Comparison of Conjugate Gradient Method on Solving Unconstrained Optimization Problems

Maulana Malik*
Department of Mathematics
Universitas Indonesia (UI)
Depok 16424, Jawa Barat, Indonesia
m.malik@sci.ui.ac.id

Mustafa Mamat, Siti Sabariah Abas, Ibrahim Mohammed Sulaiman
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (UniSZA)
Besut 22200, Terengganu, Malaysia
must@unisza.edu.my, sabariahabas@unisza.edu.my, sulaimanib@unisza.edu.my

Sukono
Department of Mathematics
Universitas Padjadjaran (Unpad)
Sumedang 45363, Jawa Barat, Indonesia
sukono@unpad.ac.id

Abdul Talib Bon
Department of Production and Operations,
Universiti Tun Hussein Onn Malaysia (UTHM)
Parit Raja 86400, Johor, Malaysia
talibon@gmail.com

Abstract

Conjugate gradient (CG) method approaches have been instrumental in solving unconstrained optimization problems. In 2020, Malik et al. have proposed a new hybrid coefficient (H-MS2), a combination of the RMIL coefficient and the new coefficient. In this paper, we propose the new method, which takes the new coefficients from H-MS2. Also, we will compare the new method and some of the classic methods that already based on the number of iterations and central processing unit (CPU) time. The new method fulfills the sufficient descent condition and global convergence properties, and it’s tested on a set functions under exact line search. The numerical results show that the new CG method has the best efficiency between all the methods tested.

Keywords
Conjugate gradient method, unconstrained optimization problems, sufficient descent condition, global convergence properties, exact line search

1. Introduction

Consider the following an unconstrained optimization problem in the form of

$$\min_{x \in \mathbb{R}^n} f(x)$$  \hspace{1cm} (1)
where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, and $\mathbb{R}^n$ denotes an $n$-dimensional Euclidean space. The conjugate gradient method is an iterative method, that is well-suited to solving problems (1) of the large scale (Polak, 1997). The iterative formula is given by

$$x_{k+1} = x_k + \alpha_k d_k, \ k = 0,1,2,...$$

where $x_k$ is the current iterate point, $d_k$ is a direction of $f$ at $x_k$, and $\alpha_k > 0$ is step-size obtained by one-dimensional line search. Step size $\alpha_k$ is obtained using several forms of line search, i.e., exact line search (Nocedal and Wright, 2006) as follows:

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha_k d_k)$$

or the strong Wolfe line search:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k, \ |g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k$$

with $0 < \delta < \sigma < 1$ (Wolfe, 1969).

The search direction $d_k$ on this gradient conjugate method uses the following rules:

$$d_k = \begin{cases} -g_k, & k = 0 \\ -g_k + \beta_k d_{k-1}, & k \geq 1 \end{cases}$$

where $g_k = \nabla f(x_k)$ is gradient $f$ at $x_k$, and $\beta_k \in \mathbb{R}$ is a scalar parameter known as the conjugate gradient coefficient.

Many formulas have been proposed to calculate the $\beta_k$. The most well-known are Hestenes-Stiefel (HS) (Hestenes and Stiefel, 1952), Fletcher-Reeves (FR) (Fletcher and Reeves, 1964), Polak-Ribiere-Polyak (PRP) (Polak and Ribiere, 1969), Conjugate Descent (CD) (Fletcher, 1987), Dai-Yuan (DY) (Dai and Yuan, 1999) and WYL (Wei-Yao-Liu) (Wei et al., 2006).

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}, \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2}, \beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}},$$

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T (g_k - g_{k-1})}, \beta_k^{WYL} = \frac{g_k^T (g_k - \frac{\|g_k\|}{\|g_{k-1}\|} g_{k-1})}{\|g_{k-1}\|^2}$$

where $\| \cdot \|$ denotes the Euclidean norm of vectors.

