A Novel Parameter-Free Filled Function Applied to Global Optimization

Ridwan Pandiya, Salmah*, Widodo, Irwan Endrayanto

Abstract—**In this paper, we mainly propose a new parameter-free filled function, which derives from two inverse trigonometric functions. First, we accomplish the analytical studies, to prove that filled function proposed in this paper possess all the properties of filled function. Secondly, based on this parameter-free filled function, a comparatively new algorithm was built. Finally, we implemented the algorithm to solve unconstrained global optimization problems. The computational results demonstrated in this paper revealed the effectiveness of the proposed filled function compared to some related results of the filled functions in the literature.**

Index Terms—**Global optimization, filled function method, nonlinear programming, global minimizer, auxiliary function approach.**

I. Introduction

A wide attention to the topic of global optimization has led many scholars to come up with new methods. The global optimization methods are divided into metaheuristic and deterministic approaches. Metaheuristic algorithm deals with the black box optimization problem (see for example [1], [2], [3]). On the other hand, problems with clearer mathematical structures are usually solved by the deterministic approach. Modified objective function method is one typical example of the deterministic approach. This technique modifies the objective function to find a new initial point once the local minimizer has been located. Tunneling function [4] and filled function [5] are both representation of a modified function approach or better known as an auxiliary function approach. Those two methods have been proven to be reliable in finding the global minima of a multidimensional function. Tunneling function method applies two phases alternately in its algorithm: minimizing and tunneling step. The prior is to minimize the objective function and the later is to find the zero point of the defined modified function. Shortly after, many researchers found that phase 2 in the tunneling algorithm was not easy, because we deal with how to obtain the zero point of function with variables. This obstacle then was improved by filled function method.

Manuscript received July 27, 2020; revised December 4, 2020.

This work was supported by Lembaga Pengelola Dana Pendidikan (LPDP), Indonesia.

R.Pandiya is a PhD candidate of the Department of Mathematics, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia (e-mail: ridwanpandiya@mail.ugm.ac.id). He is now with the Department of Informatics, Institut Teknologi Telkom Purwokerto, Purwokerto, Indonesia (e-mail: ridwanpandiya@ittelkom-pwt.ac.id).

*Salmah (corresponding author) is a professor of Mathematics Department, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia (e-mail: syalmah@yahoo.com).

Widodo is a professor of Mathematics Department, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia (email: widodo_math@ugm.ac.id)

I. Endrayanto is a senior lecturer of Mathematics Department, Universitas Gadjah Mada, Yogyakarta 55281, Indonesia (email: endrayanto@ugm.ac.id).

Newton's method, steepest descent method, conjugate gradient method, BFGS method, and so on have only the capability to obtain local minimizer. However, it does not mean that those local minimum methods cannot be exploited to locate the global one. Filled function method involves local downhill routine in its algorithm. The mechanism of the filled function method is as follows.

1) Minimizing the objective function starting from any point in its feasible domain;
2) Creating the auxiliary function (we call it filled function) of the objective function at a local minimizer obtained in the first step
3) Minimizing the filled function;
4) Using the minimizer that is obtained in the third step as an initial point and returning to the first step;

If we run the algorithm until the predetermined stopping criterion is met, the global minimizer of the cost function can lastly be obtained.

Filled function method is the original work of Renpu Ge. It firstly appeared in [5]. From [5], we know that many filled functions can be formed as long as they satisfy three conditions stated in the filled function definition. One original and specific filled function was given in [5]. Unfortunately, some conditions could be met by the two modifiable parameters in its function during the computer iteration. If the intended conditions are not satisfied by those parameters, the global optimization problem can be unsolvable. One other issue that is lacking in Ge's filled function is the involvement of the exponential term. This transcendent function makes the graph of Ge's filled function is almost flat when the feasible domain is huge [6]. When it happens, the minimizer of the filled function obtained is nothing but the pseudo minimizer. Filled functions in [6] attempted to reduce the number of parameter and the effect of the exponential function by providing 7 new filled functions. However, exponential and logarithmic functions and parameter are still incorporated. Consequently, those filled functions underwent the same deficiencies as filled function in [5] did.

One parameter filled functions have been widely conceived. Both theory and applications are developed. For example, in [7], a new filled function definition is offered. They replaced the third property of the filled function in [5]. Instead of requiring the existence of the minimizer of its filled function, they compared the value of the filled function in two regions. By this novel property, they claimed that it is easier to find a point in a basin that was lower than a basin that contained a minimizer found so far (the definition of basin can be studied in [5]). A new class of parametric filled function was also proposed in [7]. This filled function was discontinuous at some points. The aforementioned property resulted in limited use of the local downhill methods to minimize the filled function. Other filled functions that have one parameter in its function can be studied in ([8], [9], [10], [11]). Although the computational experiences perform that the methods are able to solve the global optimization problems, the parameter requires updating during the iterations. This sensation has the possibility that the global optimization process will stop at a local minimizer. The Ge's algorithm that has previously been mentioned is simplified by algorithm given in [12] because the filled function in [12] has the same minimizer of the objective function. The property of filled function given in [12] allows us to only minimize the cost function once. After that, we only need to minimize filled function simultaneously, and this increases the effectiveness the algorithm. However, the numerical results show that the algorithm provided in [12] requires more iteration to obtain the less accurate global minimum.

Filled function method was not only intended to solve the smooth global optimization without any restrictions, but also can be employed to solve any other optimization problems. For instance, filled functions in [13] and [14] are proposed to find global minimizer of the constrained global optimization problem, where all the constraints are inequality. However, filled function in [13] includes three piecewise functions and a parameter. So, the form of the function is quite complicated. Filled function can also be applied to nonlinear equations, for example, filled functions given in [15] and [16]. They solved nonlinear equations by first converting it into global optimization problem. Nevertheless, parameter in their filled functions becomes a barrier in the algorithm. The competencies of the filled function have also been proved to be reliable for solving non-smooth constrained or unconstrained global optimization (see [17], [18], [19]). The unique way of utilizing filled function method is shown in [20]. In that literature, the original multi-dimensional objective function is transformed into one dimensional function by using smoothing filled function. Based on this filled function, they proposed a new algorithm. Although this method promises a new way and evidenced to be dependable to solve unconstrained global optimization problems, but the new algorithm is complex.

It had been discussed in [5] that the ideal filled function is function that does not contain parameter. Based on this fact, parameter-free filled function was initially given in [21] and is defined as follows:

$$G\left(x, x_m^*\right)=-s\left(h(x)-h\left(x_m^*\right)\right)\left\|x-x_m^*\right\|^2, \qquad (1)$$

where $s: R \rightarrow R$ is a sign function

$$s(t)=\begin{cases}1, & t \geq 0 \\ -1, & t<0\end{cases},$$

and x_m^* is the firstly obtained local minimizer of the objective function h. From Equation (1), it is clear that the value of (1) changes at $x=x_m^*$. This discontinuous point is used as a stopping criterion in the algorithm to minimize Equation (1). Consequently, the effective well-established gradient-based local downhill procedures cannot be implemented. This reduces the plus point of Equation (1). The ones proposed in [22] come up with a relatively new parameter-free filled function. The formula is as follows:

$$G\left(x, x_m^*\right)=-s\left(h(x)-h\left(x_m^*\right)\right) \arctan \left(\left\|x-x_m^*\right\|^2\right), \qquad (2)$$

with s is a sign function as in Equation (1). Even though the use of $arctan$ function in Equation (2) prevents the overflow effect in the computational experiment, the ineffectiveness caused by the discontinuous point will be experienced by Equation (2).

