Solving a large scale nonlinear unconstrained optimization problems by using new coefficient of conjugate gradient method with exact line search direction

Talat Alkouli 1, Mustafa Mamat1*, Mohd Rivaie2, Puspa Liza Ghazali3

1Faculty of Informatics and Computing, University Sultan Zainal Abidin, 21030 Terengganu, Malaysia
2Department of Computer Science and Mathematics, Universiti Technology MARA (UITM), 23000 Terengganu, Malaysia
3Faculty of Economy and Management Sciences, University Sultan Zainal Abidin, 21030 Terengganu, Malaysia
*Corresponding author E-mail: must@unisza.edu.my

Abstract

In this paper, an efficient modification of nonlinear conjugate gradient method and an associated implementation, based on an exact line search, are proposed and analyzed to solve large-scale unconstrained optimization problems. The method satisfies the sufficient descent property. Furthermore, global convergence result is proved. Computational results for a set of unconstrained optimization test problems, some of them from CUTE library, showed that this new conjugate gradient algorithm seems to converge more stable and outperforms the other similar methods in many situations.

Keywords: Conjugate gradient; Exact line search; Global convergence; Sufficient descent condition; Unconstrained optimization.

1. Introduction

The conjugate gradient method is an important and efficient method to solve the unconstrained optimization problems, especially for large scale problems. Its wide application in many fields such as economics, industry and engineering is due to low memory requirements and global convergence properties.

\[
\min f(x), \ x \in \mathbb{R}^n
\]

where \(f: \mathbb{R}^n \to \mathbb{R}\) is continuously nonlinear differentiable function, bounded from below. A nonlinear conjugate gradient method generates a sequence \(x_k\) starting from an initial guess \(x_0 \in \mathbb{R}^n\), using the recurrence

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

where \(\alpha_k\) is the positive step size obtained by carrying out a one dimensional search, known as the ‘line searches’. The most common is the exact line search which is

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

and \(d_k\) is the search direction generated 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 the CG update parameter and \(g_k = \nabla f(x_k)\).

Different CG method relates to different choices \(\beta_k\). Table 1 organizes a chronological list of some choices for the classical CG parameter

Table 1: Various choices for the classical CG parameter

For strictly a convex quadratic objective functions, all methods in Table 1 under the exact line search have global convergence. But, for general non-quadratic functions or under the inexact line search, their behavior is relatively different [25, 26].
The global convergence property of FR method was proved by [8] when exact line search was used. This was later refuted in a counter example introduced by [9]. Meanwhile, in [11] emphasis that FR is a superior method compared to others. In [10] has also shown that the PRP method has no global convergence under some traditional line searches. In [19] success to introduce the first global convergence result of the FR method under the Strong Wolfe condition. CD method and DY method were also proved to have global convergence under Strong Wolfe line search [12, 13]. Some convergent versions were proposed by using some new complicated line searches, or through restricting the parameter to a nonnegative number [20]. However, to the best of our knowledge, the global convergence of PRP, LS and HS methods have not been proven under all mentioned line searches. The main reason is that many conjugate gradient methods cannot guarantee the descent of objective function values at each iterative.

Many researchers have studied conjugate gradient method, but the most important properties they well-studied are sufficient condition, global convergence. For example:

In [15] proposed her conjugate gradient parameter method which is an extension of LS method and the resulting algorithm was proven to have both the sufficient descent and global convergence properties under inexact line search as follows

\[ \beta_k^{\text{ARM}} = \frac{m_1 \langle g_k, g_k \rangle^2 - m_2 \langle g_k, g_k \rangle g_k \cdot g_k}{m_2 \langle g_k, g_k \rangle^2}, \]

where \( m_1 = \frac{\langle d_k, g_k \rangle + \langle g_k, g_k \rangle}{\langle d_k, g_k \rangle} \).

Table 2 provides a chronological list of some new choices for the CG update parameter.

