A MODIFIED HESTENES-STIEFEL METHOD FOR SOLVING UNCONSTRAINED OPTIMIZATION PROBLEMS

ALKHOU LI T ALAT¹, IBRAHIM MOHAMMED SULAIMAN², MUSTAFA MAMAT²,* AND MOHD RIVAIE³

¹Applied Science Department, Aqaba University College, Balqa Applied University, Jordan
²Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut, 22200, Terengganu Malaysia
³Department of Computer Science and Mathematics, Universiti Technologi MARA (UiTM), Kuala Terengganu Campus, Malaysia

Copyright © 2020 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: The conjugate gradient methods are among the most efficient methods for solving optimization models. This is due to its simplicity, low memory requirement and the properties of its global convergent. Many researchers try to improve this technique. In this paper, we suggested a modification of the conjugate gradient parameter with global convergence properties via exact minimization rule. Preliminary experiment was conducted using some unconstrained optimization benchmark problems. Numerical outcome showed that the new algorithm is efficient and promising as it performs better than other classical methods both in terms of number of iteration and CPU time.

Keywords: conjugate gradient methods, global convergence, line search technique, unconstrained optimization.

2010 AMS Subject Classification: 65K05, 90C52.

*Corresponding author
E-mail address: must@unisza.edu.my
Received July 13, 2020
1. Introduction

Conjugate gradient (CG) method is considered as an important tool for solving unconstrained optimization problems. It can be applied in many fields like industry, medical treatment and economics because of its low memory requests and global convergence properties (see [2,11,15]). Generally, the optimization problem can be express as

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

where $f: \mathbb{R}^n \to \mathbb{R}$ is smooth. The CG methods computes it iterates $x_k$ starting from an initial point $x_0 \in \mathbb{R}^n$ as follows

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

where the step-size ($\alpha_k$) can be obtained using a line search method along the search direction $d_k$. The most preferred line search algorithm is the exact minimization condition:

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

where $d_k$ is given by

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

where $\beta_k$ is a scalar and $g_k = g(x_k)$.

The first CG algorithm was suggested by Hestenes and Stiefel (HS) [12] in (1952). Later, the Hestenes-Stiefel algorithm was improved to solve (1). The HS method is characterized by its formula

$$ \beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}}. $$

Other known CG coefficients are presented in Table 1.
Table 1: Some well-known CG methods

\[
\beta_k^{FR} = \frac{g_k^T g_k}{g_k^T g_{k-1}} \quad \text{ (Fletcher–Reeves [10], 1964)}
\]

\[
\beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} \quad \text{ (Polak-Ribiere–Polyak [20,21], 1969)}
\]

\[
\beta_k^{CD} = \frac{g_k^T g_k}{d_k^T g_{k-1}} \quad \text{ (Conjugate Descent [9], 1987)}
\]

\[
\beta_k^{LS} = -\frac{g_k^T (g_k - g_{k-1})}{d_k^T g_{k-1}} \quad \text{ (Liu–Storey [16], 1991)}
\]

\[
\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}} \quad \text{ (Dai–Yuan, [8], 1999)}
\]

There are several researches about convergence properties of these methods (see [3,4,5,15,22,29,31,35]). Some convergent formulas are proposed by restricting the scalar to a nonnegative number [19]. The convergence analysis for the methods of HS, LS and PRP are yet to be established under other line searches. (see [13,32]). Some practical application of the optimization method can be referred to [28].

Recently, many researchers have studied CG methods. Table 2 provides a list of recent CG methods.

Table 2: Several choices for update CG methods parameter

\[
\beta_k^{RMI\text{L}} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2} \quad \text{ (Rivaie et al. [23], 2013)}
\]

\[
\beta_k^{ARMI} = \frac{\|g_k\|^2 - \|d_{k-1} + g_k\|}{\|d_{k-1}\|^2} \left| g_k^T g_{k-1} \right| \quad \text{ (Abashar et al. [1], 2014)}
\]

\[
\beta_k^{KM\text{AR}} = \frac{g_k^T (g_k - g_{k-1})}{g_k^T (g_k + g_{k-1})} \quad \text{ (Kamfa et al. [14], 2015)}
\]
A MODIFIED HS METHOD FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

\( \beta_k^{NRMI} = \frac{g_k^T(g_k - g_{k-1})}{g_k^T(g_k - d_{k-1})} \) \quad (Shapiee et al. [24], 2016)