The drawbacks faced by Equations (1) and (2) are then addressed by continuously differentiable filled functions suggested in [23] and [24]. The formulas are expressed in Equations 3 and 4 respectively.

$$G\left(x, x_m^*\right)=-\left\|x-x_m^*\right\|^2 \varpi\left(h(x)-h\left(x_m^*\right)\right), \qquad (3)$$

with $\varpi: R \rightarrow R$ is a single variable function,

$$\varpi(t)=\begin{cases}1, & t \geq 0 \\ -\exp\left(t^2\right)+2, & t<0\end{cases}$$

$$G\left(x, x_m^*\right)=\frac{1}{1+\left\|x-x_m^*\right\|^2} \varpi\left(h(x)-h\left(x_m^*\right)\right), \qquad (4)$$

with $\varpi: R \rightarrow R$ is

$$\varpi(t)=\begin{cases}\pi/2 & t \geq 0 \\ \pi/2-\arctan\left(t^2\right) & t<0\end{cases}.$$

Equation (3) was successful in finding global maxima of 13 problems. They have done 10 runs for each problem. However, it can be seen from its numerical results, that filled function in (3) is inefficient. This can mainly be seen from the number of iterations, function evaluation, and the successful rates. Exponential function was one of the factors causing it. Equation 4 is free from the exponential term and has been proven to be reliable for solving the given optimization problems. Both filled functions are claimed to fulfill three conditions requested by the filled function definition. However, we examine that filled functions in [23] and [24] fail to satisfy the third condition of the filled function definition. We will prove this through a counterexample, which will be explained in section 3.

After analyzing the filled functions that currently exist, this paper attempts to propose a new parameter-free filled function. This function derives from the combination of two inverse trigonometry functions. The study of the properties of the proposed filled function, including discussion of a particular condition that must be met by the intended filled function to qualify as a filled function, will be studied comprehensively. The implementation to some unconstrained global optimization problems have been done. To ensure that our filled function is better than those in the literature, some comparison has also been accomplished.

The rest of this paper is systematized as follows: Some definition and assumption supporting this study are given in Section 2. Section 3 provides a new filled function and theorems that are the properties of the proposed filled function. Implementation to some optimization problems will be done in Section 4. Comparison and discussion are proposed in Section 5. Finally, Section 6 gives some conclusion.

II. Preliminaries

This section provides some explanation of the problem that will be solved and its equivalency. Some assumptions and definitions are also given in this part. Moreover, a study of a counterexample of continuously differentiable parameter-free filled function will be offered in this section.

A. Basic Knowledge

This paper aims to solve the continuous global optimization problem without any constraints. It means that the proposed method is intended to find a global minimizer of the objective function h in the entire region of R^n. The expression can be written as follows:

$$CGO \quad \begin{cases} \min h(x) \\ x \in R^n \end{cases},$$

where h is continuously differentiable.

The CGO problem seems promising, although in the implementation step, it is almost impossible to be solved without some assumptions. Basically, we need to convert CGO into another solvable problem. In order to accomplish the conversion, in this paper, we assume that h is globally convex. By this special property, the existence of a set ψ in which the entire local minimizers of h are all the interior points of ψ can be ensured. Therefore, CGO problem can be altered into the following global optimization problem

$$BCGO \quad \begin{cases} \min h(x) \\ x \in \psi \end{cases}.$$

Other assumption, besides those already mentioned, that supports this study is that $BCGO$ has isolated finite minimizers. It means that the minimizers of h are possible to be infinite, but the number of different local minimum value is always finite.

From [5], we know that a function can be classified as a filled function of h at a converged local minimizer x_m^* if: (1) x_m^* is on the highest position of the hill of the filled function, (2) the filled function has no minimizers in the basins higher than the basin of h at x_m^*, and (3) if h has another lower minimizer than x_m^*, then there exists x' that minimizes the filled function on the line through x' and x_m^*. From criterion (3), we can conclude that, even if the Ge's filled function has minimizer, it is not guaranteed that the minimizer holds $h(x') < h(x_m^*)$. That condition is needed to make sure that the minimization process of h from x' yields better minimizer than x_m^*. Paper in [25] improves filled function definition, which is then widely used.

Definition 2.1: [25]. Function $G : R^n \to R$ is a filled function of the objective function $h : R^n \to R$ at x_m^* (a local minimizer of h), if G holds these conditions.

1) x_m^* is a strict local maximizer of G
2) G is clear from stationary points in

$$\psi_a = \{h(x) - h(x_m^*) \geq 0 | x \in \psi \backslash \{x_m^*\}\}$$

3) G has minimizer in

$$\psi_b = \{h(x) - h(x_m^*) < 0 | x \in \psi\},$$

if x_m^* is not a global minimizer of h.

In this study, we use Definition 2.1 to classified the function that is proposed to be categorized as a filled function. The following notations will be used in this paper:
(1) $\psi_a = \{h(x) - h(x_m^*) \geq 0 | x \in \psi \backslash \{x_m^*\}\}$
(2) $\psi_b = \{h(x) - h(x_m^*) < 0 | x \in \psi\}$
(3) M: is the set of all the minimizers of h
(4) $\lambda = \|x - x_m^*\|$
(5) $\lambda_1 = \|x_1 - x_m^*\|$
(6) $\lambda_2 = \|x_2 - x_m^*\|$
(7) $\beta = h(x) - h(x_m^*)$.
(8) $\beta_2 = h(x_2) - h(x_m^*)$.

B. Counterexample

The filling properties (three conditions given in Definition 2.1) fulfillment owned by the parametric filled functions in the literature is controlled by the adjustable parameter(s) in its functions. Because continuously differentiable filled functions in Equation (3) and (4) do not involve parameters, this makes those functions have no controller for satisfying all the properties mentioned in Definition 2.1. The exception is for lower-semi-continuous parameter-free filled function as in [21] and [22]. Theorem 2.3 in [23] and [24] stated that their filled function has minimizer x' in ψ_b if ψ_b is not empty (the third condition of Definition 2.1). The theorem is not always be fulfilled by Equation (3) and (4).

To prove that filled functions in [23] and [24] do not meet the third condition of Definition 2.1, now consider the following objective function:

$$h(x) = 0.1 \cos(5\pi x) + x^2. \tag{5}$$

Function (5) is globally convex (coercive) and has finite number of minimizers, they are $x_1^* = -0.5505$, $x_2^* = -0.1849$, $x_3^* = 0.1849$, and $x_4^* = 0.5505$, where the minimum values are $h(x_1^*) = 0.2318$, $h(x_2^*) = -0.0630$, $h(x_3^*) = -0.0630$, $h(x_4^*) = 0.2318$ respectively. The points x_2^* and x_3^* are the global minimizers (see Figure 1).