The global convergence properties are the most well-studied properties of CG methods. The HS method is one of the most efficient CG methods, which has good performance but no fulfill the global convergence properties under traditional line search. The FR method was developed from the HS method, which has a global convergent under exact and strong Wolfe line search (Al-Baali, 1985). The CD method has descent direction under the strong Wolfe line search (Liu et al., 2012) and fulfills the sufficient descent condition under the strong Wolfe line search. Dai and Yuan proposed the DY method, which has convergent globally under the strong Wolfe line search and Armijo line search (Zhang, 2007). Wei et al. proposed a new CG coefficient, which is a modification of the $\beta_k^{FR}$. This method known to fulfill the sufficient descent condition and global convergence properties under exact line search, Grippo-Lucidi line search, and strong Wolfe line search. For good references to studies that have described recent CG methods with important results, see Rivaie et al., 2015, Yousif, 2020, Basri and Mamat, 2018, Waziri et al., 2019, Yuan et al., 2020, Liu, 2013, and Babaie-Kafaki, 2014.

In this paper, we will present our new CG algorithm whose performance compared with classical formulas of the FR, CD, DY, and WYL. Section 2 presents a new CG algorithm to solve unconstrained optimization problems. In Section 3, we shall present the sufficient descent condition and the global convergence. Significant numerical results and discussions will be presented in Section 4. Finally, the conclusions are presented in Section 5.
2. A New CG Algorithm

Recently, Malik et al., 2020, proposed the new hybrid conjugate gradient coefficient. The coefficient defined as follows:

\[ \beta_k^{H-MS2} = \max\{0, \min\{\beta_k^{RMIL}, \beta_k^*\}\}, \]

where \( \beta_k^{RMIL} = \frac{g_k^T(g_k - g_{k-1})}{\|d_{k-1}\|^2} \) (Rivaie et al., 2012) and

\[ \beta_k^* = \frac{g_k^T(g_k - g_{k-1})}{g_k^T(g_{k-1} - d_{k-1})}. \] \hspace{1cm} (7)

In this paper, we propose to compute \( \beta_k \) using only \( \beta_k^* \).

The algorithm is given as follows:

**Algorithm 1**

**Step 1.** Initialization. Given \( x_0 \in \mathbb{R}^n \), set \( k := 0 \) and stopping criteria \( \varepsilon > 0 \).

**Step 2.** If \( \|g_k\| \leq \varepsilon \), then stop; otherwise, go to Step 3.

**Step 3.** Calculate the step size \( \beta_k \) using (7).

**Step 4.** Calculate \( d_k \) using (6).

**Step 5.** Calculate \( \alpha_k \) using (3).

**Step 6.** Set \( k := k + 1 \), and calculate the next iterate, go to step 3.

3. Convergence Analysis

This section will show the sufficient descent condition and the global convergent properties of the new method using the exact line search.

3.1 Sufficient Descent Condition

Sufficient descent condition holds when

\[ g_k^T d_k \leq -C \|g_k\|^2, \text{ for } k \geq 0 \text{ and } C > 0. \] \hspace{1cm} (8)

**Theorem 1.** Consider a CG method with search direction \( d_k \) (6), \( \beta_k^* \) given as equation (7), then, the condition (8) will hold for all \( k \geq 0 \).

**Proof.** If \( k = 0 \), then \( d_0 = -g_0 \), so \( g_0^T d_0 = g_0^T(-g_0) = -\left(\sqrt{g_0^T g_0}\right)^2 = -\|g_0\|^2 < 0 \). Hence, condition (8) holds true for \( k = 0 \). Next, we will show that for \( k \geq 1 \), condition (8) will hold true.

Multiply (6) by \( g_k^T \), then

\[ g_k^T d_k = -g_k^T g_k + \beta_k^* g_k^T d_{k-1} = -\|g_k\|^2 + \beta_k^* g_k^T d_{k-1}. \]

We know that for exact line search, \( g_k^T d_{k-1} = 0 \). Thus,

\[ g_k^T d_k = -\|g_k\|^2 < 0. \]

Hence, condition (8) holds for \( k \geq 1 \). The proof is finished.
3.2 Global Convergence Properties

Next, we will show that a new CG method with coefficient $\beta_k^*$ fulfill the convergence properties. In analyzing the global convergence properties of CG methods, the following basic assumptions are often required.

