Numerical Simulation on Solving Three-Dimensional Global Optimization Problems in cooperation of Filled Function and Search Direction

Herlina Napitupulu¹, Ismail Bin Mohd², Yuyun Hidayat³, Sukono⁴, Sudradjat Supian⁵

¹) School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu
Kuala Terengganu, Malaysia
²) Institute for Mathematical Research (INSPEM)
Universiti Putera Malaysia
Selangor, Malaysia
³) Department of Statistics
FMIPA Universitas Padjadjaran
Sumedang, Indonesia
¹, ⁴, ⁵) Department of Mathematics
FMIPA Universitas Padjadjaran
Sumedang, Indonesia

napitupuluherlina@gmail.com

Abstract. Global optimization problem still becomes an interest due to the challenge of locating the global optimum of nonlinear objective function with multiple local minima. Two challenges on solving global optimization problem are; firstly how to reach the better minimizer from the current minimizer, and secondly how to decide that the obtained minimizer is the desired global minimizer. One of the recent considered deterministic easy applied methods, which concerned in the mentioned problems, is the filled function method. The basic concept of filled function method is to build such an auxiliary function to locate a point with lower function value than the current minimizer. One of the keys to the successfully filled function method is how to decide the search direction to reach and locate a better local minimizer. In this paper, a three-dimensional filled function method and its search direction are introduced. The algorithm is presented and implemented to some benchmark test function. The numerical performance of the method on solving three-dimensional global optimization problems is presented.

1. Introduction
The objective of global optimization is to obtain the best solution in the possible presence of a multitude of local suboptimum. Theory and methods of optimization have wide applications in most all branches of science, engineering, business management, military, space technology, finance, and so forth. Research of global optimization has become a challenging topic, since the existence of multiple optima that differ from the global solution. This multimodal case cannot be solved by classical nonlinear programming techniques directly. General two difficulties of global optimization are; how to...
leave from present local minimizer to another minimizer with the lower value and how to decide that
the present minimizer is a global solution.

The main difficulty for global optimization is to escape from the current local minimizer and to
find a better one. One of the most efficient methods to deal with this issue is the filled function method
[1]. The filled function method is an approach to find the global minimizer of multi-modal functions.
Compared to the other methods, filled function method has an outstanding advantage that is simple
while its realization is relatively easy and effective at successively finding smaller local minima [2,
3, 4].

The first filled function with two adjustable parameters was proposed for smooth optimization by Ge
in 1983 and finally published in 1990, [5] which were used to solve the global minimizer of
unconstrained. Some disadvantages persist in this former function due to the existence of exponential
term and two adjustable parameters. Many modifications are proposed as the revision to the former
filled function; however, the existence of adjustable parameter(s), exponential, and the logarithmic
term still become difficulties on the computational process. However, there is no efficient criterion to
determine the appropriate parameter. It is better if it is possible to build a filled function without
parameter, as well as neither exponential nor logarithmic term. Moreover, many filled functions need
to choose more than one starting point to obtain multiple optimum solutions. Actually, the parameter
free filled function with above criteria is a desired one.

Mohd et al. [8] proposed a one dimensional parameter free filled function that is theoretically and
numerically succeeded for solving multimodal general global optimization problems using one starting
point. The extension of one dimensional filled function [8] into two-dimensional filled function [9] is
so-called Ismail-Herlina-Ridwan filled function (in short, TIHR filled function). Napitupulu and
Mohd [10] introduced types of vector direction of the second phase of filled function of their algorithm
method on solving two dimensional unconstrained global optimization problems.

In this paper, we extend parameter free filled function [8] into three-dimensional which can be used
to solve an unconstrained global optimization problem. Secondly, we introduce some types of search
direction, to be used in the second phase of filled function algorithm method. We implement the
algorithm method into benchmark test function to test the efficiency of our algorithm. Numerical and
computational results show how the performance of both the algorithm and types of proposed search
direction. The conclusion is given in the rest of the paper.

