IJAMML 2:2 (2015) 125-145 June 2015
ISSN: 2394-2258
Available at http://scientificadvances.co.in

A NEW FILLED FUNCTION FOR FINDING GLOBAL MINIMIZER OF THE UNCONSTRAINED OPTIMIZATION PROBLEM

Hongwei Lin[a], Yuelin Gao[b], Xiaoli Wang[c] and Jing Chen[a]

[a]Department of Fundamental Courses, Jinling Institute of Technology, Nanjing, Jiangsu, 211169, P. R. China

[b]Institute of Information and System Science, Beifang University for Nationalities, Yinchuan, 750021, P. R. China

[c]School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071, P. R. China

---

Abstract

In this paper, a new filled function which is a continuously differentiable function with only one parameter is proposed. The new filled function eliminates some drawbacks of the traditional filled functions. Based on the new filled function, an algorithm for solving unconstrained global optimization problems is developed. The algorithm are implemented on some test problems, and the results show the effectiveness of the proposed filled function method.

*Corresponding author.

E-mail address: linhongwei_jt@hotmail.com (Hongwei Lin).

This work was supported by the National Nature Science Foundation of China (No. 11161001, No. 61203372, No. 61375121, No. 61402350), The Research Foundation of Jinling Institute of Technology (No. jit-b-201314, No. jit-n-201309), and Ningxia Foundation for key disciplines of Computational Mathematics.

2010 Mathematics Subject Classification: 90.
Submitted by Li Li.
Received March 28, 2015; Revised April 7, 2015

Keywords: global optimization, filled function method, global minimizer, local minimizer.

1. Introduction

Consider the following unconstrained optimization problem:

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

The problem (P) has wide applications in many social practices, such as engineering, finance, management, decision science and so on. In practical problems, most of the objective functions including multiple local minimizers which leads a great challenge to solve problem (P). For finding global minimizer of problem (P), 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 the recent years, many new theoretical and computational contributions have been reported for solving problem (P). The existing approaches can be classified into two categories: deterministic methods and probabilistic methods. The former including the filled function method (FFM) [1, 2, 3], the trajectory method [4, 5], the covering method [6, 7], and the tunneling method [8] and so on. The latter refers to the clustering method [9], and the methods reported in [10] and [11], the simulated annealing method [12], and the genetic algorithms [13, 14]. Furthermore, some hybrid deterministic and probabilistic algorithms [15, 16] were proposed to solve practical problems.

Among these methods, the FFM appears to have several advantages over others mainly due to its relatively easy actualization with a process that aims at successively finding smaller local minima. In this paper, we will focus our research on the FFM and mainly issue (1). The FFM was firstly proposed by Ge in [1, 2], which was used to find the global minimizer of problem (P). and, many scholars have also done a lot of valuable work to improve this method [3, 17, 18, 19, 20]. However, there are some drawbacks for the existing filled functions, such as more than one parameters needing to be controlled, sensitivity to the parameters and ill-conditioning. For example, the filled functions proposed in [1, 2, 20] contain exponential terms or logarithmic terms which will cause ill-condition problem if the parameter is not properly chosen; the filled functions proposed in [17, 18] are non-smooth functions to which the usual classical local optimization methods can not be used; the filled functions proposed in [1, 3, 19, 24] with more than one parameter, which are difficult to control. To overcome the above shortcomings, a new filled function with only one easy to control parameter is presented. It is a continuously differentiable function. Based on this new filled function, a new filled function method is proposed.

The remainder of this paper is organized as follows. The overview of the filled function methods and some basic knowledge are given in Section 2. In Section 3, a novel filled function is proposed and its properties are analyzed. In Section 4, a new filled function method is proposed and the numerical experiments on several test problems are conducted. Finally, some concluding remarks are drawn in Section 5.

