Research Article

Nonlinear Conjugate Gradient Coefficients with Exact and Strong Wolfe Line Searches Techniques

Awad Abdelrahman¹, Mogtaba Mohammed²,³, Osman O. O. Yousif¹, and Murtada K. Elbashir⁴

¹Department of Mathematics, College of Science, Sudan University of Science and Technology, Khartoum, Sudan
²Department of Mathematics, College of Science, Majmaah University, Zulfi 11932, Saudi Arabia
³Department of Mathematics, Faculty of Mathematical and Computer Sciences, University of Gezira, Wad Madani, Sudan
⁴Department of Information Systems, College of Computer and Information Sciences, Jouf University, Sakaka 72441, Saudi Arabia

Correspondence should be addressed to Mogtaba Mohammed; mogtaba.mohammed@gmail.com

Received 15 April 2022; Revised 29 August 2022; Accepted 5 September 2022; Published 22 November 2022

Academic Editor: Sheng Du

Copyright © 2022 Awad Abdelrahman et al. 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.

Nonlinear conjugate gradient (CG) methods are very important for solving unconstrained optimization problems. These methods have been subjected to extensive researches in terms of enhancing them. Exact and strong Wolfe line search techniques are usually used in practice for the analysis and implementation of conjugate gradient methods. For better results, several studies have been carried out to modify classical CG methods. The method of Fletcher and Reeves (FR) is one of the most well-known CG methods. It has strong convergence properties, but it gives poor numerical results in practice. The main goal of this paper is to enhance this method in terms of numerical performance via a convexity type of modification on its coefficient βk. We ensure that with this modification, the method is still achieving the sufficient descent condition and global convergence via both exact and strong Wolfe line searches. The numerical results show that this modified FR is more robust and effective.

1. Introduction

An unconstrained optimization problem is solved using the nonlinear conjugate gradient method to obtain the minimal value of a given function. The CG technique is usually expressed as follows:

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

such that \( f : \mathbb{R}^n \rightarrow \mathbb{R}, f \in C^1(\mathbb{R}) \), and \( g(x) \) are given function denoting its gradient. To apply CG methods in solving (1), we start with an initial point \( x_1 \in \mathbb{R}^n \) and use the iterative form

\[
x_{k+1} = x_k + \alpha_k d_k, \quad k = 1, 2, 3, \ldots,
\]

where \( g_k = \nabla f(x_k) \) and \( \beta_k \) are scalar.

Different choices of the scalar \( \beta_k \) relate to different conjugate gradient algorithms. Over the years, several versions of this approach have been presented, some of which are now extensively utilized. \( \beta_k \) has at least six well-known formulas, which are listed below (Dai and Yaxiang [1], Conjugate descent [2], Fletcher and Colin [3], Liu and Storey [4], Hestenes and Stiefel [5], and Polak and Ribiere-Polyak [6]):

\[
\beta_{k}^{HS} = \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}},
\]

\[
\beta_{k}^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}},
\]
\[
\beta_k^{\text{PRP}} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}},
\]
\[
\beta_k^{\text{CD}} = -\frac{g_k^T g_k}{d_k^T g_{k-1}},
\]
\[
\beta_k^{\text{LS}} = -\frac{g_k^T (g_k - g_{k-1})}{d_k^T g_{k-1}},
\]
\[
\beta_k^{\text{DY}} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}}.
\]

Many authors have examined the behavior of global convergence for the \(\beta_k\)’s formulas with several line search for several years (see, for example, [1–4, 7–18]). When the objective function is a strongly convex quadratic and the line search is exact, these methods are identical, since the gradients are mutually orthogonal, and the scalars \(\beta_k\) in these methods are equal. When applied for general nonlinear functions with inexact line searches, the behavior of these methods is clearly distinct (see [11, 17, 19–21]). One of the most important properties of the CG methods is global convergence. Zoutendijk [22] proved the global convergence of the FR method via the exact line search. Although FR, DY, and CD methods have strong convergence properties, they may not perform well in practice [14]. The CD and DY methods were proved to have a global convergence under strong Wolfe line search [3, 15]. Moreover, under the exact and strong Wolfe line searches, up to our knowledge, the global convergence and sufficient descent property of some CG methods such as the PRP and the HS have not been established [2, 14]. Andrei [9] classified the CG into groups, these are scaled CG method, classical CG method, and hybrid and parameterized CG methods.