<table>
<thead>
<tr>
<th>Method</th>
<th>Formula</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_k^{\text{SMIL}} )</td>
<td>( \frac{\langle g_k, g_k \rangle}{\langle d_k, g_k \rangle} )</td>
<td>Rivaie et al. [14], 2012</td>
</tr>
<tr>
<td>( \beta_k^{\text{MAR}} )</td>
<td>( \frac{\langle g_k, g_k \rangle}{\langle g_k, d_k \rangle} )</td>
<td>Kamfa et al. [16], 2015</td>
</tr>
<tr>
<td>( \beta_k^{\text{NRM1}} )</td>
<td>( \frac{\langle g_k, g_k \rangle}{\langle g_k - g_{k-1}, d_k \rangle} )</td>
<td>Shapiee et al. [17], 2016</td>
</tr>
<tr>
<td>( \beta_k^{\text{NRM2}} )</td>
<td>( \frac{\langle g_k, g_k \rangle}{\langle g_k - g_{k-1}, g_k - g_{k-1} \rangle} )</td>
<td>Hamoda et al. [18], 2017</td>
</tr>
</tbody>
</table>

Table 2: Various choices for the new CG update parameter

For non-quadratic objective functions, the global convergence property of all methods in Table 2 was proved when an exact line search was used.

### 2. New formula for \( \beta_k \) and its properties

During the last years, much effort has been devoted to develop new modifications of conjugate gradient methods, as we mention before, which do not possess strong convergence properties, but they are also computationally superior to the classical methods. Since then a large number of variations of Conjugate Gradient algorithms have been suggested. A survey on their definition including 40 nonlinear Conjugate Gradient algorithms for unconstrained optimization is given by [21].

Lately, in [22] gave a variant of the PRP method which is called the WYL method. In [23] relied on WYL to suggest a new conjugate gradient method, NPRP, and he proved that the NPRP method satisfied descent condition and the Global convergence property under strong Wolfe line search. Subsequently, in [24] introduced an improved NPRP method known as the DPRP method.

In this section, enlightened by previous ideas [22-24], we propose our \( \beta_k \) which known as \( \beta_k^{\text{TMR}} \), where TMR denotes Tala’i, Mustafa and Rivaie. The new \( \beta_k^{\text{TMR}} \) is a modification of PRP conjugate gradient method which is as follows:

\[ \beta_k^{\text{TMR}} = \frac{\langle n(g_k - g_{k-1}), g_k \rangle}{\langle g_k, g_k \rangle^2} \]

where \( n = \frac{1}{\| g_k \|} \).

The algorithm is given as follows:

**Algorithm 2.1**

- **Step 1:** Given \( x_0 \in \mathbb{R}^n, \varepsilon \geq 0 \), set \( d_0 = -g_0 \) if \( \| g_0 \| \leq \varepsilon \) then stop.
- **Step 2:** Compute \( \alpha_k \) by (3).
- **Step 3:** Let \( x_{k+1} = x_k + \alpha_k d_k \) and \( g_{k+1} = g(x_{k+1}) \) if \( \| g_{k+1} \| \leq \varepsilon \) then stop.
- **Step 4:** Compute \( \beta_k^{\text{TMR}} \) by (5), and generate \( d_{k+1} \) by (4).
- **Step 5:** set \( k = k + 1 \) go to Step 2.

### 3. Convergent analysis

In this section, the convergent properties of \( \beta_k^{\text{TMR}} \) will be studied. For an algorithm to be converge, it must satisfy the sufficient descent condition and the global convergence properties.

#### 3.1. Sufficient descent condition

For the sufficient condition to hold,

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

**Theorem 3.1.1.**

Consider a CG method with the search direction (4) and \( \beta_k^{\text{TMR}} \) given as in (5), then condition (6) holds for all \( k \geq 0 \).

**Proof:**

If \( k = 0 \), then \( g_0^T d_0 = -C \| g_0 \|^2 \)

Hence, condition (6) holds true.

For \( k \geq 1 \), multiply (4) by \( g_{k+1} \cdot \) then

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

\[ = -\| g_{k+1} \|^2 + \beta_{k+1} g_{k+1}^T g_{k+1} \]

For exact line search, we know that \( g_{k+1}^T d_{k+1} = 0 \). Thus,

\[ g_{k+1}^T d_{k+1} = -\| g_{k+1} \|^2 \]

which implies that \( d_{k+1} \) is a sufficient descent direction. Hence,

\[ g_k^T d_k \leq -C \| g_k \|^2 \text{ holds true.} \]

The proof is completed.