2. Preliminaries

In this paper, we consider the problem (P):

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

where $f(x)$ is a twice continuously differentiable function on R^n and satisfies the condition

$$f(x) \to +\infty, \quad \text{as } \|x\| \to +\infty. \tag{1}$$

Then there exists a box $\Omega = \prod_{i=1}^{n}[l_i, u_i] \subset R^n$, whose interior contains all global minimizers of $f(x)$. Generally, we assume that Ω is known and $f(x)$ has only a finite number of minimizers in Ω. Then the problem (P) is equivalent to the following problem:

$$\min_{x \in \Omega} f(x). \tag{2}$$

So, we only consider problem (2) in the following. Additionally, the following symbols will be adopted.

k : the iteration number;

x_k : the initial point in the k-th iteration;

x_k^*: the local minimizer of $f(x)$ in the k-th iteration;

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

S_1 : the set defined by $S_1 = \{x | f(x) \geq f(x_k^*),\ x \in \Omega \setminus \{x_k^*\}\}$;

S_2: the set defined by $S_2 = \{x | f(x) < f(x_k^*),\ x \in \Omega\}$;

m: a constant defined by $m = \min_{i, j \in \{1, 2, \cdots, I\}, f(x_i^*) \neq f(x_j^*)} |f(x_i^*) - f(x_j^*)|$, where I is the number of minimizers of $f(x)$;

B_k^*: the basin of $f(x)$ at an isolated local minimizer x_k^*. It was first proposed by Ge [1].

Based on the above symbols and assumptions, the FFM was first proposed by Ge [1, 2] and undergoes the following generations. The examples of the FFM in the first generation are P-function [1] and G-function [2] which are listed as follows:

$$P(x, r, \rho) = \exp(-\|x - x_k^*\| / \rho^2) / (r + f(x)), \tag{3}$$

$$G(x, r, \rho) = -[\rho^2 \ln(r + f(x))] + \|x - x_k^*\|^p. \tag{4}$$

The P-function and G-function have a common feature, there are two adjustable parameters, r and ρ, which need to be appropriately iterated and coordinated. Because of this limitation, the second-generation filled functions were proposed. Among them, the best known is the Q-function [2] given by

$$Q(x, a) = -(f(x) - f(x_k^*)) \exp(a\|x - x_k^*\|^2). \tag{5}$$

The Q-function has only one adjustable parameter a, thus the algorithm is significantly than those in the first generation. However, the Q-function is liable to the exponential function when applied to global optimizations since its magnitude increases exponentially against parameter a. The lager a, which is required by the property of the FFM, the lager exponential may result in an overflow in the computation. To overcome this shortcoming, the H-function was proposed in [21] as follows:

$$H(x, a) = 1 / \ln(1 + f(x) - f(x_k^*)) - a\|x - x_k^*\|^2. \tag{6}$$

The H-function keeps the advantage of the Q-function that it has only one adjustable parameter and it does not include exponential term. The H-function can be regard as an example of the third-generation filled functions. Nevertheless, it is discontinues at the points which function value is equal to the one at x_k^*. Then, the most local minimization algorithm used in the filled function may lost efficiency, and the FFM will be failure. This leads to the fourth-generation (C-function) of the FFM [22]. Based on the thinking of the C-function, a new filled function which have the same properties as the C-function is constructed in this paper.

3. A New Filled Function and its Properties

The definition of the filled function is first introduced by Ge in [1]. Since the definition of the filled function was introduced, many variations of the definition of the filled function are given. In this paper, we adopt the revised definition of the filled functions as follows:

Definition 1. Suppose x_k^* is a current local minimizer of $f(x)$. A continuously differentiable function $FF(x)$ is said to be a filled function of $f(x)$ at x_k^*, if it satisfies the following properties:

(i) x_k^* is a strict local maximizer of $FF(x)$;

(ii) $FF(x)$ has no stationary point in the set S_1;

(iii) if x_k^* is not a global minimizer of $f(x)$, namely, S_2 is not empty, then there exists a point x_k' such that it is a local minimizer of $FF(x)$ on S_2.

Note that it is easier to construct a filled function by Definition 1 than by the definition in [1]. If x_k^* is not a global minimizer of $f(x)$, the by condition (p3) of Definition 1, a point $x_k' \in S_2$ will be found by minimizing $FF(x)$. This means that if we minimize $f(x)$ with initial point x', a better minimizer of $f(x)$ will be found.