**Assumption 1.** (A1) The level set $\Omega = \{ x \in \mathbb{R}^n : f(x) \leq f(x_0) \}$ is bounded, where $x_0$ is a given starting point. (A2) In an open convex set $\Omega_0$ that contains $\Omega$, $f$ is continuous and differentiable, and its gradient is Lipschitz continuous; that is, for any $x, y \in \Omega_0$, there exists a constant $L > 0$ such that $\| g(x) - g(y) \| \leq L \| x - y \|$.

This Assumption yields the following lemma, which Zoutendijk had proved (Zoutendijk, 1970).

**Lemma 1.** Suppose that Assumptions 1 hold, let $x_k$ be generated by Algorithm 1, where $d_k$ is a descent search direction, and $\alpha_k$ is obtained by (3), then the following condition, known as the Zoutendijk condition, holds

$$
\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.
$$

By using Lemma 1, the following convergent theorem for the new CG method.

**Theorem 2.** Suppose Assumption 1 hold, $\{x_k\}$ generated by Algorithm 1, where the step size $\alpha_k$ is determined by the exact line (3), then:

$$
\lim_{k \to \infty} \inf \| g_k \| = 0.
$$

**Proof.** Proof by contradiction. Suppose (9) is not correct, then there is a constant $M > 0$ such that

$$
\| g_k \| \geq M, \text{ for every } k \geq 0.
$$

(10)

Rewriting (6),

$$
d_k + g_k = \beta_k^* d_{k-1},
$$

and squaring both sides, we obtain

$$
\| d_k + g_k \|^2 = \| \beta_k^* d_{k-1} \|^2
$$

$$
\Leftarrow \| d_k \|^2 + \| g_k \|^2 + 2 g_k^T d_k = (\beta_k^*)^2 \| d_{k-1} \|^2
$$

$$
\Leftarrow \| d_k \|^2 = (\beta_k^*)^2 \| d_{k-1} \|^2 - 2 g_k^T d_k - \| g_k \|^2.
$$

Using $(g_k^T d_k)^2$ and dividing both sides, we get

$$
\frac{\| d_k \|^2}{(g_k^T d_k)^2} = \frac{(\beta_k^*)^2 \| d_{k-1} \|^2}{(g_k^T d_k)^2} - \frac{2 \| g_k \|^2}{(g_k^T d_k)^2} = \left( \frac{1}{\| g_k \|} \right)^2 + \left( \frac{1}{\| g_k \|} \right)^2 \frac{\| g_k \|^2}{(g_k^T d_k)^2} + \frac{1}{\| g_k \|^2} \leq (\beta_k^*)^2 \frac{\| d_{k-1} \|^2}{(g_k^T d_k)^2} + \frac{1}{\| g_k \|^2} \leq \frac{1}{\| g_k \|^2}.
$$

So that,

$$
\frac{\| d_k \|^2}{(g_k^T d_k)^2} \leq \frac{1}{\| g_k \|^2}
$$

or

$$
\frac{(g_k^T d_k)^2}{\| d_k \|^2} \geq \| g_k \|^2.
$$

Applying (10), then

$$
\frac{(g_k^T d_k)^2}{\| d_k \|^2} \geq M^2.
$$
Take summation, we have
\[ \sum_{k=0}^{n} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \sum_{k=0}^{n} M^2 = M^2. \]

Hence,
\[ \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \lim_{n \to \infty} \sum_{k=0}^{n} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \lim_{n \to \infty} (n + 1) M^2 = \infty. \]

This contradicts the Zoutendijk condition in Lemma 1. Therefore, the proof is completed.

