfrac{P(x, x^*, \gamma) - P(x^*, x^*, \gamma)}{\|x - x^*\|^2} & x \neq x^* \\ \gamma & x = x^*. \end{cases} \tag{9}$$

To some degree, $J(x, x^*, \gamma)$ can be regarded as the slop of $P(x, x^*, \gamma)$ at x according to the reference point x^* with $\|x - x^*\|^2$. $J(x, x^*, \gamma)$ can help us understand the properties of $P(x, x^*, \gamma)$ and also can be used to estimate whether the obtained solutions are better than x^*.

Proposition 1. *If $f(x)$ is continuous, then both $S(x, x^*)$ and $P(x, x^*, \gamma)$ are continuous.*

Proposition 2. *Suppose that x^* is the best solution found so far. Then, x^* is the unique global maximizer of $P(x, x^*, \gamma)$.*

Proof. For $\forall x \in D$,

$$\begin{aligned} P(x, x^*, \gamma) &= S(x, x^*) - \gamma \|x - x^*\|^2 \\ &\leq f(x^*) - \gamma \|x - x^*\|^2 \\ &\leq f(x^*). \end{aligned} \tag{10}$$

Assume that there exists another point $y \neq x^*$, such that, for any $x \in D$, $P(x, x^*, \gamma) \leq P(y, x^*, \gamma)$. While

$$\begin{aligned} P(y, x^*, \gamma) &= S(y, x^*) - \gamma \|y - x^*\|^2 \\ &\leq f(x^*) - \gamma \|y - x^*\|^2 \\ &< f(x^*) \\ &= P(x^*, x^*, \gamma), \end{aligned} \tag{11}$$

thus $\exists x^* \in D$, such that $P(y, x^*, \gamma) < P(x, x^*, \gamma)$, which conflicts with the assumption.

Therefore, x^* is the unique global maximizer of $P(x, x^*, \gamma)$. □

Proposition 3. *For $\forall x \in D$, $J(x, x^*, \gamma) \geq \gamma$.*

Proof. For $\forall x \in D$, it can be concluded from formula (8) that $S(x, x^*) \leq S(x^*, x^*) = f(x^*)$. Therefore, for $x = x^*$, it is obvious that $J(x^*, x^*, \gamma) = \gamma$. For $x \neq x^*$,

$$\begin{aligned} J(x, x^*, \gamma) &= \frac{P(x^*, x^*, \gamma) - P(x, x^*, \gamma)}{\|x^* - x\|^2} \\ &= \frac{S(x^*, x^*) - S(x, x^*)}{\|x^* - x\|^2} + \gamma \geq \gamma. \end{aligned} \tag{12}$$

□

Proposition 4. *For $\forall x \in D$,*

$$\begin{aligned} f(x) < f(x^*) &\Longleftrightarrow J(x, x^*, \gamma) > \gamma \\ f(x) \geq f(x^*) &\Longleftrightarrow J(x, x^*, \gamma) = \gamma. \end{aligned} \tag{13}$$

Proof. For $\forall x \in D$, if $x = x^*$, it is obvious that $f(x) = f(x^*) \Leftrightarrow J(x, x^*, \gamma) = \gamma$ holds; if $x \neq x^*$, then

$$\begin{aligned} f(x) < f(x^*) &\Longleftrightarrow S(x, x^*) < S(x^*, x^*) \\ &\Longleftrightarrow P(x, x^*, \gamma) + \gamma \|x^* - x\|^2 \\ &< P(x^*, x^*, \gamma) \\ &\Longleftrightarrow P(x^*, x^*, \gamma) - P(x, x^*, \gamma) \\ &> \gamma \|x^* - x\|^2 \\ &\Longleftrightarrow \frac{P(x^*, x^*, \gamma) - P(x, x^*, \gamma)}{\|x^* - x\|^2} > \gamma \\ &\Longleftrightarrow J(x, x^*, \gamma) > \gamma, \\ f(x) \geq f(x^*) &\Longleftrightarrow S(x, x^*) = S(x^*, x^*) \\ &\Longleftrightarrow P(x, x^*, \gamma) + \gamma \|x^* - x\|^2 \\ &= P(x^*, x^*, \gamma) \\ &\Longleftrightarrow \frac{P(x^*, x^*, \gamma) - P(x, x^*, \gamma)}{\|x^* - x\|^2} = \gamma \\ &\Longleftrightarrow J(x, x^*, \gamma) = \gamma. \end{aligned} \tag{14}$$

□

Following **Proposition 4**, it is obvious that better solutions are with larger slop than worse solutions.

Denote

$$\begin{aligned} S_0 &= \{x \mid J(x, x^*, \gamma) = \gamma, x \in D\}, \\ S_0' &= \{x \mid f(x) \geq f(x^*), x \in D\} \\ S_1 &= \{x \mid J(x, x^*, \gamma) > \gamma, x \in D\}, \\ S_1' &= \{x \mid f(x) < f(x^*), x \in D\}. \end{aligned} \tag{15}$$

Then,

$$\begin{gathered} S_0 \cap S_1 = S_0' \cap S_1' = \Phi \\ S_0 \cup S_1 = S_0' \cup S_1' = D \\ S_0 = S_0', \qquad S_1 = S_1'. \end{gathered} \tag{16}$$

Proposition 5. *Suppose that $B_1 \subset S_1$ is a basin of $S(x, x^*)$ and x_1^* is the minimizer in B_1, then*

$$J(x_1^*, x^*, \gamma) = \max_{x \in B_1} \{J(x, x^*, \gamma)\}. \tag{17}$$

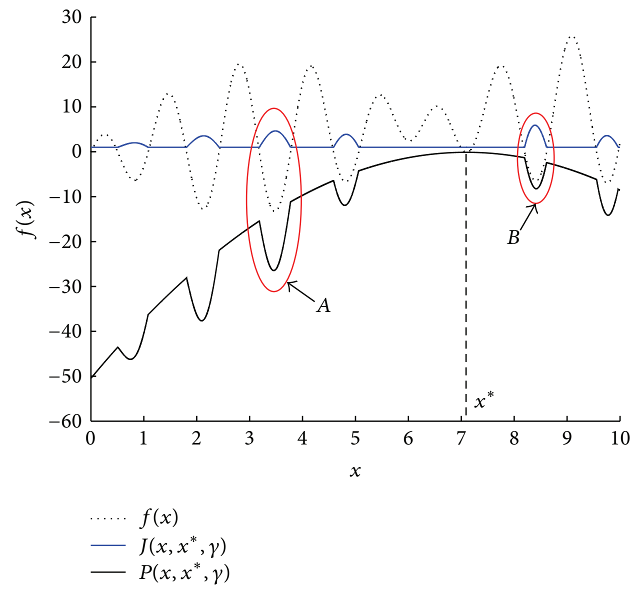


Figure 9: Minmum-escape function at the best solution x^*. The dot-line indicates the original function $f(x)$; the black and blue solid line indicate $P(x, x^*, \gamma)$ and $J(x, x^*, \gamma)$, respectively.