In order to construct a new filled function, a functions of one variable is introduced firstly:

$$h_c(t) = \begin{cases} -1, & t \geq 0, \\ 2\left(\dfrac{2t^3}{c^3} + \dfrac{3t^2}{c^2} \right) - 1, & -c < t < 0, \\ 1, & t \leq -c, \end{cases}$$

where $c > 0$ is a constant. It is obvious that $h_c(t)$ is a continuous differentiable function over R. Based on function $h_c(t)$, we propose a novel filled function for problem (2) at a local minimizer x_k^* as follows:

$$FF(x, x_k^*, P) = \|x - x_k^*\|^2 h_P(f(x) - f(x_1^*)), \tag{7}$$

where $P > 0$ is a parameter. We can see that the formula (7) is continuously differentiable. The following theorems indicate that the function in (7) is a filled function by Definition 1.

Theorem 1. *Suppose* x_k^* *is a local minimizer of* $f(x)$ *and* $FF(x, x_k^*, P)$ *is defined by* (7), *then* x_k^* *is a strict local maximizer of* $FF(x, x_k^*, P)$ *on* Ω *for all* $P > 0$.

Proof. Since x_k^* is a local minimizer of $f(x)$, there exists $\delta > 0$, such that $f(x) \geq f(x_k^*)$ for any $x \in \Omega \cap U(x_k^*, \delta) \setminus \{x_k^*\}$. By (7), we have

$$FF(x, x_k^*, P) = -\|x - x_k^*\|^2 < 0 = FF(x_k^*, x_k^*, P). \tag{8}$$

Namely, x_k^* is a strict local maximizer of $FF(x, x_k^*, P)$ on Ω.

Theorem 2. *Suppose* x_k^* *is a local minimizer of* $f(x)$, *x is a point such that in set* S_1, *then x is not a stationary point of* $FF(x, x_k^*, P)$ *for all* $P > 0$.

Proof. By $x \in S_1$, one has $x \neq x_1^*$ and

$$FF(x, x_k^*, P) = -\|x - x_k^*\|^2, \nabla FF(x, x_k^*, P) = -2\|x - x_k^*\| \neq 0.$$

Namely, *x* is not a stationary point of $FF(x, x_1^*, P)$.

By the continuously differentiability of $FF(x, x_k^*, P)$ and definition of Ω, we know that $\forall y S1 \setminus \partial\Omega$ is not a local minimizer of $FF(x, x_k^*, P)$.

Theorem 3. *Suppose* x_k^* *is a local minimizer but not a global minimizer of* $f(x)$ *on* Ω, *it means that* S_2 *is not empty, then there exists a point* $x' \in S_2$ *such that* x' *is a local minimizer of* $FF(x, x_k^*, P)$ *when* $P < m$.

Proof. Since x_k^* is a local minimizer of $f(x)$, and x_k^* is not a global minimizer of $f(x)$, there exists an x^* which is an another local minimizer of $f(x)$ such that $f(x^*) < f(x_k^*)$.

By the definition of m and $P < m$, we have $f(x^*) < f(x_k^*) < -m < -P$. By the definition of the function h, one has $FF(x^*, x_k^*, P) = -\|x^* - x_k^*\|^2$ and $\nabla FF(x^*, x_k^*, P) = -2\|x^* - x_k^*\|$.

Take (x^*, x_k^*) as a direction, then $(x^*, x_k^*)^T \nabla FF(x^*, x_k^*, P) = 2\|x^* - x_k^*\|^2 > 0$.

Since x_k^* is a local minimizer of $f(x)$, there exists $\delta > 0$, such that $f(x) \geq f(x_k^*)$ for any $x \in \Omega \cap U(x_k^*, \delta) \setminus \{x_k^*\}$. Then, along the direction (x^*, x_k^*), there exists a point $y \in \Omega \cap U(x_k^*, \delta) \setminus \{x_k^*\}$, such that $FF(y, x_k^*, P) = -2\|y - x_k^*\|$. So $(x^*, x_k^*)^T \nabla FF(y, x_k^*, P) < 0$.

