

Research Article

A New Filled Function Method with One Parameter for Global Optimization

Fei Wei and Yuping Wang

School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Yuping Wang; wyp9000@yahoo.com.cn

Received 22 November 2012; Accepted 10 May 2013

Academic Editor: Salvatore Alfonzetti

Copyright © 2013 F. Wei and Y. Wang. 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.

The filled function method is an effective approach to find the global minimizer of multidimensional multimodal functions. The conventional filled functions are numerically unstable due to exponential or logarithmic term and sensitive to parameters. In this paper, a new filled function with only one parameter is proposed, which is continuously differentiable and proved to satisfy all conditions of the filled function definition. Moreover, this filled function is not sensitive to parameter, and the overflow can not happen for this function. Based on these, a new filled function method is proposed, and it is numerically stable to the initial point and the parameter variable. The computer simulations indicate that the proposed filled function method is efficient and effective.

1. Introduction

More and more practical problems in science, economics, engineering, and other fields can be formulated as global optimization problems. Lots of researchers have been attracted to the field of global optimization. In recent years, many new theoretical and computational contributions have been reported for solving global optimization problems. Global optimization is mainly concerned with the characteristics and algorithms for the multimodal functions. In general, the existing approaches can be classified into two categories: deterministic methods ([1–16], e.g.) and probabilistic methods ([17–22], e.g.). The typical examples of the former are the filled function method (FFM) [3–9, 13–15], the trajectory method [2, 10], the tunneling method [12], and the covering method [1, 16], whereas ones of the latter are the clustering method [17], the methods reported in [18, 19], the simulated annealing method [22], and genetic algorithms [20, 21]. Furthermore, some hybrid deterministic and probabilistic algorithms ([23], e.g.) were proposed to solve practical problems.

However, the existence of multiple local minima of a general nonconvex objective function makes global optimization become a great challenge. For global optimization problems, there are two major issues.

- (1) How to find a lower minimizer of the objective function from a known local minimizer.
- (2) How to evaluate the convergence and, accordingly, design the stopping criteria.

In this paper, we will focus our research on the FFM and mainly issue (1). Among the existing methods for global optimization problems, the FFM appears to have several advantages over others mainly due to its relatively easy realization with a process that aims at successively finding smaller local minima. The FFM was firstly proposed by Ge in [3], which was used to solve the global minimizer of unconstrained multiextremum function. Later, many scholars have also done a lot of valuable work to improve this method ([5–9, 13–15, 24–28], e.g.). However, conventional filled functions are often nondifferentiable ([13, 25, 26], e.g.), need more than one adjustable parameter ([15], e.g.), or contain ill-conditioned terms ([3, 4, 9], e.g.). To overcome these shortcomings, some parameter-free filled functions ([13, 25], e.g.) and some filled functions without ill-conditioned terms ([26, 27], e.g.) are proposed; however, they are usually nondifferentiable, which often results in additional local minimizers. And some continuously differentiable filled functions with one parameter have been proposed ([8], e.g.), but the parameter is not easy to adjust. To deal with this problem, a new class of filled function

with one parameter, which is continuously differentiable and the parameter is easy to adjust, is proposed in this paper. Based on this, a new filled function method with randomly and uniformly local search scheme is proposed, and the algorithm is numerical stability. In addition, the proposed method can be used to solve the multidimensional multimodal problem.

2. Overview of the FFM

In this paper, we consider the following global optimization problem:

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

where $f(x) : R^n \rightarrow R$ is a twice continuously differentiable function. Suppose $f(x)$ satisfies the condition $f(x) \rightarrow +\infty$ as $\|x\| \rightarrow +\infty$. Then there exists a closed bounded domain Ω called an operating region that contains all local minimizers of $f(x)$. Then the global optimization Problem (P) can be rewritten into an equivalent form as follows:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega = [l, u] = \{x \mid x \in l \leq x \leq u, l, u \in R^n\}. \end{aligned} \quad (\text{P1})$$

Because Ω can be estimated before Problem (P) is solved, so we can assume that Ω is known without loss of generality. We only consider Problem (P1) in the following.

2.1. Basic Concepts and Assumptions. In this paper, we adopt the following symbols:

- k : the iteration number;
- x'_k : the initial point in the k th iteration;
- x_k^* : the local minimizer of the objective function in the k th iteration;
- f_k^* : the function value at x_k^* ;
- B_k^* : the basin of $f(x)$ at an isolated local minimizer x_k^* ;
- x^* : the global minimizer of the objective function.

Assumption 1. The function $f(x)$ in (P1) is continuously differentiable in R^n , $f(x)$ has only a finite number of minimizers in Ω , and therefore every minimizer is an isolated minimizer.

The basin B_k^* of $f(x)$ at a local minimizer x_k^* is defined in [3, 4] as a connected domain, it contains x_k^* , and the steepest descent trajectory of $f(x)$ converges to x_k^* from any initial point in B_k^* . The minimal radius of B_k^* at an isolated minimizer x_k^* is defined as

$$\bar{R} = \inf_{x \notin B_k^*} \|x - x_k^*\|. \quad (1)$$

Radius \bar{R} is not zero if $\nabla^2 f(x_k^*)$ is positive definite. The basin of $f(x)$ at x_k^* is said to be lower than another basin of

$f(x)$ at another local minimizer x_1^* if and only if $f(x_k^*) < f(x_1^*)$. The hill of $f(x)$ at x_k^* is the basin of $-f(x)$ at its isolated minimizer x_k^* .

The definition of the filled function was first proposed by Ge in [3] as follows.

Definition 2. Suppose x_k^* is a current local minimizer of $f(x)$. $P(x, x_k^*)$ is said to be a filled function of $f(x)$ at x_k^* , if it satisfies the following properties:

- (i) x_k^* is a strictly maximizer of $P(x, x_k^*)$, and the whole basin B_k^* of $f(x)$ at x_k^* becomes a part of a hill of $P(x, x_k^*)$;
- (ii) $P(x, x_k^*)$ has no minimizers or stable points in any basin of $f(x)$ higher than B_k^* ;
- (iii) if $f(x)$ has a lower basin than B_k^* , then there is a point x'_{k+1} in such a basin that minimizes $P(x, x_k^*)$ on the line through x and x_k^* .

Based on the filled functions, a global optimization problem can be solved via a two-phase cycle.

In Phase 1, we start from an initial point and use any local minimization method to find a local minimizer x_k^* of $f(x)$.

In Phase 2, we construct a filled function at x_k^* and minimize the filled function in order to identify a point x'_{k+1} with $f(x'_{k+1}) < f(x_k^*)$. If such a point x'_{k+1} is found, x'_{k+1} is certainly in a lower basin than B_k^* . We can then use x'_{k+1} as an initial point in Phase 1 again, and hence we can find a better local minimizer x_{k+1}^* of $f(x)$ with $f(x_{k+1}^*) < f(x_k^*)$. This process repeats until no better solution can be found. The final local minimum will be a global minimizer of $f(x)$.

2.2. Overview of the FFM. As a deterministic yet a universal global optimization technique, the development of the FFM undergoes the following generations. The representative examples of the FFM in the first generation are P -functions [3] and G -functions [4] given by

$$P(x, r, \rho) = \exp \frac{(-\|x - x_k^*\|^2 / \rho^2)}{[r + f(x)]}, \quad (2)$$

$$G(x, r, \rho) = -(\rho^2 \ln [r + f(x)] + \|x - x_k^*\|^2).$$

The first-generation filled functions share a common feature: there are two adjustable parameters, r and ρ , which greatly affect the performance of the algorithms and need to be appropriately adjusted. However, how to adjust the parameters is a very difficult task. Due to these drawbacks, the second-generation filled functions were proposed, and they have only one parameter. Among them, the representative examples are Q -function [4] and R -function [29]:

$$Q(x, a) = -[f(x) - f(x_k^*)] \exp(a\|x - x_k^*\|^2), \quad (3)$$

where the adjustable parameter is a .