Proof. Follow that $S_1 = S_1'$, $B_1 \subset S_1'$. Consider $\forall x \in B_1$, $S(x^*, x^*) > S(x, x^*) \geq S(x_1^*, x^*)$,

$$\begin{aligned} J(x, x^*, \gamma) &= \frac{S(x^*, x^*) - S(x, x^*) + \gamma \|x^* - x\|^2}{\|x^* - x\|^2} \\ &= \frac{S(x^*, x^*) - S(x, x^*)}{\|x^* - x\|^2} + \gamma \\ &\leq \frac{S(x^*, x^*) - S(x_1^*, x^*)}{\|x^* - x\|^2} + \gamma \\ &= J(x_1^*, x^*, \gamma). \end{aligned} \tag{18}$$

Therefore,

$$J(x_1^*, x^*, \gamma) = \max_{x \in B_1} \{J(x, x^*, \gamma)\}. \tag{19}$$

□

According to **Proposition 5**, one might conclude that if $\tilde{x}^*$ is the global minimizer of $S(x, x^*)$, then $J(\tilde{x}^*, x^*, \gamma) = \max_{x \in S_1} \{J(x, x^*, \gamma)\}$ holds. Unfortunately, this conclusion does not always hold. We will give a counter example to show that the point $\tilde{x}^*$, which satisfies $J(\tilde{x}^*, x^*, \gamma) = \max_{x \in S_1} \{J(x, x^*, \gamma)\}$, is not the global minimizer of $S(x, x^*)$. Here, we still use the example function in this section, where $x \in [0, 10]$ and the best point is $x^* = 7.102$. In **Figure 9**, we denote the basins as A and B, which are surrounded by Ellipses A and B, respectively. From **Figure 9**, it can be seen vividly that the global minimum is included in A, and B is an ordinary basin worse than the global minimum. What is unusual is that $J(x, x^*, \gamma)$ does not achieve its global maximum at the global minimum of $f(x)$ but achieves its global maximum at a local minimum of $f(x)$ in B. It might seem strange. Next, we will discuss the reason of this strange phenomenon.

For the convenience of analyzing this phenomenon, the domain of the original problem is expanded to R^n and the objective function is supposed to be an uniformly continuous function, which satisfies that, for $\forall x \in R^n$, $\exists M > 0$ such that $|f(x)| \leq M$. Under this assumption, we can get the following theorems.

Proposition 6. *If $\exists M > 0$, such that, for $\forall x$, $|f(x)| \leq M$ holds, then*

$$\lim_{x\to\infty} J(x, x^*, \gamma) = \gamma. \tag{20}$$

Proof. Consider

$$\begin{aligned}\lim_{x\to+\infty} J(x, x^*, \gamma) &= \lim_{x\to+\infty} \frac{P(x^*, x^*, \gamma) - P(x, x^*, \gamma)}{\|x^* - x\|^2} \\ &= \gamma + \lim_{x\to+\infty} \frac{S(x^*, x^*) - S(x, x^*)}{\|x^* - x\|^2} \\ &= \gamma.\end{aligned} \tag{21}$$

□

Proposition 7. *For $x_1, x_2 \in S_1$, suppose that $x_1 \neq x_2$ and $\|x^* - x_1\|^2 < \|x^* - x_2\|^2$; if $f(x_1) = f(x_2)$, then*

$$J(x_1, x^*, \gamma) > J(x_2, x^*, \gamma). \tag{22}$$

Proof. If $f(x_1) = f(x_2)$, then $S(x_1, x^*) = S(x_2, x^*)$ and $P(x_1, x^*, \gamma) > P(x_2, x^*, \gamma) > \gamma$ hold. Then

$$\begin{aligned}J(x_1, x^*, \gamma) &= \frac{P(x^*, x^*, \gamma) - P(x_1, x^*, \gamma)}{\|x^* - x_1\|^2} \\ &= \frac{P(x^*, x^*, \gamma) - P(x_2, x^*, \gamma)}{\|x^* - x_1\|^2} \\ &> \frac{P(x^*, x^*, \gamma) - P(x_2, x^*, \gamma)}{\|x^* - x_2\|^2} \\ &= J(x_2, x^*, \gamma).\end{aligned} \tag{23}$$

□

Propositions 6 and 7 can explain why this strange phenomenon occurs. Proposition 6 means that, for $\forall x \in S_1$, the further away from x^*, $J(x, x^*, \gamma)$ will be closer to γ. If x is sufficient far away from x^*, then the value of $J(x, x^*, \gamma)$ is so close to γ that its variation will be tiny. From Proposition 7, we can infer that the closer a basin is to x^*, the greater $J(x, x^*, \gamma)$ will change. If a local minimum $\widehat{x}$ is closer to the reference point x^* than the global minimum $\widetilde{x}^*$ then $J(\widehat{x}, x^*, \gamma) > J(\widetilde{x}^*, x^*, \gamma)$ might take place.

From Propositions 1, 2, and 4, it can be seen that the designed minimum-elimination function can efficiently eliminate the solutions worse than the best one found so far. In this way, the original problem can be converted to another global optimization problem sharing the same global optimum, which will be good for solution algorithm to avoid trapping into local minimum. The current best solution can be converted to the global maximizer by the designed minimum-escape function. During the search process, the best solution found so far will not be visited twice. The designed minimum-escape function can help the solution algorithm to explore the whole domain by reducing the objective function value in proportion to the distance to the best solution found so far. And Proposition 4 provides a method to estimate whether the point is better than the current best solution.

Estimating the parameter γ is another issue. In most existing auxiliary function methods, estimating parameters is an important and hard task. However, the parameter γ in the proposed function is not sensitive and can be adopted easily. In general, when x^* is the best solution found so far, the search will go away from x^*. However, when the search goes to a point x far away from x^* (usually x is away from x^* in order to jump out the current basin), the main problem arisen will be that $\gamma\|x - x^*\|$ is too large to result in arithmetic overflow. So γ should be small in order to avoid this phenomenon. Thus, in the proposed algorithm, it will be no problem when γ is taken not too large. In this paper, the parameter γ is suggested to be taken in $(0, 1]$. In experiments, the influence of parameter γ will be tested.

4. Solution Algorithm

Based on the strategies described above, we design a new minimum-elimination-escape function method, named MEEF for short. In the proposed method, the search process is divided into global search and local search. The gradient-free descent method is employed here to do global search and the gradient-based descent method is employed to do local search. Using gradient-free descent methods can accelerate the speed of finding a better solution, and inaccuracy line searching methods are suggested. The gradient-based descent methods can ensure that the found better solution can be improved to the precise local optima. In the proposed method, Amijo method is employed, and BFGS method is employed to update the found solution.

The executable algorithm MEEF is designed as Algorithm 8.

Algorithm 8. Minimum-Elimination-Escape Function Method (MEEF)

(1) Given initial point x_0, preset the parameter $\gamma > 0$. Let $i = 0$.

(2) Start from x_i and use a gradient-based descent method to minimize the original objective function till a local minimum x^* is obtained; if stop condition is satisfied, stop; otherwise, go to Step 3.

(3) Construct the minimum-escape function $P(x, x^*, \gamma)$ according to (8) with respect to x^* in Step 2. Randomly generate K uniformly distributed vectors e_k $(k = 1 \sim K)$ on the unit ball; $j = 0$.

(4) Consider $j = j + 1$. Start from x^* and use a gradient-free method along with direction e_j, if there exists a point x' such that $J(x', x^*, \gamma) > \gamma$, $i = i + 1$, $x_i = x'$, then go to Step 5; else, go to Step 4.

5. Numerical Experiments and Comparison

5.1. Benchmark Problems. We choose 19 standard benchmark problems from [3] to [17] to test the performance of the designed algorithms, which are listed in the appendix. In these selected test problems, the dimensions are from 2 to 30. It is needed to point out that although f_9 and f_{10} are almost the same, both of these two test problems were reserved for comprehensive comparison with other methods.

5.2. Experimental Settings. In experiments, the algorithm was set as follows.

(i) The Amijo line search method was employed as a gradient-free method to do global search in Step 4, and the BFGS method was taken as a gradient-based method to do the local search in Step 2.

(ii) In Amijo method, the initial step length α is adopted as $\alpha = \max_{1\leq i\leq n}\{U_i - L_i\} \times 1.0e - 4$, where n is the dimension of the problems and L, U are the lower bound and upper bound, respectively.

(iii) The parameter $\gamma = 1.0e - 4$. When analyzing the influence of γ, γ is taken as $1.0e - 4$ and $1.0e - 9$, respectively.

(iv) Number of direction vectors $K = 2n$, where n is the dimension of the problems.