4. Numerical Experiments

In this section, we will show the new method’s numerical results compared with the other method of the FR, CD, DY, and WYL. We will use some of the test problems considered in (Andrei, 2008) under low, medium, and high dimensions as in (Malik et al., 2020), namely, 2, 3, 4, 10, 50, 100, 500, 1000 and 10,000 to show the efficiency. The function used is an artificial function. Artificial functions are used to see algorithmic behavior in different situations such as the length of the narrow valleys, unimodal functions, and functions with large number of significant local optimal.

There are thirty-one nonlinear functions to be tested in this paper, as listed in Table 1. Furthermore, for each test dimension of the test function, one of which is the initial point that suggested by Andrei (Andrei, 2008). The comparison each method is based on the number of iterations (NOI) and the time in seconds required for running each of the test problems (CPU). The tests' evaluation was based on Nocedal-Wright line search algorithm for exact condition and coded in MATLAB under stopping criterion is set to \( \|g_k\| \leq 10^{-6} \). The test was carried out on a laptop with processor intel® Core™ i7-8550U CPU @ 1.80GHz (8CPUs), ~2.0 GHz, 16 GB for RAM and Windows 10 Professional 64bit operating system.

<table>
<thead>
<tr>
<th>No</th>
<th>Test Function</th>
<th>No</th>
<th>Test Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extended White &amp; Holst</td>
<td>17</td>
<td>Six hump camel</td>
</tr>
<tr>
<td>2</td>
<td>Extended Rosenbrock</td>
<td>18</td>
<td>Booth</td>
</tr>
<tr>
<td>3</td>
<td>Extended Freudenstein &amp; Roth</td>
<td>19</td>
<td>Trecanni</td>
</tr>
<tr>
<td>4</td>
<td>Extended Beale</td>
<td>20</td>
<td>Zettl</td>
</tr>
<tr>
<td>5</td>
<td>Extended Wood</td>
<td>21</td>
<td>Shallow</td>
</tr>
<tr>
<td>6</td>
<td>Raydan 1</td>
<td>22</td>
<td>Generalized Quartic</td>
</tr>
<tr>
<td>7</td>
<td>Extended Tridiagonal 1</td>
<td>23</td>
<td>Quadratic QF2</td>
</tr>
<tr>
<td>8</td>
<td>Diagonal 4</td>
<td>24</td>
<td>Generalized Tridiagonal 1</td>
</tr>
<tr>
<td>9</td>
<td>Extended Himmelblau</td>
<td>25</td>
<td>Generalized Tridiagonal 2</td>
</tr>
<tr>
<td>10</td>
<td>FLETCHCR</td>
<td>26</td>
<td>POWER</td>
</tr>
<tr>
<td>11</td>
<td>Extended Powell</td>
<td>27</td>
<td>Quadratic QF1</td>
</tr>
<tr>
<td>12</td>
<td>NONSCOMP</td>
<td>28</td>
<td>Extended quadratic penalty QP2</td>
</tr>
<tr>
<td>13</td>
<td>Extended DENSCHNB</td>
<td>29</td>
<td>Extended quadratic penalty QP1</td>
</tr>
<tr>
<td>14</td>
<td>Extended Penalty</td>
<td>30</td>
<td>Matyas</td>
</tr>
<tr>
<td>15</td>
<td>Hager</td>
<td>31</td>
<td>Dixon and Price</td>
</tr>
<tr>
<td>16</td>
<td>Extended Maratos</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Comparison Between the New Method, FR, CD, DY, and WYL