Consider

$$R(x, a) = \frac{1}{1 + \|x - x_k^*\|} \cdot \varphi_q(f(x) - f(x_k^*) + q), \quad (4)$$

where

$$\varphi_q(t) = \begin{cases} \exp\left(-\frac{q^3}{t}\right), & t \neq 0, \\ 0, & t = 0, \end{cases} \quad (5)$$

$$0 < q < \min_{x_k^* \in L(P) \setminus G(P)} (f(x_k^*) - f(x_g^*)),$$

and $x_g^* \in G(P)$; $L(P)$ and $G(P)$ stand for the set of local minimizers of $f(x)$ and global minimizers of $f(x)$, respectively.

These filled functions only have one parameter than those in the first generation. However, these functions are liable to be ill-conditioned in practice since their function values increase exponentially due to an exponential function. As the adjustable parameter becomes larger and larger, which is required by the FFM itself, the rapidly increasing exponential function value may result in an overflow in the computation. To overcome this drawback, H -function was proposed by Liu [27]:

$$H(x, a) = \frac{1}{\ln[1 + f(x) + f(x_k^*)]} - a\|x - x_k^*\|^2. \quad (6)$$

The H -function retains the advantage of the Q -function with only a single parameter and without exponential terms. The performance of the H -function in numerical experiments for a large set of testing functions was quite satisfactory [27]. $H(x, a)$ can be regarded as the third generation due to the absence of the exponential term. Nevertheless, the H -function has a drawback which is discontinuous at $x \in S = \{x \mid f(x) = f(x_k^*)\}$.

However, on the one hand, the continuity and differentiability on the FFM are required for the convergence analysis in theory [15]. On the other hand, computationally, most local minimization algorithms for the numerical nonlinear programming require the gradient information in their procedures (readers can refer to [30] or [31] for detail). Thus, it is very necessary to develop a continuously differentiable filled function with as few parameters as possible. There was already some work in this area ([8], e.g.), but the parameters of the filled functions are not easy to adjust. Based on this consideration, a continuously differentiable filled function with one parameter is designed, and the parameter, which can be as large as possible, is relatively easy to adjust.

3. A New Filled Function and Its Properties

Definition 2 relies on the concept of the basin and hill of $f(x)$, which requires that the minima in the operating region are isolated, and Definition 2 also requires that there exists a minimal point of $f(x)$ along a line. This is more difficult to be guaranteed. Therefore, many improvements in the definition are given in the literatures ([6, 14, 28], e.g.), which make it more convenient to construct a new filled function. In this section, we use the definition in [28] for Problem (P1).

Definition 3. Suppose x_k^* is a current local minimizer of $f(x)$. $P(x, x_k^*)$ is said to be a filled function of $f(x)$ at x_k^* , if it

satisfies the following properties:

- (i) x_k^* is a strictly maximizer of $P(x, x_k^*)$;
- (ii) for any $x \in \Omega_1$, one has $0 \notin \nabla P(x, x_k^*)$, and $\Omega_1 = \{x \in \Omega \mid f(x) \geq f(x_k^*), x \neq x_k^*\}$;
- (iii) if $\Omega_2 = \{x \mid f(x) < f(x_k^*), x \in \Omega\}$ is not empty, then there exists a point $x_k' \in \Omega_2$ such that x_k' is a local minimizers of $P(x, x_k^*)$.

Note that Definition 3 about the filled function is different from the definition mentioned in [3]. It is much easier to construct a new filled function by Definition 3, and the local optimal solution of the filled function can be easily found. For example, suppose that x_k^* is not a global minimizer, then, by condition (iii) of Definition 3, we can find a point $x_k' \in \Omega_2$ by minimizing $P(x, x_k^*)$. Therefore, we can obtain a lower local minimizer x_{k+1}^* of $f(x)$ by searching $f(x)$ starting at x_k' via local search schemes. In the process of minimizing $P(x, x_k^*)$, it does not require, unlike Definition 2, that x_{k+1}^* must be on the straight line through x_k' and x_k^* . So the design of the filled function is much easier and flexible.

In order to find a global minimizer of $f(x)$, the major issue of the filled function method is to find a lower minimizer of $f(x)$ or justify whether the obtained local minimizer is a global minimizer of $f(x)$. This heavily relies on the performance of the filled function used.

In this section, we propose a new filled function for Problem (P1) at a local minimizer x_k^* as follows:

$$P(x, x_k^*) = -\|x - x_k^*\|^2 g(f(x) - f(x_k^*)),$$

$$g(t) = \begin{cases} \frac{\pi}{2}, & t \geq 0, \\ r \cdot \arctan(t^2) + \frac{\pi}{2}, & t < 0, \end{cases} \quad (7)$$

where r is an adjustable positive real number as large as possible, used as the weight factor.

Note that the proposed filled function has some advantages: first, it has one parameter r which is a positive real number as large as possible, thus it is easy to adjust, and, second, it is continuously differentiable, which makes it more easily solved by the existing local optimization method; finally, $\arctan(t^2) \in [0, \pi/2)$ is bounded, which ensures that the calculation of $P(x, x_k^*)$ will not overflow and is of numerical stability. The following theorems show that $P(x, x_k^*)$ satisfies Definition 3.

Theorem 4. Suppose that x_k^* is a local minimizer of $f(x)$, then x_k^* is a strictly local maximizer of $P(x, x_k^*)$.

Proof. Since x_k^* is a local minimizer of $f(x)$, then there exists a small positive real number ε , and a neighborhood $\delta = U(x_k^*, \varepsilon)$, such that, for all $x \in \delta$, when $x \neq x_k^*$, then $f(x) > f(x_k^*)$, and $P(x, x_k^*) = -\pi/2\|x - x_k^*\|^2 < 0 = P(x_k^*, x_k^*)$. Thus, x_k^* is a strictly local maximizer of $P(x, x_k^*)$. \square

Theorem 4 clearly shows that $P(x, x_k^*)$ satisfies the property (i) of Definition 3.

Theorem 5. Suppose Assumption 1 is satisfied, x_k^* is a local minimizer of $f(x)$, and for any $x \in \Omega_1 = \{x/f(x) \geq f(x_k^*)\}$, $x \in \Omega$, $x \neq x_k^*$, one has $0 \notin \nabla P(x, x_k^*)$.

Proof. For any $x \in \Omega_1$, $f(x) \geq f(x_k^*)$, and for $x \neq x_k^*$, one has $\nabla P(x, x_k^*) = -\pi(x - x_k^*) \neq 0$. Consequently, $0 \notin \nabla P(x, x_k^*)$. \square

Theorem 6. Suppose Assumption 1 is satisfied, x_k^* is a local minimizer of $f(x)$, and $\Omega_2 = \{x/f(x) < f(x_k^*), x \in \Omega\}$ is not empty; then there exists a point $x'_k \in \Omega_2$, such that x'_k is a local minimizer of $P(x, x_k^*)$.