\( \beta_k^{RMAR} = \frac{g_k^T(g_k - \|g_k\|/\|d_{k-1}\| d_{k-1})}{\|d_{k-1}\|^2} \) \quad (Mamat et al. [17], 2017)

\( \beta_k^{MMM} = \frac{\|g_k\|}{d_{k-1}(d_{k-1} - g_k)} \) \quad (Mandara et al. [18], 2018)

2. NEW FORMULA FOR \( \beta_k \)

In early 21st century, tremendous efforts have been made by researchers to improve the CG methods. The researchers suggested numerous variants of CG methods with strong convergence properties and efficient numerical results. A survey of the CG methods is given by Andrei [6].

Lately, Wei et al. [30] introduce a variation of the PRP coefficient referred to the WYL method.

\( \beta_k^{WYL} = \frac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2} \frac{\|g_k\|}{\|g_{k-1}\|} \).

Motivated by the ideas of [12,30], we introduce our \( \beta_k \) known as \( \beta_k^{TM*} \), where TM* represents Tala’t and Mustafa. The new \( \beta_k^{TM*} \) is a variant of HS method which is as follows:

\( \beta_k^{TM*} = \frac{g_k^T(m(g_k - g_{k-1}))}{m(g_k - g_{k-1})^T d_{k-1}} \), where \( m = \frac{\|g_{k-1}\|}{\|g_k\|} \).

The algorithm of the proposed coefficient is as follows:

**Algorithm 1:** Algorithm for CG coefficient \( \beta_k^{TM*} \)

Stage 1: Initialization. Given \( x_0 \in R^n, \varepsilon \geq 0 \), set \( d_0 = -g_0 \) if \( \|g_0\| \leq \varepsilon \) then stop.

Stage 2: Compute \( \alpha_k \) by (3).

Stage 3: Let \( x_{k+1} = x_k + \alpha_k d_k \), \( g_{k+1} = g(x_{k+1}) \) if \( \|g_{k+1}\| \leq \varepsilon \) then stop.

Stage 4: Calculate \( \beta_k \) by (5), and produce \( d_{k+1} \) by (4).

Stage 5: let \( k = k + 1 \) go to Stage 2.
3. CONVERGENCE ANALYSIS

An important condition for the convergence analysis of any CG algorithm is satisfying the sufficient descent condition (SDC) [2,27].

2.1. Sufficient descent condition

For the SDC to hold,

\[ g_k^T d_k \leq -C \|g_k\|^2 \quad \text{for} \quad k \geq 0 \quad \text{and} \quad C > 0. \]  (6)

**Theorem 1**

Consider a CG method with \( d_k \) defined by (4) and \( \beta_k^{TM^*} \) specified as (5), then (6) holds for all \( k \geq 0 \).

**Proof.**

If \( k = 0 \), then \( g_0^T d_0 = -C \|g_0\|^2 \). So, condition (6) holds true. For \( k \geq 1 \),

\[
g_k^T d_k = g_k^T (-g_k + \beta_k d_{k-1})
\]

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

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

\[ g_k^T d_k = -\|g_k\|^2. \]  (7)

Hence, \( g_k^T d_k \leq -C \|g_k\|^2 \) holds true. The proof is completed.  

2.2. Global convergence

To establish the convergence properties of the method of \( \beta_k^{TM^*} \), we need to simplify \( \beta_k^{TM^*} \) to make the proof easier. From (5) we can see that

\[
\beta_k^{TM^*} = \frac{g_k^T (m(g_k - g_{k-1}))}{m(g_k - g_{k-1})^T d_{k-1}}
\]

\[ = \frac{m \|g_k\|^2 - m g_k^T g_{k-1}}{m(g_k - g_{k-1})^T d_{k-1}} \leq \frac{\|g_k\|^2}{(g_k - g_{k-1})^T d_{k-1}}. \]
Hence, we get

\[ \beta_k^{TM^*} \leq \frac{\|g_k\|^2}{(g_k - g_{k-1})^T d_{k-1}}. \]

For the convergence of CG methods, next assumptions are always needed.