2. Parameter Free Filled Function

This paper considers the following global optimization problem:

$$\min_{x \in D} f(x)$$

(1)

where $D \subset \mathbb{R}^n$ is a closed bounded domain containing all global minimizers of $f(x)$ in its interior. It
is assumed in this paper that $f(x)$ has only a finite number of local minimizers. When $f : \mathbb{R}^n \to \mathbb{R}$ is
coercive, that is, $f(x) \to +\infty$ as $\|x\| \to +\infty$ then a global optimization problem (2) can be always
reduced into an equivalent problem (1). Let $x^*$ denote the isolated local minimizer of $f(x)$.

$$\min_{x \in \mathbb{R}^n} f(x).$$

(2)

A non-parametric filled function of one-dimensional in [8] basically from the idea of integration, the formula is as follows

$$F_{ig}(x, x^*) = \begin{cases} \int_x^{x^*} (f(s) - f(x^*)) ds & (x \geq x^*) \\ \int_{x^*}^x (f(s) - f(x^*)) ds & (x < x^*) \end{cases}$$

(3)
Others researchers are proposed the n-dimensional parameter free filled functions as done in [6], [4], and [7]. The features of filled function [5, 6, 7] are no parameter included and no exponential or logarithmic term exists in the formula. However, these functions method has disadvantages, that is, always needs more than one starting point to obtain more than one global minimizers.

In this paper we proposed a three-dimensional parameter free filled function method, which is extension from (3), together with its search direction. The attention is to find the global minimizer(s) of multimodal function \( f(x) \) on domain \( D \subseteq \mathbb{R}^3 \). The properties of three dimensional parameter free filled function are explained in the following section.

3. Three-Dimensional Parameter Free Filled Function

Consider a function \( f : X \subseteq \mathbb{R}^3 \rightarrow R \) where \( X \) is a box defined by
\[
X = \left\{ x = (x_1, x_2, x_3)^T \mid x_{ij} \leq x_i \leq x_{iS} \wedge x_{jL} \leq x_j \leq x_{jS} \right\}
\]
and consider a function \( F : D \subseteq X \) where a box \( D \) is defined by
\[
D = \left\{ x = (x_1, x_2, x_3)^T \mid x_{iL} \leq x_i \leq x_{iS} \wedge x_{jL} \leq x_j \leq x_{jS} \right\}
\]
where \( x_{ij} \) and \( x_{iS} \) (\( j = 1, 2, 3 \)) are the infimum (denoted by \( i \)) and supremum (denoted by \( S \)) of the interval \( x_j\) respectively. Assume that \( \mathbf{x}^* = (x_1^*, x_2^*, x_3^*)^T \in D \) is a current local isolated minimizer of \( f \). The domain \( D \) can be divided into eight sub domains \( D_1, D_2, D_3, D_8 \) such that
\[
D = \bigcup_{i=1}^{8} D_i = (D_1 \cup D_2 \cup ... \cup D_8),
\]
Three-dimensional free filled function \( F \), defined on sub domain \( D_i \) (\( i = 1, 2, ..., 8 \)), is given by
\[
F_i \left( \mathbf{x}, \mathbf{x}^* \right) = - \frac{1}{b_i} \int_{a_i}^{b_i} \left( f(x_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) dx_i \quad (a_i \leq b_i)
\]
where \( a_j \) and \( b_j \) \((j=1,2,3)\) are described in Table 1.

| \( j \) | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| \( a_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) |
| \( b_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) | \( x_1 \) |
| \( a_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) |
| \( b_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) | \( x_2 \) |
| \( a_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) |
| \( b_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) | \( x_3 \) |

The definition of three-dimensional parameter free filled function is given in Definition 1 and its validity is proved by Theorem 1-3.

**Definition 1** \( F(x,x) \) is called a three-dimensional parameter free filled function of \( f(x) \) at local isolated minimizer \( x^* \) of \( f \) iff

(i). \( x^* \) is a maximizer of \( F = F(x,x^*) \leq f(x^*,x^*) \)

(ii). \( F \) has no stationary point in a set \( H_i = \{ x \mid f(x) > f(x^*,x^*) \} \)

(iii). There is a point \( x \in H_i = \{ x \mid f(x) \leq f(x^*,x^*) \} \) such that \( x \) is a stationary point of \( F \), where \( a_j \) and \( b_j \) are as in Table 1

Theorem 1

If \( x^* = (x_1^*,x_2^*,x_3^*)^T \in \mathbb{R}^3 \) is an isolated minimizer of the objective function \( f(x) \) on \( D = \bigcup_{i=1}^{8} D_i \), then \( x^* \) is a maximizer of filled function \( F \).