Proof. Let $\Omega_3 = \{x/f(x) \leq f(x_k^*), x \in \Omega\}$, $\partial\Omega_2 = \{x/f(x) = f(x_k^*), x \in \Omega\}$; then $\Omega_3 = \Omega_2 \cup \partial\Omega_2$, where $\partial\Omega_2$ is the boundary of Ω_2 (and Ω_3). Since $f(x)$ is continuous and both Ω_3 and $\partial\Omega_2$ are contained in Ω , then both Ω_3 and $\partial\Omega_2$ are bounded closed sets. For any $x \in \partial\Omega_2$,

$$P(x, x_k^*) = -\frac{\pi}{2} \cdot \|x - x_k^*\|^2. \quad (8)$$

Since $\partial\Omega_2$ is bounded closed set, then there exists $\tilde{x} \in \partial\Omega_2$, such that

$$\min_{x \in \partial\Omega_2} P(x, x_k^*) = P(\tilde{x}, x_k^*) = -\frac{\pi}{2} \cdot \|\tilde{x} - x_k^*\|^2. \quad (9)$$

Note that, for any $x \in \Omega_2$, $f(x) < f(x_k^*)$, and

$$P(x, x_k^*) = -\|x - x_k^*\|^2 \times \left[\frac{\pi}{2} + r \cdot \arctan(f(x) - f(x_k^*))^2 \right]. \quad (10)$$

Since Ω_2 is not empty, there exists a point $x_1 \in \Omega_2 = \text{int } \Omega_3$ such that

$$P(x_1, x_k^*) = -\|x_1 - x_k^*\|^2 \times \left[\frac{\pi}{2} + r \cdot \arctan(f(x_1) - f(x_k^*))^2 \right], \quad (11)$$

when

$$r > \frac{\pi}{2 \cdot \arctan(f(x_1) - f(x_k^*))^2} \left(\frac{\|\tilde{x} - x_k^*\|^2}{\|x_1 - x_k^*\|^2} - 1 \right) \triangleq \bar{r}, \quad (12)$$

for example, $r = \bar{r} + 1$; one has $P(x_1, x_k^*) < P(\tilde{x}, x_k^*)$. In other words, when $r = \bar{r} + 1$, at least there exists a point $x_1 \in \text{int } \Omega_3$, such that $P(x_1, x_k^*) < P(\tilde{x}, x_k^*)$. Since $P(x, x_k^*)$ is continuous on closed bounded set Ω_3 , $P(x, x_k^*)$ must have a global minimizer x'_k on Ω_3 for this fixed r . Note that $P(x'_k, x_k^*) \leq P(x_1, x_k^*) < P(\tilde{x}, x_k^*)$, and \tilde{x} is a global minimizer of $P(x, x_k^*)$ on $\partial\Omega_3$ (and $\partial\Omega_2$), then $x'_k \notin \partial\Omega_3$, and thus $x'_k \in \text{int } \Omega_3 = \Omega_2$, that is, x'_k is also a global minimizer of $P(x, x_k^*)$ on Ω_2 ; The proof is completed. \square

Theorems 4–6 state that the proposed filled function satisfies the conditions of Definition 3.

4. Filled Function Algorithm

4.1. A New Local Search Method: RULS. Conventional local optimization method minimizes function $f(x)$ directly from the initial point x'_k , and then a local minimizer x_k^* is obtained. In the process, the function is called repeatedly when the derivative and the search direction are calculated; in addition, the number of function evaluations is also increased in the one-dimensional search process. More importantly, with the increase of dimension, the number of iterations and the number of function evaluations are also increasing. Thus, the amount of computation for the traditional local search methods is very large, which would affect the computational efficiency. In this subsection, a new local search strategy called randomly and uniformly local search (RULS) is given, which can make the initial point closer to the local optimal solution, thereby the number of iterations and the number of function evaluations will be reduced, and the convergence speed will be accelerated.

Algorithm RULS. The details for RULS are as follows.

Step 1. $x'_k \in \Omega = [l, u] = \{x/l \leq x \leq u, l, u \in R^n\}$ is an initial point for the k th iteration, and M is the set of all points that have been used so far. Calculate $a = (\sum_{i=1}^{|M|} \|x'_k - x_i\|)/|M|$, $x_i \in M$, where $|M|$ denotes the elements number of the set M .

Step 2. $\Gamma \in \Omega$ is a neighborhood at x'_k , where $\Gamma = [\Gamma_l, \Gamma_u] = [x'_k - a/m \cdot e, x'_k + a/m \cdot e]$, $m > 1$, and $e = (1, \dots, 1)^T \in R^n$.

Set $\Gamma_l = x'_k - a/m \cdot e$, and $\Gamma_u = x'_k + a/m \cdot e$.

For $i = 1$ to n .

If $\Gamma_l(i) < l(i)$ then

$\Gamma_l(i) = l(i)$

Endif

If $\Gamma_u(i) > u(i)$ then

$\Gamma_u(i) = u(i)$

Endif

Endfor

Randomly and uniformly select $b * n$ points $Y_k = \{y_1, y_2, \dots, y_{bn}\}$ in Γ , where n is the dimension of the problem, and b is the number of points selected in each dimension, $M = \{M, Y_k\}$.

Step 3. Calculate the function value of these points $Y_k = \{y_1, y_2, \dots, y_{bn}\}$ to find the points \bar{Y} corresponding to the smallest function value, where

$$\bar{Y} = \underset{y_i \in Y_k, i \in \{1, 2, \dots, bn\}}{\operatorname{argmin}} \{f(y_1), f(y_2), \dots, f(y_{bn})\}. \quad (13)$$

Let $X_1 = \underset{x'_k, y \in \bar{Y} \cup x_k}{\operatorname{argmin}} \{f(x'_k), f(y)\}$ as a set of the initial points.

Step 4. Starting from any point $z_k \in X_1$, minimize $f(x)$ by using a local optimization method to obtain a local minimizer

sequence $X_2 = \{z_k^*\}$ and a local minimum value sequence $F_2 = \{f(z_k^*)\}$, set $M = \{M, X_2\}$.

Step 5. Calculate $f(x_k^*) = \min\{f(z_k^*)\}$ to obtain a local minimizer x_k^* , $M = \{M, x_k^*\}$.

Explanation of Algorithm RULS. uniformly and randomly generate some points near the initial point x'_k , select one of the points corresponding to the smallest function value as the new initial point to minimize the function $f(x)$, and obtain a local minimizer x_k^* . Diagram is shown in Figures 4(d₁), 4(d₂), and 4(d₃).

4.2. A New Filled Function Algorithm (NFFA). Based on the results of the previous content, a new filled function algorithm is proposed as follows.

Algorithm 1 (NFFA)

Step 1 (initialization step).

- Choose a tolerance $\varepsilon > 0$, for example, $\varepsilon := 1.0e - 3$.
- Choose a large integer constant B , a positive real number as large as possible r , and a small constant $\delta := 1.0e - 3$.
- Choose a positive integer $C < B$.
- Set $f(x_{k-1}^*) = +\infty$, and $k := 1$.

Step 2. Randomly and uniformly select $b * n$ points Y_k in the operating region, where $Y_k = \{y_1, y_2, \dots, y_{bn}\}$, $y_i \in \Omega$, $i \in \{1, 2, \dots, bn\}$, n is the dimension of the problem, and b is the number of points selected in each dimension. Calculate the function value of these points, and find the points corresponding to the smallest function value $\bar{Y} = \arg\min_{y_i \in Y_k, i \in \{1, 2, \dots, bn\}} \{f(y_1), f(y_2), \dots, f(y_{bn})\}$. Select for all $x_k \in \bar{Y}$.

Step 3. Starting from x_k , minimize $f(x)$ by using RULS to obtain a local minimizer x_k^* ; then go to Step 6.

Step 4. Construct a filled Function (7) at the local minimizer x_k^* , and Go to Step 5.

Step 5. Set $x = x_k^* + \delta e^i$, use x as the initial point to minimize $P(x, x_k^*)$ by using RULS, and find minimizers x' of $P(x, x_k^*)$. Let $x_{k+1} = x'$ and $k = k + 1$, and then go to Step 3.

Step 6 (termination step). If $k > C$ and $|f(x_k^*) - f(x_{k-C}^*)| \leq \varepsilon$, ($k = 1, 2, \dots$) or $k > B$, the algorithm will stop, and $x^* = x_k^*$ is taken as a global minimizer of $f(x)$; otherwise, increase r , and go to Step 4.

Some Explanations about the above Filled Function Algorithm

- In minimization to $f(x)$ and $P(x, x_k^*)$, a local optimization method has to be selected firstly. In our algorithm, the MATLAB function “fmincon” is used.
- In Step 5, the smaller δ has to be chosen accurately, in the algorithm; the δ is selected to guarantee that $\|\nabla P(x, x_k^*)\|$ is greater than a threshold.

(3) Step 5 means that if local minimizer x' of $P(x, x_k^*)$ is found in Ω with $f(x') < f(x_k^*)$, we can use x' as the initial point to minimize $f(x)$ and obtain a better local minimizer of $f(x)$.

(4) In Step 6, it is necessary to increase r , because the lower local minimizer may not be found when r is not large enough.