**Assumption 1.**

i. \( f(x) \) is bounded below on the level set \( R^n \) and differentiable in a neighborhood \( N \) of the level set \( \ell = \{x \in R^n: f(x) \leq f(x_0)\} \) at the initial point \( x_0 \).

ii. The gradient \( g(x) \) is Lipschitz continuous in \( N \), i.e.,

\[ \exists L > 0 \text{ s.t. } \|g(x) - g(y)\| \leq L\|x - y\| \quad \forall x, y \in N. \]

Under this Assumption, we have the next lemma, that was proven by Zoutendijk [33].

**Lemma 1.**

Let Assumption 1 holds true for any CG method of the form (1), with search direction \( d_k \) and \( \alpha_k \) fulfills (3). Then the following condition knowns as the Zoutendijk condition, holds

\[ \sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \]

which is equivalent to

\[ \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty. \]

**Lemma 2**

Let Assumption 1 holds, \( \{x_k\} \) generated by the Algorithm 1, and \( \alpha_k \) is calculated by (3). Then Lemma 1 holds for all \( k \geq 0 \).

**Proof.**

Let \( \forall k, \ g_k \neq 0 \). If \( k = 0 \) then

\[ g_0^T d_0 = g_0^T (-g_0) = -\|g_0\|^2. \]
Let a point \( x_k \) and \( d_k \) is not a descent direction then we have \( x_k = x_{k-1} \), which implies \( g_k = g_{k-1} \). From (5), we have
\[
\beta_k^T M^* = 0.
\]
That means those points become the steepest descent directions and denoted by \( N_1 = \{ x_k | \beta_k^T M^* = 0 \} \) and the other points are denoted by \( N_2 = \{ x_k | \beta_k^T M^* \neq 0 \} \). For all points in \( N_1 \), from Lemma 1, we have
\[
\sum_{x_k \in N_1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{9}
\]
The same as the above proof, for the points \( N_2 \), we also have
\[
\sum_{x_k \in N_2} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{10}
\]
So
\[
\sum_{k \geq 1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \sum_{x_k \in N_1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} + \sum_{x_k \in N_2} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty.
\]
The proof is completed. \( \blacksquare \)

By Lemma 1 and using (8), we obtain the following convergence theorem.

**Theorem 2**

Suppose that Assumption 1 is holds for any CG method in the form of (2), (4), and (8), where \( \alpha_k \) is obtained by (3). If the descent condition holds true. Then either
\[
\lim_{k \to \infty} \|g_k\| = 0 \quad \text{or} \quad \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty
\]

**Proof:**

From (4)
\[ \|d_k\|^2 = -\|g_k\|^2 - 2g_k^T d_k + \beta_k^2 \|d_{k-1}\|^2 \]

and from (7), implies
\[ \|d_k\|^2 = \|g_k\|^2 + \beta_k^2 \|d_{k-1}\|^2, \]

applying (8), we have
\[ \|d_k\|^2 = \|g_k\|^2 + \frac{\|g_k\|^4}{((g_k - g_{k-1})^T d_{k-1})^2} \|d_{k-1}\|^2, \]

therefore,
\[ \frac{\|d_k\|^2}{\|g_k\|^4} - \frac{1}{((g_k - g_{k-1})^T)^2} = \frac{1}{\|g_k\|^2}. \]

Also,
\[ \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \leq \sum_{k=1}^{\infty} \|g_k\|^2 \]

that is, we have
\[ \lim_{k \to \infty} \|g_k\| = 0. \]