#### 3.2. Global convergence properties

In this section, we will show that CG methods with \( \beta_k^{\text{TMR}} \) converge globally. However, we first need to simplify our new \( \beta_k^{\text{TMR}} \) so that our convergence proof will be significantly easier. In (5), we can see that
\[
\beta_k^{\text{TMR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} - \frac{\|\frac{g_k - g_{k-1}}{\|g_k - g_{k-1}\|^2}\|^2}{\|g_{k-1}\|^2} \\
= \frac{\|g_k\|^2}{\|g_{k-1}\|^2} - \frac{\|\frac{g_k}{\|g_k\|} - \frac{g_{k-1}}{\|g_{k-1}\|}\|^2}{\|g_{k-1}\|^2} \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2}
\]

Hence, we obtain

\[0 \leq \beta_k^{\text{TMR}} \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2}\]  

(8)

The following straightforward assumptions are always needed in the analysis of global convergence properties of CG methods.

**Assumption A**

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

**Assumption B**

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

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

Under these assumptions, we have the following lemma which was proven by [8].

**Lemma 3.2.1.**

Suppose that Assumptions A and B holds true for any iteration method of the form (1), where \(d_k\) is a descent search direction and \(\alpha_k\) satisfies the exact minimization rule. Then, the following condition, known as the Zoutendijk condition holds

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

The proof of this lemma can be seen in [8].

**Lemma 3.2.2**

Suppose that Assumptions A and B hold, \(\{x_k\}\) generated by the Algorithm 2.1, where the step size is determined by the exact line search (3). Then, Lemma 3.2.1 holds for all \(k \geq 0\).

Proof:

Suppose that \(\forall k \neq 0\), if \(k = 0\), then \(g_k^T d_0 = g_k^T(-g_0) = -\|g_{k=0}\|^2\).

If at a point \(x_k\), \(d_k\) is not a descent direction then by the exact line search, we have

\[x_{k+1} = x_k, \text{ which implies } g_{k+1} = g_k\]

From (5), we have \(\beta_k^{\text{TMR}} = 0\).

This means that at those points the directions will turn out to be the steepest descent directions. Those points are denoted by \(N_1 = \{x_k | \beta_k^{\text{TMR}} = 0\}\) and the other points are denoted by \(N_2 = \{x_k | \beta_k^{\text{TMR}} \neq 0\}\).

For all the points in \(N_1\), since the directions are the steepest descent directions, from Lemma 3.2.1, we have

\[\sum_{x_k \in N_1} \left(\frac{g_k^T d_k}{\|d_k\|^2}\right)^2 < \infty\]

The same as the above proof, for the points \(N_2\), we also have

\[\sum_{x_k \in N_2} \left(\frac{g_k^T d_k}{\|d_k\|^2}\right)^2 < \infty\]

The proof is completed.

By using Lemma 3.2.1, we can obtain the following convergent theorem of the CG method using (8).

**Theorem 2.**

Suppose that Assumptions A and B holds true. For any CG method in the form of (1) and (4) combined with (8), where \(\alpha_k\) is obtained by the exact minimization rule defined in (3). Also, the descent condition hold true. Then, either

\[\lim_{k \to \infty} \|g_k\| = 0 \text{ or } \sum_{k=1}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty\]

Proof:

From (4)

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

From (7)

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

From (8)

\[\|d_k\|^2 = \|g_k\|^2 + \frac{1}{\|g_k\|} \|d_{k-1}\|^2\]

Therefore

\[\|d_k\|^2 - \|d_{k-1}\|^2 = \frac{1}{\|g_k\|} \|g_k\|^2\]

So

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

i.e.

\[\lim_{k \to \infty} \|g_k\| = 0\]

Hence, the proof is completed.

4. Results and discussion

In order to test the efficiency of TMR, we compare this new method with two classical methods (FR, PRP) and two of the recent modifications (WYL, RMIL). Table 4 shows the computational performance of a MATLAB program on a set of unconstrained optimization test problems. We select 34 test functions from [29].