5. Numerical Experiments

5.1. Test Problems. In this section, the proposed algorithm is tested on Problem 1 and Problems 2–9 taken from [14].

Problem 1 (one-dimensional function). We have

$$\begin{aligned} \min \quad & f(x) = \sin x + \sin 2x - \cos 4x, \\ \text{s.t.} \quad & -2 \leq x \leq 4. \end{aligned} \quad (14)$$

The global minimizer is $x^* = -1.4523$, and the global optimal value is $f(x^*) = -2.1175$.

Problem 2 (two-dimensional function). We have

$$\begin{aligned} \min \quad & f(x) = [1 - 2x_2 + c \sin(4\pi x_2) - x_1]^2 \\ & + [x_2 - 0.5 \sin(2\pi x_1)]^2, \\ \text{s.t.} \quad & 0 \leq x_1 \leq 10, \quad -10 \leq x_2 \leq 0, \end{aligned} \quad (15)$$

where $c = 0.2, 0.5$, and 0.05 . The global minimum value is $f(x^*) = 0$ for all c .

Problem 3 (three-hump back camel function). We have

$$\begin{aligned} \min \quad & f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6} \cdot x_1^6 - x_1x_2 + x_2^2, \\ \text{s.t.} \quad & -3 \leq x_1 \leq 3, \quad -3 \leq x_2 \leq 3. \end{aligned} \quad (16)$$

The global minimizer is $x^* = (0, 0)^T$, and the global optimal value is $f(x^*) = 0$.

Problem 4 (six-hump back camel function). We have

$$\begin{aligned} \min \quad & f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} \cdot x_1^6 - x_1x_2 - 4x_2^2 + 4x_2^4, \\ \text{s.t.} \quad & -3 \leq x_1 \leq 3, \quad -3 \leq x_2 \leq 3. \end{aligned} \quad (17)$$

The global minimizers are $x^* = (-0.0898, -0.7127)^T$ and $x^* = (0.0898, 0.7127)^T$, and the global optimal value is $f(x^*) = -1.0316$.

Problem 5 (Treccani function). We have

$$\begin{aligned} \min \quad & f(x) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2, \\ \text{s.t.} \quad & -3 \leq x_1 \leq 3, \quad -3 \leq x_2 \leq 3. \end{aligned} \quad (18)$$

This problem has two global minimizers in total, which are $x^* = (0, 0)^T$ and $x^* = (-2, 0)^T$, and the global optimal value is $f(x^*) = 0$.

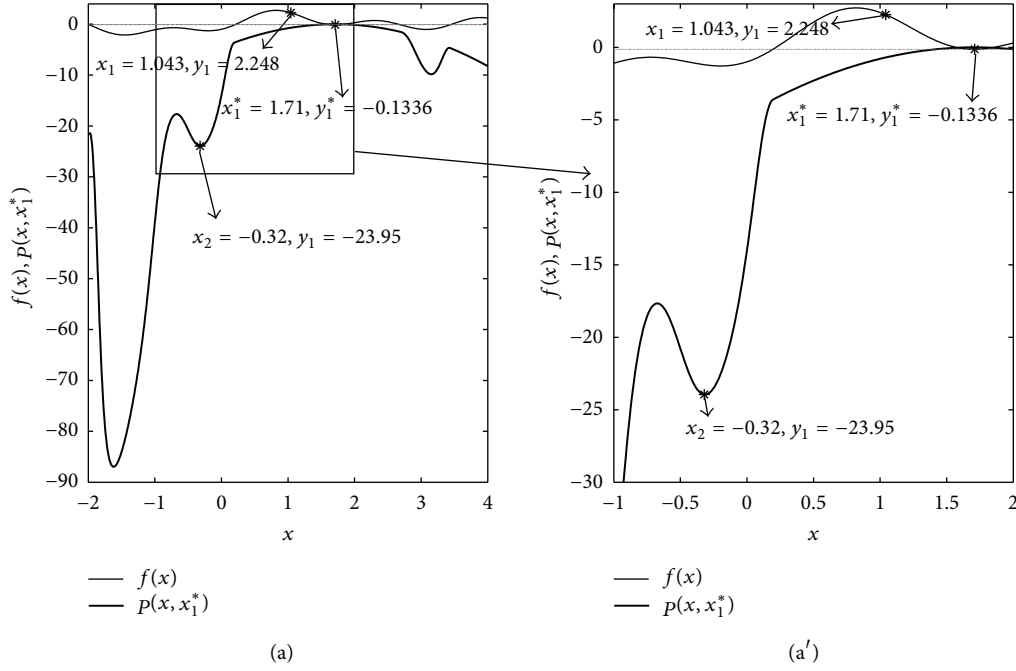


FIGURE 1: ((a) and (a')) The process of Steps (1), (2), and (3) in Section 5.2.1.

Problem 6 (Goldstein and Price function). We have

$$\begin{aligned} \min \quad & f(x) = g(x)h(x), \\ \text{s.t.} \quad & -3 \leq x_1 \leq 3, \quad -3 \leq x_2 \leq 3, \end{aligned} \quad (19)$$

where $g(x) = 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)$, and $h(x) = 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)$.

The global minimizer is $x^* = (0, -1)^T$ and the global optimal value is $f(x^*) = 3$.

Problem 7 (two-dimensional Shubert function). We have

$$\begin{aligned} \min \quad & f(x) = \left\{ \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \right\} \\ & \times \left\{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \right\}, \\ \text{s.t.} \quad & 0 \leq x_1 \leq 10, \quad 0 \leq x_2 \leq 10. \end{aligned} \quad (20)$$

This problem has two global minimizers in total, which are $x^* = (4.8581, 5.4829)^T$ and $x^* = (5.4829, 4.8581)^T$. The global minimum value is $f(x^*) = -186.7309$.

Problem 8 (Shekel's function). We have

$$\begin{aligned} \min \quad & f(x) = -\sum_{i=1}^5 \left[\sum_{j=1}^4 (x_j - a_{ij})^2 + c_i \right]^{-1}, \\ \text{s.t.} \quad & 0 \leq x_j \leq 10, \quad j = 1, 2, 3, 4, \end{aligned} \quad (21)$$

TABLE 1: The coefficients $a_{ij}, c_i, i = 1, 2, 3, 4, 5, j = 1, 2, 3, 4$.

| i | a_{i1} | a_{i2} | a_{i3} | a_{i4} | 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 |

where the coefficients $a_{ij}, i = 1, 2, 3, 4$, and $5, j = 1, 2, 3$, and 4 , are given in Table 1. The global minimizer is $x^* = (4.0000, 4.0001, 4.000, 4.0001)^T$ and the global optimal value is $f(x^*) = -10.1529$.

Problem 9 (n -dimensional function). We have

$$\min \quad f(x) = \frac{\pi}{n} \cdot [10 \sin^2 \pi x_1 + g(x) + (x_n - 1)^2], \quad (22)$$

$$\text{s.t.} \quad -10 \leq x_i \leq 10, \quad i = 1, 2, \dots, n,$$

where $g(x) = \sum_{i=1}^{n-1} [(x_i - 1)^2(1 + 10 \sin^2 \pi x_{i+1})]$. The global minimizer of this problem is $x^* = (1, \dots, 1)^T$ for all n .

5.2. The Experimental Results. In the following, the procedure for solving Problem 1 is given, respectively, by using the traditional filled function method and the proposed algorithm, and a comparison of these two algorithms is also given.