Formulas (4) to (9) are in the classical group. One of the most important among them is the FR method, in which \(\beta_k\) satisfying \(|\beta_k| \leq \beta_k^{\text{FR}}\), provided that the line search satisfies the strong Wolfe line search. If \(\beta_k\) satisfies \(|\beta_k| \leq c\beta_k^{\text{FR}}\) (where \(c > 1\)), they have given an example to indicate that even exact line search cannot guarantee the global convergence property of the dependent FR conjugate gradient method. Nostriipour and Keyvan [23] proved that the dependent FR conjugate gradient method is globally convergent with the strong Wolfe line search if \(\beta_k\) satisfies

\[
\sigma |\beta_k| \leq \bar{\sigma},
\]

\[
\|g_k\| \sum_{j=1}^{k-1} l_i \leq c_k,
\]

where \(0 < \sigma < 1, 0 < \bar{\sigma} < 1/2, l_i = |\beta_i/\beta_i^{\text{FR}}|, \) and \(c > 0\).

Zhang and Donghui [24] proposed a modified FR method (called MFR) in which the direction \(d_k\) is defined by

\[
d_k = \begin{cases} 
-g_k, & k = 0, \\
-d_k, & k > 0,
\end{cases}
\]

which is a descent direction independent of the line search.

Competitive numerical and global convergence results are obtained by newly introduced or modified techniques; see for instance [10] and the references therein.

This paper is organized as follows: Section 2 is devoted to the modification and introduce the algorithm for the modified FR method. In Section 3, we have determined sufficient descent and global convergence properties under exact and strong Wolfe line searches. Section 4 provides preliminary numerical results and considerations. Section 5 shows the conclusions.

### 2. Motivation and Properties

Several authors have attempted to modify classical CG methods such as FR, PRP, HS, and LS in order to produce new modifications with sufficient descent and global convergence properties. In addition to that, the new modifications are expected to have an efficient numerical performance. It is well known that the FR method has poor numerical performance but has strong convergence properties. The main aim from this paper is to overcome this flaw, using the following modification to the FR formula:

\[
\beta_k^{\text{MFR}} = -\frac{\|g_k\|^2}{(1-\theta)\|d_{k-1}\|^2 + \theta \|g_{k-1}\|^2},
\]

where \(0 < \theta < 1, \theta\) is a scalar parameter, which is to be determined later. Note that if \(\theta = 1\), then \(\beta_k^{\text{MFR}} = \beta_k^{\text{FR}}\) and if \(\theta = 0\), then \(\beta_k^{\text{MFR}} = \|g_k\|^2/\|d_{k-1}\|^2\). On the other hand, if \(0 < \theta < 1\), then \(\beta_k^{\text{MFR}}\) is the modification of \(\beta_k^{\text{FR}}\), where \(\|\cdot\|\) means the norm in \(R\). (12) satisfies the following inequalities:

\[
\beta_k^{\text{MFR}} \leq \frac{\|g_k\|^2}{\theta \|g_{k-1}\|^2}.
\]

**Algorithm 1.** Step 1. Initialization. Given \(x_0 \in R^n, d_0 = -g_0, \epsilon > 0\) is tolerance, \(k = 0\), when \(g_0 = 0\) stop.

Step 2. Evaluate \(\beta_k\) according to (12).
Step 3. Evaluate $d_k$ according to (3); if $\|g_k\| \leq \varepsilon$, then stop; otherwise, go to the next step.

Step 4. Evaluate an $\alpha_k > 0$ using exact line search, i.e., $\alpha_k$ that satisfies

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

(14)

and inexact line searches, i.e., $\alpha_k$ that satisfies

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k,$$

(15)

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

(16)

where $0 < \delta < \sigma < 1$.

Step 5. Renew the point according to (2), if $g_k^T d_k > 0$, then stop.

Step 6. Set $k = k + 1$, and go to step 2.

The ideal method to choose the step length is via exact line search in practice, some approximation methods called inexact line search such as strong Wolfe are used to define the step length that give suitable reductions in the objective function with minimal cost. However, the convergence properties of some CG methods such as FR, RMIL, and RMIL+ have been established under exact line search (see [3, 6, 25]).

3. Convergent Analysis

In this section, we will examine the convergence properties of $\beta_k^{\text{NMFR}}$. The main feature of Algorithm 1 is achieving sufficient descent conditions and global convergence properties according exact and strong Wolfe line searches.