Proof.

Since \( f(x_1,x_2,x_3) - f(x_1^*,x_2^*,x_3^*) \geq 0 \) and Table 1, then for all sub domains \( D_i \ (i=1,2,...,8) \), we have

\[
\int_{a_i}^{b_i} f(x_1,x_2,x_3) - f(x_1^*,x_2^*,x_3^*) \, ds_i \geq 0 \quad \text{for } a_i \leq b_i
\]

These imply for all \( x \in D_i \), \( F_i(x,x^*) \leq 0 \) \((i=1,...,8)\). Therefore \( F(x,x^*) \leq 0 = F(x^*,x^*) \).

Theorem 2

If \( (x_1^*,x_2^*,x_3^*)^T \) is a local minimizer of \( f(x_1,x_2,x_3) \) on \( D = \bigcup_{i=1}^{8} D_i \) and

\[
H_i = \{ x \mid f(x) > f(x^*), \ x \in D \ \backslash \{ x^* \} \}
\]
then \( F \) has no stationary point in the set \( H_1 \).

**Proof.**

Suppose that \( \mathbf{x} \in H_1 \) satisfies \( f(\mathbf{x}) \geq f(\mathbf{x}^*) \). Consider \( F(\mathbf{x},\mathbf{x}^*) \) in sub domain \( D_1 \). Since \( f \) is a twice differentiable function, then

\[
\frac{\partial F_1}{\partial x_i} = -(f(x_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*)) < 0
\]

\[
\frac{\partial F_1}{\partial x_2} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_2} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

\[
\frac{\partial F_1}{\partial x_3} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_3} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

Therefore, \( \nabla F_{1i} \neq 0 \), means that \( F_{1i} \) has no stationary point in \( H_1 \).

For the gradient of \( F_{12} \), also

\[
\frac{\partial F_{12}}{\partial x_1} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_1} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

\[
\frac{\partial F_{12}}{\partial x_2} = -(f(x_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*)) < 0
\]

\[
\frac{\partial F_{12}}{\partial x_3} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_3} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

Thus \( \nabla F_{12} \neq 0 \), means that \( F_{12} \) has no stationary point in \( H_1 \). Therefore, \( \nabla F_{12} \neq 0 \), means that \( F_{12} \) has no stationary point in \( H_1 \).

For the gradient of \( F_{13} \), also

\[
\frac{\partial F_{13}}{\partial x_1} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_1} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

\[
\frac{\partial F_{13}}{\partial x_2} = -(f(x_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*)) < 0
\]

\[
\frac{\partial F_{13}}{\partial x_3} = -\left( \int_{s_i}^{s_j} \frac{\partial}{\partial x_3} \left( f(s_1, x_2, x_3) - f(x_1^*, x_2^*, x_3^*) \right) ds_i \right) 
\]

Thus \( \nabla F_{13} \neq 0 \), means that \( F_{13} \) has no stationary point in \( H_1 \). Therefore, in sub-domain \( D_1 \) there is no stationary point of \( F \) in \( H_1 \). The similar proof can be applied to \( F_{2}, \ldots, F_{n} \). □

**Theorem 3**

If \( H_2 = \left\{ \mathbf{x} \mid f(\mathbf{x}) \leq f(\mathbf{x}^*), 0 \leq \mathbf{w} \leq W, \mathbf{x} \in \mathbb{D} \left\{ \mathbf{x}^* \right\} \right\} \neq \emptyset \) where

\[
W = \left\{ \frac{\partial}{\partial x_i} \left( \int_{s_i}^{s_j} f(\mathbf{x}) ds_i \right) \right\}_{l \neq j, (l = 1, 2, 3)}
\]

for specified \( j \) and \( a_j, b_j \) are as in Table 1, then there is a point \( \mathbf{x} = (x_1, x_2, x_3)^T \in H_2 \) such that \( \mathbf{x} \) is a stationary point of \( F(\mathbf{x}, \mathbf{x}^*) \).