TABLE 2: The results obtained by the proposed algorithm.

| No. | $r(>)$ | q | x_k^* | f -mean | f -best | f -std | Ratio (%) |
|---------------------------------|--------|----------|--|-----------------|-----------------|-----------------|-----------|
| 2 ($c = 0.2$) ($n = 2$) | 100 | 78.5510 | (1.5909; -0.2703) (0.9997; -0.0005) (1.8784; -0.3458) (0.9820; -0.0565) | $3.6840e - 010$ | $6.2589e - 017$ | $1.8044e - 009$ | 100 |
| 2 ($c = 0.5$) ($n = 2$) | 100 | 67.5000 | (1; 0) (1.5872; -0.2606) (1.8973; -0.3005) | $1.0221e - 012$ | 0 | $2.0430e - 012$ | 100 |
| 2 ($c = 0.05$) ($n = 2$) | 100 | 80.7500 | (1.8513; -0.4021) (1.5975; -0.2874) | $2.9402e - 10$ | $2.8417e - 012$ | $2.5634e - 008$ | 100 |
| 3 ($n = 2$) | 100 | 67.9796 | (0; 0) | $4.7567e - 009$ | 0 | $1.2455e - 009$ | 100 |
| 4 ($n = 2$) | 100 | 84.2449 | (0.0898; 0.7127) (-0.0898; -0.7127) | -1.0316 | -1.0316 | $1.2574e - 007$ | 100 |
| 5 ($n = 2$) | 100 | 53.1224 | (0; 0) (-2; 0) | $3.5872e - 008$ | 0 | $3.5668e - 008$ | 100 |
| 6 ($n = 2$) | 1000 | 64.9796 | (0; -1) | 3.0000 | 3.0000 | $8.6282e - 008$ | 100 |
| 7 ($n = 2$) | 100 | 625.4898 | (4.8581; 5.4829) (5.4829; 4.8581) | -186.7309 | -186.7309 | $1.0480e - 007$ | 100 |
| 8 ($n = 2$) | 100 | 357.1429 | (0.0000; 0.0000) | -10.1529 | -10.1529 | $6.7489e - 010$ | 100 |
| 9 ($n = 2$) | 100 | 107.6939 | (1; 1) | $1.8548e - 007$ | 0 | $1.1284e - 007$ | 100 |
| 9 ($n = 3$) | 100 | 90.1429 | (1; 1; 1) | $5.6906e - 011$ | 0 | $2.5855e - 007$ | 95.92 |

TABLE 3: The results obtained by the proposed algorithm and the algorithm in [14].

| No. | The proposed algorithm | | | The algorithm of the literature [14] | | |
|---------------------------------|------------------------|--|----------|--------------------------------------|--|----------|
| | q | x_k^* | times | q | x_k^* | times |
| 2 ($c = 0.2$) ($n = 2$) | 78.5510 | (1.5909; -0.2703) (0.9997; -0.0005) (1.8784; -0.3458) (0.9820; -0.0565) | 0.341403 | 500.3 | (1.8784; -0.3458) | 1.097546 |
| 2 ($c = 0.5$) ($n = 2$) | 67.5000 | (1; 0) (1.5872; -0.2606) (1.8973; -0.3005) | 0.175545 | 641.5 | (1.0000; -2.2205e - 14) | 0.548026 |
| 2 ($c = 0.05$) ($n = 2$) | 80.7500 | (1.8513; -0.4021) (1.5975; -0.2874) | 0.548172 | 262.3 | (1.8513; -0.4020) | 0.726323 |
| 3 ($n = 2$) | 67.9796 | (0; 0) | 0.135797 | 564 | (0; 0) | 1.017142 |
| 4 ($n = 2$) | 84.2449 | (0.0898; 0.7127) (-0.0898; -0.7127) | 0.162335 | 551 | (0.0898; 0.7127) (-0.0898; -0.7127) | 0.453289 |
| 5 ($n = 2$) | 53.1224 | (0; 0) (-2; 0) | 0.350461 | 632 | (0; 0) | 0.613838 |
| 6 ($n = 2$) | 64.9796 | (0; -1) | 0.135588 | 852.7 | (0; -1) | 0.356531 |
| 7 ($n = 2$) | 625.4898 | (4.8581; 5.4829) (5.4829; 4.8581) | 0.126097 | 6468 | (5.4829; 4.8581) | 0.274596 |
| 8 ($n = 4$) | 357.1429 | (4.0000; 4.0001; 4.0000; 4.0001) | 0.425600 | 513 | (4.0000; 4.0001; 4.0000; 4.0001) | 0.701534 |
| 9 ($n = 2$) | 107.6939 | (1; 1) | 0.126054 | 663 | (1; 1) | 0.707344 |
| 9 ($n = 3$) | 90.1429 | (1; 1; 1) | 0.135507 | 624.7 | (1; 1; 1) | 0.212676 |
| 9 ($n = 7$) | 6028 | (1.0000; 1.0000; 1.0001; 1.0001; 1.0001; 1.0001; 1.0003) | 1.067376 | 14678 | (1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000) | 1.105676 |
| 9 ($n = 10$) | 6027 | (1.0000; 1.0001; 1.0000; 1.0000; 1.0003; 1.0000; 1.0000; 1.0000; 1.0000; 1.0001) | 0.123788 | 31351 | (1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000; 1.0000) | 0.282408 |

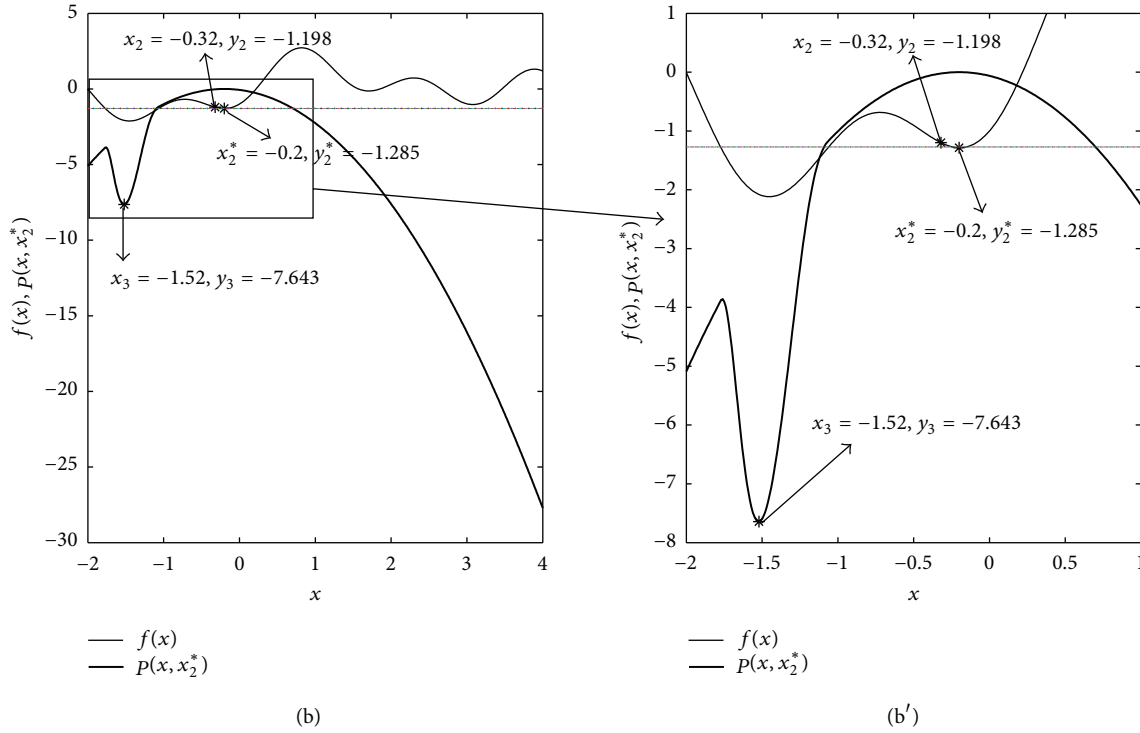


FIGURE 2: ((b) and (b')) The process of Step (4) in Section 5.2.1.