3.1. Convergent Analysis via the Exact Line Search. In this subsection, we show that our modified (12) coefficient satisfies global convergence according to the exact line search.

**Theorem 1.** Assume that (2) and (3) are generated by Algorithm 1 and $\alpha_k$ is determined by the exact line search (14), where $\beta_k = \beta_k^{\text{NMFR}}$ is given as (12); there exists $c > 0$, such that

$$g_k^T d_k \leq -c\|g_k\|^2 \quad \forall k \geq 0.$$  

(17)

The proof of this Theorem 1 is obvious, from (3), and multiply by $g_{k+1}$; then,

$$g_{k+1}^T d_{k+1} = g_{k+1}^T (-g_{k+1} + \beta_{k+1} d_k) = -\|g_{k+1}\|^2 + \beta_{k+1} g_{k+1}^T d_k.$$  

(18)

When $g_{k+1}^T d_k = 0$, then (17) holds true for all $k \geq 1$ and becomes

$$g_k^T d_k = -\|g_k\|^2.$$  

(19)

**3.1.1. Global Convergence Properties.** In this subsection, we show that our modified (12) coefficient satisfies global convergence according to the exact line search.

**Assumption 2.**

(i) There exists some positive constant $c$ such that $c \geq \|f(x)\|$ for all $x \in R^n$ and $f \in C^1(N)$ for some neighborhood $N$ of $\Gamma = \{x \in R^n | f(x) \leq f(x_0)\}$. Also, assume that there exists a positive constant $C_0$ such that

$$\|x - y\| \leq C_0, \text{ for any } x, y \in N.$$  

(20)

(ii) There exists positive constant $C_1$

$$\|g(x) - g(y)\| \leq C_1\|x - y\|, \text{ for all } x, y \in N.$$  

(21)

From the above assumption, we can easily see that

$$\|g(x)\| \leq C_2 \quad \text{for any } x \in \Gamma \text{ and some positive constant } C_2.$$  

(22)

The following lemma will be used in our analysis (see Zoutendijk [22]).

**Lemma 3.** Assume that Assumption 2 holds, for any conjugate gradient method as in (2), such that $d_k$ is a descent search direction and $\alpha_k$ satisfies the exact line search. Then,

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

(23)

**Theorem 4.** Assume that Assumption 2 holds, for conjugate gradient method as in (2) and (3) such that $\alpha_k$ is achieved via exact line search. Furthermore, assume that the sufficient descent condition is satisfied. Then,

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

or

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

(24)

**Proof.** The proof will be conducted by contradiction argument. So, assume that the statement of Theorem 4 is false. Thus, there exists some positive constant $c$, where

$$\|g_k\| \geq c.$$  

(25)

We rewrite (3) as

$$d_{k+1} + g_{k+1} = \beta_{k+1} d_k.$$  

(26)

Taking the square on both sides of above equation, we
get
\[ \|d_{k+1}\|^2 = -\|g_{k+1}\|^2 - 2g_{k+1}^T d_{k+1} + (\beta_k)^2 \|d_k\|^2. \]  
(27)

Dividing both sides of (27) by \( \|g_{k+1}\|^4 \), we obtain
\[ \frac{\|d_{k+1}\|^2}{\|g_{k+1}\|^4} = -\frac{1}{\|g_{k+1}\|^2} \frac{2g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^4} + \frac{(\beta_k)^2 \|d_k\|^2}{\|g_{k+1}\|^4}. \]  
(28)

From (19) and (13), we obtain
\[ \frac{\|d_{k+1}\|^2}{\|g_{k+1}\|^4} = \frac{1}{\|g_{k+1}\|^4} \frac{\|d_k\|^2}{\|g_{k+1}\|^4} \leq \frac{\|d_k\|^2}{\|g_{k+1}\|^4}. \]  
(29)

Recursively using (29) and noticing that \( \|d_1\|^2 = -g_1^T d_1 = \|g_1\|^2 \), we get
\[ \frac{\|d_{k+1}\|^2}{\|g_{k+1}\|^4} \leq \left( \frac{1}{\theta^2} \right)^{k+1} \frac{1}{\|g_1\|^4}. \]  
(30)

Hence,
\[ \frac{\|d_{k+1}\|^2}{\|g_{k+1}\|^4} \leq \frac{k}{\theta^2 c^2}, \]  
(32)
\[ \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^4} \geq \frac{\theta^2 c^2}{k}. \]  
(33)

As a result of (32) and (25), it is clear that
\[ \sum_{k=0}^{\infty} \frac{\|g_{k+1}\|^4}{\|d_{k+1}\|^4} = \infty, \]  
(34)

which contradicts Lemma 3’s Zoutendijk condition; hence, the proof is completed. \( \square \)

3.2. Convergent Analysis according to Strong Wolfe Line Searches. In this subsection, we show that our modified (12) coefficient satisfies sufficient descent condition and global convergence according to strong Wolfe line search (15) and (16).