In this test, we choose \(\varepsilon = 10^{-6}\) and stopping criteria is set to \(\|g_k\| \leq \varepsilon\) as in [28] suggested. Three initial points are chosen starting from a point closer to the solution point to a point further away from the solution point, so that it can be used to test the global convergence of new CG coefficient. The dimensions \(n\) of 33 problems are 2, 4, 10, 100 and 1000.

In some cases, the computation stopped due to the failure of the line search to find the positive step size and thus it was considered as a fail. Numerical results are compared relative to the number iteration and CPU time. We use the performance profile intro-
duced by [27] to get the performance results that shown in Fig. 1 and 2. The CPU processor used was Intel(R) Core TM i3-M350 (2.27GHz) with RAM 4 GB.

Table 1: List of Problem Functions

<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),(10,10),(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>LEON</td>
<td>2</td>
<td>(2.2),(4.4),(8.8)</td>
</tr>
<tr>
<td>4</td>
<td>MATHYS</td>
<td>2</td>
<td>(5.5),(10,10),(15,15)</td>
</tr>
<tr>
<td>5</td>
<td>BOOTH</td>
<td>2</td>
<td>(10,10),(25,25),(100,100)</td>
</tr>
<tr>
<td>6</td>
<td>RAYDAN</td>
<td>2</td>
<td>(3.3),(13,13),(22,22)</td>
</tr>
<tr>
<td>7</td>
<td>ZETTL</td>
<td>2</td>
<td>(5.5),(20,20),(50,50)</td>
</tr>
<tr>
<td>8</td>
<td>TRECKANN</td>
<td>2</td>
<td>(5.5),(10,10),(50,50)</td>
</tr>
<tr>
<td>9</td>
<td>EXTENDED WOOD</td>
<td>4</td>
<td>(5...5),(20...20),(30...30)</td>
</tr>
<tr>
<td>10</td>
<td>CLOVILLE</td>
<td>2</td>
<td>(2...2),(4...4),(10...10)</td>
</tr>
<tr>
<td>11</td>
<td>HAGER</td>
<td>2</td>
<td>(7.7),(15,15),(22,22)</td>
</tr>
<tr>
<td>12</td>
<td>DIXON &amp; PRICE</td>
<td>2</td>
<td>(6.6),(30,30),(125,125)</td>
</tr>
<tr>
<td>13</td>
<td>EXTENDED POOLEW</td>
<td>100,100</td>
<td>(2...2),(4...4),(8...8)</td>
</tr>
<tr>
<td>14</td>
<td>GENERALIZED TRIDIAGONAL</td>
<td>2</td>
<td>(3.3),(21,21),(90,90)</td>
</tr>
<tr>
<td>15</td>
<td>GENERALIZED TRIDIAGONAL2</td>
<td>100,100</td>
<td>(15,15),(30,30),(150,150)</td>
</tr>
<tr>
<td>16</td>
<td>EXTENDED PENALTY</td>
<td>100,100</td>
<td>(2.2),(20,20),(100,100)</td>
</tr>
<tr>
<td>17</td>
<td>ROBENSTOCK</td>
<td>100,100</td>
<td>(3.3),(15,15),(75,75)</td>
</tr>
<tr>
<td>18</td>
<td>SHALLOW</td>
<td>100,100</td>
<td>(2.2),(12,12),(200,200)</td>
</tr>
<tr>
<td>19</td>
<td>EXTENDED WHITE &amp; HOLST</td>
<td>100,100</td>
<td>(3...3),(7,7),(10,10)</td>
</tr>
<tr>
<td>20</td>
<td>EXTENDED FREUDENTSTEIN &amp; ROTH</td>
<td>100,100</td>
<td>(2...2),(5...5),(7...7)</td>
</tr>
<tr>
<td>21</td>
<td>EXTENDED BEALE</td>
<td>100,100</td>
<td>(-1...-1),(3...3),(10...10)</td>
</tr>
<tr>
<td>22</td>
<td>PERTURBED QUADRATIC</td>
<td>100,100</td>
<td>(1...1),(5...5),(10...10)</td>
</tr>
<tr>
<td>23</td>
<td>EXTENDED TRIDIAGONAL1</td>
<td>100,100</td>
<td>(6...6),(12...12),(17,17)</td>
</tr>
<tr>
<td>24</td>
<td>DIAGONAL 4</td>
<td>100,100</td>
<td>(1...1),(20,20),(40,40)</td>
</tr>
<tr>
<td>25</td>
<td>EXTENDED HIMMELBLAU</td>
<td>100,100</td>
<td>(10...10),(50,50),(125,125)</td>
</tr>
<tr>
<td>26</td>
<td>FLETCHER</td>
<td>100,100</td>
<td>(3...3),(12...12),(15,15)</td>
</tr>
<tr>
<td>27</td>
<td>EXTENDED DENSCHN</td>
<td>100,100</td>
<td>(5...5),(30,30),(50,50)</td>
</tr>
<tr>
<td>28</td>
<td>EXTENDED BLOCK TRIDIAGONAL BD1</td>
<td>100,100</td>
<td>(1...1),(5...5),(10...10)</td>
</tr>
<tr>
<td>29</td>
<td>GENERALIZED QUARTIC</td>
<td>100,100</td>
<td>(7...7),(70,70),(140,140)</td>
</tr>
<tr>
<td>30</td>
<td>QUADRATIC Q2</td>
<td>100,100</td>
<td>(4...4),(16,16),(40,40)</td>
</tr>
<tr>
<td>31</td>
<td>EXTENDED QUADRATIC PENALTY QP2</td>
<td>100,100</td>
<td>(10...10),(20,20),(30,30)</td>
</tr>
<tr>
<td>32</td>
<td>QUARTIC</td>
<td>100,100</td>
<td>(8...8),(16,16),(32,32)</td>
</tr>
<tr>
<td>33</td>
<td>SUM SQUARES</td>
<td>100,100</td>
<td>(1...1),(5...5),(10...10)</td>
</tr>
<tr>
<td>34</td>
<td>QUADRATIC QF1</td>
<td>100,100</td>
<td>(3...3),(5...5),(10...10)</td>
</tr>
</tbody>
</table>