It means that the function value of $FF(x^*, x_k^*, P)$ is first decrease then increase. Since $FF(x, x_k^*, P)$ is continuously differentiable, thus a point x' which along the direction (x^*, x_k^*) is a minimizer of $FF(x, x_k^*, P)$. By Theorem 2, one has $FF(x', x_k^*, P) < FF(x_k^*, x_k^*, P) = 0$, namely, $x' \in S_2$.

From Theorems 2 and 3, we know that if x_k^* is not a global minimizer of $f(x)$ on Ω, there exists a local minimizer x' of $FF(x, x_k^*, P)$ on Ω which satisfies $x' \in S_2$ when parameter P is taken as small as possible. Meanwhile, we known that $FF(x, x_k^*, P) < 0$.

4. Filled Function Algorithm and Numerical Experiments

4.1. Filled function algorithm and some explanations

Based on the theorems and discussions in the above section, a filled function algorithm for finding a global minimizer of $f(x)$ and some explanations of the algorithm are given. The algorithm is described firstly.

The filled function algorithm

Step 0: Choose an lower bound Lbp of P (e.g., take it as 10^{-6}) and a constant $0 < \rho < 1$ (e.g., take it as $\rho = 0.1$); give the initial P (e.g., take it as 10), respectively; some directions d_i, $i = 1, 2, \cdots, k\,(k \geq 2n)$ are given in advance, where d_i almost uniformly distribute over the unit sphere $\{x| \, \|x\| = 1\}$, n is the dimension of the optimization problems. Set $k := 1$, and choose a point $x_k \in S$.

Step 1: Minimize $f(x)$ starting from the initial point $x_k \in \Omega$ and obtain a minimizer x_k^* of $f(x)$.

Step 2: Construct

$$FF(x, x_k^*, P) = \|x - x_k^*\|^2 h_P(f(x) - f(x_1^*))$$

set $i = 1$.

Step 3: If $i \leqslant 2n$, then set $x = x_k^* + \delta d_i$ and go to Step 4; otherwise, go to Step 5.

Step 4: Use x as the initial point to minimize $FF(x, x_k^*, P)$ and denote the sequence points generated by a local optimization algorithm as x_j, $j = 1, 2, \cdots$. If $\exists_{j_0} \in \{1, 2, \cdots\}$ such that $x_{j_0} \notin \Omega$, set $i = i + 1$ and go to Step 3; otherwise, find a minimizer x' of $FF(x, x_k^*, P)$ and set $x_{k+1} = x'$, $k = k + 1$, go to Step 1.

Step 5: If $P > Lbp$, then decrease P by setting $P := \rho \times P$ and go to Step 2; otherwise, the algorithm stops and x_k^* is taken as a global minimizer of problem (P).

Some explanations about the above filled function algorithm are necessary.

(1) In minimization of $f(x)$ and $FF(x, x_k^*, P)$, a local optimization method needs to be selected firstly. In our algorithm, the quasi-Newton (BFGS) method is chosen to minimize $f(x)$ and $FF(x, x_k^*, P)$.

(2) In Step 3, δ needs to be selected carefully. A large δ may cause losing the better solution of the original problem, while a small δ may cause the local optimization to fail to progress in the minimization of $FF_p(x)$. In our algorithm, δ is selected to guarantee that $\|\nabla FF(x, x_k^*, P)\|$ is greater than a threshold (e.g., take δ as 0.01). For specific problems, the selection of δ is related to the number of minimizers of the objective function and the size of the feasible region. The fewer minimizers of the objective function and the larger size of the feasible region, the larger δ should be.

(3) Step 4 means that if a local minimizer x' of $FF(x, x_k^*, P)$ 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 $f(x)$.

4.2. Numerical experiments

In this section, the proposed algorithm is tested on some benchmark problems taken from the literature [23].