(v) Selection of initial points. In analysis of the influence of parameter γ and initial points, the initial points in experiments are chosen as listed in Tables 2, 3, 4, and 7. In comparison with other methods, the initial point is generated randomly for each test problem.

(vi) Stopping criterion: when $|f_{\text{obtain}} - f_{\text{global}}| \leq 1.0e - 10$ holds, where f_{obtain} is the function value obtained by our algorithm and f_{global} is the known global optimal function value of the test functions, the execution of the algorithm is stopped.

5.3. Experimental Results and Comparisons. The proposed algorithm MEEF is essentially a random search method. Its performance needs to be analyzed in different aspects. For an auxiliary function method, its performance is often influenced by its parameters and initial points, which should not be avoided. In numerical experiments, the influences of the parameter γ and the initial points are tested first. Then, comparisons between MEEF and other methods are made.

In each of Table 2 to Table 8, P represents the related benchmark problem used in this paper, D the dimension of the benchmark problem, FE the number of function evaluations, f_{mean} the mean function value, f_{best} the best function value, f_{std} the standard deviation of function value, Succ the ratio of successful runs, and "—" no result reported.

5.3.1. Influence of Parameter γ. For an auxiliary function method, the parameters often influence its performance significantly. Sometimes, improper parameters might inccrease the complexity of the original problems and affect the global search of the algorithm [14]. In order to analyze the influence of parameter γ, we took different values for parameter γ from the suggested interval and used the same initial points in the numerical experiments. Tables 2 and 3 show the results of MEEF using the same initial points with parameter $\gamma = 1.0e - 4$ and $1.0e - 9$, respectively.

From Tables 2 and 3, it can be seen obviously that MEEF can find the optimal or close-to-optimal solutions for all test problems with different parameter value. Comparing the results listed in Tables 2 and 3, one can find that, with different parameter value, the MEEF can obtain almost the same results using the same function evaluations. This indicates that taking from the suggested interval, the parameter γ has no big influence for MEEF.

5.3.2. Influence of Initial Points. As well known, the initial point plays an important role for a deterministic optimization method. Auxiliary function methods are a kind of optimization methods which execute deterministic methods repeatedly on the transformed objective function and/or the original objective function to find the global optima of the problems. Thus, It is inevitable that the initial point influences the performance of an auxiliary function method. For another reason, MEEF is essentially a random search method; the initial point might have more influence on its performance. In the experiments, we use different initial points to test their influence. Table 4 shows the results of MEEF on the test problems starting from different initial points. From Table 4, we can find that MEEF can find optimal or close-to-optimal solutions for all test problems. Comparing the results listed in Tables 2 and 4, the influence brought by initial points is mainly on the number of function evaluations. At this point, it is reasonable and appears to be very similar to deterministic methods. However, for f_{17}, when the initial point was far away from the global optimal solution, it will take much computational cost and time cost to reach the stopping condition. Sometimes, MEEF failed in finding close-to-optimal solution for f_{17}.

Table 7 shows the results of MEEF on f_{16} and f_{19} with dimension from 2 to 50 starting from different initial points. From same initial points, the number of function evaluations increases with dimension of the problem, which validates intuitionistic deduction on the relationship between function dimension and the number of function evaluations.

5.3.3. Comparisons. For comparison, Ge's Fill Function method (FF) [3], Levy's Tunneling function method (Tun) [8], Wang's auxiliary function method (NAF) [14], the Cut-Peak function method (C-P) [11], SCM method [10], TRUST method [18], Multilevel Single Linkage method (MSL) [19], Diffusion Equation method (DE) [20], Ma's filled function method (PFFF) [21], and Wei's filled function methods (NFFA [22] and NFFM [23]) are selected. These methods can find the optimal or close-to-optimal solutions effectively. The results of these methods reported in the related literature

Table 1

i	a_i				c_i	i	a_i				c_i
1	4	4	4	4	0.1	6	2	9	2	9	0.6
2	1	1	1	1	0.2	7	5	5	3	3	0.3
3	8	8	8	8	0.2	8	8	1	8	1	0.7
4	6	6	6	6	0.4	9	6	2	6	2	0.5
5	3	7	3	7	0.4	10	7	3.6	7	3.6	0.5

Table 2: Results obtained by the MEEF method with parameter $\gamma = 1.0e-4$.

P	n	Initial point	Obtained solution	Function value	FE
f_1	2	(2, 3)	(0.089842003437579, −0.712656409125114)	−1.031628453489877	85
f_2	2	(2, 2)	(3.141592688391699, 2.275000558005582)	0.397887357730086	24
f_3	2	(2, 3)	(0.000000101085690, −1.000000080645316)	3.000000000007129	123
f_4	2	(0.5, 0.5)	$(0, 0.177012710045332)e-8$	−2.000000000000000	9
f_5	2	(2, 2)	(0.999999975527650, 0.999999946318032)	$6.101167528810935e-16$	39
f_6	2	(2, 2)	$(-0.206247122387113, -0.641651061630765)e-6$	$3.644531508398336e-13$	55
f_7	2	(2, 2)	$(0.255229333496938, -0.434186565450557)e-6$	$4.490860908315311e-13$	72
f_8	2	(−5, −5)	(−7.708313690302231, −7.083506338327442)	−186.7309088310084	117
f_9	2	(−5, −5)	(−1.425128417092666, −0.800321092273561)	−186.7309088310210	147
f_{10}	2	(−5, −5)	(−1.425128512018681, −0.800320977448782)	−186.7309088309708	120
f_{11}	4	(5, . . ., 5)	(4.000037215924857, 4.000133145585854, 4.000037215925353, 4.000133145585680)	−10.153199679054000	175
f_{12}	4	(5, . . ., 5)	(4.000571087341658, 4.000688525752843, 3.999491847214348, 3.999606246757180)	−10.402940565956264	215
f_{13}	4	(5, . . ., 5)	(4.000744184774035, 4.000591649142067, 3.999664021647593, 3.999514044541852)	−10.536409814138384	200
f_{14}	6	(−5, . . ., −5)	(0.999999975220856, 1.000000087429692, 1.000000087650645, 1.000000087650650, 1.000000087650650, 1.000000087666372)	$5.214584135632792e-14$	63
f_{15}	6	(−5, . . ., −5)	(1.000000115352228, 1.000028015306673, 1.000000362452354, 1.000004773697615, 1.000015817395051, 0.999995146352257)	$3.543622987114415e-11$	434
f_{16}	6	(−5, . . ., −5)	(1.000000043566668, 0.999999279798989, 0.999999684929615, 0.999999795460065, 0.999999641721513, 1.000000155571707)	$8.329925614010338e-14$	84
	30	(−5, . . ., −5)	(1.000000, . . ., 1.000000)	$7.853766183846135e-12$	403
f_{17}	30	(400, . . ., 400)	(420.96874, . . ., 420.96874)	−12569.48661817255	186
f_{18}	30	(2, . . ., 2)	(0, . . ., 0)	0	93
f_{19}	30	(−50, . . ., −50)	(−2.903534, . . ., −2.903534)	−1174.984119718874	496

are used directly for a comparison. Table 5 presents the results obtained by the proposed algorithm and the compared methods.

From Table 5 we can see that our algorithm outperforms NFFM, NFFA, PFFF, and the cut-peak function method. Comparing the results listed in Tables 2 and 3 with the results of C-P method reported in [11], NFFA in [22] and NFFM in [23], it can be seen that MEEF can find better solutions using much fewer function evaluations starting from different initial points for all test problems. Also, the termination precision of MEEF is much higher than that of both C-P methods, NFFA and NFFM.