3.2.1. Sufficient Descent Condition

**Theorem 5.** Let Assumption 1 hold true, for \( \sigma < 1/2 \) and \( \theta = 0.3 \). Then, Algorithm 1 assures the sufficient descent condition (17) such that
\[ c = 2 - \frac{\theta}{\theta - \sigma}. \]  
(35)

Proof. Multiplying (3) by \( g_{k+1} \) and (13) and taking the absolute value of the second term in (36), we obtain
\[ g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \]  
(36)

Dividing both sides by \( \|g_{k+1}\|^2 \), we get
\[ \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{\|d_k\|^2}{\|g_{k+1}\|^2}, \]  
(37)

From (16) and the Cauchy-Schwartz inequality, we get
\[ g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{\|d_k\|^2}{\|g_{k+1}\|^4} \]  
(38)

By repeating this process and the fact that \( \|d_i\| = \|g_i\| \), we get
\[ \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 + \frac{\|d_k\|^2}{\|g_{k+1}\|^2}, \]  
(39)

Therefore, from (39), we can deduce that (17) holds for \( k \geq 0 \). The proof is completed. \( \square \)

3.2.2. Global Convergence. This subsection is devoted to the global convergence in the case of inexact line search technique. The following lemma is collected from [22].

**Lemma 6.** Let Assumption 1 and let any conjugate gradient method be in the form (2), where the descent direction is \( d_{k+1} \) and \( \alpha_k \) satisfies the strong Wolfe line search (15) and (16).

\[ \sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^4} < \infty. \]  
(40)