**Proof.**

Consider \( F_{11} \) in sub domain \( D_1 \). Suppose that for \( \mathbf{x} \in H_2 \), \( f(x_1, x_2, x_3) = f(x_1^*, x_2^*, x_3^*) \) holds and satisfies

\[
\frac{\partial}{\partial x_i} \left( \int_{s_i}^{s_j} f(s_1, x_2, x_3) ds_i \right) = 0 = \frac{\partial}{\partial x_j} \left( \int_{s_j}^{s_i} f(s_1, x_2, x_3) ds_i \right)
\]

Then
Therefore, there exists a stationary point of $F_{1,j}$ for $x \in H_2$. The analog proof also applied for $F_{1,i}$ and $F_{1,j}$ for $x \in H_2$ there is a stationary point of $F$. Similar proof can be applied to $F_2, F_3, ..., F_8$. □

4. Algorithm Method

In this section the algorithm method is presented which is in cooperation of three dimensional parameter free filled function and the search direction. The calculation method consist of two main phases, that are the minimization of objective function and filled function construction.

Data (initialization) specify initial point $x_0$, domain $D_i$, real number $d > 0$ and set, $i = 1, j = 1, m = 1$

step 1 specify initial step size of steepest descent $\alpha_0 > 0$, minimize $f(x)$ starting at $x_0$ and to obtain a local minimizer $x_m$

step 2 construct $F$ function $F_{ij}(x, x_m)$ at $x_m$

step 3 if $i \leq n$ then choose direction $e_i$ for respected defined dimension and go to step 4 else stop

step 4 set $c := 1$

step 5 set initial point $x_0^e = x_m + cde_i$

if $x_0^e \in D_i$ then go to step 6
else if $j < n$ then $j := j + 1$, go to step 4
else $j = 1; i := i + 1$; and go to step 3

step 6 if $f(x_0^e) < f(x_m^e)$ then set $x_0 := x_0^e$, go to step 1
else if one of cond$_1$ true then go to step 7
else $c := c + 1$ and go to step 5

step 7 by Newton’s method with initial $x_0^e$ solve $\nabla F_{ij}(x, x_m^e) = 0$ to obtain $x^e$

step 8 if cond$_2$ and cond$_3$ hold then set $x_m := x^e$, $m := m + 1$, go to step 1
else $c := c + 1$ go to step 5

Note:
cond$_1$: at least there is exists a $j$ ($j = 1, 2, 3$) such that the following condition holds

$$\left(\frac{\partial F_{ij}(x = x_m^e + cde_i)}{\partial x_j}\right)^2 < \left(\frac{\partial F_{ij}(x = x_m^e + (c - 1)d e_i)}{\partial x_j}\right)^2$$

cond$_2$: at least one of following conditions holds for a specified $j$

(i). $\frac{\partial^2}{\partial x_j^2} F_{ij}(x = x^e) > 0$ and (ii). $\left|\frac{\partial^2}{\partial x_j^2} F_{ij}(x = x^e)\right| < 10^{-4}$

cond$_3$: $|f(x^e) - f(x_m^e)| < 10^2$

The explanation of this algorithm is as follows. The initial point is chosen for iteration steepest descent to obtain the current minimizer $x^e$. After $x^e$ is obtained then entering phase 2, built a filled function according to its domain. Symbol $m$ represent number of current minimizer obtained. After that, set the
initial point to minimize the filled function i.e., \( x_0^f = x_0^* + cd\epsilon \) such that satisfies \( x_0^f \in D \) and if not thus generate \( c \) and repeat to checking the initial point, if this point out of its domain, thus do \( c := c + 1 \) or evaluated other filled function in the same domain \( i \), if \( j > 2 \), do \( i := i + 1 \) means calls the next filled function in the next domain, and do repeat to check the initial point again, until this point satisfies the conditions given, if \( j > 2 \) then stop. If the initial point satisfy the condition, doing minimizing filled function till obtained the point which has the equal value with current minimizer. If it is not satisfied but still in its domain, generate the integer number \( c \) again and return to step 11. If the point \( x^f \) satisfies the conditions go to step 2, set \( m := m + 1 \), and do iteration steepest descent again. If the next minimizer obtained is has the lower or equal value compared to the current minimizer obtained, thus set it to be the current minimizer to entering second phase.