Problem 1. (Two-dimensional function)

$$\min f(x) = [1 - 2x_2 + c\sin(4\pi x_2) - x_1]^2 + [x_2 - 0.5\sin(2\pi x_1)]^2,$$

$$\text{s.t. } 0 \le x_1 \le 10, -10 \le x_2 \le 0,$$

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

Problem 2. (Three-hump back camel function)

$$\min f(x) = 2x_1^2 - 1.05x_1^4 + \frac{1}{6}x_1^6 - x_1x_2 + x_2^2,$$

$$\text{s.t. } -3 \le x_1 \le 3, -3 \le x_2 \le 3.$$

The global minimizer is $x^* = (0, 0)^T$.

Problem 3. (Six-hump back camel function)

$$\min f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 - x_1x_2 - 4x_2^2 + 4x_2^4,$$

$$\text{s.t.} -3 \le x_1 \le 3, -3 \le x_2 \le 3.$$

The global minimizer is $x^* = (-0.0898, -0.7127)^T$ or $x^* = (0.0898, 0.7127)^T$.

Problem 4. (Treccani function)

$$\min f(x) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2,$$

$$\text{s.t.} -3 \le x_1 \le 3, -3 \le x_2 \le 3.$$

The global minimizers are $x^* = (0, 0)^T$ and $x^* = (-2, 0)^T$.

Problem 5. (Goldstein and Price function function)

$$\min f(x) = g(x)h(x),$$

$$\text{s.t.} -3 \le x_1 \le 3, -3 \le x_2 \le 3,$$

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 $x^* = (0, -1)^T$.

Problem 6. (Two-dimensional Shubert function)

$$\min f(x) = \{\sum_{i=1}^{5} i\cos[(i+1)x_1] + i\}\{\sum_{i=1}^{5} i\cos[(i+1)x_2] + i\},$$

$$\text{s.t. } 0 \le x_1 \le 10, 0 \le x_2 \le 10.$$

This problem has 760 minimizers in total. The global minimum value is $f(x^*) = -186.7309.$

Problem 7. (Shekel's function)

$$\min f(x) = -\sum_{i=1}^{5}[\sum_{j=1}^{4}(x_j - a_{i,j})^2 + c_i]^{-1},$$

$$\text{s.t. } 0 \le x_j \le 10, \quad j = 1, 2, 3, 4,$$

where the coefficients $a_{i,j}, c_i, i = 1, 2, 3, 4, 5, j = 1, 2, 3, 4$ are given in the Table 1.

Table 1. The coefficient for Problem 7

i	$a_{i,1}$	$a_{i,2}$	$a_{i,3}$	$a_{i,4}$	c_i
1	4.0	4.0	4.0	4.0	0.1
2	1.0	1.0	1.0	1.0	0.2
3	8.0	8.0	8.0	8.0	0.3
4	6.0	6.0	6.0	6.0	0.4
5	3.0	7.0	3.0	7.0	0.5

All local minimizers are approximately equal to $(a_{i,1}\ a_{i,2}\ a_{i,3}\ a_{i,4})^T$ with function value $-1/c_i, i = 1, 2, 3, 4, 5$.

Problem 8. (n-dimensional function)

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

$$\text{s.t. } -10 \le x_j \le 10, \quad i = 1, 2, \cdots, 4,$$

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 $x^* = (1, \cdots, 1)$ for all n.

The proposed algorithm is executed on the above 8 test problems and the performance is compared with that of the algorithm in [22]. The minimizers obtained by the above two algorithms are listed in Table 2 to Table 16. In these tables, we adopt the following symbols:

x_0: A initial point;

The initial value of P is taken as 10 and Lbp is taken as 10^{-6} for all problems;

NFFM: The proposed algorithm in this paper;

NFA: The algorithm proposed in [23].