**Theorem 7.** Let Assumption 2 hold. Then, Algorithm 1 is
<table>
<thead>
<tr>
<th>No.</th>
<th>Function</th>
<th>DIM</th>
<th>FR IN/CPU</th>
<th>PRP IN/CPU</th>
<th>CD IN/CPU</th>
<th>NMFR IN/CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Extended White &amp; Holst</td>
<td>10</td>
<td>631/4.449</td>
<td>11/0.106</td>
<td>341/2.473</td>
<td>28/0.246</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>177/2.312</td>
<td>20/0.292</td>
<td>308/4.012</td>
<td>17/0.246</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>205/1.368</td>
<td>30/0.226</td>
<td>341/2.300</td>
<td>26/0.202</td>
</tr>
<tr>
<td>2</td>
<td>Extended Rosenbrock</td>
<td>100</td>
<td>204/1.629</td>
<td>30/0.278</td>
<td>316/2.554</td>
<td>26/0.242</td>
</tr>
<tr>
<td>3</td>
<td>Extended Freudenstein &amp; Roth</td>
<td>10</td>
<td>F/F</td>
<td>F/F</td>
<td>57/0.445</td>
<td>34/0.269</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>117/1.114</td>
<td>F/F</td>
<td>42/0.446</td>
<td>59/0.573</td>
</tr>
<tr>
<td>4</td>
<td>Extended Beale</td>
<td>4</td>
<td>66/0.566</td>
<td>8/0.083</td>
<td>66/0.571</td>
<td>27/0.243</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>19/1.637</td>
<td>9/0.776</td>
<td>19/1.641</td>
<td>10/0.859</td>
</tr>
<tr>
<td>5</td>
<td>Raydan1</td>
<td>10</td>
<td>52/0.469</td>
<td>24/0.220</td>
<td>52/0.470</td>
<td>26/0.247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>190/2.244</td>
<td>104/1.257</td>
<td>186/2.210</td>
<td>131/1.563</td>
</tr>
<tr>
<td>6</td>
<td>Extended Tridiagonal1</td>
<td>1000</td>
<td>48/3.760</td>
<td>14/1.104</td>
<td>48/3.763</td>
<td>76/5.948</td>
</tr>
<tr>
<td>7</td>
<td>Diagonal4</td>
<td>1000</td>
<td>5/0.124</td>
<td>4/0.099</td>
<td>5/0.122</td>
<td>5/0.118</td>
</tr>
<tr>
<td>8</td>
<td>Extended Himmelblau</td>
<td>10000</td>
<td>8/1.65</td>
<td>5/0.124</td>
<td>8/1.501</td>
<td>8/1.488</td>
</tr>
<tr>
<td>9</td>
<td>FLETCHCR</td>
<td>10</td>
<td>190/2.244</td>
<td>104/1.257</td>
<td>186/2.210</td>
<td>131/1.563</td>
</tr>
<tr>
<td>10</td>
<td>Diagonal2</td>
<td>1000</td>
<td>14/1.141</td>
<td>30/0.303</td>
<td>147/1.405</td>
<td>30/0.300</td>
</tr>
<tr>
<td>11</td>
<td>NONSCOMP</td>
<td>4</td>
<td>1725/14.289</td>
<td>344/2.972</td>
<td>1726/14.316</td>
<td>546/4.730</td>
</tr>
<tr>
<td>12</td>
<td>Extended DENSCHNB</td>
<td>1000</td>
<td>11/0.135</td>
<td>9/0.109</td>
<td>11/0.135</td>
<td>10/0.122</td>
</tr>
<tr>
<td>13</td>
<td>Extended Penalty</td>
<td>10</td>
<td>10/0.118</td>
<td>7/0.093</td>
<td>10/0.113</td>
<td>8/0.093</td>
</tr>
<tr>
<td>14</td>
<td>Hager</td>
<td>1000</td>
<td>99/1.386</td>
<td>36/0.521</td>
<td>94/1.335</td>
<td>36/0.521</td>
</tr>
<tr>
<td>15</td>
<td>ARWHEAD</td>
<td>4</td>
<td>6/0.070</td>
<td>7/0.086</td>
<td>6/0.084</td>
<td>6/0.070</td>
</tr>
<tr>
<td>16</td>
<td>Extended Maratos</td>
<td>1000</td>
<td>159/1.543</td>
<td>F/F</td>
<td>37/0.315</td>
<td>52/0.579</td>
</tr>
<tr>
<td>17</td>
<td>Six hump</td>
<td>2</td>
<td>4/0.065</td>
<td>4/0.066</td>
<td>4/0.061</td>
<td>4/0.056</td>
</tr>
<tr>
<td>18</td>
<td>Three hump</td>
<td>2</td>
<td>6/0.085</td>
<td>7/0.123</td>
<td>6/0.084</td>
<td>7/0.083</td>
</tr>
<tr>
<td>19</td>
<td>Booth</td>
<td>2</td>
<td>3/0.049</td>
<td>3/0.042</td>
<td>3/0.051</td>
<td>3/0.039</td>
</tr>
<tr>
<td>20</td>
<td>Trecani</td>
<td>2</td>
<td>5/0.068</td>
<td>5/0.067</td>
<td>5/0.076</td>
<td>5/0.066</td>
</tr>
<tr>
<td>21</td>
<td>Zettl</td>
<td>2</td>
<td>16/0.169</td>
<td>9/0.109</td>
<td>16/0.173</td>
<td>9/0.109</td>
</tr>
<tr>
<td>22</td>
<td>Shallow</td>
<td>1000</td>
<td>1216/30.119</td>
<td>14/0.369</td>
<td>1171/28.969</td>
<td>18/0.479</td>
</tr>
<tr>
<td>23</td>
<td>Generalized Quartic</td>
<td>10000</td>
<td>6/0.092</td>
<td>6/0.095</td>
<td>6/0.091</td>
<td>6/0.088</td>
</tr>
<tr>
<td>24</td>
<td>Quadratic QF2</td>
<td>10</td>
<td>875/8.867</td>
<td>31/0.341</td>
<td>746/7.476</td>
<td>35/0.382140/</td>
</tr>
<tr>
<td>25</td>
<td>Leon</td>
<td>2</td>
<td>1681/14.026</td>
<td>43/0.399</td>
<td>557/4.697</td>
<td>36/0.343</td>
</tr>
<tr>
<td>26</td>
<td>Generalized Tridiagonal 1</td>
<td>1000</td>
<td>151/2.339</td>
<td>25/0.403</td>
<td>26/0.422</td>
<td>24/0.377</td>
</tr>
<tr>
<td>27</td>
<td>Generalized Tridiagonal 2</td>
<td>10</td>
<td>40/1.937</td>
<td>36/0.411</td>
<td>42/0.485</td>
<td>33/0.378</td>
</tr>
<tr>
<td>28</td>
<td>POWER</td>
<td>10</td>
<td>20/0.207</td>
<td>21/0.218</td>
<td>20/0.217</td>
<td>131/1.296</td>
</tr>
</tbody>
</table>
convergent, that is,
\[
\lim_{k \to \infty} g_k = 0,
\]
or
\[
\sum_{k=1}^{\infty} \frac{\|g_k\|}{\|d_k\|^2} < \infty.
\]  
\[\text{(43)}\]