Hence, the proof is completed. ■

4. Numerical Results

To illustrate the efficiency of the proposed \( TM^* \), we compare its performance with that of FR, WYL and RMIL methods based on iteration number and CPU time. Table 3 displays some classical test problems, dimensions and the initial points considered for the experiments. Most of the selected test functions are from Andrei [6]. We choose \( \varepsilon = 10^{-6} \) and the termination criteria is set as \( \|g_k\| \leq \varepsilon \) as suggested by Hillstron [13]. Three random initial guesses are used; starting from the points near the solution points, to a point far from it. All standard optimization test problems are tested in a small to large-scale dimension. If the line search fails to obtain the positive \( \alpha_k \) in some cases, the computation stopped [25,26]. The performance was displayed in Figure 1 and Figure 2 based on performance profile introduced by [8].
<table>
<thead>
<tr>
<th>NO</th>
<th>Function</th>
<th>Dim</th>
<th>Initial point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIX HUMP</td>
<td>2</td>
<td>(0.5,0.5), (8,8), (40,40)</td>
</tr>
<tr>
<td>2</td>
<td>THREE HUMP</td>
<td>2</td>
<td>(-1.1), (-1.1), (2,-2)</td>
</tr>
<tr>
<td>3</td>
<td>QUADRATIC QF1</td>
<td>2</td>
<td>(3,3), (5,5), (10,10)</td>
</tr>
<tr>
<td>4</td>
<td>DIAGONAL 2</td>
<td>2</td>
<td>(1,1), (5,5), (15,15)</td>
</tr>
<tr>
<td>5</td>
<td>LEON</td>
<td>2</td>
<td>(2,2), (4,4), (8,8)</td>
</tr>
<tr>
<td>6</td>
<td>MATYAS</td>
<td>2</td>
<td>(5,5), (10,10), (15,15)</td>
</tr>
<tr>
<td>7</td>
<td>BOOTH</td>
<td>2</td>
<td>(10,10), (25,25), (100,100)</td>
</tr>
<tr>
<td>8</td>
<td>RAYDAN</td>
<td>2</td>
<td>(3,3), (13,13), (23,23)</td>
</tr>
<tr>
<td>9</td>
<td>ZETTL</td>
<td>2</td>
<td>(5,5), (20,20), (50,50)</td>
</tr>
<tr>
<td>10</td>
<td>TRECANNI</td>
<td>2</td>
<td>(5,5), (10,10), (50,50)</td>
</tr>
<tr>
<td>11</td>
<td>EXTENDED WOOD</td>
<td>4</td>
<td>(5,...,5), (20,...,20), (30,...,30)</td>
</tr>
<tr>
<td>12</td>
<td>CLOVILLE</td>
<td>4</td>
<td>(2,...,2), (4,...,4), (10,...,10)</td>
</tr>
<tr>
<td>13</td>
<td>HAGER</td>
<td>2</td>
<td>(7,7), (15,15), (20,20)</td>
</tr>
<tr>
<td>14</td>
<td>EXTENDED PENALTY</td>
<td>2</td>
<td>(40,40), (80,80), (100,100)</td>
</tr>
<tr>
<td>15</td>
<td>DIXON &amp; PRICE</td>
<td>2, 4</td>
<td>(6,6), (30,30), (125,125), (30,...,30), (125,...,125)</td>
</tr>
<tr>
<td>16</td>
<td>ARWHEAD</td>
<td>2, 10</td>
<td>(8,8), (24,24), (48,48), (24,...,24), (48,...,48)</td>
</tr>
<tr>
<td>17</td>
<td>QUARTC</td>
<td>2, 10</td>
<td>(8,8), (16,16), (30,30), (16,...,16), (30,...,30)</td>
</tr>
<tr>
<td>18</td>
<td>QUADRATIC QF2</td>
<td>2, 10</td>
<td>(4,4), (40,40), (80,80), (40,...,40), (80,...,80)</td>
</tr>
<tr>
<td>19</td>
<td>EX QUADRATIC PENALTY QP2</td>
<td>2, 10</td>
<td>(10,10), (20,20), (30,30), (10,...,10), (30,...,30)</td>
</tr>
<tr>
<td>20</td>
<td>EXTENDED POWELL</td>
<td>100, 1000</td>
<td>(2,...,2), (4,...,4), (8,...,8)</td>
</tr>
<tr>
<td>21</td>
<td>GENERLIZED TRIDIAGONAL 1</td>
<td>2, 10</td>
<td>(3,3), (21,21), (90,90), (21,...,21), (90,...,90)</td>
</tr>
<tr>
<td>22</td>
<td>GENERLIZED TRIDIAGONAL 2</td>
<td>10, 100</td>
<td>(15,...,15), (30,...,30), (150,...,150)</td>
</tr>
<tr>
<td>23</td>
<td>ROSENBROCK</td>
<td>2, 10, 100, 1000</td>
<td>(3,...,3), (15,15), (75,75), (3,...,3), (75,...,75)</td>
</tr>
<tr>
<td>24</td>
<td>SHALLOW</td>
<td>2, 10, 100, 1000</td>
<td>(-2,-2), (12,12), (200,200), (200,...,200)</td>
</tr>
<tr>
<td>25</td>
<td>EXTENDED WHITE &amp; HOLST</td>
<td>2, 10, 100, 1000</td>
<td>(-3,-3), (6,6), (10,10), (-3,...,3), (6,...,6), (10,...,10)</td>
</tr>
<tr>
<td>26</td>
<td>EXT FREUDENTSTEIN &amp; ROTH</td>
<td>2, 10, 100, 1000</td>
<td>(2,2), (25,25), (30,30), (2,...,2), (25,...,25), (30,...,3)</td>
</tr>
<tr>
<td>27</td>
<td>EXTENDED BEALE</td>
<td>2, 10, 100, 1000</td>
<td>(-1,-1), (7,7), (11,11), (-1,...,-1), (7,...,7), (11,...,11)</td>
</tr>
<tr>
<td>28</td>
<td>PERTURBED QUADRATIC</td>
<td>2, 10, 100, 1000</td>
<td>(1,1), (5,5), (10,10), (1,...,1), (5,...,5), (10,...,10)</td>
</tr>
<tr>
<td>29</td>
<td>EXTENDED TRIDIAGONAL 1</td>
<td>2, 10, 100, 1000</td>
<td>(25,25), (50,50), (75,75), (25,...,25), (50,...,50),</td>
</tr>
<tr>
<td>30</td>
<td>DIAGONAL 4</td>
<td>2, 10, 100, 1000</td>
<td>(1,1), (20,20), (40,40), (1,...,1), (20,...,20), (40,...,4)</td>
</tr>
<tr>
<td>31</td>
<td>EXTENDED HIMMELBLAU</td>
<td>2, 10, 100, 1000</td>
<td>(10,10), (50,50), (125,125), (10,...,10), (125,...,12)</td>
</tr>
<tr>
<td>32</td>
<td>FLETCHCR</td>
<td>2, 10, 100, 1000</td>
<td>(12,12), (15,15), (35,35), (12,...,12), (35,...,35)</td>
</tr>
<tr>
<td>33</td>
<td>EXTENDED DENSCHNB</td>
<td>2, 10, 100, 1000</td>
<td>(5,5), (30,30), (50,50), (5,...,5), (30,...,30), (50,...,5)</td>
</tr>
<tr>
<td>34</td>
<td>EXT BLOCK DIAGONAL BD1</td>
<td>2, 10, 100, 1000</td>
<td>(1,1), (5,5), (10,10), (1,...,1), (5,...,5), (10,...,10)</td>
</tr>
<tr>
<td>35</td>
<td>GENERLIZED QUARTIC</td>
<td>2, 10, 100, 1000</td>
<td>(7,7), (70,70), (140,140), (7,...,7), (140,...,140)</td>
</tr>
<tr>
<td>36</td>
<td>SUM SQUARES</td>
<td>2, 10, 100, 1000</td>
<td>(1,1), (5,5), (10,10), (1,...,1), (5,...,5), (10,...,10)</td>
</tr>
</tbody>
</table>
Using Dolan and More profile we compare and evaluate the performance of the 4 algorithms. Supposing \( n_s \) solvers and \( n_p \) problems exists, for every problem \( p \) and solver \( s \), Dolan and More defined by:

\[
\tau_{p,s} = \text{calculating time (NO.IT. or CPU time) necessary to solve problem} \ p \ \text{by solver} \ s.
\]

Both figures above illustrate that \( TM^* \) is the best solver, as it can solve all of the test problems and reach 100% percentage. Comparing with 90% for FR, 97% for WYL, and 96% for RMIL of the given test problems. To sum up, our numerical outcomes show that the \( TM^* \) technique is efficient, modest to the typical CG method and owns nice convergence properties under exact line search.

5. CONCLUSION
In this paper, we present a new modification of the CG coefficient that guaranteed the sufficient descent condition provided exact line search is used. The global convergence of the proposed MS method was established under the exact line search. Numerical results reported have shown that the proposed coefficient is efficient and robust when compared to other CG methods. Future research can focus on investigating the performance of improved version of the conjugate gradient coefficient giving a wider scope on step length.
ACKNOWLEDGMENTS
The authors are grateful to Malaysian government for supporting this research under the grant number (FRGS/1/2017/STGO6/Unisza/01/1) and also Universiti Sultan Zainal Abidin, Malaysia.

CONFLICT OF INTERESTS
The authors declare that there is no conflict of interests.

REFERENCES
A MODIFIED HS METHOD FOR UNCONSTRAINED OPTIMIZATION PROBLEMS

(1997).