Table 2. Results for Problem 1 with $c = 0.2$, $x_0 = (6, -2)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(5.7221, -1.8806)^T$	2.5070	$(5.7221, -1.8806)^T$	2.5070
2	$(3.7387, -1.2649)^T$	0.6165	$(4.7387, -1.7417)^T$	1.6212
3	$(1.8786, -0.3458)^T$	1.0871e-007	$(4.7096, -1.3985)^T$	1.3566
4			$(3.7387, -1.2649)^T$	0.61647
5			$(2.7380, -0.78836)^T$	0.088673
6			$(1.8784, -0.34585)^T$	0

Table 3. Results for Problem 1 with $c = 0.5$, $x_0 = (0, 0)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(0.0420, -0.0948)^T$	0.5175	$(0.042023, -0.094772)^T$	0.51745
2	$(1.0568, 0.1746)^T$	1.6489e-010	$(0.99991, -1.2524\text{e-}4)^T$	2.2389e-7
3			$(1.0000, -2.2205\text{e-}14)^T$	0

Table 4. Results for Problem 1 with $c = 0.05,\ x_0 = (10, -10)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(8.7299, -3.2965)^T$	9.0733	$(8.7299, -3.2965)^T$	9.0733
2	$(7.7280, -0.8347)^T$	60.6031	$(7.7280, -2.8347)^T$	6.5031
3	$(6.7248, -2.3724)^T$	4.3943	$(6.7248, -2.3724)^T$	4.3943
4	$(5.7198, -1.9162)^T$	2.7434	$(5.7198, -1.9162)^T$	2.7534
5	$(4.7129, -1.4891)^T$	1.5351	$(4.7129, -1.4891)^T$	1.5351
6	$(2.7300, -0.7934)^T$	0.1022	$(3.7305, -1.2306)^T$	0.611844
7	$(1.0000, -0.0000)^T$	4.1270e-012	$(2.7300, -0.79341)^T$	0.10216
8			$(1.8513, -0.40209)^T$	0

Table 5. Results for Problem 2 with initial point $(-2, -1)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(-1.7476, -0.8738)^T$	0.2986	$(-1.7476, -0.87378)^T$	0.29864
2	$(-0.0001, -0.0001)^T$	1.0379e-008	$(0, 0)^T$	0

Table 6. Results for Problem 2 with initial point $(2, 1)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(1.7476, 0.8738)^T$	0.2986	$(1.7476, 0.87378)^T$	0.29864
2	$(0.0001, 0.0001)^T$	1.5137e-008	$(0, 0)^T$	0

Table 7. Results for Problem 3 with initial point $(-2, 1)^T$

	NFFM		NFA	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(-1.6071, 0.5687)^T$	2.1043	$(-1.6071, 0.56865)^T$	2.1043
2	$(0.0898, 0.7127)^T$	-1.0316	$(0.089842, 0.71266)^T$	-1.0316

Table 8. Results for Problem 3 with initial point $(2, -1)^T$

	NFFM		NFA	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(1.6071, -0.5687)^T$	2.1043	$(1.6071, -0.56865)^T$	2.1043
2	$(-0.0898, -0.7127)^T$	-1.0316	$(-0.089842, -0.71266)^T$	-1.0316

Table 9. Results for Problem 3 with initial point $(-2, -1)^T$

	NFFM		NFA	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(1.7036, -0.79608)^T$	-0.21546	$(1.7036, -0.79608)^T$	-0.21546
2	$(-0.0898, -0.7127)^T$	-1.0316	$(-0.089842, -0.71266)^T$	-1.0316

Table 10. Results for Problem 4 with initial point $(-1, 0)^T$

	NFFM		NFA	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(-1.0000, 0)^T$	1.0000	$(-1.0000\ 0)^T$	1.0000
2	$(-0.0000, -0.0000)^T$	9.8600e-017	$(0, 0)^T$	0

Table 11. Results for Problem 5 with initial point $(-1, -1)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(-0.6000, -0.4000)^T$	30.0000	$(-0.60000, -0.40000)^T$	30.000
2	$(0.0000, -1.0000)^T$	3.0000	$(0, -1.0000)^T$	3.0000

Table 12. Results for Problem 6 with initial point $(1, 1)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(2.0467, 2.0467)^T$	0	$(1.0865, 1.0865)^T$	2.8841e-17
2	$(3.2800, 4.8581)^T$	-46.511	$(1.3200, 1.8703, e-12)^T$	-13.052
3	$(4.2760, 4.8581)^T$	-79.411	$(1.3200, 4.8581)^T$	-37681
4	$(5.4892, 4.8581)^T$	-186.7309	$(3.2800, 4.8581)^T$	-46.511
5			$(4.2760, 4.8581)^T$	-79.411
6			$(5.4892, 4.8581)^T$	-186.73