<table>
<thead>
<tr>
<th>Funct.</th>
<th>Dim.</th>
<th>Point</th>
<th>New</th>
<th>FR</th>
<th>CD</th>
<th>DY</th>
<th>WYL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>NOI</td>
<td>CPU</td>
<td>NOI</td>
<td>CPU</td>
<td>NOI</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>(-1.2,1)</td>
<td>23</td>
<td>0.959</td>
<td>26</td>
<td>0.8585</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(10,10)</td>
<td>39</td>
<td>1.4218</td>
<td>293</td>
<td>9.7359</td>
<td>222</td>
</tr>
<tr>
<td>Func, Dim.</td>
<td>Point</td>
<td>New</td>
<td>FR</td>
<td>CD</td>
<td>DY</td>
<td>WYL</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>53</td>
<td>30</td>
<td>1000</td>
<td>30</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>12</td>
<td>50</td>
<td>1000</td>
<td>50</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>10</td>
<td>40</td>
<td>1000</td>
<td>40</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>9</td>
<td>30</td>
<td>1000</td>
<td>30</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>7</td>
<td>20</td>
<td>1000</td>
<td>20</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>6</td>
<td>10</td>
<td>1000</td>
<td>10</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>5</td>
<td>5</td>
<td>1000</td>
<td>5</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>4</td>
<td>1</td>
<td>1000</td>
<td>1</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>3</td>
<td>0.5</td>
<td>1000</td>
<td>0.5</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>2</td>
<td>0.1</td>
<td>1000</td>
<td>0.1</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>1</td>
<td>0.01</td>
<td>1000</td>
<td>0.01</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.1</td>
<td>0.1</td>
<td>1000</td>
<td>0.1</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.05</td>
<td>0.05</td>
<td>1000</td>
<td>0.05</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.01</td>
<td>0.01</td>
<td>1000</td>
<td>0.01</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.005</td>
<td>0.005</td>
<td>1000</td>
<td>0.005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.001</td>
<td>0.001</td>
<td>1000</td>
<td>0.001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.0005</td>
<td>0.0005</td>
<td>1000</td>
<td>0.0005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.0001</td>
<td>0.0001</td>
<td>1000</td>
<td>0.0001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.00005</td>
<td>0.00005</td>
<td>1000</td>
<td>0.00005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.00001</td>
<td>0.00001</td>
<td>1000</td>
<td>0.00001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.000005</td>
<td>0.000005</td>
<td>1000</td>
<td>0.000005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.000001</td>
<td>0.000001</td>
<td>1000</td>
<td>0.000001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.0000005</td>
<td>0.0000005</td>
<td>1000</td>
<td>0.0000005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.0000001</td>
<td>0.0000001</td>
<td>1000</td>
<td>0.0000001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.00000005</td>
<td>0.00000005</td>
<td>1000</td>
<td>0.00000005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.00000001</td>
<td>0.00000001</td>
<td>1000</td>
<td>0.00000001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.000000005</td>
<td>0.000000005</td>
<td>1000</td>
<td>0.000000005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>5000</td>
<td>(1,1)</td>
<td>0.000000001</td>
<td>0.000000001</td>
<td>1000</td>
<td>0.000000001</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>10000</td>
<td>(1,1)</td>
<td>0.0000000005</td>
<td>0.0000000005</td>
<td>1000</td>
<td>0.0000000005</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1000</td>
<td>(1,1)</td>
<td>0.0000000001</td>
<td>0.0000000001</td>
<td>1000</td>
<td>0.0000000001</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>
The numerical results are combined using the profile results described in Dolan and More (Dolan and More, 2002). The profile results are illustrated in Figures 1 and 2. The results in Figure 1 and Figure 2 are obtained in the following way:

\[ r_{p,s} = \frac{a_{p,s}}{\min\{a_{p,s} : s \in S\}} \]

where \( r_{p,s} \) is performance ratio, \( a_{p,s} \) is the number of iterations or CPU time, \( P \) is set to test, and \( S \) is set of solvers on the test set \( P \). Overall profile results can be obtained in the following ways:

\[ \rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\} \]

The function \( \rho_s(\tau) \) is the distribution function for the performance ratio, and \( \rho_s(\tau) \) is the probability for solver \( s \in S \), that a performance ratio \( r_{p,s} \) is within a factor \( \tau \in \mathbb{R} \) of the best possible ratio, and \( n_p \) is the number of functions.

![Figure 1. Performance Profile Based on Number of Iterations](image-url)
Based on performance profiles plots in Figure 1 and Figure 2, where Figure 1 shows that the new method at the top curve, and it also in Figure 2. Corresponding in Table 2, the numerical results based on the number of iterations and CPU time shows that, the DY only reaches 96.25%, the WYL 96.25% and the FR, CD, and NEW method 100%. Since the curve of the new method on the top curve and reaches all the test function, then the new method is more efficient compared with other methods.