So as to compare with NFFA in detail, we executed our proposed method MEEF on f_{14} with different dimensions according to [22]. The comparison results are shown in Table 6. Table 6 shows vividly that MEEF can find the close-to-optimal solutions of f_{14} successfully for all dimensions in experiments. The precision of the results obtained by MEEF is much higher than that of NFFA. However, when the dimension is more than 12, the function evaluation numbers of MEEF are lager than NFFA, which shows that NFFA outperforms MEEF. It is worth mentioning that the stopping precision of NFFA is set to be $1.0e-3$, which is much larger than that of MEEF. Also, NFFA cannot obtain the approximate optimal solutions successfully in each run. What is more, the parameter of NFFA was needed to be adjusted according to dimensions and problems as reported in [22]. MEEF does not need to adjust its parameter specially. Thus,

Table 3: Results obtained by the MEEF method with parameter $\gamma = 1.0e - 9$.

P	n	Initial point	Obtained solution	Function value	FE
f_1	2	(2, 3)	(−0.089842002528404, 0.712656392613011)	−1.031628453489876	82
f_2	2	(2, 2)	(3.141592041900419, 2.275001484633560)	0.397887357732550	24
f_3	2	(2, 3)	(0.000000079909768, −1.000000070623033)	3.000000000004941	121
f_4	2	(0.5, 0.5)	$(0, 0.177012710045332)e - 8$	−2.000000000000000	9
f_5	2	(2, 2)	(0.999999975527650, 0.999999946318032)	$6.101167528810935e - 16$	39
f_6	2	(2, 2)	$(0.971130213666137, 0.706031037637329)e - 8$	$1.699019537462052e - 16$	55
f_7	2	(2, 2)	$(0.255229333496938, -0.434186565450557)e - 6$	$4.490860908315311e - 13$	72
f_8	2	(−5, −5)	(−7.708313676331958, −0.800321082369111)	−186.7309088310150	117
f_9	2	(−5, −5)	(−1.425128417571605, −0.800321090609564)	−186.7309088310209	153
f_{10}	2	(−5, −5)	(−1.425128416153344, −0.800321091763070)	−186.7309088310196	123
f_{11}	4	(5, . . ., 5)	(4.000037215924857, 4.000133145585854, 4.000037215925353, 4.000133145585680)	−10.153199679054000	175
f_{12}	4	(5, . . ., 5)	(4.000572886900829, 4.000689336118056, 3.999489677757608, 3.999606130253441)	−10.402940566818309	220
f_{13}	4	(5, . . ., 5)	(4.000746305174065, 4.000589588385656, 3.999664311018934, 3.999511307520479)	−10.536409815258031	200
f_{14}	6	(−5, . . ., −5)	(0.999999975220856, 1.000000087429692, 1.000000087650645, 1.000000087650650, 1.000000087650650, 1.000000087666372)	$5.214584135632792e - 14$	63
f_{15}	6	(−5, . . ., −5)	(0.999999825677932, 1.000021645871587, 1.000005196859773, 1.000037076494750, 0.999975987700201, 1.000005012447259)	$8.099295276500388e - 11$	435
f_{16}	6	(−5, . . ., −5)	(1.000000043566668, 0.999999279798989, 0.999999684929615, 0.999999795460065, 0.999999641721513, 1.000000155571707)	$8.329925614010338e - 14$	84
	30	(−5, . . ., −5)	(1.000000, . . ., 1.000000)	$7.853766183846135e - 12$	403
f_{17}	30	(400, . . ., 400)	(420.96874, . . ., 420.96874)	−12569.48661817255	186
f_{18}	30	(2, . . ., 2)	(0, . . ., 0)	0	93
f_{19}	30	(−50, . . ., −50)	(−2.903534, . . ., −2.903534)	−1174.984119718874	496

it is easy to conclude that MEEF outperforms NFFA over stability and validity.

Compared with Wang's auxiliary function method NAF, MEEF outperforms this method on f_3, f_7, and f_{10} from Table 5. However, from Table 8, it can be seen obviously that MEEF outperforms NAF on f_{16} and f_{19} with dimension from 2 to 50.

From Table 5, one can see that the performance of MEEF equals that of SCM. But for f_{13}, MEEF could find the global optimal solution successfully, and SCM failed. For other methods, MEEF has much better performance.

Since only the number of function evaluations of the methods listed in Table 5 is reported in the related papers, we compared the number of function evaluations with that of the methods listed in Table 5. From Tables 2, 4, and 5, it can be seen obviously that the proposed algorithm outperforms the compared method in Table 5. Compared with Wang's method, the proposed method uses fewer function evaluations on f_2, f_7, f_9, $f_{10\sim 13}$. The proposed method outperforms SCM except f_1. And the proposed method outperforms other methods on function evaluations of the problems listed in Table 5.

Numerical experiments on the problems f_{16} and f_{19} with dimensions from 2 to 50 were made by Zhu et al. [10] and Wang et al. [14] to evaluate the performance of their methods. In this paper, we do the same experimental settings for f_{16} and f_{19} as in [10, 14] to fair comparison, and the results can be found in Tables 7 and 8. In experiments, the proposed algorithm was executed with two different initial points for each of f_{16} and f_{19} and terminated when $|f_{\text{obtain}} - f_{\text{global}}| < 1.0e - 10$ holds. The final obtained solutions and numbers of function evaluations are recorded in Table 7. For f_{16}, it can be seen vividly that the influence of initial points is obvious not only on the number of function evaluations but also on the precision of the results. During the experiments on f_{16}, our algorithm could obtain the global optimum starting with other different initial points. For f_{19}, it can be seen from Table 7 that both numbers of function evaluations and obtained solutions are with no great difference for MEEF starting with two selected initial points. However, it should be pointed out that MEEF can find the global solution with the preset precision with 90% success in 30 runs when the initial points were taken as $x_0 = (50, \ldots, 50)$ and

Table 4: Results obtained by the MEEF method starting from different initial points.

P	n	Initial point	Obtained solution	Function value	FE
f_1	2	(−2, −2)	(−0.089842004424893, 0.712656395105758)	−1.031628453489877	89
f_2	2	(8, 8)	(9.424776894809153, 2.474997849423675)	0.397887357736760	30
f_3	2	(−2, −2)	(0.000000010826836, −0.999999992529544)	3.000000000000036	97
f_4	2	(−0.7, −0.6)	(0.836906625737857, −0.650879325753485)e − 8	−1.999999999999982	74
f_5	2	(−2, −2)	(0.999999408887528, 0.999998918185347)	3.544550322092880e − 13	45
		(−2, 2)	(1.000000321772440, 0.999999706335175)	5.427186156663280e − 13	39
f_6	2	(−2, −2)	(0.025802684495041, 0.484606092900819)e − 7	2.236704842114040e − 15	55
		(−2, 2)	(0.025802684495041, 0.484606092900819)e − 7	1.206260598539464e − 11	33
f_7	2	(−2, −2)	(−2.000000119209290, 0.000000000000016)	2.236704842114040e − 15	9
		(−2, 2)	(−2.000000119209290, −0.000000000000016)	2.236704842114040e − 15	9
f_8	2	(5, 5)	(5.482864160335367, 4.858056875878652)	−186.7309088310190	52
		(8, 8)	(18.049234762369981, 4.858057086084300)	−186.7309088309161	93
f_9	2	(8, 8)	(−1.425128433538681, −0.800321104507429)	−186.7309088310213	123
f_{10}	2	(8, 8)	(−1.425128424119843, −0.800321101249553)	−186.7309088310201	91
f_{11}	4	(3, 3, 3, 3)	(4.000037114918254, 4.000133242814511, 4.000037114918254, 4.000133242814511)	−10.153199679057712	80
f_{12}	4	(3, 3, 3, 3)	(4.000572886907811, 4.000689336992803, 3.999489678498883, 3.999606128564901)	−10.402940566818305	110
f_{13}	4	(3, 3, 3, 3)	(4.000746502821329, 4.000592906692932, 3.999663365362384, 3.999509769213728)	−10.536409816691679	110
f_{14}	6	(5, . . ., 5)	(1.000000001648912, 1.000002074290066, 1.000002058608007, 1.000002058496111, 1.000002058496307, 1.000002058512318)	1.112809898134190e − 11	49
		(8, . . ., 8)	(1.000000053505098, 0.999999912139717, 0.999999918053981, 0.999999918047487, 0.999999918031023, 0.999999905452657)	1.687131064818418e − 13	63
f_{15}	6	(5, . . ., 5)	(1.000000102099649, 0.999999675649315, 0.999998985096000, 1.000005834707141, 0.999992602921426, 1.000002804125038)	3.233163366392778e − 012	357
		(8, . . ., 8)	(0.999999977831993, 1.000001966743495, 0.999999885358024, 1.000000409193240, 1.000000132358755, 0.999999852114368)	1.353844335746799e − 013	413
f_{16}	6	(4, . . ., 4)	(1.000000178908328, 1.000007873947206, 1.000002877836001, 0.999999355316356, 0.999984951143500, 1.000000910594055)	2.983417815754360e − 12	835
		(8, . . ., 8)	(1.000000346074765, 0.999995860881688, 1.000000027091785, 1.000006231036376, 1.000001238636247, 0.999998410125881)	6.132259614323958e − 12	266
f_{17}	30	(500, . . ., 500)	(420.96874, . . ., 420.96874)	−12569.48661817271	217
		(321, . . ., 321)	(420.96873, . . ., 420.96873)	−12569.48661817220	310
f_{18}	30	(−3, . . ., −3)	(0.15432, . . ., 0.15432)e − 9	0	217
		(−5, . . ., −5)	(−0.78662277565, . . ., −0.78662277565)e − 8	3.730349362740526e − 13	217
		(5, . . ., 5)	(−0.78662277565, . . ., −0.78662277565)e − 8	3.730349362740526e − 13	217
f_{19}	30	(50, . . ., 50)	(−2.9035340, . . ., −2.9035340)	−1174.984971113140	652