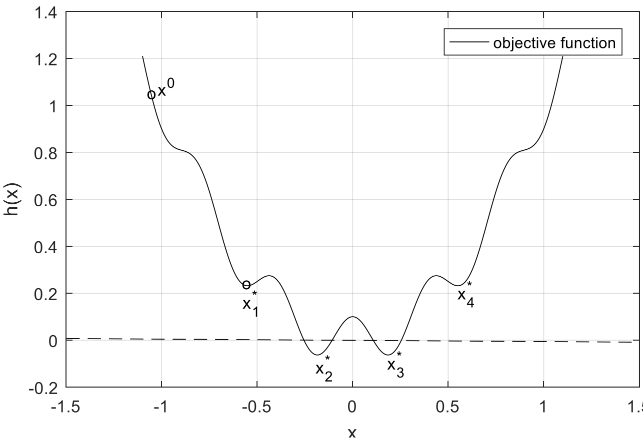

Fig. 1: The Graph of Equation (5).

If we move from initial point x^0 by using any local minimization method, we will attain the first local minimizer x_1^*. A filled function of $h(x)$ is then built at x_1^*. Figures 2 and 3 are the geometrical interpretation of Liu's filled function in [23] and Ahmed's filled function in [24], respectively. Both filled functions have no minimizer in the entire feasible

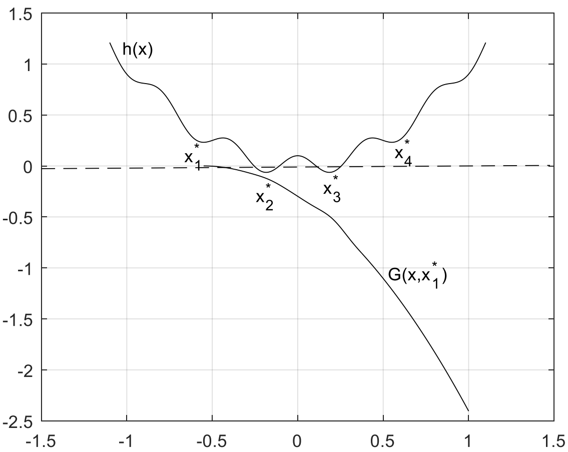

Fig. 2: The Graph of Equation (3).

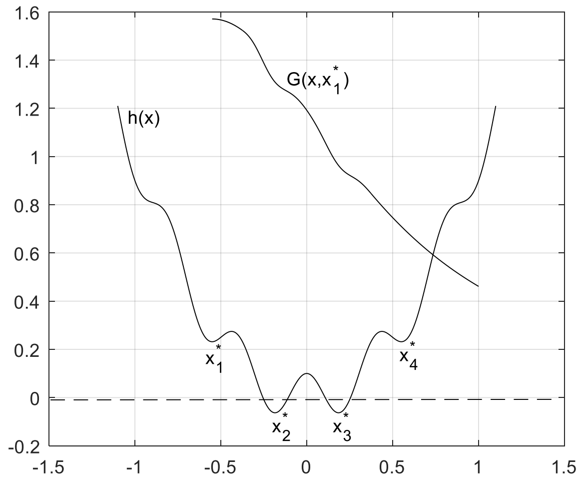

Fig. 3: The Graph of Equation (4).

domain. Therefore, if we implement the filled function algorithm, the global minimizer of $h(x)$ will fail to be localized.

Based on this reality, continuously differentiable parameter-free filled function requires a specific condition to categorized as a filled function. We provide the intended condition in Equation (9) and is proved in Theorem 3.6.

III. A New Filled Function and Its Properties

Conditions 2 and 3 of Definition 2.1 are difficult to be fulfilled by two parameter filled functions, due to two adjustable parameters in its function. To address this difficulty, Liu in [26] recommended the impelling function. The idea is to construct a hill in the region ψ_b, so that the minimization process of the filled function in ψ_a, which is a concave function, starting from the neighborhood of x_m^*, will be halted by the hill. Based on this reason, the minimizer of the filled function is the point near or equal to x' where $h(x') = h(x_m^*)$. The character of concave impelling function given in [26] is then adopted by the filled function given in [23]. However, as already mentioned in Section 1, the exponential term causes some troubles in the numerical experiment. Furthermore, [26] stated that the concavity property of a function may bring some undesirable consequences in the numerical minimization. This property, by the way, is owned by the parameter-free filled function given in [23]. These facts motivated us to propose a new type of filled function. Our filled function is free from any parameters and is convex in the region ψ_a. Our new filled function is defined as follows:

$$G(x, x_m^*) = -\arcsin\left(\frac{\lambda^\alpha}{1+\lambda^\alpha}\right) \times \left[\arctan\left((\min[0,\beta])^2\right) + 1\right], \quad (6)$$

where α is an even positive integer.

Remark 1: Without loss of generality, in this paper, we take $\alpha = 2$. From Equation (6),

$$G(x, x_m^*) = -\arcsin\left(\frac{\lambda^2}{1+\lambda^2}\right), \quad (7)$$

$\forall x \in \psi_a$, and

$$G(x, x_m^*) = -\arcsin\left(\frac{\lambda^2}{1+\lambda^2}\right)\left[\arctan\left(\beta^2\right) + 1\right], \quad (8)$$

$\forall x \in \psi_b$.

Before some theorems are given, we provide a lemma. This lemma reveals that the value of G in Equation (6) is negative for all $x \in \psi$. This property relates to the monotonically increasing behavior of the $arcsin$ function. The lemma is as follows.

Lemma 3.1: If $x_m^* \in M$, then $G(x, x_m^*) \leq 0$, $\forall x \in \psi$.

Proof: If $x \in \psi_a$, then it follows Remark 1, the value of $G(x, x_m^*)$ is Equation (7). We know that the value of $\frac{\lambda^2}{1+\lambda^2}$ contains in the interval $(0, 1)$. The $arcsin$ function has monotonic increasing property and $\arcsin(0) = 0$. So, $\arcsin\left(\frac{\lambda^2}{1+\lambda^2}\right)$ is always positive. Consequently, Equation (7) is negative. On the other hand, if $x \in \psi_b$, the value of $G(x, x_m^*)$ is Equation (8). Because $arctan$ is monotone increasing and $\arctan(0) = 0$. So, $\arctan\left(\beta^2\right) + 1 > 1$. This implies $G(x, x_m^*) < 0$ for all $x \in \psi_b$. The conclusion is $G(x, x_m^*) < 0$, $\forall x \in \psi = \psi_a \cup \psi_b$.

■

The property of $\arcsin(0) = 0$ implies that $G(x, x_m^*) = 0$ holds when $x = x_m^*$. Theorem below provides to guarantee that the property (1) of Definition 2.1 is fulfilled.

Theorem 3.2: If $x_m^* \in M$, then x_m^* is a maximizer of G.

Proof: Since $x_m^* \in M$, there exists $B_\sigma(x_m^*) = \{x \in R^n | \|x_m^* - x\| < \sigma\}$ where $h(x) \geq h(x_m^*)$, $\forall x \in B_\sigma(x_m^*) \cap \psi$. Because $h(x) \geq h(x_m^*)$, then $B_\sigma(x_m^*) \subseteq \psi_a$. It follows Remark 1, the value of filled function is Equation (7). From Lemma 3.1, we know that $G(x, x_m^*) < 0$. Because $G(x_m^*, x_m^*) = 0$, so, $G(x, x_m^*) < G(x_m^*, x_m^*)$. This proves that x_m^* is a maximizer of G.