5. Conclusion

In this paper, we propose a new conjugate gradient method. Under some assumptions, the global convergent and the sufficient descent condition of the new conjugate gradient method when it is applied under exact line search have been established. A numerical experiment has shown the new method under exact line search can be used successfully in the practical computations and have the best performance compared with the other CG methods, namely FR, CD, DY, and WYL methods.

Acknowledgments

We would like to thank the reviewer for their suggestions and comments. This work is supported the Ph.D. mathematics study group on optimization field in Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia

References


**Biographies**

**Maulana Malik** is currently a Lecturer at the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Indonesia, since 2016. He received his Bachelor of Science (2009), and Master of Science (2012) in Mathematics from Universitas Indonesia (UI), Indonesia, and he is currently (2019-present) a Ph.D. student at Universiti Sultan Zainal Abidin (UniSZA), Terengganu, Malaysia. His current research focuses on optimization include the conjugate gradient (CG), hybrid CG, and spectral CG methods. Maulana Malik is a member of Indonesian Mathematical Society (IndoMS), and in IAENG is a new member (236191) has been received in April 2019.

**Mustafa Mamat** is currently a Professor in the Faculty of Informatics and Computing at the Universiti Sultan Zainal Abidin (UniSZA) Malaysia since 2013. He was first appointed as a Lecturer at the Universiti Malaysia Terengganu (UMT), in 1999. He obtained his Ph.D. from the UMT in 2007 with a specialization in optimization. He was appointed as a Senior Lecturer in 2008 and as an Associate Professor in 2010 also the UMT. To date, he has published more than 380 research papers in various international journals and conferences. His research interest in applied
mathematics, with a field of concentration of optimization, includes conjugate gradient methods, steepest descent methods, Broyden's family, and quasi-Newton methods.

**Siti Sabariah Abas** is a lecturer at the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA) Malaysia. She obtained her Ph.D. from the Universiti Sains Malaysia (USM) in 2016 with a field in the numerical analysis include fluid dynamics.

**Ibrahim Mohammed Sulaiman** is currently a post-doctoral researcher at the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA) Malaysia, from 2019 till date. He obtained his Ph.D. from UniSZA in 2018, specializing in the field of fuzzy systems. He has published research papers in various international journals and attended international conferences. His research interest includes numerical research, Fuzzy nonlinear systems and unconstrained optimization.

**Sukono** was born in Ngawi, East Java, Indonesia, on April 19, 1956. Master’s in Actuarial Sciences at Institut Teknologi Bandung, Indonesia in 2000, and Ph.D. in Financial Mathematics at the Universitas Gajah Mada, Yogyakarta Indonesia, in 2011. The current work is the Chairman of the Master Program in Mathematics, Universitas Padjadjaran, Bandung Indonesia. The research is in the field of financial mathematics and actuarial science. Dr. Sukono is a member of Indonesian Mathematical Society (IndoMS), a member of Indonesian Operations Research Association (IO Ara), and in IAENG is a new member was received in February 2016.

**Abdul Talib Bon** is a professor of Production and Operations Management in the Faculty of Technology Management and Business at the Universiti Tun Hussein Onn Malaysia since 1999. He has a Ph.D. in Computer Science, which he obtained from the Universite de La Rochelle, France, in the year 2008. His doctoral thesis was on topic Process Quality Improvement on Beltline Moulding Manufacturing. He studied Business Administration in the Universiti Kebangsaan Malaysia, for which he was awarded the MBA in the year 1998. He’s a bachelor’s degree and a diploma in Mechanical Engineering, which his obtained from the Universiti Teknologi Malaysia. He received his postgraduate certificate in Mechatronics and Robotics from Carlisle, United Kingdom in 1997. He had published more 150 International Proceedings and International Journals and eight books. He is a member of MSORSM, IIF, IEOM, IIE, INFORMS, TAM, and MIM.