TABLE 4: The results obtained by the proposed algorithm for solving Problem 9 with different dimensions.

| No. | $r(>)$ | q | f -mean | f -best | f -std | Ratio (%) |
|----------------|------------|-----------|-----------------|-----------------|-----------------|-----------|
| 9 ($n = 7$) | 1000 | 6028 | $7.5895e - 008$ | $8.2743e - 008$ | 0 | 100 |
| 9 ($n = 10$) | 1000 | 6027 | 0.0012 | $8.3180e - 007$ | 0.0020 | 97.10 |
| 9 ($n = 12$) | $1.0e + 5$ | 641.0518 | $1.5597e - 005$ | $2.0935e - 010$ | $1.1098e - 008$ | 93.47 |
| 9 ($n = 15$) | $1.0e + 6$ | 972.1050 | $2.4116e - 005$ | $1.8709e - 013$ | $1.0032e - 005$ | 89.78 |
| 9 ($n = 20$) | $1.0e + 9$ | 1321.3011 | $1.1602e - 004$ | $8.2346e - 008$ | $5.0653e - 006$ | 85.57 |
| 9 ($n = 30$) | $1.0e + 9$ | 2666.1220 | 0.3012 | $5.3080e - 005$ | $3.0120e - 004$ | 89.87 |

5.2.1. *Conventional Filled Function Method for Solving Problem 1.* The procedure of the conventional filled function method is as follows.

- (1) Randomly select point $x_1 = 1.043$ in the operating region Ω ; then minimize $f(x)$ from x_1 by using local minimization function “fmincon” to obtain a local minimizer $x_1^* = 1.71$.
- (2) Construct a filled Function (7) at $x_1^* = 1.71$, where

$$f(x) = \sin x + \sin 2x - \cos 4x,$$

$$g(t) = \begin{cases} \frac{\pi}{2}, & t \geq 0, \\ 5 \cdot \arctan(t^2) + \frac{\pi}{2}, & t < 0. \end{cases} \quad (23)$$

- (3) Minimize $P(x, x_1^*)$ to obtain a local minimizer $x_2 = -0.32$ of $P(x, x_1^*)$; the specific processes of Steps (1), (2), and (3) are shown in Figure 1. Note that Figure 1(a') is to enlarge the rectangular box in

Figure 1(a), behind Figures 2(b') and 3(c'), and so on, also said this, therefore, not to repeat explained.

- (4) Minimize $f(x)$ from $x_2 = -0.32$ to obtain a local minimizer $x_2^* = -0.2$ of $f(x)$. Construct a filled function $P(x, x_2^*)$ at x_2^* , where $r = 5$, and then minimize the filled function $P(x, x_2^*)$ to obtain a local minimizer $x_3 = -1.15$ of $P(x, x_2^*)$ (as shown in Figure 2).
- (5) Minimize $f(x)$ from $x_3 = -1.15$ to obtain a local minimizer $x_3^* = -1.4523$. Construct a filled function $P(x, x_3^*)$ at x_3^* , where $r = 5$, which is shown in Figure 3, and then minimize $P(x, x_3^*)$. Without any stable point is found. Then the iteration terminates, and set $x^* = x_3^*$. The global minimizer $x^* = -1.4523$ and the global minimum value $f(x^*) = -2.1175$ are obtained.

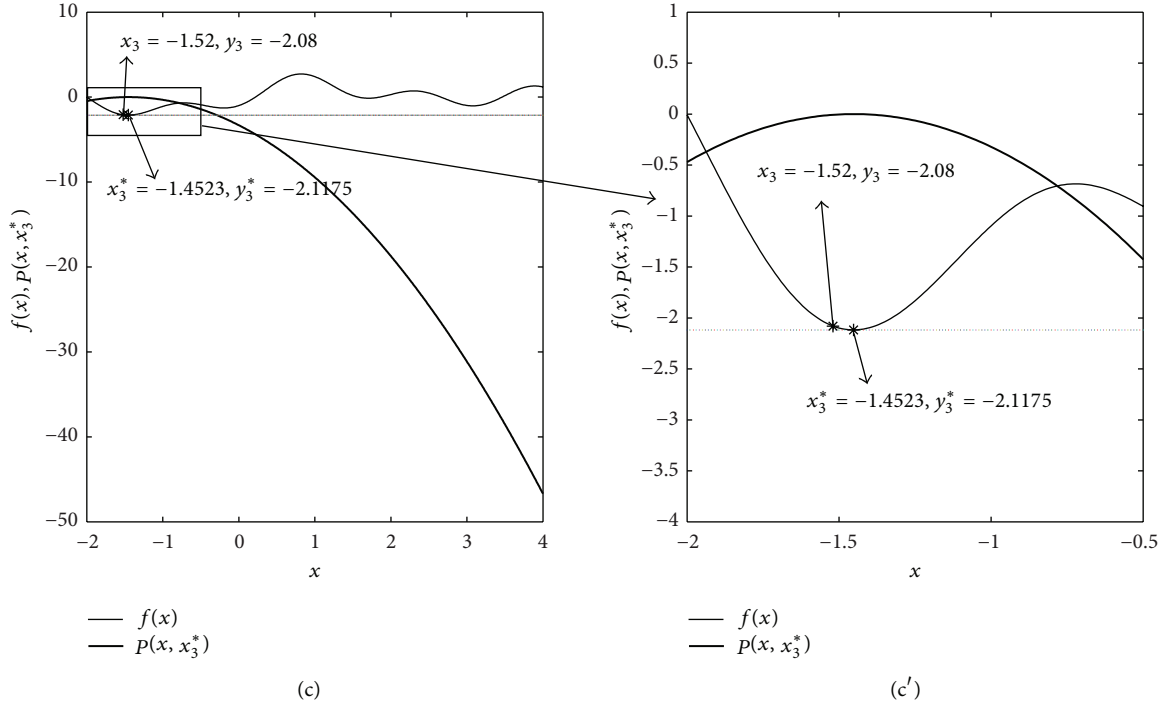
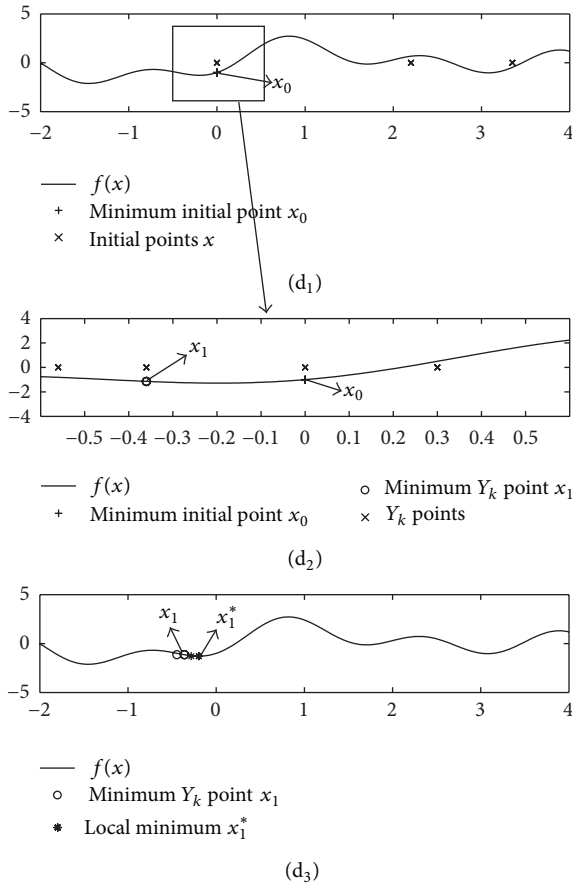


FIGURE 3: ((c) and (c')) The process of Step (5) in Section 5.2.1.

FIGURE 4: ((d₁), (d₂), and (d₃)) The process of Steps (1), (2), and (3) in Section 5.2.2.

Conventional filled function algorithm is run independently 50 times; the mean number of function evaluations is 18.7119.

5.2.2. The Proposed Filled Function Algorithm for Solving Problem 1. The procedure of the proposed filled function algorithm for solving Problem 1 is as follows.

- (1) Uniformly and randomly generate three points $x_{01} = 0$, $x_{02} = 2.2$, and $x_{03} = 3.35$ in the operating region, set $M = \{x_{01}, x_{02}, x_{03}\}$, and calculate the function value $f(x)$ of the three points. Find the point $x'_0 = 0$ corresponding to the smallest function value, as shown in (d₁) of Figure 4.

- (2) Calculate $a = \sum_{i=1}^{|M|} \|x'_0 - x_i\| / |M| = (\|0 - 2.2\| + \|0 - 3.35\|) / 2 = 2.775$ by using the rule in RULS, where $x_i \in M, i \in \{1, 2, \dots, |M|\}$; let $m = 5$, and then

$$\Gamma = \left[x'_0 - \frac{a}{m}, x'_0 + \frac{a}{m} \right] = [x'_0 - 0.555, x'_0 + 0.555]. \quad (24)$$

Randomly select three points Y_k in Γ , calculate their value of $f(x)$ to find the point x_1 corresponding to the smallest function value, as shown in (d₂) of Figure 4, and set $M = \{M, x_1\}$.

- (3) Minimize $f(x)$ from x_1 by using a local optimization method to find a local minimizer $x_1^* = -0.2$ and a minima $f(x_1^*) = -1.285$, as shown in (d₃) of Figure 4, and set $M = \{M, x_1^*\}$.

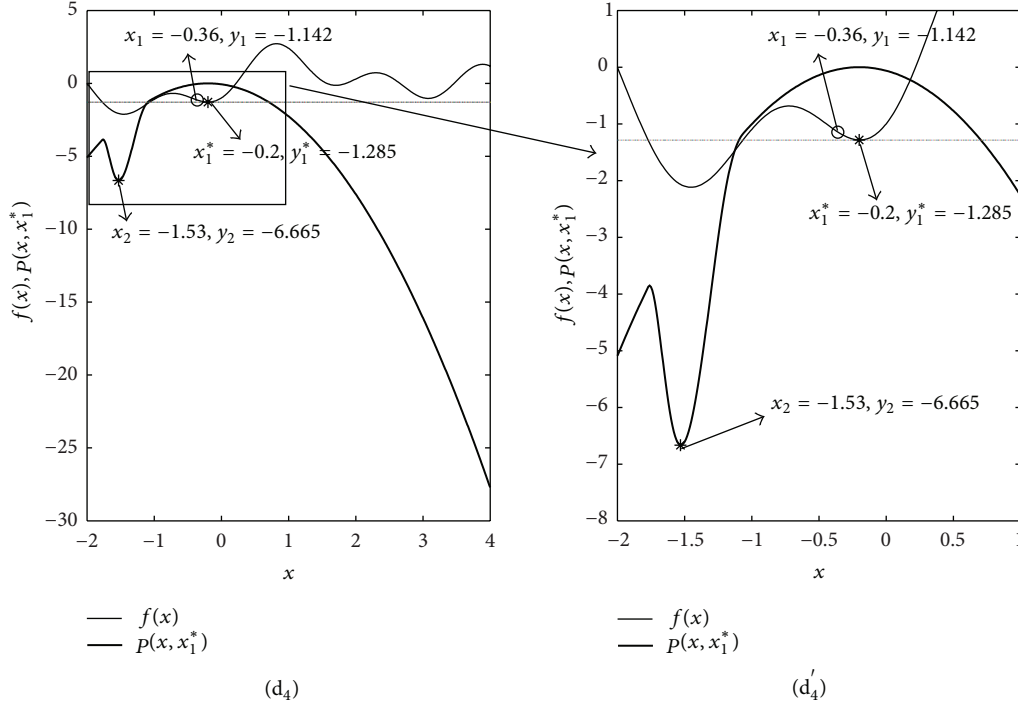


FIGURE 5: ((d_4) and (d_4')) The process of Step (4) in Section 5.2.2.

Steps (2) and (3) describe the specific process of RULS, as shown in (d_2) and (d_3) of Figure 4.

- (4) Construct a filled Function (7) at $x_1^* = -0.2$, as shown in Figure 5, where $f(x) = \sin x + \sin 2x - \cos 4x$,

$$g(t) = \begin{cases} \frac{\pi}{2}, & t \geq 0, \\ 4 \cdot \arctan(t^2) + \frac{\pi}{2}, & t < 0. \end{cases} \quad (25)$$

- (5) Optimize the function $P(x, x_1^*)$ at $x_1^* = -0.2$ by using RULS method to obtain a minimizer $x_2 = -1.53$ of $P(x, x_1^*)$, as shown in (d_4) of Figure 4. Set $x_0' = x_2$ and $M = \{M, x_2^*\}$, and then minimize $f(x)$ by using RULS to obtain $x_2^* = -1.4523$ and $f(x_2^*) = -2.1175$.
- (6) Construct a filled function $P(x, x_2^*)$ at x_2^* , where $r = 4$, as shown in Figure 6.
- (7) Function $P(x, x_2^*)$ without a stable point in $\{x \in \Omega \mid x \neq x_2^*\}$, so the iteration terminates. Set $x^* = x_2^*$. The global minimizer $x^* = -1.4523$ and the global minimum value $f(x^*) = -2.1175$ are obtained.

The proposed algorithm runs independently 50 times; the mean number of function evaluations is 13.6168, which is significantly less than the former result that uses conventional filled function method.

Both experimental results of Sections 5.2.1 and 5.2.2 and the mean number of function evaluations show that the proposed filled function method is more efficient.

5.2.3. The Proposed Filled Function Method for Solving the Problems Taken from [14]. The proposed algorithm is executed on the above test Problems 2–9. The results obtained by the proposed algorithm and the comparison with [14] are listed in Tables 2, 3, and 4.

The symbols used in Tables 2, 3, and 4 are given as follows:

No.: the number of the test problems;

n : the dimension of the test problems;

r : the parameter of the filled function;

k : the iteration number;

q : the total number of function and gradient evaluations of $f(x)$ and $P(x, x_k^*)$;

x_k^* : the local minimizer of the objective function in the k th iteration;

f_k^* : the function value at x_k^* ;

x^* : the global minimizer of the objective function;

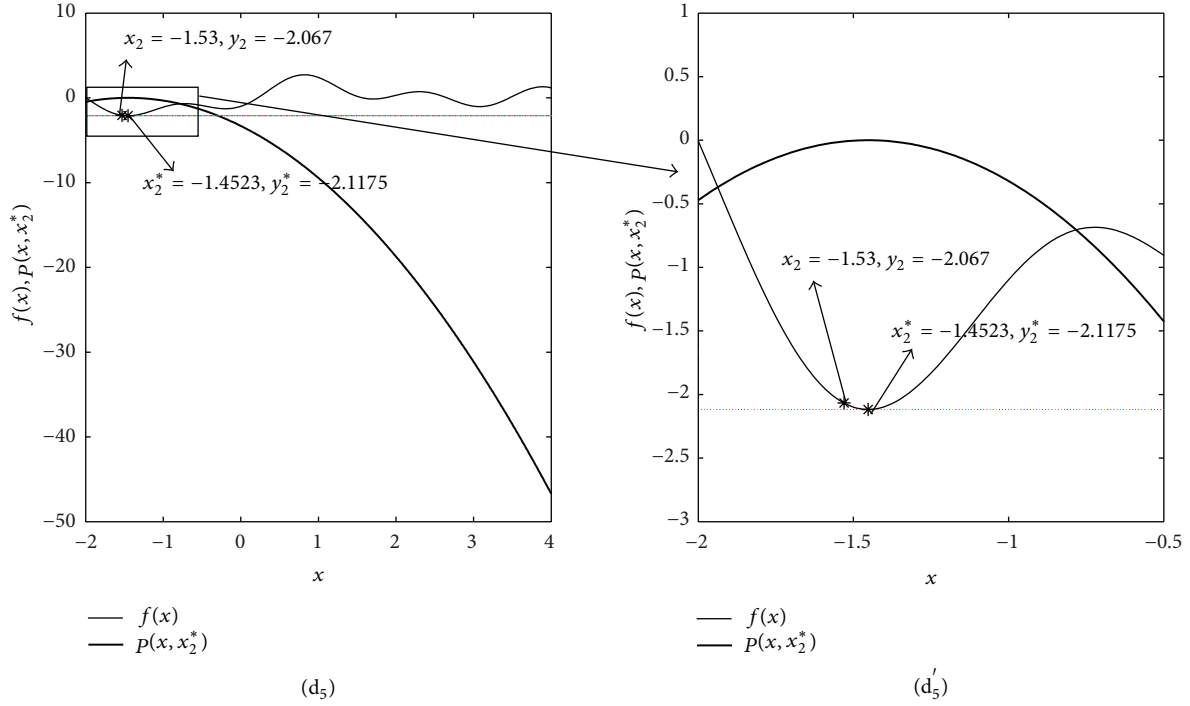
times: the once running time (second);

f -mean: the mean function value, respectively, in the 50 runs;

f -best: the best function value, respectively, in the 50 runs;

f -std: the standard deviation of function value, respectively, in the 50 runs;

ratio: the ratio of the successful runs, that is, the ratio of the runs finding the true optimal solution in the 50 runs.

FIGURE 6: ((d_5) and (d_5')) The process of Step (5) in Section 5.2.2.

It can be seen from Tables 2, 3, and 4 that the proposed algorithm has the following advantages.

(1) *More Minimizers Are Obtained.* In the proposed algorithm, the initial points are randomly and uniformly generated. This is helpful to get more different global minimizers in the 50 runs. For example, for Problems 2, 5, and 7, more optimal solutions are obtained by the proposed algorithm than those obtained by the algorithm in the literature [14].

(2) *More Effective and Efficient.* The optimal solutions of all test problems can be found by the proposed algorithm. This indicates the effectiveness of the proposed algorithm. In addition, it can be seen from Table 3 that both the mean number of function evaluations q and the once running time obtained by the proposed algorithm are smaller than that obtained by the algorithm in [14]. This also indicates the efficiency of the proposed algorithm.

(3) *More Stability.* It can be seen from column ratio in Table 2 that the ratio of the successful runs is more than 95.92%, and most of them are 100%, which shows that the proposed algorithm is stable. In addition, it can be seen from columns f -mean and f -std that difference between f -mean and f -best is small, and f -std is also small. This also indicates that the proposed algorithm is stable and robust to the initial points and parameter variation.

(4) *Applicable to Multidimensional Problems.* In Table 4, Problem 9 with different dimensions is tested, and the numerical results indicate that the proposed algorithm is suitable for solving multidimensional problems.

6. Conclusions

The filled function method is an approach to find the global minima of multi-modal functions. The existing filled functions have some drawbacks such as being nondifferentiable at some point in search domain, including the exponential term, containing more than one adjustable parameter, and being discontinuous. In this paper, a filled function with one parameter is designed, which is continuously differentiable, and it can overcome the existing shortcomings. Based on this, a new filled function method with randomly and uniformly local search scheme is proposed, and the algorithm is numerically stable. The results of the computer simulations indicate that the proposed filled function method is effective and efficient.

Acknowledgment

The paper is supported by The National Natural Science Foundation of China (no. 61272119).

References

- [1] P. Basso, "Iterative methods for the localization of the global maximum," *SIAM Journal on Numerical Analysis*, vol. 19, no. 4, pp. 781–792, 1982.
- [2] F. H. Branin, Jr., "Widely convergent method for finding multiple solutions of simultaneous nonlinear equations," *IBM Journal of Research and Development*, vol. 16, pp. 504–522, 1972.
- [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] R. P. Ge and Y. F. Qin, "A class of filled functions for finding global minimizers of a function of several variables," *Journal of Optimization Theory and Applications*, vol. 54, no. 2, pp. 241–252, 1987.
- [5] A.-F. Ling, C.-X. Xu, and F.-M. Xu, "A discrete filled function algorithm for approximate global solutions of max-cut problems," *Journal of Computational and Applied Mathematics*, vol. 220, no. 1–2, pp. 643–660, 2008.
- [6] W. Wang, Y. Shang, and L. Zhang, "A filled function method with one parameter for box constrained global optimization," *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 54–66, 2007.
- [7] S. He, W. Chen, and H. Wang, "A new filled function algorithm for constrained global optimization problems," *Applied Mathematics and Computation*, vol. 217, no. 12, pp. 5853–5859, 2011.
- [8] H. Lin, Y. Wang, and L. Fan, "A filled function method with one parameter for unconstrained global optimization," *Applied Mathematics and Computation*, vol. 218, no. 7, pp. 3776–3785, 2011.
- [9] Y. Zhang and Y.-T. Xu, "A one-parameter filled function method applied to nonsmooth constrained global optimization," *Computers & Mathematics with Applications*, vol. 58, no. 6, pp. 1230–1238, 2009.
- [10] J. A. Snymann and L. P. Fatti, "A multistart global minimization algorithm with dynamic search trajectories," *Journal of Optimization Theory and Applications*, vol. 54, no. 1, pp. 121–141, 1987.
- [11] S. Lucidi and V. Piccialli, "New classes of globally convexized filled functions for global optimization," *Journal of Global Optimization*, vol. 24, no. 2, pp. 219–236, 2002.
- [12] A. V. Levy and A. Montalvo, "The tunneling algorithm for the global minimization of functions," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 15–29, 1985.
- [13] 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.
- [14] L.-S. Zhang, C.-K. Ng, D. Li, and W.-W. Tian, "A new filled function method for global optimization," *Journal of Global Optimization*, vol. 28, no. 1, pp. 17–43, 2004.
- [15] X. Liu, "A class of continuously differentiable filled functions for global optimization," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 38, no. 1, pp. 38–47, 2008.
- [16] R. H. Mladineo, "An algorithm for finding the global maximum of a multimodal, multivariate function," *Mathematical Programming*, vol. 34, no. 2, pp. 188–200, 1986.
- [17] L. Bai, J. Y. Liang, C. Y. Dang, and F. Y. Cao, "A cluster centers initialization method for clustering categorical data," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8022–8029, 2012.
- [18] Y. Wang, "A uniform enhancement approach for optimization algorithms: smoothing function method," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 7, pp. 1111–1131, 2010.
- [19] R. Qing-dao-er-ji and Y. Wang, "A new hybrid genetic algorithm for job shop scheduling problem," *Computers & Operations Research*, vol. 39, no. 10, pp. 2291–2299, 2012.
- [20] Y. W. Leung, Y. Wang, Y. W. Leung, and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41–53, 2001.
- [21] Y. W. Leung, "Multiobjective programming using uniform design and genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 30, no. 3, pp. 293–304, 2000.
- [22] C. Dang, W. Ma, and J. Liang, "A deterministic annealing algorithm for approximating a solution of the min-bisection problem," *Neural Networks*, vol. 22, no. 1, pp. 58–66, 2009.
- [23] E. F. Campana, G. Liuzzi, S. Lucidi, D. Peri, V. Piccialli, and A. Pinto, "New global optimization methods for ship design problems," *Optimization and Engineering*, vol. 10, no. 4, pp. 533–555, 2009.
- [24] Z. Y. Wu, F. S. Bai, H. W. J. Lee, and Y. J. Yang, "A filled function method for constrained global optimization," *Journal of Global Optimization*, vol. 39, no. 4, pp. 495–507, 2007.
- [25] L. An, L.-S. Zhang, and M.-L. Chen, "A parameter-free filled function for unconstrained global optimization," *Journal of Shanghai University*, vol. 8, no. 2, pp. 117–123, 2004.
- [26] C. Gao, Y. Yang, and B. Han, "A new class of filled functions with one parameter for global optimization," *Computers & Mathematics with Applications*, vol. 62, no. 6, pp. 2393–2403, 2011.
- [27] X. Liu, "Finding global minima with a computable filled function," *Journal of Global Optimization*, vol. 19, no. 2, pp. 151–161, 2001.
- [28] Y. Zhang, L. Zhang, and Y. Xu, "New filled functions for non-smooth global optimization," *Applied Mathematical Modelling*, vol. 33, no. 7, pp. 3114–3129, 2009.
- [29] Y.-I. 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.
- [30] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1987.
- [31] M. Avriel, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1976.