Proof. The proof will be conducted by contradiction argument. So, assume that the statement of Theorem 7 is not true; then, there exists a constant \( \epsilon > 0 \), such that \( \|g_k\| \geq \epsilon \).

\[\|g_k\| \geq \epsilon.\]  
\[\text{(44)}\]

Equation (3) can be written as
\[
d_{k+1} = -g_{k+1} + \beta_{k+1} d_k,
\]
and multiplying (45) on both sides by \( d_{k+1} \), we obtain
\[
\|d_{k+1}\|^2 = -g^T_{k+1} d_{k+1} + \beta_{k+1} d^T_{k+1} d_k.
\]  
\[\text{(45)}\]

Dividing both sides of (46) by \( \|g_{k+1}\|^4 \) with the help of (13), we get
\[
\left\|\frac{d_{k+1}}{\|g_{k+1}\|}\right\|^2 \leq \frac{1}{\|g_{k+1}\|^2} \left( \frac{g^T_{k+1} d_{k+1}}{\|g_{k+1}\|} + \frac{\beta_{k+1} d^T_{k+1} d_k}{\|g_{k+1}\|^2} \right).
\]  
\[\text{(47)}\]

From Cauchy-Schwartz inequality, we get
\[
\left\|\frac{d_{k+1}}{\|g_{k+1}\|}\right\|^2 \leq \left\|\frac{g_{k+1}}{\|g_{k+1}\|}\right\|^2 + \frac{\beta_{k+1} d^T_{k+1} d_k}{\|g_{k+1}\|^2}.
\]  
\[\text{(48)}\]

Table 1: Continued.

<table>
<thead>
<tr>
<th>No.</th>
<th>Function</th>
<th>DIM</th>
<th>FR IN/CPU</th>
<th>PRP IN/CPU</th>
<th>CD IN/CPU</th>
<th>NMFR IN/CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Quadratic QF</td>
<td>10</td>
<td>10/0.127</td>
<td>10/0.125</td>
<td>10/0.119</td>
<td>25/0.285</td>
</tr>
<tr>
<td>30</td>
<td>Extended quadratic penalty QP</td>
<td>1000</td>
<td>F/F</td>
<td>41/2.460</td>
<td>122/9.486</td>
<td>73/3.679</td>
</tr>
<tr>
<td>31</td>
<td>Extended quadratic penalty QP</td>
<td>100</td>
<td>314/3.803</td>
<td>F/F</td>
<td>14/0.209</td>
<td>14/0.207</td>
</tr>
<tr>
<td>32</td>
<td>Quartic</td>
<td>4</td>
<td>223/2.513</td>
<td>615/6.911</td>
<td>223/2.510</td>
<td>260/2.960</td>
</tr>
<tr>
<td>33</td>
<td>Colville</td>
<td>4</td>
<td>34/0.369</td>
<td>74/0.775</td>
<td>34/0.3661</td>
<td>59/0.626</td>
</tr>
<tr>
<td>34</td>
<td>Dixon and Price</td>
<td>100</td>
<td>76/0.906</td>
<td>42/0.533</td>
<td>74/0.897</td>
<td>42/0.531</td>
</tr>
<tr>
<td>35</td>
<td>Extended Wood</td>
<td>4</td>
<td>F/F</td>
<td>267/2.979</td>
<td>F/F</td>
<td>998/12.978</td>
</tr>
</tbody>
</table>

Figure 1: Performance profile according to the iterations number based on exact line search.

Figure 2: Performance profile based on the CPU time based on exact line search.
Theorem 2.5. Assume that \( g_k \) is a descent direction, and \( \theta \) is a positive constant. Then,

\[
\frac{\|d_{k+1}\|}{\|g_{k+1}\|} \leq \frac{1}{\theta}.
\]

Substituting (51) into (49), we get

\[
\frac{\|d_{k+1}\|}{\|g_{k+1}\|} \leq \frac{1}{\theta} \frac{1}{\theta^2} \|g_k\|.
\]

where \( \gamma = 1/2 - 1/c \).