In [27] offered a model to evaluate and compare the performance of the set solvers $S$ on a test set $P$. Supposing $P_S$ solvers and $P_p$ problems exists, for each problem $p$ and solver $s$, they defined $t_{p,s}$ = computing time (the number of iterations, CPU time or others) required to solve problems $P$ by solver $S$.

Lacking a standard form for comparisons, they compared the performance of problem $p$ by solver $s$ with the best performance for any solver to the same problem using the performance ratio

$$r_{p,s} = \frac{1}{\min\{t_{p,s} : s \in S\}}$$

Thus, $p_s(t)$ is the probability for solver $s \in S$ that a performance ratio $r_{p,s}$ was within a factor $t \in R$ of the best possible ration. Then, function $p_s(t)$ is the cumulative distribution function for the performance ratio. The performance profile

$$p_s : R \rightarrow [0,1]$$

for a solver was a non-decreasing, piecewise, and continuous from the right. The value of $p_s(1)$ is the probability that the solver will win over the rest of the solvers. In general, a solver with high values of $P(t)$ or at the top right of the figure is better or represent the best solver.

**Fig. 1:** Performance profile based on number of iterations

**Fig. 2:** Performance profile based on CPU time

Figures show the performance profile of the five methods we used based on number of iterations and CPU time.

Both figures depict that TMR outperforms on the other methods, since it can solve all of the test problems and reach 100% percent.
age. In contrast, FR, PRP, WYL and RMIL only reach 90%, 97%, 98% and 96% respectively in solving the given test problems. It can be seen that TMR possesses restart properties and clearly dominates the other methods. To sum up, our numerical results suggest a new efficient conjugate gradient method. The applications of CG can be refer to [30, 31].

5. Conclusion and further research

Even though, the CG methods are appropriate and efficient for solving nonlinear optimization problems. Many studies on CG methods have led to a new variety of these methods. Meanwhile, their performance has been shown to be slightly better than the previous methods; they seem to be more difficult and complicated. In this paper, we have proposed a new and simple $\beta_k$. Besides that, our numerical results show that the TMR method is competitive to the standard conjugate gradient methods and possesses good convergence properties with exact line search. For further research, we intend to study the convergence of the TMR using other line search.

References