$x_0 = (100, \ldots, 100)$. This indicates that the performance of MEEF on f_{19} is influenced by the initial points greatly.

In the following experiments, the precision for f_{19} was set to be 1.0×10^{-1} which was the same as that of NAF [14], and the initial points were taken randomly. From Table 8, it can be seen that MEEF outperforms other 4 methods on the number of function evaluations obviously form f_{16}. But for f_{19}, the comparison results seem to be a little complex. MEEF uses fewer function evaluations than NAF in 15 test cases. From Table 8, it can be seen that SCM seems to have better performance than MEEF on f_{19}. It should be pointed out that in [10], the precision of the obtained results was not mentioned. Thus, we can not judge which one has better performance. For the tunneling function method [8], only 4 test cases were executed without any solution precision. We can not do comprehensive comparison.

5.3.4. A Phenomenon for Discussion. Careful readers might find a strange phenomenon in Table 8. That is the number of function evaluations of SCM and NAF on f_{16} and f_{19} does not present promptly increasing trend with the increasing of the dimension. The number of function evaluations varies greatly. For example, for f_{19}, NAF uses 14956 function evaluations to obtain the global solution with the preset precision when the problem dimension is 21. While the number of function evaluations is down sharply to 93 when

Table 5: Comparison of different methods on f_1 to f_{18}.

P	n	MEEF	NFFM	NFFA	PFFF	C-P	NAF	SCM	Tun	TRUST	DE	FF	MSL
f_1	2	53	74	84.2449	518	497	28	55	1496	—	120	184	—
f_2	2	34	202	—	1819	619	26	57	—	60	—	189	206
f_3	2	83	80	64.9796	8140	343	86	98	—	—	120	148	184
f_4	2	9	814	—	337	153	—	—	—	—	—	—	—
f_5	2	38	—	—	—	—	—	—	—	—	—	—	—
f_6	2	43	74	67.9796	545	—	28	46	—	—	—	429	—
f_7	2	29	72	53.1224	595	—	30	61	—	—	—	313	—
f_8	2	134	78	625.4898	5280	—	37	103	12160	256	—	290	—
f_9	2	149	—	—	—	—	114	166	2912	—	—	234	—
f_{10}	2	163	—	—	—	555	205	232	2180	—	—	439	—
f_{11}	4	178	116	357.1429	—	—	110	156	—	—	12000	390	404
f_{12}	4	212	209	—	—	—	165	159	—	—	12000	410	Fail
f_{13}	4	203	507	—	—	—	183	Fail	—	—	12000	Fail	564
f_{14}	6	63	—	—	—	319	—	—	—	—	—	—	—
f_{15}	6	434	—	—	—	497	—	—	—	—	—	—	—
f_{16}	6	84	—	—		365	—	—	—	—	—	—	—
f_{17}	30	186	—	—	—	523	—	—	—	—	—	—	—
f_{18}	30	93	18613	—	—	2049	—	—	—	—	—	—	—

Table 6: Comparison of MEEF and NFFA on f_{14} with different dimensions.

n	Method	FE	f_{mean}	f_{best}	f_{std}	Succ
7	MEEF	1922.88	$3.4441e-12$	$2.6303e-15$	$2.8989e-12$	100%
	NFFA	6028	$7.5895e-8$	$8.2743e-8$	0	100%
10	MEEF	2412.96	$3.1753e-12$	$4.1488e-14$	$3.0139e-12$	100%
	NFFA	6027	0.0012	$8.3180e-7$	0.0020	97.10%
12	MEEF	3198.52	$2.4606e-12$	$2.2468e-14$	$2.3517e-12$	100%
	NFFA	641.0518	$1.5597e-5$	$2.0935e-10$	$1.1098e-8$	93.47%
15	MEEF	3623.68	$2.9584e-12$	$1.1703e-13$	$2.9320e-12$	100%
	NFFA	972.1050	$2.4116e-5$	$1.8709e-13$	$1.0032e-5$	89.78%
20	MEEF	5582.64	$2.5215e-12$	$1.4919e-14$	$2.9127e-12$	100%
	NFFA	1321.3011	$1.1602e-4$	$8.2346e-8$	$5.0653e-6$	85.57%
30	MEEF	14039.28	$3.1787e-12$	$1.2915e-14$	$2.9895e-12$	100%
	NFFA	2666.1220	0.3012	$5.3080e-5$	$3.0120e-4$	89.87%

the dimension is 22. This phenomenon is against our intuitive idea that the number of function evaluations increases with the dimension of the problem. In traditional opinion, the higher the dimension is, the more local optimum there will be which will take much computational cost. It might be a strange and interesting topic that why this unreasonable phenomenon happened.

6. Conclusions

In this paper, a new minimum-elimination-escape function method for global optimization is proposed, which can avoid the "Mexican hat" efficiently. In the proposed method, the minimum-elimination function is constructed to eliminate the solutions worse than the best one found so far. In this way, the influence brought by the local optimum will be reduced. Flattened by the minimum-elimination function, the original problem can be transformed to another optimization problem, which shares the same global optimum with the original problem with fewer local optimum. A new minimum-escape function with one parameter is constructed for the flattened problem. The properties of minimum-escape function are analyzed theoretically. Based on the two proposed functions, a minimum-elimination-escape function method for multimodal optimization is constructed. Theoretical analysis and experimental results indicate that the minimum-elimination-escape function method is insensitive to its unique parameter, which can be set easily. In the experiments, the proposed algorithm can find the global optimal solutions of all 19 selected problems. The numerical experimental results show that the proposed algorithm can converge rapidly to the global optimum with high precision. Compared with 11 existing methods, we found that the proposed algorithm performs much more stably and effectively.