Table 13. Results for Problem 7 with initial point $(1, 1, 1, 1)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(1.0001, 1.0002, 1.0001, 1.0002)^T$	-5.0552	$(1.0001, 1.0002, 1.0001, 1.0002)^T$	-5.0552
2	$(4.0000, 4.0000, 4.0000, 4.0000)^T$	-10.1529	$(4.0000, 4.0001, 4.0000, 4.0001)^T$	-10.153

Table 14. Results for Problem 7 with initial point $(6, 6, 6, 6)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(5.9987, 6.0002, 5.9987, 6.0002)^T$	– 2.6822	$(5.9987, 6.0003, 5.9987, 6.0003)^T$	– 2.6829
2	$(4.0000, 4.0001, 4.0000, 4.0001)^T$	– 10.1529	$(7.9996, 7.9996, 7.9996, 7.9996)^T$	– 5.1008
3			$(4.0000, 4.0001, 4.0000, 4.0001)^T$	– 10.153

Table 15. Results for Problem 8 with $n = 7$ and initial point $(2, 2, 2, 2, 2, 2, 2)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000)^T$	2.3538e-013	$(1.9899, 1.9897, 1.9896, 1.9896 1.9896, 1.9896, 1.9898)^T$	– 3.1095
2			$(1.0000, 1.0000, 1.0000, 1.0000 1.0000, 1.0000, 1.0000)^T$	0

Table 16. Results for Problem 8 with $n = 10$ and initial point $(6, 6, 6, 6, 6, 6, 6, 6, 6, 6)^T$

	NFFM		*NFA*	
k	x_k^*	f_k^*	x_k^*	f_k^*
1	$(0.0101, 0.0103, 0.0103, 0.0104, 0.0103,$ $0.0102, 1.0000, 6.0000, 6.0000, 6.0000)^T$	2.6653	$(5.9490, 5.9979, 5.9980, 5.9980, 5.9980$ $5.9980, 5.9980, 5.9980, 5.9980, 5.9980)^T$	78.432
2	$(1.1615, 1.1651, 0.4418, 0.9258, 0.9638,$ $-0.4809, 0.9926, 6.0000, 6.0000, 6.0000)^T$	2.4443	$(-1.9696, 5.9943, 5.9980, 5.99805.9980$ $5.9980, 5.9980, 5.9980, 5.9980, 5.9980)^T$	73.450
3	$(1.9900, 1.0000, 1.0000, 1.0000, 1.0000,$ $1.0000, 1.0000, 6.0000, 6.0000, 6.0000)^T$	0.4443	$(-0.97956, 5.9871, 5.9980, 5.9980, 5.9980$ $5.9980, 5.9980, 5.9980, 5.9980, 5.9980)^T$	71.884
4	$(1.0000, 1.0000, 1.0000, 1.0000, 1.0000,$ $1.0000, 1.0000, 1.0000, 1.0000, 1.0000)^T$	0	$(0.012709, 5.9476, 5.9979, 5.9980, 5.9980$ $5.9980, 5.9980, 5.9980, 5.9980, 5.9980)^T$	70.890
5			$(1.0000, 1.0000, 1.0000, 1.0000, 1.0000,$ $1.0000, 1.0000, 1.0000, 1.0000, 1.0000)^T$	0

From Table 2 to Table 16, we can see that for all test problems, both the proposed algorithm (NFFM) and the algorithm proposed in [23] (NFA) can find the global optimal solutions for all test problems. However, the NFFM used no more iterations to find the global optimal solution. In particular, for test Problem 1 in all three cases, for Problem 6 and Problem 8, the MFFM used fewer iterations to find the optimal solutions than the NFA. For example, for Problem 1 in the case that $c = 0.05$ and $x_0 = (10, -10)$ shown in Table 4, the NFFM only needs 7 iterations to find the global optimal solution, but the NFA needs 8 iterations to find the global optimal solution. For Problem 6, there are 760 local minimizers, and the NFFM only needs 4 iterations to find the global optimal solutions, but the NFA needs 6 iterations to find the global optimal solution. Thus, the proposed algorithm is more efficient than the algorithm in [23].