Recursively using (52) and noticing that \( \|d_k\| = \|g_k\| \), we have

\[
\frac{\|d_{k+1}\|}{\|g_{k+1}\|} \leq \frac{1}{\theta} \frac{1}{\theta^2} \frac{1}{\theta^3} \frac{1}{\theta^4} \frac{1}{\theta^5} \cdots \frac{1}{\theta^k},
\]

which contradicts (43); hence, the proof is completed.

4. Numerical Results and Discussion

Most of the test problems used in this study are taken from Andrei [8], and they are used to evaluate the efficiency of the NMFR method to that of FR and PRP under exact line search and to that of FR, PRP, and CD under strong Wolfe line search. The step-size is computed using the exact and strong Wolfe line search techniques, and numerical results are compared based on the number of iterations and CPU time. For all test problems, the stopping criteria are set to be \( \|g_k\| \leq \epsilon \), where \( \epsilon = 10^{-6} \); for each test problem, various starting points are used, as suggested by Hillstrom [26]. All runs are performed on a PC ACER (Intel® Core™ i3-3217u CPU @ 1.8 GHZ, with 4.00 GB RAMS, Windows 7 Ultimate). Every problem mentioned in Table 1 is solved using Matlab10 subroutine programming. The performance results are shown in Figures 1–4, respectively, using a performance profile introduced by Dolan and Jorge [13].

In the list of problem functions, IN indicates the number of iterations, and CPU indicates the CPU time as shown in Table 1. F indicates that the test problem function is failing. Also, in some cases, it means that the computation came to a halt when a line search failed to locate a positive step-size, and it was deemed a failure.

We offer the concept of a way of evaluating and examine the effectiveness of set solvers \( s \) on a test set \( p \) using the performance profile. Assuming there are \( n_s \) solvers and \( n_p \),
problems; they characterized $t_{ps}$ as the computing time (computational time, CPU time, or other factors) needed to tackle problems $p$ by solver $s$. They used the performance ratio $r_{ps} = t_{ps}/\min \{t_{ps} : s \in S\}$ to compare solver $s$ performance on problem $p$ to the best performance by any solver on this problem.

We let $r_{M} \geq r_{ps}$ for each chosen $p$ and $s$ such that $r_{ps} = r_{M}$, whenever $s$ is not a solution of problem $p$. The solution for performance $s$ of presented problems has to be reliable, and we wish to have the entire evaluation of solution for performance $s$. Thus, we define $P(t_i) = (1/n_p)\text{size}\{p \in P : r_{ps} \leq t\}$, where $P(t_i)$ stands for the probability of solution for performance $s$ in $S$ and $P_s$ is a cumulative distribution function for $r_{ps}$. The value $P(1)$ is the probability that the solver will win over the rest of the solvers. In all, a solver with high values of $P(t_i)$ or at the top right of the figures is preferable or represents robust solver.

Clearly from Figures 1 and 2, we see that NMFR has better performance, since it solved all the test problems and reached 100% percentage. These CG coefficients could also be divided into three categories: the first category of which consists of NMFR, while the second consists of PR, and the third consists of PRP. It is easy to see that their performance is much better. Although the performance of the third category seems to be much better than NMFR, it could only solve 86% of the problems, whereas the performance of the second category only reached 89%. Hence, we considered NMFR as an efficient performance and robust method with the others because it can solve all the problems.

Also, Figures 3 and 4 show that the curve of NMFR is higher than that of PRP, CD, and FR. This implies that the NMFR approach outperforms the other three methods significantly. Furthermore, the NMFR approach solves all problems; meanwhile, the PRP method solves about 86 percent of problems, and the CD and FR methods solve about 89 percent. As a result, we can infer that NMFR is the preferred approach because it has the highest curve and solves all problems.

5. Conclusion

In this article, we have proposed a new and simple modification for $\beta_k^{FR}$ that is easy to implement, known as $\beta_k^{NMFR}$. Numerical results show that $\beta_k^{NMFR}$ has efficient performance compared to other standard CG methods. In contrast to $\beta_k^{FR}$, we have seen that $\beta_k^{NMFR}$ shows good numerical performance at each step. We have also proved that $\beta_k^{NMFR}$ converges globally based on the exact and strong Wolfe line searches.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