Table 7: Results obtained by MEEF for f_{16} and f_{19} with different dimensions and initial points.

n	f_{16}				f_{19}			
	$x_0 = (-5, \ldots, -5)^n$		$x_0 = (8, \ldots, 8)^n$		$x_0 = (-50, \ldots, -50)^n$		$x_0 = (-100, \ldots, -100)^n$	
	FE	Function value	FE	Function value	FE	Function value	FE	Function value
2	18	$1.085446757e-14$	18	$1.297419473e-13$	64	−78.33233141	70	−78.33233141
3	32	$3.881689583e-12$	52	$6.581268930e-14$	85	−117.4984971	93	−117.4984971
4	50	$1.354820184e-11$	110	$2.251127378e-12$	106	−156.6646628	116	−156.6646628
5	78	$4.197601362e-14$	150	$4.168584952e-13$	127	−195.8308285	139	−195.8308285
6	84	$8.329925614e-14$	266	$1.472052530e-11$	148	−234.9969942	162	−234.9969942
7	112	$6.235422186e-14$	376	$2.189048047e-12$	169	−274.1631599	185	−274.1631599
8	126	$1.306299412e-13$	270	$5.815899531e-12$	190	−313.3293256	208	−313.3293256
9	140	$6.703769837e-14$	300	$2.936541720e-11$	211	−352.4954913	231	−352.4954913
10	143	$7.356304059e-12$	352	$3.516108327e-12$	232	−391.6616570	254	−391.6616570
11	168	$5.459150817e-14$	408	$6.176002998e-12$	253	−430.8278227	277	−430.8278227
12	182	$9.150503163e-14$	455	$8.683242009e-12$	274	−469.9939884	300	−469.9939884
13	196	$6.270859476e-14$	476	$3.718745017e-12$	295	−509.1601541	323	−509.1601541
14	210	$4.792250528e-14$	495	$2.219453850e-11$	316	−548.3263199	346	−548.3263199
15	224	$4.098385496e-14$	544	$8.168107717e-12$	337	−587.4924856	369	−587.4924856
16	221	$2.315381989e-12$	527	$2.298317761e-11$	358	−626.6586513	392	−626.6586513
17	234	$2.138104564e-12$	594	$3.817915433e-11$	379	−665.8248170	415	−665.8248170
18	266	$6.000741134e-14$	646	$2.690515855e-11$	400	−704.9909827	438	−704.9909827
19	260	$2.123271916e-12$	680	$5.573695353e-12$	421	−744.1571484	461	−744.1571484
20	294	$4.108817487e-14$	735	$3.857485816e-13$	442	−783.3233141	484	−783.3233141
21	286	$1.795219586e-12$	726	$2.271437801e-12$	463	−822.4894798	507	−822.4894798
22	299	$1.642006180e-12$	736	$1.894022176e-11$	484	−861.6556455	530	−861.6556455
23	312	$1.848577193e-13$	840	$2.361769572e-12$	505	−900.8218112	553	−900.8218112
24	325	$1.483992255e-12$	800	$8.836913596e-12$	526	−939.9879769	576	−939.9879769
25	364	$4.463518375e-14$	884	$2.175002083e-12$	547	−979.1541426	599	−979.1541426
30	403	$7.853766184e-12$	1054	$3.914811669e-12$	652	−1174.984971	714	−1174.984971
40	533	$4.426171374e-12$	1394	$6.979357443e-12$	862	−1566.646628	944	−1566.646628
50	663	$1.099481484e-11$	1938	$2.005274943e-12$	1123	−1958.308285	1174	−1958.308285

Appendix

Benchmarks

(i) Six-hump camel-back ($n = 2$) [24]:

$$f_1(x) = 4x_1^2 - 2.1x_1^4 + \frac{x_1^6}{3} + x_1x_2 - 4x_2^2 + 4x_2^4, \tag{A.1}$$

where $-3 \leq x_1, x_2 \leq 3$. The global minimizers are $x^* = (0.08983, -0.7126)$ and $(-0.08983, 0.7126)$ with the global optimal value being $f(x^*) = -1.031628$.

(ii) Branin ($n = 2$) [25]:

$$f_2(x) = \left(x_2 - \frac{1.275x_1^2}{\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{0.125}{\pi}\right)\cos(x_1) + 10, \tag{A.2}$$

where $-5 \leq x_1 \leq 10, 0 \leq x_2 \leq 15$. The global minimizers are $x^* = (-3.142, 12.275), (3.142, 2.275)$, and $(9.425, 2.425)$ with the global optimal value being $f(x^*) = 0.3979$.

(iii) Goldstein-Price (G-P) ($n = 2$) [26]:

$$\begin{aligned} f_3(x) = {} & \left[1 + (x_1 + x_2 + 1)^2 \cdot \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \\ & \times \left[30 + (2x_1 - 3x_2)^2 \cdot \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right], \end{aligned} \tag{A.3}$$

where $-3 \leq x_1, x_2 \leq 3$. the global minimizer is $x^* = (0, -1)$ with the global optimal value being $f(x^*) = 3$.

(iv) Rastrigin ($n = 2$) [7]:

$$f_4(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \tag{A.4}$$

where $-1 \leq x_1, x_2 \leq 1$. This function has about 50 minimizers; the global minimizer is $x^* = (0, 0)$ with the global optimal value being $f(x^*) = -2$.

Table 8: Comparison of different methods on f_{16} and f_{19} with different dimensions.

n	f_{16}, $\epsilon = 1.0e-10$					f_{19}, $\epsilon = 0.1$			
	MEEF	NAF	SCM	FF	Tun	MEEF	NAF	SCM	Tun
2	128	2206	143	252	2653	63	44	26	38
3	324	2591	331	339	6955	108	20	27	22
4	575	3294	674	1012	3865	120	47	29	21
5	942	3817	1312	938	10715	132	43	27	21
6	1513	4795	824	2262	12786	154	75	32	—
7	1986	5062	675	2951	16063	264	1216	117	—
8	2555	7064	1058	3634	—	207	2223	105	—
9	3339	5366	2736	3623	—	300	2845	74	—
10	4468	8839	1567	2969	—	396	1490	126	—
11	4698	7308	4396	—	—	216	1645	83	—
12	6120	11388	1821	—	—	208	3501	132	—
13	7168	13025	1666	—	—	546	1489	303	—
14	8160	8239	21830	—	—	675	3305	581	—
15	9344	18329	2127	1555	—	1200	51	469	—
16	13600	17144	2987	—	—	1071	1173	795	—
17	11577	21982	38603	—	—	1170	6828	250	—
18	13851	20821	3752	—	—	1064	4743	2234	—
19	10647	32047	4509	—	—	1880	1147	1297	—
20	15740	22713	3593	—	—	2835	77	1306	—
21	10890	29370	3486	4668	—	1628	14956	566	—
22	15272	30881	6703	—	—	1541	93	1370	—
23	39576	31113	23058	—	—	1824	11756	1619	—
24	27000	17109	2558	—	—	975	6789	883	—
25	25636	36202	23141	2361	—	1716	1663	1540	—
30	12245	36983	13732	—	—	1395	1339	1928	—
40	16810	10188	10975	—	—	2419	1917	2073	—
50	36567	63058	11736	—	—	3927	2670	1164	—

ϵ denotes the stopping precision.

(v) Simplified Rosenenbrock problem ($n = 2$) [3]:

$$f_5(x) = 0.5\left(x_1^2 - x_2\right)^2 + \left(x_1 - 1\right)^2, \qquad \text{(A.5)}$$

where $-3 \le x_1, x_2 \le 3$. The global minimizer is $x^* = (1, 1)$ with the optimal value being $f(x^*) = 0$.

(vi) Three-hump camelback problem ($n = 2$) [3, 10]:

$$f_6(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} - x_1x_2 + x_2^2, \qquad \text{(A.6)}$$

where $-3 \le x_1, x_2 \le 3$. The global minimizer is $x^* = (0, 0)$ with the optimal value being $f(x^*) = 0$.

(vii) Treccani problem ($n = 2$) [3, 10]:

$$f_7(x) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2, \qquad \text{(A.7)}$$

where $-3 \le x_1, x_2 \le 3$. The global minimizer is $x^* = (-2, 0)$ and $(0, 0)$ with the optimal value being $f(x^*) = 0$.