5. The Search Direction

How to find the right direction is one of the important things in the filled function method. Various functions may have different pattern of location of the local or global minimizer. For two-dimensional case, it can be easily seen through contour plot. In this research we use search direction to locate another better point, which is outside the current minimizer to another the desired initial point of Newton's method. It is well known that Newton's iteration will well behave if the initial point is near to the desired point. If search direction is not properly chosen, Newton's method will not work well, then may causing execution time on searching minimizer or saddle point of filled function becoming longer, or make the desired point does not obtain even if the desired point(s) exist in the evaluated sub domain.

Now let us consider the second phase of filled function's algorithm. The proposed search direction is focused on finding a new point, say \( x_0^e \), which exist outside the neighborhood of current minimizer, \( x^* \), but near to the minimizer or saddle point of filled function, say \( x^f \). Since our consideration is on three-dimensional function, then it is possible that will exist four triple of a sign of axis direction. In the section of algorithm method, we use following search direction for each respected sub-domain \( D_i \) with \( i = (1, 2, ..., 8) \).

\[
e = \{ e_1, e_2, e_3, e_4 \} = \{(1,1,1)^T, (1,-1,1)^T, (1,-1,-1)^T, (-1,1,1)^T, (-1,-1,1)^T, (-1,-1,-1)^T \}
\]  

In this paper, we introduce the search direction to be implemented to algorithm method, in the form of equation (12).

\[
V^{(p)} = (\text{sign}_1 v^{(p)}_1, \text{sign}_2 v^{(p)}_2, \text{sign}_3 v^{(p)}_3)^T
\]  

The \( \text{sign}_j \) \( (j = 1, 2, 3) \) in (12) symbolizing a positive or negative sign of \( v^{(p)} \) in search direction \( V^{(p)} \) for each respected sub-domain \( D_i \) \( (i = 1, 2, ..., 8) \). Let \( n \) is the number of variable of \( f \). Suppose that \( D(i) \) is decimal (denoted by \( (D(i))_{10} \)) and \( b_j \) is a binary number with value only 1 or 0 (denoted by \( (b_1 b_2 b_3)_{2} \)). Then it can be written that \( (D(i))_{10} = (b_1 b_2 b_3)_{2} \). Let \( D(i) = i - 1 \), then we define \( \text{sign}_j \) \( (j = 1, ..., n) \) for the given \( i \) as

\[
b_j \begin{cases}
0, \text{sign}_j = + \\
1, \text{sign}_j = -
\end{cases}
\]

For example, for \( i = 1 \) then \( \text{sign}_j \) for all \( j = 1, 2, 3 \) are positive, and for \( i = 2 \) the \( \text{sign}_1 = \text{sign}_2 \) is positive and \( \text{sign}_3 \) is negative, and so on.
In this paper, we introduce the search direction in the form of equation (12), where \( v^{(p)}_j = \sin \theta_j \) for \( 0 \leq \theta_j \leq \pi \). To see the performance of the filled function algorithm method, we restrict restricted the simulation by only use the search direction with combination \( \theta \) value are \( \theta = \pi/6 \) and \( \theta = \pi \). We present the 36 combination of search direction, and the explanation on how search directions defined is given in the following explanation and algorithm.

**Case 1:** \((1 \leq p \leq 6)\)

Suppose that \( p(i) \) is decimal (denoted by \( (p)_{10} \)) and \( q_j \) is a binary number with value only 1 or 0 (denoted by \( (q_j)_{2} \)). Then it can be written that \((p)_{10} = (q_1 q_2 q_3)_{2} \). Thus the defined \( v^{(p)}_j(j=1,\ldots,n) \),

\[
\begin{align*}
\text{if } q_j = 0, & \text{ then } v^{(p)}_j = \sin(\pi) \\
\text{if } q_j = 1, & \text{ then } v^{(p)}_j = \sin(\pi/6)
\end{align*}
\]

**Case 2:** \((7 \leq p \leq 36)\)

Define \( p := 7 \)

for \((p1:=1; p1++ , p1 \leq 6)\) do

for \((p2:=1; p2++ , p2 \leq 6)\) do

if \((p2 == p1)\) then \((p2 := p2++)\)

\(V^{(p)} = \begin{cases} 
V^{(p1)}, & \text{if } c = c_o \\
V^{(p2)}, & \text{if } c = c_e 
\end{cases} \)

\(p := p++;\)

end do

end do

The previous algorithm shows that the search direction with \(7 \leq p \leq 36\) are the combination of the search direction with \( 1 \leq p \leq 6\). The \( c_o \) and \( c_e \) respectively means an even and an odd value of \( c \) (a constant value of initial point in step 5 algorithm method).

Now we have 36 type of search direction to replace former search direction \( e_i \) in Step 5 of algorithm. In further, the numerical and computational results are explained in the following section.

### 6. Results

To test the performance of algorithm method with several given search directions we use benchmark test function 3-dimensional sine square function, with initial point \( x_0 = (-2,-2,-2)^T \).

\[
f(x) = \frac{\pi}{3} \left[ 10 \sin^2(\pi x_i) + \sum_{i=1}^{n-1} (x_i - 1)^2 \left( 1 + 10 \sin^2(\pi x_{i+1}) \right) + (x_n - 1)^2 \right]
\]

\(-10 \leq x_i \leq 10, i = 1,2,3 \quad (13)\)

The program is written in visual C++ program, Processor Intel ® Core i3, 2048 MB RAM. In minimizing \( f(x) \), steepest descent method with elimination line search step size (see [11]) is used, which its step size is made to be absolute. Stopping criteria for steepest descent method that we use is \( |\nabla f| \leq 10^{-3} \), while for Newton’s method is \( |\nabla F| < 2 \cdot 10^{-4} \).

### 6.1. Numerical Results

The detail numerical results of the given problem are presented in Table 2, with the explanation of symbols given as follows.

\( k \) : number of iteration
$F_j$: the evaluated Three-Dimensional Parameter Free Filled Function  
\(\alpha_0\): initial step size of steepest descent method  
\(c\): integer number to be multiplied to search direction  
\(x_0\): initial point for steepest descent method in \(f(x)\)  
\(x_k\): \(k\)-th obtained minimizer  
\(x_0^F\): initial point for Newton’s method in \(F_j\)  
\(x^F\): point obtained by Newton’s method or satisfied \(f(x^F) < f(x_k^*)\)  
\(f_0, f_1^*, f_0^F, f^F\): function value of \(x_0, x_k^*, x_0^F, x^F\)

### Table 2. Numerical Results with given \(e\)

| \(k\) | \(a_0, F_{ij}, c, d\) | Results |
|---|---|---|
| 0 | \(a_0 = 0.001\) | \(x_0 = (-2, -2, -2)^T, f_0 = 28.287\)  
\(x_1 = (-1.96892, -1.99575, -1.99582)^T, f_1^* = 28.1544\) |
| 1 | \(F_{ij}, c = 7\) | \(x_0 = (-1.26892, -1.64575, -1.29582)^T, f_0 = 114.62\)  
\(x_1 = (-1.96892, -1.10412, -0.0864502)^T, f_1^* = 28.1555\) |
| 2 | \(F_{ij}, c = 5\) | \(x_0 = (-1.46894, -0.747303, 0.502532)^T, f_0 = 84.6167\)  
\(x_1 = (-1.96895, 1.00115, 1.4996)^T, f_1^* = 14.5577\) |
| 3 | \(F_{ij}, c = 10\) | \(x_0 = (-1.46894, -0.747303, 0.502532)^T, f_0 = 84.6167\)  
\(x_1 = (0.999608, 0.99999, 1)^T, f_1^* = 1.71823 \times 10^{-7}\) |

6.2. Comparison  
The general computational results are presented in Table 3. The following symbols are used in the table. The SD refers to types of search direction.  

\(VD\): type of search direction  
\(iter\): number of succeeded iteration  
\(time/time_1\): execution time until global minimizer obtained/ program terminated in second  
\(nf/nf_1\): number of function evaluation of \(f\) until global minimizer obtained/ program terminated  
\(nF'/nF'_1\): number of \(\nabla F_j\) until global minimizer obtained/ program terminated  
\(T/T_1\): total number of \(nf + nF' + nF'\) / total number of \(nf_1 + nF'_1 + nF'_1\)

### Table 3. General Numerical Results

| \(VD\) | \(iter\) | \(time/time_1\) (second) | \(nf/nf_1\) | \(nF'/nF'_1\) | \(T/T_1\) |
|---|---|---|---|---|---|
| \(e\) | 10 | 12.558/41.449 | 3004/3336 | 1502/1668 | 21672/38062 | 26178/43066 |
| \(\nu^{(1)}\) | 8 | 6.708/15.1 | 6904/7566 | 3452/3783 | 9733/27247 | 20089/38596 |
Based on number of previously, by applying particular type of search direction in filled function algorithm method, the performed by V
(21)
, V
(20)
, V
(19)
, V
(18)
, V
(17)
, V
(16)
, V
(15)
, V
(14)
, V
(13)
, V
(12)
, V
(11)
, V
(10)
, V
(9)
, V
(8)
, V
(7)
, V
(6)
, V
(5)
, V
(4)
, V
(3)
, V
(2)
, V
(1)
, V
(0)
 methods which is more efficient. Besides, it is observed from the results of numerical experiment that the global minimizer is obtained is performed by V
(2)
 with 1.762 second, while the least total time until program terminated is performed by V
(23)
 with 9.188 second. Based on number of T, V
(21)
 performs the least number with total is 548. Based to total number of T, V
(15)
 performs the least number with total is 17412.

7. Conclusion

Three dimensional parameter free filled function together with some types of search direction are introduced and explained in this paper. The objective is on solving the three dimensional unconstrained global optimization problem. According to the numerical performance discussed previously, by applying particular type of search direction in filled function algorithm method, the method is more efficient. Besides, it is observed from the results of numerical experiment that the search direction of filled function method does works better for some case and does not for another. Based on number of T, V
(21)
 performs the least number with total is 548. Based to total number of T, V
(15)
 performs the least number with total is 17412. However, this chapter gives insight, that, it is possible to look at the search direction as the way of speeding up the search on this algorithm method.
Acknowledgments
This work was supported by: 1) Grant research of Riset Kompetensi Dosen Unpad (RKDU) under project leader of Dr. Sukono MM, MSi.; 2) Unpad Academic Leadership Grant (ALG) under the project leader of Prof. Dr. Sudradjat Supian, MS.; 3) Fundamental Research Grant Scheme (FRGS Vot. No. 59255). Ministry of Higher Education Malaysia

References
[1] Lin H, Gao Y, and Wang Y 2013 A continuously differentiable filled function method for global optimization, Numerical Algorithm (doi: 10.1007/s11075-013-9746-3)
[2] Wang X, and Zhou G 2006 A New Filled Function for Unconstrained Global Optimization, Applied Mathematics and Computation 174 pp 419–429.
[3] Wei F and Wang Y, A 2013 New Filled Function Method with One Parameter for Global Optimization, Hindawi: Mathematical Problems in Engineering. (http://dx.doi.org/10.1155/2013/532325)
[4] Ma S, Yang Y, and Liu H 2010 A Parameter Free Filled Function For Unconstrained Global Optimization, J. Applied Mathematics and Computation, 215, pp. 3610-3619.
[5] Ge R 1990 A Filled Function Method For Finding A Global Minimizer of A Function of Several Variables”, J. Mathematical Programming, 46 (2), pp.191-204.
[6] An L, Zhang L, Chen M 2004 A Parameter-Free Filled Function for Unconstrained Global Optimization J. Shanghai Univ. Engl. Ed. 8 (2) pp 117–123.
[7] Shang Y, Li P, Xie H, 2010 A New Parameter Free Filled Function for Unconstrained Global Optimization The Third Int. Joint Conf. on Computational Sciences and Optimization pp 65-69 (doi: 10.1109/CSO.2010.59)
[8] Mohd I B, Wen G K, and B. D. Yosza 2008 Fungsi Pengisian Sebagai Satu Kaedah Pilihan Dalam Pengoptimuman Sejagat Jurnal KALAM (Karya Asli Lorekan Ahli Matematik) 1(1) pp 6-14.
[9] Napitupulu H, Mohd I B, and Pandiya R 2014 Proc. of the 3rd int. Conf. on Global Optimization and Its Applications Kuala Terengganu: Penerbit UMT pp 218-140
[10] Napitupulu H, and Mohd I B 2016 Vector Direction of Filled Function Method on Solving Unconstrained Global Optimization Problem, AIP Conf. Proc. 1707, 050011 (doi: 10.1063/1.4940843)
[11] Antoniou A, and Lu W S 2007 Practical Optimization, Algorithm and Engineering Application. New York: Springer