■

Filled function algorithm uses x_m^* as a starting guess to minimize filled function. We can use any local minimization methods. However, if gradient-based procedure is used, x_m^* cannot be directly exploited because the gradient vector of G at x_m^* is a zero vector. So, the point close to x_m^* will be worked as an initial point. Our proposed filled function is convex in the region ψ_a. This can be intuitively seen if it is drawn for function with one variable. The next theorem is another advantageous property of the proposed filled function.

Theorem 3.3: Let $x_m^* \in M$. If $x_1, x_2 \in \psi_a$ such that $\|x_1 - x_m^*\| < \|x_2 - x_m^*\|$, then $G(x_1, x_m^*) > G(x_2, x_m^*)$.

Proof: Let x_1 and x_2 be two points arbitrarily taken from ψ_a. From Remark 1, $G(x_1, x_m^*) = -\arcsin\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right)$ and $G(x_2, x_m^*) = -\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)$. The difference between $G(x_1, x_m^*)$ and $G(x_2, x_m^*)$ is as follows:

$$\begin{aligned} G(x_1, x_m^*) - G(x_2, x_m^*) &= -\arcsin\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right) \\ &\quad + \arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right). \end{aligned}$$

Since $\|x_1 - x_m^*\| < \|x_2 - x_m^*\|$, then

$$\frac{\lambda_2^2}{1+\lambda_2^2} > \frac{\lambda_1^2}{1+\lambda_1^2} > 0.$$

We know that $arcsin$ function is monotonically increasing function. Therefore, $G(x_1, x_m^*) - G(x_2, x_m^*) > 0$.

■

Theorem 2 assures that the iteration process of minimization of the proposed filled function will run successfully, unless there exists a stationary point in ψ_a. Fortunately, we have the following theorem that guarantees the situation is not going to happen.

Theorem 3.4: Let $x_m^* \in M$. If $x_1 \in \psi_a$, then $\nabla G(x_1, x_m^*) \neq 0$.

Proof: Let x_1 be arbitrary taken from ψ_a. From Remark 1, we have

$$G(x_1, x_m^*) = -\arcsin\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right).$$

The vector gradient of G at x_1 is as follows:

$$\begin{aligned} \nabla G(x_1, x_m^*) &= -\left(\frac{2(x_1 - x_m^*)}{1+\lambda_1^2} - \frac{2(x_1 - x_m^*)\lambda_1^2}{(1+\lambda_1^2)^2}\right) \\ &\quad \times \frac{1}{\sqrt{-\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right)^2 + 1}}. \end{aligned}$$

Define $d = (x_1 - x_m^*) \neq 0 \in R^n$. Since $x_m^* \in \psi^0$, where ψ^0 is the interior of ψ, then d is a feasible direction. Thus,

$$\begin{aligned} \nabla G(x_1, x_m^*) &= -\left(\frac{2\lambda_1^2}{1+\lambda_1^2} - \frac{2\lambda_1^4}{(1+\lambda_1^2)^2}\right) \times \\ &\quad \frac{1}{\sqrt{-\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right)^2 + 1}} \\ &= -\frac{2\lambda_1^2}{(1+\lambda_1^2)^2} \frac{1}{\sqrt{-\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right)^2 + 1}} < 0. \end{aligned}$$

Because $x_1 \in \psi_a$, then $x_1 \neq x_m^*$. Since $d^T \nabla G(x_1, x_m^*) < 0$, then $\nabla G(x_1, x_m^*) \neq 0$. This proves the theorem.

■

From Theorems 3.2 to 3.4, we have proved that Equation (6) is descent in the region ψ_a. How the minimization of Equation (6) converges to a minimizer will be assured by the following theorems. Theorem 3.5 studies the value of Equation (6) in ψ_a and ψ_b.

Theorem 3.5: Let $x_m^* \in M$ be not a global minimizer of h. If $x_1 \in \psi_a$ and $x_2 \in \psi_b$, such that $\|x_1 - x_m^*\| < \|x_2 - x_m^*\|$. Then, $G(x_1, x_m^*) > G(x_2, x_m^*)$.

Proof: Since $x_1 \in \psi_a$ and $x_2 \in \psi_b$, then from Remark 1, we have

$$G(x_1, x_m^*) = -\arcsin\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right)$$

and

$$\begin{aligned} G(x_2, x_m^*) = &\left(-\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)\right) \\ &\times \left[\arctan\left(\beta_2^2\right) + 1\right]. \end{aligned}$$

The inequality below comes from the fact that $\|x_1 - x_m^*\| < \|x_2 - x_m^*\|$ and $arcsin$ is a monotonically increasing function.

$$\arcsin\left(\frac{\lambda_1^2}{1+\lambda_1^2}\right) < \arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right).$$

We know that $\arctan(t) \geq 0$ as $t \geq 0$. Thus,

$$\arctan\left(\beta_2^2\right) + 1 \geq 1.$$

Therefore, $G(x_1, x_m^*) > G(x_2, x_m^*)$.

■

The remark below is needed to provide cloudless interpretation of how the theorems are constructed so far.

Remark 2: The explanations of Theorems 3.2 - 3.5 are reviewed as follows:

1) Theorem 3.2 exposes that the point x_m^* is a hilltop.
2) If we move from x_m^*, the entire region of ψ_a is a downhill slope. This fact is stated by Theorem 3.3.
3) How our movement will not encounter uphill areas along the region ψ_a is guaranteed by Theorem 3.4.
4) If ψ_b is not empty and we remain moving to the region ψ_b, then we will continue to descent. Theorem 3.5 declares this.
5) At some points, we will find an uphill area in ψ_b. Theorem 3.6 will guarantee this condition.

From Remark 2, it is clear that the convergence of minimization process of filled function in Equation (6) will hold if we can find the ascent region in ψ_b. This will be proven in Theorem 3.6. The following inequality will be used in Theorem 3.6.

$$A > B, \tag{9}$$

where

$$A = \left[\frac{2d^T \nabla h(x)(-\beta_2)}{\beta_2^2 + 1}\right] \arcsin\left(\left\|\frac{\lambda_2^2}{1+\lambda_2^2}\right\|\right),$$

and

$$B = 2\lambda_2^2 \times \left[\arctan\left(\beta_2^2\right) + 1\right]$$

.

Theorem 3.6: Let $x_m^* \in M$. Suppose that x_m^* is not a global minimizer and $x_2 \in \psi_b$. If Equation (9) holds and $d^T \nabla h(x_2) > 0$, then d is an ascent direction of G at x_2.