(viii) Two-dimensional Shubert problem I ($n = 2$) [3, 8, 10]:

$$f_8(x) = \prod_{j=1}^{2}\left(\sum_{i=1}^{5} i\cos\left((i+1)x_j + i\right)\right), \qquad \text{(A.8)}$$

where $-10 \le x_1, x_2 \le 10$. This function has 760 minimizers and 18 of them are global minimizers with the global optimal value $f(x^*) = -186.730909$.

(ix) Two-dimensional Shubert problem II ($n = 2$) [3, 8, 10]:

$$f_9(x) = \prod_{j=1}^{2}\left(\sum_{i=1}^{5} i\cos\left((i+1)x_j + i\right)\right) + 0.5\left(x_1 + 1.42513\right)^2 + \left(x_2 + 0.80032\right)^2, \qquad \text{(A.9)}$$

where $-10 \le x_1, x_2 \le 10$. This function has 760 minimizers and only one global minimizer $x^* = (-1.42513, -0.80032)$ with the global optimal value $f(x^*) = -186.730909$.

(x) Two-dimensional Shubert problem III ($n = 2$) [3, 8, 10]:

$$f_{10}(x) = \prod_{j=1}^{2}\left(\sum_{i=1}^{5} i\cos\left((i+1)x_j + i\right)\right) + \left(x_1 + 1.42513\right)^2 + \left(x_2 + 0.80032\right)^2, \quad \text{(A.10)}$$

where $-10 \le x_1, x_2 \le 10$. This function has 760 minimizers and only one global minimizer $x^* = (-1.42513, -0.80032)$ with the global optimal value $f(x^*) = -186.730909$.

(xi) Shekel's Family ($n = 4$) [14]:

$$f_{11\sim 13}(x) = -\sum_{i=1}^{m}\left(\left(x - a_i\right)\left(x - a_i\right)^T + c_i\right)^{-1}, \quad \text{(A.11)}$$

where $0 \le x_{1\sim 4} \le 10$. These functions have m local minima ($m = 5, 7, 10$ for f_{11}, f_{12}, f_{13}, resp.) and $x_{\text{local\_opt}} \approx a_i$ with local optimal value $f(x_{\text{local\_opt}}) \approx 1/c_i$. The global minimizer $f(x^*) < -10$, $f_{11}(x^*) = -10.1532$, $f_{12}(x^*) = -10.4029$, $f_{13}(x^*) = -10.5364$. a_{ij}, c_i are taken as shown in Table 1.

(xii) Sine-square I ($n = 6$) [27]:

$$f_{14}(x) = \frac{\pi}{n}\left[10\sin^2\left(\pi x_1\right) + \left(x_n - 1\right)^2 + \sum_{i=1}^{n-1}\left(x_i - 1\right)^2\left(1 + 10\sin^2\left(\pi x_{i+1}\right)\right)\right], \quad \text{(A.12)}$$

where $-10 \le x_i \le 10$, for $i = 1 \sim n$. This function has about 60 minimizers, the global minimizer is $x^* = (1, 1, 1, 1, 1, 1)$ with the global optimal value being $f(x^*) = 0$.

(xiii) Sine-square II ($n = 6$) [27]:

$$f_{15}(x) = \frac{\pi}{n}\left[10\sin^2\left(\pi y_1\right) + \left(y_n - 1\right)^2 + \sum_{i=1}^{n-1}\left(y_i - 1\right)^2\left(1 + 10\sin^2\left(\pi y_{i+1}\right)\right)\right], \quad \text{(A.13)}$$

where $y_i = 1 + ((x_i - 1)/4)$ and $-10 \le x_i \le 10$, for $i = 1 \sim n$. This function has about 30 minimizers, the global minimizer is $x^* = (1, 1, 1, 1, 1, 1)$ with the global optimal value being $f(x^*) = 0$.

(xiv) Sine-square III [27]:

$$f_{16}(x) = \frac{1}{10}\left[\sin^2\left(3\pi x_1\right) + \left(x_n - 1\right)^2\left(1 + \sin^2\left(2\pi x_n\right)\right) + \sum_{i=1}^{n-1}\left(x_i - 1\right)^2\left(1 + \sin^2\left(3\pi x_{i+1}\right)\right)\right], \quad \text{(A.14)}$$

where $-10 \le x_i \le 10$, for $i = 1 \sim n$. This function has about 180 minimizers, the global minimizer is $x^* = (1, \ldots, 1)$ with the global optimal value being $f(x^*) = 0$.

(xv) Generalized Schwefel's Problem ($n = 30$) [17]:

$$f_{17}(x) = -\sum_{i=1}^{n} x_i \sin\left(\sqrt{\left|x_i\right|}\right), \quad \text{(A.15)}$$

where $-500 \le x_i \le 500$, for $i = 1 \sim n$. The global minimizer is $x^* = (420.9687, \ldots, 420.9687)$ with the global optimal value being $f(x^*) = -418.98n$.

(xvi) Generalized Rastrigin's function ($n = 30$) [17]:

$$f_{18} = \sum_{i=1}^{n}\left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right), \quad \text{(A.16)}$$

where $-5.12 \le x_i \le 5.12$, for $i = 1 \sim n$. The global minimizer is $x^* = (0, \ldots, 0)$ with the global optimal value being $f(x^*) = 0$.

(xvii) Test function ($n = 30$) [10, 14]:

$$f_{19}(x) = 0.5\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right), \quad \text{(A.17)}$$

where $x_i \in [-100, 100]$, $i = 1 \sim n$. This function has 2^n local minimizers and only one global minimizer $x^* = (-2.903534, \ldots, -2.903534)$ with global optimal value being $f(x^*) = -39.1662n$.

Disclosure

This paper is an extended version of our conference paper [28]. The extension includes the following: (1) we proposed a new minimum-elimination function; (2) we analyse the properties of our proposed auxiliary function theoretically; (3) we conduct new experiment on more benchmarks to evaluate the robust and effectiveness of our method; (4) comparisons with recently proposed auxiliary function methods are made.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors thank the anonymous reviewers for their constructive remarks and valuable comments for improving the quality of the manuscript. This work was partly supported by Fundamental Research Funds for Central Universities (JB141004), the National Natural Science Foundations of China (61272119 and 61472311), and Research Project of Leshan Normal University (Z1402).

References

[1] C. A. Floudas and C. E. Gounaris, "A review of recent advances in global optimization," *Journal of Global Optimization*, vol. 45, no. 1, pp. 3–38, 2009.

[2] R. Ge, "A filled function method for finding a global minimizer of a function of several variables," in *Proceedings of the Biennial Conference on Numerical Analysis*, Dundee, Scotland, 1983.

[3] R. P. Ge, "A filled function method for finding a global minimizer of a function of several variables," *Mathematical Programming*, vol. 46, no. 2, pp. 191–204, 1990.

[4] Z. Xu, H.-X. Huang, P. M. Pardalos, and C.-X. Xu, "Filled functions for unconstrained global optimization," *Journal of Global Optimization*, vol. 20, no. 1, pp. 49–65, 2001.

[5] Y. L. Shang, D. G. Pu, and A. P. Jiang, "Finding global minimizer with one-parameter filled function on unconstrained global optimization," *Applied Mathematics and Computation*, vol. 191, no. 1, pp. 176–182, 2007.

[6] X. Liu, "Several filled functions with mitigators," *Applied Mathematics and Computation*, vol. 133, no. 2-3, pp. 375–387, 2002.

[7] X. Liu and W. Xu, "A new filled function applied to global optimization," *Computers & Operations Research*, vol. 31, no. 1, pp. 61–80, 2004.

[8] A. V. Levy and S. Gomez, "The tunneling method applied to global optimization," in *Numerical Optimization*, P. T. Boggs, R. H. Byrd and, and R. B. Schnabel, Eds., pp. 213–244, SIAM, 1985.