5. Concluding Remarks

The filled function method is a popular approach for the global optimization. Existing filled functions have some drawbacks such as being non-differentiable at some point in search domain, containing more than one parameter, ill-conditioning and so on. In this paper, a continuously differentiable filled function with one parameter is presented. It can overcome these shortcomings to a certain degree. From the numerical experiments, we can see that the proposed algorithm is reliable and efficient.

References

[1] R. Ge, A filled function method for finding a global minimizer of a function of several variables, Mathematical Programming 46 (1990), 191-204.

[2] R. Ge, The theory of filled function method for finding global minimizers of nonlinearly constrained minimization problems, Journal of Computational Mathematics 5 (1987), 1-9.

[3] Xian Liu and Wilsun Xu, A new filled function applied to global optimization, Computer and Operation Research 31 (2004), 61-80.

[4] F. H. Branin, Widely convergent methods for finding multiple solutions of simultaneous nonlinear equations, IBM Journal of Research and Development 16 (1972), 504-522.

[5] J. A. Snyman and L. P. Fatti, A multi-start global minimization algorithm with dynamic search trajectories, Journal of Optimization Theory and Applications 54 (1987), 121-141.

[6] P. Basso, Iterative methods for the localization of the global maximum, SIAM Journal on Numerical Analysis 19 (1982), 781-792.

[7] R. H. Mladineo, An algorithm for finding the global maximum of a multimodal, multivariate function, Mathematical Programming 34 (1986), 188-200.

[8] A. V. Levy and A. Montalvo, The tunneling algorithm for the global minimization of functions, SIAM Journal on Scientific and Statistical Computing 6 (1985), 15-29.

[9] 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 39 (2012), 8022-8029.

[10] N. Baba, Global optimization of functions by the random optimization method, Int. J. Control 30 (1979), 1061-1065.

[11] C. Dorea, Limiting distribution for random optimization methods, SIAM J. Control Optim. 24 (1986), 76-82.

[12] C. Dang, W. Ma and J. Liang, A deterministic annealing algorithm for approximating a solution of the min-bisection problem, Neural Networks 22 (2009), 58-66.

[13] Y. W. Leung and Y. Wang, An orthogonal genetic algorithm with quantization for global numerical optimization, IEEE Transactions on Evolutionary Computation 5 (2001), 41-53.

[14] Y. W. Leung, Multiobjective programming using uniform design and genetic algorithm, IEEE Transactions on Systems, Man and Cybernetics C 30 (2000), 293-304.

[15] R. Storn and K. Price, Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997), 341-359.

[16] 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 10 (2009), 533-555.

[17] X. L Wang and G. B. Zhou, A new filled function for unconstrained global optimization, Applied Mathematics and Computation 174 (2006), 419-429.

[18] Yong Jun Wang and Jiang She Zhang, A new constructing auxiliary function method for global optimization, Mathematical Computer Modelling 47 (2008), 1396-1410.

[19] X. Liu, The barrier attribute of filled functions, Applied Mathematics and Computation 149 (2004), 641-649.

[20] Hongwei Lin and Huirong Li, A new filled function with one parameter to solve global optimization, Open Journal of Optimization 4 (2015), 10-20.

[21] Xian Liu, Finding global minima with a computable filled function, Journal of Global Optimization 19 (2001), 151-161.

[22] Xian Liu, A class of continuously differentiable filled functions for global optimization, IEEE Transactions on Systems, Man and Cybernetics-Part A: System and Humans 38 (2008), 38-47.

[23] Lian-Sheng Zhang, Chi-Kong Ng, Duan Li and Wei-Wen Tian, A new filled function method for global optimization, Journal of Global Optimization 28 (2004), 17-43.

[24] Fei Wei, Yuping Wang and Hongwei Lin, A new filled function method with two parameters for global optimization, Journal of Optimization Theory and Applications 163 (2014), 510-527.

■