Proof: Since $x_2 \in \psi_b$, it follows Remark 1, the value of G is the following:

$$G(x_2, x_m^*) = -\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)\left[\arctan\left(\beta_2^2\right)+1\right].$$

The vector gradient of G at x_2 is

$$\begin{aligned}\nabla G(x_2, x_m^*) &= -2(x_2 - x_m^*) \times \frac{\arctan\left(\beta_2^2\right)+1}{\left(1+\lambda_2^2\right)^2\sqrt{1-\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)^2}} \\ &\quad - \left[\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)\right]\frac{2\beta_2 \nabla h(x_2)}{1+\beta_2^2}\end{aligned}$$

The directional derivative of G for vector $d = (x_2 - x_m^*)$ is given below

$$\begin{aligned}d^T \nabla G(x_2, x_m^*) &= \left[\frac{-2\lambda_2^2}{\left(1+\lambda_2^2\right)^2}\right]\frac{\arctan\left(\beta_2^2\right)+1}{\sqrt{1-\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)^2}} \\ &\quad + \left[\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)\right] \times \\ &\quad \frac{2(-\beta_2) d^T \nabla h(x_2)}{1+\beta_2^2} \\ &> \left(-2\lambda_2^2\right)\left(\arctan\left(\beta_2^2\right)+1\right) \\ &\quad + \left[\arcsin\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)\right] \times \\ &\quad \frac{2(-\beta_2) d^T \nabla h(x_2)}{1+\beta_2^2}\end{aligned}$$

which derives from $\left(1+\lambda_2^2\right)^2\sqrt{1-\left(\frac{\lambda_2^2}{1+\lambda_2^2}\right)^2} = \left(1+\lambda_2^2\right)\sqrt{1+2\lambda_2^2} > 1$. Since (9) holds,

$$d^T \nabla G(x_2, x_m^*) > 0.$$

So, d is an ascent direction of G at x_2 ■

IV. Filled Function Algorithm

Our goal in this section is to offer filled function algorithm, which involves our new parameter-free filled function. All process of the algorithm is ensured by all properties provided in the previous section. Our algorithm (we call it NPFF) is given below.

Algorithm 1: NPFF Algorithm

- S-0: Choose $\gamma_0 \in (0, \gamma)$, where $\gamma \in R$; choose $P > 0$ (e.g. $P = 0.1$); set coordinate directions e_i, $i = 1, 2, ..., 2n$, where n is the number of variables of h; set $m = 1$.
- S-1: Minimize h starting from $x_m \in \psi$. This step yields x_m^*.
- S-2: Build filled function G of h at x_m^* (Equation (6)) and set $i = 1$.
- S-3: **while** $\gamma_0 < \gamma$ **do** $x_m = x_m^* + \gamma_0 e_i$ and go to Step 4, otherwise x_m^* is taken as a global minimizer.
- S-4: **if** $i \le 2n$, **then** minimize G starting from x_m to obtain x_i' and go to Step 5, otherwise, $\gamma_0 = \gamma_0 + P$, set $i = 1$ and go to Step 3.
- S-5: **if** $h(x'_i) < h(x_m^*)$, **then** $x_m \leftarrow x_i'$ and go to Step 1, otherwise set $i = i + 1$ and go to Step 4.

NPFF Algorithm is divided into two part: initial part and looping part. Minimizing the objective function with a specified initial guess and building filled function of the objective function at a converged local minimizer are done before entering the looping stage. The looping part begins with generating $x_m = x_m^* + \gamma_0 e_i$, where x_m^* is the first local minimizer and e_i is a direction. 3. The point x_m^* is then used as the initial point for minimizing filled function. The value γ_0 is added into x_m to avoid the minimization traps on x_m^*, because x_m^* is a maximizer of filled function. If there is i where the minimization of the filled function yields a minimizer x_i' such that $h(x'_i) < h(x_m^*)$, then x_i' is a new initial point to minimize the objective function so that a new better minimizer is obtained. From this step, we replace x_m^* by a new minimizer and the iteration will go back to Step 2. If for all the value of i, no minimizers of filled function hold $h(x'_i) < h(x_m^*)$, then we regenerate x_m by adding the positive small real number P to γ_0 and algorithm will go back to Step 3. On the other hand, if $\gamma_0 \ge \gamma$ and no minimizer of filled function satisfies $h(x'_i) < h(x_m^*)$, then the algorithm stops and the minimizer of the objective function found so far is taken as a global minimizer.

V. Numerical Experiment

NPFF Algorithm is then implemented to solve well-known global optimization problems. These problems are commonly used in the global optimization literature. For minimizing both cost function and filled function, we used BFGS method. In the implementation stage, the set ψ was known in advance. The benchmark global optimization problems are as follows:

Problem 1 (Two-dimensional Rastrigin function)

$$\begin{aligned}&\min\ h(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \\ &s.t.\ -1 \le x_1 \le 1,\ -1 \le x_2 \le 1.\end{aligned}$$

This is a very common problem on unconstrained global optimization literature. The global minimizer of Rastrigin function is $x_m^* = (0.0000, 0.0000)$ with $h(x_m^*) = -2$.

Problem 2 (Two-dimensional function)

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

NPFF algorithm succeeded to identify the global minimum value $h(x_m^*) = 0.0000$ for all c.

Problem 3 (Three-hump camel-back function)

$$\begin{aligned}&\min\ h(x) = 2x_1^2 - 1.05x_1^4 + \tfrac{1}{6}x_1^6 - x_1x_2 + x_2^2, \\ &s.t.\ -3 \le x_1 \le 3,\ -3 \le x_2 \le 3.\end{aligned}$$

The global optimum solution of this problem is $x_m^* = (0.0000, 0.0000)$, where $h(x_m^*) = 0.0000$.

Problem 4 (Six-hump camel-back function)

$$\begin{array}{l} \min\ h(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 - x_1x_2 \\ \qquad -4x_2^2 + 4x_2^4, \\ s.t.\ -3 \le x_1 \le 3,\ -3 \le x_2 \le 3. \end{array}.$$

Six-hump camel-back function has two global minimizers. They are $x_m^* = (0.0898, 0.7127)$ and $x_m^* = (-0.0898, -0.7127)$ with $h(x_m^*) = -1.0316$.

Problem 5 (Treccani function)

$$\begin{array}{l} \min\ \ h(x) = x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2, \\ s.t.\ -3 \le x_1 \le 3,\ -3 \le x_2 \le 3. \end{array}.$$

Treccani function has two global minimizers. They are $x_m^* = (0.0000, 0.0000)$ and $x_m^* = (-2.0000, 0.0000)$, where $h(x_m^*) = 0.0000$.

Problem 6 (Two-dimensional Shubert function)

$$\begin{array}{l} \min\ h(x) = \left\{ \sum\limits_{i=1}^{5} i\cos\left[(i+1)\,x_1\right] + i \right\} \\ \qquad \times \left\{ \sum\limits_{i=1}^{5} i\cos\left[(i+1)\,x_2\right] + i \right\}, \\ s.t.\ -10 \le x_1 \le 10,\ -10 \le x_2 \le 10. \end{array}$$

Shubert function is a crucial problem in global optimization field. This problem has about 760 minimizers. NPFF algorithm was successful in obtaining the global minimum value $h(x_m^*) = 186.7309$.

Problem 7 (Shekel's function)

$$\begin{array}{l} \min\ \ h(x) = -\sum\limits_{i=1}^{5} \left[\sum\limits_{j=1}^{4} (x_j - a_{i,j})^2 + c_i \right]^{-1} \\ s.t.\ 0 \le x_j \le 10,\ \ j = 1,2,3,4, \end{array},$$

with the coefficients $a_{i,j}$ and c_i are provided in Table I

TABLE I: The coefficients 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

The global minimizer of this problem is $x_m^* = (4, 4, 4, 4)$ with $h(x_m^*) = -10.1532$.

Problem 8 (Goldstein and Price function)

$$\begin{array}{l} \min\ \ h(x) = f_1(x).f_2(x) \\ s.t.\ \ -3 \le x_1 \le 3,\ \ -3 \le x_2 \le 3, \end{array}$$

with

$$\begin{array}{l} f_1(x) = 1 + (x_1 + x_2 + 1)^2 \\ \qquad \times \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right) \end{array}$$

and

$$\begin{array}{l} f_2(x) = 30 + (2x_1 - 3x_2)^2 \\ \qquad \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right) \end{array}$$

NPFF algorithm has successfully obtained the global minimizer $x_m^* = (0.0000, -1.0000)$ of this problem, where $h(x_m^*) = 3.0000$.

Problem 9 (n-dimensional Sine-square function)

$$\begin{array}{l} \min\ h(x) = \frac{\pi}{n}\left(P + S + T\right) \\ s.t.\ -10 \le x_i \le 10,\ i = 1,2,...,n. \end{array},$$

where $P = 10\sin^2(\pi x_1)$,

$$S = \sum_{i=1}^{n-1} \left[(x_i - 1)^2 \left(1 + 10\sin^2(\pi x_{i+1})\right) \right],$$

$T = (x_n - 1)^2$.

We solved this problem up to 50 dimensions. The global minimum value is $h(x_m^*) = 0.0000$.

Problem 10 (n-dimensional Rastrigin function)

$$\begin{array}{l} \min\ \ h(x) = 10n + \sum\limits_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right), \\ s.t.\ \ -5.15 \le x_i \le 5.12,\ \ i = 1,2,...,n. \end{array}$$

This function was solved for $n = 20,\ 30,\ 50$. The global minimizer is uniformly expressed as $x_m^* = (0, 0, 0, ..., 0)$, where $h(x_m^*) = 0.0000$.

Problem 11 (Ackley function)

$$\begin{array}{rcl} \min\ h(x) & = & -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) \\ & & -\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) \\ & & +20 + e, \end{array}$$

$$s.t.\ -32.768 \le x_i \le 32.768,\ i = 1,2,...,n.$$

This function was solved for $n = 10$ and $n = 20$. The global minimizer: $x_m^* = (0, ..., 0)$, $h(x_m^*) = 0$.

Problem 12 (Branin function)

$$\begin{array}{rcl} \min\ h(x) & = & \left(x_2 - 1.275\frac{x_1^2}{\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 \\ & & +10\left(1 - \frac{0.125}{\pi}\right)\cos(x_1) + 10, \end{array}$$

$$s.t.\ -5 \le x_i \le 15,\ i = 1,2.$$

The Branin function has three global minimum points: $x_m^* = (9.4247, 2.4750)$, $x_m^* = (3.1416, 2.2750)$, and $x_m^* = (-3.1416, 12.2750)$. The global minimum value is $h(x_m^*) = 0.3979$.

Problem 13 (Bohachevsky function)

$$\min\ h(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$

$$s.t.\ -100 \le x_i \le 100,\ i = 1,2.$$

This function attains its minimum global at $x_m^* = (0,0)$, with $h(x_m^*) = 0$.

Problem 14 (Beale function)

$$\begin{aligned} \min\ h(x) &= (1.5 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 \\ &\quad + \left(2.625 - x_1 + x_1x_2^3\right)^2 \end{aligned}$$

$$s.t. -4.5 \le x_i \le 4.5,\ i = 1, 2.$$

The global minimum point is at $x_m^* = (3, 0.5)$, $h(x_m^*) = 0$.

The notations below are used in Tables II - VII.
Prob: problem number.
n: the number of variables.
I_t: the number of iteration to get global minimizer.
H_f: the number of function evaluation of the objective function and filled function.
T: the CPU time for algorithm to obtain the global minimizer.
h_m^*: global minimum value of the h.
G_f: the number of gradient evaluation of the objective function and filled function.

Table II is an overall numerical result of the filled function proposed in this paper. Those data are gained by implementing the algorithm given in the previous section. We solve 14 global optimization problems that frequently used in global optimization literature. The information of the local minimizer(s) and their value of the problems are widely available in many resources. So, the computational results obtained can later be compared to other filled function algorithms, which will also be reviewed in this section. In Table II, I_t means the number of local minimum points successfully visited by our algorithm, where the last local minimizer obtained is the global one. The dimension of the problem solved ranges from 2 to 50. In [24], they addressed Problem 9 and Problem 10 up to 1000 dimensions. However, those two problems have symmetric feasible region. The feature owned by those two problems means that the higher dimension of the problems does not indicate the level of the efficiency of the algorithm. We used the same initial points as used in [12], except for problem 2 ($c = 0.2$), 9, 11, 12, 13, and 14 . For those problems, we deploy the initial point acquired randomly in its feasible region. If the starting point is more than 1, the value displayed in Table II is the average. Table II shows that the parameter-free filled function given in this study can be employed to find the global solution of the unconstrained global optimization problems.

The numerical results obtained will be compared with various filled functions that currently exist. The first comparison is with the filled functions [5] (RPFF Algorithm) and [27] (WPFF Algorithm), which are presented in Table III. These two functions are often used as an object of comparison to examine for evaluating the filled function method's effectiveness. We compare problems 1-6 and problem 9 up to 10 dimensions. The numerical performance of the RPFF and WPFF algorithms are adopted from [27]. Table III reveals that our filled function is much more effective in terms of the number of iterations function evaluations.

TABLE II: Computational Results

Prob	n	I_t	H_f	T	$h(x_m^*)$
1	2	3	322	0.233071	-2.0000
2 (c=0.2)	2	2	177	0.103109	3.8015e-18
2 (c=0.5)	2	2	127	0.136122	2.4982e-18
2 (c=0.05)	2	2	110	0.085900	6.3690e-19
3	2	2	45.5	0.095338	1.7812e-23
4	2	2	42.3	0.048961	-1.0316
5	2	2	146	0.112724	1.9330e-18
6	2	2	102	0.079625	-186.7309
7	4	2	332.5	0.192953	-10.1532
8	2	2	193	0.136824	3.0000
9	2	2	59	0.049324	1.1533e-18
	3	3	154	0.119019	3.2955e-20
	5	2	231	0.491824	1.1384e-15
	7	3	495	0.225745	3.6199e-17
	10	2	395	0.115396	9.9173e-17
	20	2	1043	0.473937	9.0621e-14
	30	3	2703	1.915886	1.4398e-14
	50	5	2278	8.412125	1.6860e-15
10	2	3	147	0.118253	0.0000
	3	4	487	0.379145	0.0000
	5	5	803	0.764473	0.0000
	7	6	854	0.502052	0.0000
	10	11	3688	1.966561	0.0000
	20	19	12417	7.590710	0.0000
	30	28	25821	23.614725	0.0000
	50	49	3205	139.035894	0.0000
11	10	18	6302	3.890256	9.0552e-10
	20	36	23113	17.768747	6.8313e-09
12	2	1	139	0.088063	0.3979
13	2	2	767	0.434860	0
14	2	1	52	0.340522	5.5785e-16

TABLE III: The Comparison of the Results

Prob	n	RPFF		WPFF		NPFF	
		I_t	H_f	I_t	H_f	I_t	H_f
1	2	4	21276	4	11097	3	322
2 (c=0.2)	2	15	27500	5	903	2	177
2 (c=0.5)	2	34	54505	3	47204	2	127
2 (c=0.05)	2	35	38424	4	11476	2	110
3	2	31.5	58328.5	3	8783	2	45.5
4	2	25	20437	3	4613	2	42.3
5	2	14	15759	2	2544	2	146
6	2	20	103988	3	8061	2	102
9	2	22	107899	6	83516	2	59
	3	6	248407	10	127695	3	154
	5	16	1229860	3	329956	2	231
	7	21	1443686	19	222630	3	495
	10	16	1829898	35	276386	2	395

In the next comparison, we want to assess the NPFF algorithm with another parameter-free filled function [22] (MPFF Algorithm). Three indicators are utilized to capture the efficiency of the NPFF algorithm; they are the number of iterations, function evaluations, and gradient evaluations. In [22], they employed the BFGS method and quasi-newton method to minimize the objective function and the filled function. At the same time, we only used the BFGS method to minimize both the objective and the filled function. This is why, as can be seen in Table IV, the number of gradient evaluations of the NPFF algorithm is all 0 for all problems. The numerical performance for the number of function evaluations of the NPFF algorithm, for all tested problems, is better than the MPFF algorithm. However, for some problems, the NPFF algorithm has a larger number of iterations than the MPFF algorithm. For example, for Problem 1, the MPFF algorithm needs only 2 iterations to arrive at the global minimizer, while the NPFF algorithm takes 3 steps. Other examples are for Problems 3,4,5 and

Problem 9 (n=2).

TABLE IV: The Comparison of the Results

Prob	n	MPFF			NPFF		
		I_t	H_f	G_f	I_t	H_f	G_f
1	2	2	337	206	3	322	0
2 (c=0.2)	2	3	4012	991	2	177	0
2 (c=0.5)	2	3	5097	1022	2	127	0
2 (c=0.05)	2	2	2507	246	2	110	0
3	2	1	545	37	2	45.5	0
4	2	1	518	79	2	42.3	0
5	2	1	595	72	2	146	0
6	2	5	5280	756	2	102	0
9	2	1	536	103	2	59	0
	3	3	6083	734	3	154	0
	5	2	7839	1140	2	231	0
	7	2	10130	4259	3	495	0
	10	4	29463	6856	2	395	0

The next two comparisons were made to portray the efficacy of the NPFF algorithm with another continuously differentiable parameter-free filled functions and the current filled function which has the same minimum point of the objective function.

Table V is the comparison between our algorithm and continuously differentiable parameter-free filled functions proposed in [23] and [24]. For convenience, we call them LPFF and APFF algorithms, respectively. Both NPFF and LPFF algorithms used the BFGS method to minimize objective function and filled function, while the APFF algorithm employed the quasi-Newton method. Table V is reported to capture the behavior of each filled function judged based on two indicators, namely the number of iteration and their function evaluation. From Table V, we can say that, in general, the number of iterations of the NPFF algorithm is better than the LPFF algorithm, and it is worst but not significantly if it is compared with the APFF algorithm. Our filled function tends to produce more local minimizer, especially if the objective functions is a symmetric function such as Rastrigin function (Problem 1 and Problem 10). However, our algorithm is much better than two other algorithms when viewed from the aspect of the number of function evaluations.

TABLE V: The Comparison of the Results

Prob	n	NPFF		LPFF		APFF	
		I_t	H_f	I_t	H_f	I_t	H_f
1	2	3	322	3.7	642.1	2	409
2 (c=0.2)	2	2	177	3	635	2	581
2 (c=0.5)	2	2	127	2.7	455.8	2	380
2 (c=0.05)	2	2	110	2.9	619.8	2	588
3	2	2	45.5	2	379	2	282
4	2	2	42.3	2	224.7	2	263
5	2	2	146	2	214.3	2	186
6	2	2	102	4.2	627.1	5	652
7	4	2	332.5	3	820.1	2	613
8	2	2	193	3	321.2	1	323
9	2	2	59	3.2	596.8	2	505
	3	3	154	3.4	1023.1	3	793
	5	2	231	3.2	2025.4	2	903
	7	3	495	4.7	3799.6	3	1121
	10	2	395	10.3	11747	3	1639
	20	2	1043	23.9	43254.3	3	5355
	30	3	2703	10	44194.1	3	3866
	50	5	2278	34.3	249833	3	5897
10	2	3	147	2.9	687.6	3	564
	3	4	487	2	998.1	3	650
	5	5	803	2	1906	6	1770
	50	49	3205	-	-	3	5642

The comparison between the proposed filled function and that given in [12] (GFFM algorithm) is displayed in Table VI. The main reason for comparing our algorithm with that given in [12] is because GFFM algorithm cuts one step of the traditional filled function algorithm. This is because their filled function has the same minimizer of the objective function. For this reason, we want to evaluate whether the GFFM algorithm influences the effectiveness of the results obtained.

The data which are compared between NPFF and GFFM algorithms are based on the aspect of the value of global minimum, the number of iterations, and the function evaluation of both cost function and filled function. Generally, function evaluation of GFFM algorithm is better than the algorithm given in this paper. From 15 problems displayed in Table VI, it can be seen that NPFF algorithm had 7 problems that had better function evaluation than GFFM algorithm did. However, the difference of the function evaluation between NPFF and GFFM algorithm is not too significant.

From the results displayed in Table VI, it can also be seen that for all problems solved, the number of iterations of NPFF algorithm was less than or equal to the GFFM algorithm. For example, for problem 2 (c = 0.2), NPFF algorithm has 2 iterations to obtain global minimizer, while GFFM had 3 iterations. Similarly, for problem 2 (c = 0.05), the number of iterations of our algorithm was far less than that of the GFFM algorithm. Two other problems, specifically problems 6 and 7, also had better number of iterations. In addition, NPFF algorithm has a better accuracy rate of global minimum value compared to GFFM algorithm for almost all problems. This level of accuracy is mainly seen in problems in which the global minimum value is 0. For example, the global minimum values for Problems 2–3 were far more accurate than those obtained in [12]. The only exception was for Problem 5 and Problem 9 (n=10), in which our global minimum value was 1.9330e-18 and 9.9173e-17, respectively, while the global minimum value in [12] was 0 for both problems.

TABLE VI: Comparison of the NPFF and GFFM algorithm

Prob	NPFF			GFFM		
	$h(x_m^*)$	I_t	$1H_f$	$h(x_m^*)$	I_t	H_f
2 (c = 0.2)	3.8015e-18	2	177	2.8229e010	3	131
2 (c = 0.5)	2.4982e-18	2	127	1.5257e009	2	88
2 (c = 0.05)	6.3690e-19	2	110	4.3885e011	7	236
3	1.7812e-23	2	61	1.5130e008	2	49
3	1.7812e-23	2	30	1.5130e008	2	49
4	-1.0316	2	41	-1.0316	2	60
4	-1.0316	2	43	-1.0316	2	60
4	-1.0316	2	43	-1.0316	2	92
5	1.9330e-18	2	146	0	2	34
6	-186.7309	2	102	186.7309	4	84
7	-10.1532	2	397	-10.1532	2	160
7	-10.1532	2	268	-10.1532	3	251
8	3.0000	2	193	3.0000	2	121
9 (n = 7)	3.6199e-17	3	495	7.4665e-011	2	690
9 (n = 10)	9.9173e-17	2	395	0	4	883

The last comparison was conducted to test our method's effectiveness compared with the results published in [20] (SFFM algorithm). SFFM algorithm applies the smoothing technique and the transformation of the multi-dimensional global optimization problem into a one-dimensional problem. From Table VII, it can be seen that the global minimum value generated by the NPFF algorithm has a higher level of accuracy than the accuracy level of the SFFM algorithm.

The superiority of the NPFF algorithm can also be seen from function evaluations, which are lower than those in the SFFM algorithm.

TABLE VII: Comparison of the NPFF and SFFM algorithm

Prob	NPFF		SFFM	
	$h\left(x_m^*\right)$	H_f	$h\left(x_m^*\right)$	H_f
2 (c = 0.2)	3.8015e-18	177	6.4583e16	518
2 (c = 0.5)	2.4982e-18	127	2.3665e15	522
2 (c = 0.05)	6.3690e-19	110	4.3800e16	306
3	1.7812e-23	45.5	1.3537e15	360
4	-1.0316	42.3	1.0316	384
5	1.9330e-18	146	2.3139e16	364
6	-186.7309	102	186.7309	480
7	-10.1532	332.5	10.1532	388
8	3.0000	193	3.0000	400
9	3.2955e-20	154	5.7060e14	244
10	0	3688	3.7682e12	332
12	0.3979	139	0.3979	204
13	0	767	1.0504e13	280
14	5.5785e-16	52	4.0766e14	112

VI. CONCLUSION

This paper offered a new parameter-free filled function. This paper has no limitations which usually happen to parametric filled function. The counterexample to other continuously differentiable parameter free filled function was illustrated. Furthermore, the general computational results were mined. The comparison result confirms that the proposed filled function was better than the current filled function.

REFERENCES

[1] Z. Tian, G. Wang and Y. Ren, "AMOAIA: Adaptive Multi-objective Optimization Artificial Immune Algorithm," *IAENG International Journal of Applied Mathematics*, vol. 49, no. 1, pp. 14-21, 2019.

[2] J. Wang, "A Novel Firefly Algorithm for Portfolio Optimization Problem," *IAENG International Journal of Applied Mathematics*, vol. 49,no. 1, pp. 45-50, 2019.

[3] W. C. Ho, J. M. Su and C. Y. Chang, "Maximal Market Potential of Feeder Bus Route Design Using Particle Swarm Optimization," *IAENG International Journal of Applied Mathematics*, vol. 49, no. 4, pp. 470-477, 2019.

[4] A. V. Levy and A. Montalvo, "The Tunneling Algorithm for The Global Minimization of Functions," *SIAM Journal On Scientific Computing*, vol. 45, pp. 15-29, 1985.

[5] R. P. Ge, "A Filled Function Method for Finding a Global Minimizer of A Function of Several Variables," *Mathematical Programming*, vol. 46, pp. 191-204, 1990.

[6] R. P. Ge and Y. F. Qin, "A Class of Filled Functions for Finding A Global Minimizer of A Function of Several Variables," *Journal of Optimization Theory and Applications*, vol. 54, pp. 241-252, 1987.

[7] C. Gao, Y. Yang and B. Han, "A New Class of Filled Functions with One Parameter for Global Optimization," *Computers and Mathematics with Applications*, vol. 62, pp. 2393-2403, 2011.

[8] 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, pp. 16-31, 2007.

[9] H. Lin., Y. Wang and L. Fan, "A Filled Function Method with One Parameter for Unconstrained Global Optimization," *Applied Mathematics and Computation*, vol. 218, pp. 3776-3785, 2011.

[10] H. Lin., Y. Gao and Y. Wang, "A Continuously Differentiable Filled Function Method for Global Optimization," *Numerical Algorithms*, vol. 66, pp. 511-523, 2014.

[11] J. Liu., Y. Wang, S. Wei, X. Sui and W. Tong, "A Filled Flatten Function Method Based on Basin Deepening and Adaptive Initial Point for Global Optimization," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, doi: 10.1142/S0218001420590119, 2020.

[12] H. Lin., Y. Gao, X. Wang and S. Su, "A Filled Function which Has The Same Local Minimizer of The Objective Function," *Optimization Letters*, vol. 13, pp. 761-776, 2019.

[13] H. Lin., Y. Wang, Y. Gao and X. Wang, "A Filled Function Method for Global Optimization with Inequality Constraints," *Computational and Applied Mathematics*, vol. 37, pp. 1524-1536, 2018.

[14] N. Yilmaz and A. Sahiner, "A New Auxiliary Function Approach for Inequality Constrained Global Optimization Problems," *An International Journal of Optimization and Control: Theories and Applications*, vol. 9, pp. 31-38, 2019.

[15] Y. Lin., Y. Yang and M. Mammadov, "A New Filled Function Method for Nonlinear Equations," *Applied Mathematics and Computation*, vol. 210, pp. 411-421, 2009.

[16] Z. Y. Wu., M. Mammadov, F. S. Bai and Y. J. Yang, "A Filled Function Method for Nonlinear Equations," *Applied Mathematics and Computation*, vol. 189, pp. 1196-1204, 2007.

[17] W. X. Wang., Y. L. Shang, L. S. Zhang and Y. Zhang, "Global Minimization of Non-Smooth Unconstrained Problems with Filled Function," *Optimization Letters*, vol. 7, pp. 435-446, 2013.

[18] Y. Zhang and Y. T. Xu, "A One-Parameter Filled Function Method Applied to Nonsmooth Constrained Global Optimization," *Computers and Mathematics with Applications*, vol. 58, pp. 1230-1238, 2009.

[19] Y. Zhang, Y. Xu and L. Zhang, "A Filled Function Method Applied to Nonsmooth Constrained Global Optimization," *Journal of Computational and Applied Mathematics*, vol. 232, pp. 415-426, 2009.

[20] A. Sahiner and S. A. Ibrahem, "A New Global Optimization Technique by Auxiliary Function Method in A Directional Search," *Optimization Letters*, vol. 13, pp. 309-323, 2019.

[21] L. An, L. S. Zhang and M. L. Chen, "A Parameter-Free Filled Function for Unconstrained Global Optimization," *Journal of Shanghai University (English Edition)*, vol. 8, pp. 117-123, 2004.

[22] S. Ma, Y. Yang and H. Liu, "A Parameter Free Filled Function for Unconstrained Global Optimization," *Applied Mathematics and Computation*, vol. 215, pp. 3610-3619, 2010.

[23] H. Liu, Y. Wang, S. Guan and X. Liu, "A New Filled Function Method for Unconstrained Global Optimization," *International Journal of Computer Mathematics*, vol. 94, pp. 2283-2296, 2017.

[24] A. I. Ahmed, "A New Parameter Free Filled Function for Solving Unconstrained Global Optimization Problems," *International Journal of Computer Mathematics*, doi:10.1080/00207160.2020.1731484, 2020.

[25] Y. Yang and Y. Shang, "A New Filled Function Method for Unconstrained Global Optimization," *Applied Mathematics and Computation*, vol. 173, pp. 501-512, 2006.

[26] X. Liu, "The Impelling Function Method Applied to Global Optimization," *Applied Mathematics and Computation*, vol. 151, pp. 745-754, 2004.

[27] C. Wang, Y. Yang and J. Li, "A New Filled Function Method for Unconstrained Global Optimization," *Journal of Computational and Applied Mathematics*, vol. 225, pp. 68-79, 2009.