[9] D. J. Wales and H. A. Scheraga, "Global optimization of clusters, crystals, and biomolecules," *Science*, vol. 285, no. 5432, pp. 1368–1372, 1999.

[10] W. Zhu and Q. Fu, "A sequential convexification method (SCM) for continuous global optimization," *Journal of Global Optimization*, vol. 26, no. 2, pp. 167–182, 2003.

[11] Y. Wang, W. Fang, and T. Wu, "A cut-peak function method for global optimization," *Journal of Computational and Applied Mathematics*, vol. 230, no. 1, pp. 135–142, 2009.

[12] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, 2004.

[13] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Improving particle swarm optimizer by function stretching," in *Advances in Convex Analysis and Global Optimization*, N. Hadjisavvas and P. Pardalos, Eds., vol. 54 of *Non-Convex Optimization and Applications*, pp. 445–457, Kluwer Academic Publishers, Norwell, Mass, USA, 2001.

[14] Y.-J. Wang and J.-S. Zhang, "A new constructing auxiliary function method for global optimization," *Mathematical and Computer Modelling*, vol. 47, no. 11-12, pp. 1396–1410, 2008.

[15] Z.-H. Huang, X.-H. Miao, and P. Wang, "A revised cut-peak function method for box constrained continuous global optimization," *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 224–233, 2007.

[16] L. C. Dixon, J. Gomulka, and S. E. Hersom, "Reflections on the global optimization problem," in *Optimization in Action*, L. C. Dixon, Ed., pp. 398–435, Academic Press, New York, NY, USA, 1976.

[17] C. Barróny and S. Gómez, "The exponential tunneling method," *Reportes de Investigación IIMAS*, vol. 1, no. 3, pp. 1–23, 1991.

[18] B. C. Cetin, J. Barhen, and J. W. Burdick, "Terminal repeller unconstrained subenergy tunneling (TRUST) for fast global optimization," *Journal of Optimization Theory and Applications*, vol. 77, no. 1, pp. 97–126, 1993.

[19] A. H. G. R. Kan and G. T. Timmer, "Stochastic global optimization methods. Part II: multi-level methods," *Mathematical Programming*, vol. 39, no. 1, pp. 57–78, 1987.

[20] J. Kostrowicki and L. Piela, "Diffusion equation method of global minimization: performance for standard test functions," *Journal of Optimization Theory and Applications*, vol. 69, no. 2, pp. 269–284, 1991.

[21] S. Ma, Y. Yang, and H. Liu, "A parameter free filled function for unconstrained global optimization," *Applied Mathematics and Computation*, vol. 215, no. 10, pp. 3610–3619, 2010.

[22] F. Wei and Y. Wang, "A new filled function method with one parameter for global optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 532325, 12 pages, 2013.

[23] F. Wei, Y. Wang, and H. Lin, "A new filled function method with two parameters for global optimization," *Journal of Optimization Theory and Applications*, vol. 163, no. 2, pp. 510–527, 2014.

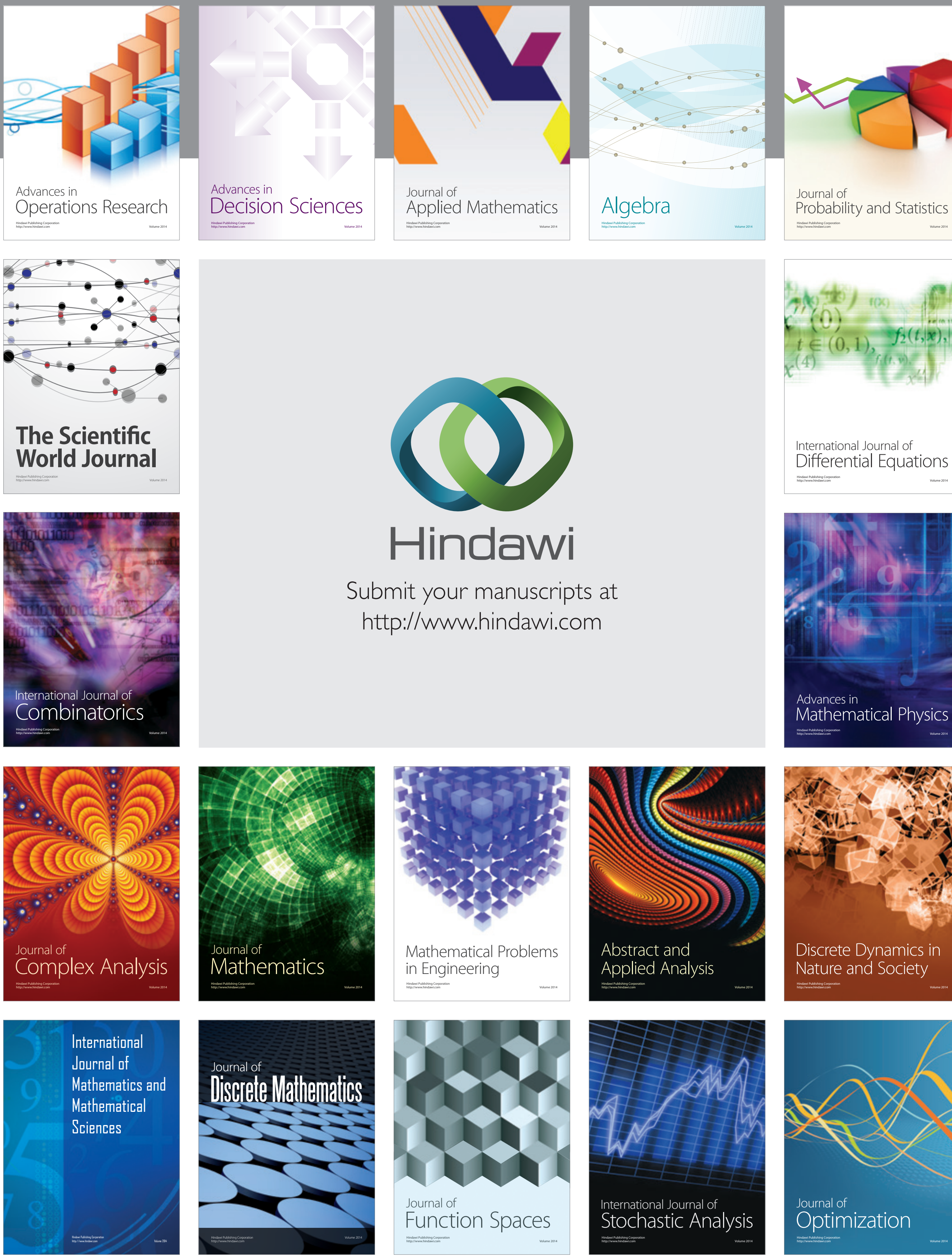
[24] F. Branin, "Wildly convergent methods for finding multiple solutions of simultaneous nonlinear equations," *IBM Journal of Research and Development*, vol. 16, no. 5, pp. 504–522, 1972.

[25] J. Branin and S. K. Hoo, "A method for finding multiple extrema of a function of n variables," in *Numerical Methods for Non-Linear Optimization*, P. M. Lootsma, Ed., pp. 231–237, Academic Press, New York, NY, USA, 1972.

[26] L. Dixon and G. P. Szegö, *Towards Global Optimization*, vol. 2, North-Holland Publishing, Amsterdam, The Netherlands, 1978.

[27] Y. M. Liang, L. S. Zhang, M. M. Li, and B. S. Han, "A filled function method for global optimization," *Journal of Computational and Applied Mathematics*, vol. 205, no. 1, pp. 16–31, 2007.

[28] L. Fan, X. Liu, and L. Jia, "A minimum-elimination-escape function method for multimodal optimization problems," in *Proceedings of the 10th International Conference on Computational Intelligence and Security (CIS '14)*, pp. 312–316, Kunming, China, November 2014.



Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
Hindawi
Submit your manuscripts at
http://www.hindawi.com
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization