Nonlinear Functional Analysis and Applications
Vol. 26, No. 4 (2021), pp. 811-837
ISSN: 1229-1595(print), 2466-0973(online)

https://doi.org/10.22771/nfaa.2021.26.04.10
http://nfaa.kyungnam.ac.kr/journal-nfaa

A NEW CLASS OF NONLINEAR CONJUGATE GRADIENT METHOD FOR UNCONSTRAINED OPTIMIZATION MODELS AND ITS APPLICATION IN PORTFOLIO SELECTION

Maulana Malik[1,2], Ibrahim Mohammed Sulaiman[3], Mustafa Mamat[1,4], Siti Sabariah Abas[1] and Sukono[5]

[1]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin
Terengganu 22200, Malaysia
e-mail: m.malik@sci.ui.ac.id, must@unisza.edu.my, sabariahabas@unisza.edu.my

[2]Department of Mathematics, Universitas Indonesia
Depok 16424, Indonesia

[3]School of Quantitative Sciences, College of Arts and Sciences, Universiti Utara Malaysia
Kedah 06010, Malaysia
e-mail: kademi1985@gmail.com

[4]Institute of Engineering Mathematics, Universiti Malaysia Perlis
Arau 02600, Malaysia

[5]Department of Mathematics, Universitas Padjadjaran
Jatinangor 45363, Indonesia
e-mail: sukono@unpad.ac.id

Abstract. In this paper, we propose a new conjugate gradient method for solving unconstrained optimization models. By using exact and strong Wolfe line searches, the proposed method possesses the sufficient descent condition and global convergence properties. Numerical results show that the proposed method is efficient at small, medium, and large dimensions for the given test functions. In addition, the proposed method was applied to solve practical application problems in portfolio selection.

[0]Received March 23, 2021. Revised April 28, 2021. Accepted May 9, 2021.

[0]2010 Mathematics Subject Classification: 65K10, 90C52, 90C26.

[0]Keywords: Conjugate gradient method, unconstrained optimization, descent condition, global convergence, portfolio selection.

[0]Corresponding author: Maulana Malik(m.malik@sci.ui.ac.id).

1. Introduction

Many applications in real life can be modeled into optimization models. Obtaining the solution of these models requires the use of numerical methods. One of the efficient methods available for solving optimization problems is the conjugate gradient (CG) method. This method applies to numerous problems in data science, optical physics, image and signal processing, social science, economics, chemistry, and other fields (see,[1, 2, 6, 36, 39]).

In unconstrained optimization models, we minimize the objective function, independent of any limitations on the values of the variables. Mathematically, this type of problem is formulated as

$$\min f(x),\ \ x \in \mathbb{R}^n, \tag{1.1}$$

where $x \in \mathbb{R}^n$ is a real vector with $n \geq 1$ components and $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth function whose gradient $\nabla f(x) = g(x)$ is available.

To obtain the solution of (1.1), the nonlinear CG method generate a sequence $\{x_k\}$, $k \geq 0$, using the following iterative formula:

$$x_{k+1} = x_k + \alpha_k d_k\ ,\ \ k = 0, 1, 2, ..., \tag{1.2}$$

where x_0 is a randomly selected starting point, α_k is the step-size determined by certain line search [22] and d_k is the search direction computed as:

$$d_k := \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_k d_{k-1}, & k \geq 1. \end{cases} \tag{1.3}$$

Through out this study, $g_k = g(x_k)$ will denote the gradient of f at point x_k, g_k^T is the transpose of g_k and β_k is the real scalar CG parameter [21]. At present, there are several line search methods that can be use to obtain the value of α_k. Some of the popular and frequently use line search methods include, the exact [21], Armijo [5], weak Wolfe [34, 35] and strong Wolfe line searches [24]. The exact line search is computed such that α_k satisfies:

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \tag{1.4}$$

and the Armijo line search is computed such that α_k satisfies:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k. \tag{1.5}$$

The weak Wolfe line search is computed such that α_k satisfies Armijo condition (1.5) and curvature condition

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \tag{1.6}$$

Lastly, the strong Wolfe line search is computed such that α_k fulfills the condition (1.5) and following condition:

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \leq -\sigma g_k^T d_k, \tag{1.7}$$

where $0 < \delta < \sigma < 1$. The step-size α_k plays an essential when investigating the sufficient descent condition

$$g_k^T d_k \leq -c\|g_k\|^2, \ c > 0, \tag{1.8}$$

and global convergence properties

$$\lim_{k\to\infty} \inf \|g_k\| = 0. \tag{1.9}$$

In the study of optimization problems, the famous CG methods used for obtaining the solution of the models are Fletcher-Reeves (FR) method [9], the Polak-Ribiére-Polyak (PRP) method [23, 25], the Dai-Yuan (DY) method [7], the Conjugate Descent (CD) method [10], the Wei-Yao-Liu (WYL) method [33] and the Rivaie-Mustafa-Ismail-Leong (RMIL) method [27]. The parameters β_k of these methods are formulated as follows:

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

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

where $\|.\|$ denotes the Euclidean norm of vectors and d_{k-1}^T is the transpose of d_{k-1}.

The convergence properties of the above methods have been studied by many researchers. Al-Baali [3] shows that the FR method satisfies the descent condition (1.8) and converges globally under strong Wolfe line search. Also, for the strongly convex objective function that is twice continuously differentiable, the PRP method would converge globally under exact line search [40]. However, the global convergence of the PRP method cannot be guaranteed under Wolfe and Armijo line searches [29]. The DY method is a modification of the FR method and its global convergence was established under Wolfe line search [7]. Fletcher in [9] proposed the CD method and prove the convergence under some suitable conditions, though, the numerical performance of the method is very poor [31].

Recently, there has been tremendous effort to modify the PRP method in order to improve its efficiency. Among these modifications are the WYL [33] and RMIL [27]. The WYL parameter was developed by replacing the numerator while retaining the denominator of the PRP. The global convergence of the WYL method has been studied under the exact, Wolfe, and strong Wolfe line search [33]. On the other hand, the RMIL parameter was constructed by replacing the denominator while retaining the original numerator of PRP. The convergence result of the RMIL method was established under the exact line search [27]. The above literature led to the study of several CG parameters for solving optimization problems (1.1) (see Liu et al. [13], Rivaie et al. [26], Waziri et al. [32], Yousif [38], Mtagulwa and Kaelo [20], Yao et al. [37], Zhu et al [42], and Malik et al. [14, 15, 16, 17, 18, 19, 30]).

In [41], Zhang presents a modification of the WYL method and called it the NPRP method with the formula given as:

$$\beta_k^{NPRP} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|}\left|g_k^T g_{k-1}\right|}{\|g_{k-1}\|^2}, \tag{1.10}$$

where the NPRP CG method possesses the sufficient descent condition and global convergence properties under strong Wolfe line search with parameter $\sigma \in \left(0, \frac{1}{2}\right)$. The numerical results obtained show that the NPRP method is efficient for the given problems considered from the CUTE library.

Inspired by the efficient numerical performance of the NPRP method, this paper present a new CG parameter for solving (1.1). The proposed parameter is presented in the next section and the rest part of the paper is structured as follows. In Section 3, we analyze the sufficient descent condition and global convergence properties under the exact and strong Wolfe line searches. The numerical results and comparison with other methods are presented in Section 4. Application for solving minimizing risk problems in portfolio selection is provided in Section 5. Finally, the conclusion is in Section 6.

2. A New parameter and algorithm

In this section, we propose a new modification of the NPRP method. The proposed parameter considered replacing $\dfrac{\|g_k\|}{\|g_{k-1}\|}$ in the numerator of NPRP by $\dfrac{\|g_k\|}{\|d_{k-1} - g_{k-1}\|}$, and adding a negative $\left|g_k^T g_{k-1}\right|$, while maintaining the denominator of NPRP method. The formula of the new parameter is as follows:

$$\beta_k^{MSMSS} = \begin{cases} A, & \text{if } B, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where

$$A = \frac{\|g_k\|^2 - \dfrac{\|g_k\|}{\|d_{k-1} - g_{k-1}\|}\left|g_k^T g_{k-1}\right| - \left|g_k^T g_{k-1}\right|}{\|g_{k-1}\|^2}$$

and

$$B = \|g_k\|^2 > \left(\frac{\|g_k\|}{\|d_{k-1} - g_{k-1}\|} + 1 \right) \left| g_k^T g_{k-1} \right| ,$$

where MSMSS denotes Malik-Sulaiman-Mustafa-Sabariah-Sukono.

Next, we present the algorithm of the proposed method for solving (1.1) below.

Algorithm 2.1. (MSMSS Method)

Step 1. Given $x_0 \in \mathbb{R}^n$, stopping tolerance $\epsilon > 0$, set $k := 0$, for strong Wolfe line search given σ and δ.
Step 2. Compute $\|g_k\|$. If $\|g_k\| \leq \epsilon$ then stop. Else, go to Step 3.
Step 3. Compute β_k using (2.1).
Step 4. Compute d_k using (1.3).
Step 5. Compute α_k using the exact line search (1.4) or the strong Wolfe line search (1.5) and (1.7).
Step 6. Set $k := k + 1$ and compute the next iterate x_{k+1} using equation (1.2).
Step 7. Go to Step 2.

3. Convergence analysis

In this section, we show that the proposed MSMSS method fulfills the sufficient descent condition and the global convergence properties using both the exact and strong Wolfe line searches. Because, for any conjugate gradient method to be considered efficient and robust, it must satisfy the sufficient descent conditions and the property of global convergence.

To analyze the convergence of the CG method, we need the following assumptions [21].

Assumption 3.1. The level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ at x_0 is bounded, that is, there exist a constant $r > 0$ such that $\|x\| \leq r$ for all $x \in \Omega$.

Assumption 3.2. In any neighborhood Ω_0 of Ω, f is continuous and differentiable, and its gradient $g(x)$ is Lipschitz continuous with Lipschitz constant $L > 0$, that is,

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \text{for all} \;\; y \in \Omega_0.$$

3.1. Convergence analysis based on the exact line search.

In this subsection, we will study the convergence analysis of the MSMSS method under exact line search.

Theorem 3.3. *Let the sequence $\{x_k\}$ be generated by Algorithm 2.1 under exact line search. Then the search direction d_k satisfies the sufficient descent condition.*

Proof. We can prove this theorem for $k = 0$ and $k \geq 1$. If $k = 0$, we obtain $d_0 = -g_0$, then $g_0^T d_0 = -\left(\sqrt{g_0^T g_0}\right)^2 = -\|g_0\|^2$. Hence, there exist $c = 1 > 0$ such that condition (1.8) is satisfied. For $k \geq 1$, multiplying (1.3) by g_k^T, we have

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

By using the properties of the exact line search, $g_k^T d_{k-1} = 0$, we have,

$$g_k^T d_k = -\|g_k\|^2. \tag{3.1}$$

This implies that condition (1.8) holds where $c = 1 > 0$. □

The following lemma would be applied to prove the global convergence properties.

Lemma 3.4. *The value* β_k^{MSMSS} *is* $0 \leq \beta_k^{MSMSS} \leq \dfrac{\|g_k\|^2}{\|g_{k-1}\|^2}$.

Proof. From (2.1), we have two cases.

Case 1 : for $\|g_k\|^2 > \left(\dfrac{\|g_k\|}{\|d_{k-1} - g_{k-1}\|} + 1\right) \left|g_k^T g_{k-1}\right|$, then $\beta_k^{MSMSS} > 0$. Furthermore,

$$\beta_k^{MSMSS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}-g_{k-1}\|} \left|g_k^T g_{k-1}\right| - \left|g_k^T g_{k-1}\right|}{\|g_{k-1}\|^2} \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2}.$$

Case 2 : for $\|g_k\|^2 \leq \left(\dfrac{\|g_k\|}{\|d_{k-1} - g_{k-1}\|} + 1\right) \left|g_k^T g_{k-1}\right|$, then $\beta_k^{MSMSS} = 0$.

Hence, we have $0 \leq \beta_k^{MSMSS} \leq \dfrac{\|g_k\|^2}{\|g_{k-1}\|^2}$ and thus, completes the proof. □

The following lemma is Zoutendijk condition [43].

Lemma 3.5. *Suppose that Assumption* 3.1 *and Assumption* 3.2 *holds. Consider any CG method with* (1.2) *and* (1.3)*, where* α_k *is obtained by the exact line search or the strong Wolfe line search. Then the following condition, known as the Zoutendijk condition holds,*

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{3.2}$$

The following theorem would be used to establish the global convergence of MSMSS method under the exact line search.

Theorem 3.6. *Consider the CG method in the form* (1.2) *and* (1.3)*, where* β_k *is calculated by* (2.1) *and* α_k *is determined by the exact line search. Suppose that Assumption* 3.1 *and Assumption 3.2 holds, and the sufficient descent condition* (1.8) *holds. Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.3}$$

Proof. Assume that the condition (3.3) does not holds. By definition of inferior limit, there exist constant η such that

$$\|g_k\| \geq \eta, \text{ for } k \text{ sufficiently large.}$$

It means,

$$\frac{1}{\|g_k\|^2} \leq \frac{1}{\eta^2}. \tag{3.4}$$

By rewriting (1.3), we have

$$d_k + g_k = \beta_k^{MSMSS} d_{k-1}.$$

By squaring the two sides of the above, we get

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

By using $(g_k^T d_k)^2$ and dividing both sides, we obtain

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

Based on Lemma 3.4 and (3.1), we have

$$\begin{aligned}
\frac{\|d_k\|^2}{(g_k^T d_k)^2} &\leq \frac{\|g_k\|^4 \|d_{k-1}\|^2}{\|g_{k-1}\|^4 (g_k^T d_k)^2} + \frac{1}{\|g_k\|^2} = \frac{\|g_k\|^4 \|d_{k-1}\|^2}{\|g_{k-1}\|^4 \|g_k\|^4} + \frac{1}{\|g_k\|^2} \\
&\leq \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} + \frac{1}{\|g_k\|^2}.
\end{aligned} \tag{3.5}$$

We know that $\frac{\|d_0\|^2}{(g_0^T d_0)^2} = \frac{1}{\|g_0\|^2}$ and together with (3.4), (3.1) and (3.5) yields

$$\begin{aligned} \frac{\|d_k\|^2}{(g_k^T d_k)^2} &= \frac{\|d_k\|^2}{\|g_k\|^4} \\ &\le \frac{\|d_{k-1}\|^2}{\|g_{k-1}\|^4} + \frac{1}{\|g_k\|^2} \\ &\le \frac{\|d_{k-2}\|^2}{\|g_{k-2}\|^4} + \frac{1}{\|g_{k-1}\|^2} + \frac{1}{\|g_k\|^2} \\ &\le \frac{\|d_{k-3}\|^2}{\|g_{k-3}\|^4} + \frac{1}{\|g_{k-2}\|^2} + \frac{1}{\|g_{k-1}\|^2} + \frac{1}{\|g_k\|^2} \\ &\vdots \\ &\le \sum_{i=0}^{k} \frac{1}{\|g_i\|^2} = \frac{1}{\|g_0\|^2} + ... + \frac{1}{\|g_k\|^2} \\ &\le \frac{k}{\eta^2}, \end{aligned}$$

that is,

$$\frac{(g_k^T d_k)^2}{\|d_k\|^2} \ge \frac{\eta^2}{k}.$$

Take summation, we have

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \ge \eta^2 \sum_{k=0}^{\infty} \frac{1}{k},$$

where $\sum_{k=0}^{\infty} \frac{1}{k}$ is harmonic series and we know that the harmonic series is divergent. Hence,

$$\sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} \ge \infty,$$

which contradicts the Zoutendijk condition (3.2) in Lemma 3.5. Hence, the condition (3.3) is true. □

3.2. **Convergence Analysis Using Strong Wolfe Line Search.** In this sub-section, we study the sufficient descent condition and convergence properties of the MSMSS method under the strong Wolfe line search.

The following theorem will be used to show that our proposed method satisfies the sufficient descent condition.

Theorem 3.7. *Let the sequence* $\{x_k\}$ *be generated by Algorithm* 2.1*, where* α_k *is determined by the strong Wolfe line search with* $0 < \sigma < \frac{1}{4}$*. Then,*

$$\frac{-1}{1-\sigma} < \frac{g_k^T d_k}{\|g_k\|^2} < \frac{2\sigma - 1}{1-\sigma} \text{ , for all } k \geq 0. \tag{3.6}$$

Hence, the sufficient descent condition (1.8) *holds.*

Proof. For $k = 0$, we get $d_0 = -g_0$, hence

$$\frac{-1}{1-\sigma} < \frac{g_0^T d_0}{\|g_0\|^2} = -\frac{g_0^T g_0}{\|g_0\|^2} = -1 < \frac{2\sigma - 1}{1-\sigma}.$$

So that (3.6) is true for $k = 0$, furthermore, we can write $g_0^T d_0 = -\|g_0\|^2$. Hence, the sufficient descent condition (1.8) holds.

Now, we will prove for $k > 0$. Assume that (3.6) is true for $k = n, n \in \mathbb{N}$. We obtain

$$\frac{-1}{1-\sigma} < \frac{g_n^T d_n}{\|g_n\|^2} < \frac{2\sigma - 1}{1-\sigma}. \tag{3.7}$$

We will prove that (3.6) is true for $k = n + 1$. From (1.3), we get

$$d_{n+1} = -g_{n+1} + \beta_{n+1}^{MSMSS} d_n.$$

By multiplying the both sides by g_{n+1}^T, we have

$$g_{n+1}^T d_{n+1} = -\|g_{n+1}\|^2 + \beta_{n+1}^{MSMSS} g_{n+1}^T d_n,$$

By dividing by $\|g_{n+1}\|^2$ yields,

$$\frac{g_{n+1}^T d_{n+1}}{\|g_{n+1}\|^2} = -1 + \beta_{n+1}^{MSMSS} \frac{g_{n+1}^T d_n}{\|g_{n+1}\|^2} \frac{\|g_n\|^2}{\|g_n\|^2}. \tag{3.8}$$

From the strong Wolfe line search condition (1.7) and multiplying both side by β_{n+1}^{MSMSS}, we have

$$\left|\beta_{n+1}^{MSMSS} g_{n+1}^T d_n\right| \leq -\sigma \left|\beta_{n+1}^{MSMSS}\right| g_n^T d_n. \tag{3.9}$$

From (3.8), (3.9) and using the absolute value properties,

$$\begin{aligned} -1 + \sigma \left|\beta_{n+1}^{MSMSS}\right| \frac{\|g_n\|^2}{\|g_{n+1}\|^2} \frac{g_n^T d_n}{\|g_n\|^2} \quad &\leq \quad \frac{g_{n+1}^T d_{n+1}}{\|g_{n+1}\|^2} \\ &\leq \quad -1 - \sigma \left|\beta_{n+1}^{MSMSS}\right| \frac{\|g_n\|^2}{\|g_{n+1}\|^2} \frac{g_n^T d_n}{\|g_n\|^2}. \end{aligned}$$

By using $\beta_k^{MSMSS} \leq \dfrac{\|g_{n+1}\|^2}{\|g_n\|^2}$, we get

$$-1 + \sigma \frac{\|g_{n+1}\|^2}{\|g_n\|^2} \frac{\|g_n\|^2}{\|g_{n+1}\|^2} \frac{g_n^T d_n}{\|g_n\|^2} \leq \frac{g_{n+1}^T d_{n+1}}{\|g_{n+1}\|^2} \leq -1 - \sigma \frac{\|g_{n+1}\|^2}{\|g_n\|^2} \frac{\|g_n\|^2}{\|g_{n+1}\|^2} \frac{g_n^T d_n}{\|g_n\|^2},$$

this implies,

$$-1+\sigma\frac{g_n^Td_n}{\|g_n\|^2}\leq\frac{g_{n+1}^Td_{n+1}}{\|g_{n+1}\|^2}\leq -1-\sigma\frac{g_n^Td_n}{\|g_n\|^2}.$$

By applying (3.7), we obtain

$$-1+\sigma\left(\frac{-1}{1-\sigma}\right)<\frac{g_{n+1}^Td_{n+1}}{\|g_{n+1}\|^2}< -1-\sigma\left(\frac{-1}{1-\sigma}\right).$$

Hence,

$$\frac{-1}{1-\sigma}<\frac{g_{n+1}^Td_{n+1}}{\|g_{n+1}\|^2}<\frac{2\sigma-1}{1-\sigma}.$$

This show that (3.6) is true for $k=n+1$. Hence, (3.6) is true for all $k\geq 0$. Denotes $c=\dfrac{2\sigma-1}{\sigma-1}$ and $0<\sigma<\frac{1}{4}$, then, $0<c<1$, and from (3.6), we obtain

$$(c-2)\|g_k\|^2<g_k^Td_k<-c\|g_k\|^2.$$

This implies that (1.8) holds and thus completes the proof. □

The following property, lemma, and theorem are needed to prove the convergence properties of the MSMSS method under the strong Wolfe line search.

This property is proposed by Gilbert and Nocedal [11] and has been widely used in the global convergence analysis of CG methods.

Property 3.8. *Suppose a CG method with form* (1.2) *and* (1.3)*. Then, for all* $k\geq 1$*,*

$$0<\gamma\leq\|g_k\|\leq\bar{\gamma}, \tag{3.10}$$

where γ *and* $\bar{\gamma}$ *are two positive constants. The method possess the Property* 3.8*, if there exist constants* $b>1,\lambda>0$ *such that for all* k*;*

$$\|s_k\|\leq\lambda\rightarrow|\beta_k|\leq\frac{1}{2b},$$

where $s_k=\alpha_kd_k$*.*

The following lemma shows that the MSMSS method possess the Property 3.8.

Lemma 3.9. *Suppose a CG method with form* (1.2) *and* (1.3)*. Let Assumption* 3.1 *and Assumption* 3.2 *holds. Then, the MSMSS method possess the Property* 3.8*.*

Proof. Setting the values b and λ as follows:

$$b=\frac{\bar{\gamma}^2}{\gamma^2}>1\ ,\ \ \lambda=\frac{\gamma^4}{2\bar{\gamma}^3L}>0.$$

For the first case

$$\beta_k^{MSMSS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}-g_{k-1}\|}|g_k^T g_{k-1}| - |g_k^T g_{k-1}|}{\|g_{k-1}\|^2}.$$

From Lemma 3.4 and (3.10), we have

$$\left|\beta_k^{MSMSS}\right| \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \leq \frac{\bar{\gamma}^2}{\gamma^2} = b.$$

From the Assumption 3.2, the inequality holds, if $\|s_k\| = \|\alpha_k d_k\| = \|x_{k+1} - x_k\| \leq \lambda$, then, based on the Cauchy-Schwartz inequality, we obtain

$$\begin{aligned}
\left|\beta_k^{MSMSS}\right| &= \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}-g_{k-1}\|}|g_k^T g_{k-1}| - |g_k^T g_{k-1}|}{\|g_{k-1}\|^2} \\
&\leq \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}-g_{k-1}\|}\|g_k\|\|g_{k-1}\| - \|g_k\|\|g_{k-1}\|}{\|g_{k-1}\|^2} \\
&\leq \frac{\|g_k\|^2 - \|g_k\|\|g_{k-1}\|}{\|g_{k-1}\|^2} = \frac{\|g_k\|\left(\|g_k\| - \|g_{k-1}\|\right)}{\|g_{k-1}\|^2} \\
&\leq \frac{\|g_k\|\|g_k - g_{k-1}\|}{\|g_{k-1}\|^2} \\
&\leq \frac{\|g_k\| L\lambda}{\|g_{k-1}\|^2} \leq \frac{\bar{\gamma} L\lambda}{\gamma^2} = \frac{1}{2b}.
\end{aligned}$$

For the second case, $\beta_k^{MSMSS} = 0$. It is obvious, that MSMSS method possess the Property 3.8. The proof is completed. □

The following theorem will be used to study the global convergence of the MSMSS method. Since the proof is similar to that of Theorem 4.3 in [11], so we omit it here.

Theorem 3.10. *Suppose any CG method with form* (1.2) *and* (1.3) *that satisfies the following conditions:*

(1) $\beta_k \geq 0$.
(2) *The search directions satisfy the sufficient descent condition.*
(3) *The Zoutendijk condition holds.*
(4) *The Property* 3.8 *holds.*

Then $\liminf_{k\to\infty} \|g_k\| = 0$.

From Lemma 3.4, Theorem 3.7, Lemma 3.5, and Lemma 3.9, we found that the MSMSS method satisfies all four conditions in Theorem 3.10 under strong Wolfe line search, this implies $\liminf_{k\to\infty} \|g_k\| = 0$. Hence, the MSMSS method is globally convergent.

4. Numerical experiments

In this section, we present the numerical experiments for MSMSS, RMIL, FR, CD, DY, WYL, and NPRP CG methods using exact and strong Wolfe line searches with $\sigma = 0.001$ and $\delta = 0.0001$. The comparison with other existing methods is to demonstrate the efficiency of the proposed method both in terms of the number of iterations (NOI) and central processing unit (CPU) time (in second) on several benchmark functions considered by Andrei [4] and Jamil and Yang [12]. For each test function, we considered different initial points with dimensions ranging from small, medium, and large. Most of the functions and initial points used are suggestions from Andrei [4] and dimensions varies from 2, 3, 4, 10, 50, 100, 500, 1,000, 5,000 and 10,000. The functions considered in this study are articial. We know that articial functions are functions used to detect algorithmic behavior under different conditions such as local optimal functions, bowl-shaped functions, plate-shaped functions, valley-shaped functions, unimodal functions, and other functions.

The study considered thirty-seven nonlinear test functions with ninety-eight problems for evaluation as presented in Table 1. The evaluation of the test function is based on the exact line search (1.4) and the strong Wolfe line search (1.5) and (1.7). All the algorithms are coded in MATLAB R2019a with stopping criteria set as $\|g_k\| < 10^{-6}$. The numerical experiment is considered to fail if the number of iterations exceeds 10,000 or no solution is reached. A personal laptop with specifications; processor Intel Core i7, 16 GB RAM, and operating system Windows 10 Pro 64 bit was used to run the program. In the following Tables 2 and 3, we denote P: Problems, N: Number of iterations (NOI) and C: CPU time.

Table 1. List of the test functions, dimensions, and initial points.

Prob.	Test Functions	Dimensions	Initial points
1	Extended White & Holst	1,000	(-1.2,1,...,-1.2,1)
2	Extended White & Holst	1,000	(10,...,10)
3	Extended White & Holst	10,000	(-1.2,1,...,-1.2,1)
4	Extended White & Holst	10,000	(5,...,5)
5	Extended Rosenbrock	1,000	(-1.2,1,...,-1.2,1)
6	Extended Rosenbrock	1,000	(10,...,10)
7	Extended Rosenbrock	10,000	(-1.2,1,...,-1.2,1)
8	Extended Rosenbrock	10,000	(5,...,5)
9	Extended Freudenstein & Roth	4	(0.5,-2,0.5,-2)
10	Extended Freudenstein & Roth	4	(5,5,5,5)
11	Extended Beale	1,000	(1,0.8,...,1,0.8)

(Continued on next page)

Table 1 – *Continued*

Prob.	Test Functions	Dimensions	Initial points
12	Extended Beale	1,000	(0.5,...,0.5)
13	Extended Beale	10,000	(-1,...,-1)
14	Extended Beale	10,000	(0.5,...,0.5)
15	Extended Wood	4	(-3,-1,-3,-1)
16	Extended Wood	4	(5,5,5,5)
17	Raydan 1	10	(1,...,1)
18	Raydan 1	10	(10,...,10)
19	Raydan 1	100	(-1,...,-1)
20	Raydan 1	100	(-10,...,-10)
21	Extended Tridiagonal 1	500	(2,...,2)
22	Extended Tridiagonal 1	500	(10,...,10)
23	Extended Tridiagonal 1	1,000	(1,...,1)
24	Extended Tridiagonal 1	1,000	(-10,...,-10)
25	Diagonal 4	500	(1,...,1)
26	Diagonal 4	500	(-20,...,-20)
27	Diagonal 4	1,000	(1,...,1)
28	Diagonal 4	1,000	(-30,...,-30)
29	Extended Himmelblau	1,000	(1,...,1)
30	Extended Himmelblau	1,000	(20,...,20)
31	Extended Himmelblau	10,000	(-1,...,-1)
32	Extended Himmelblau	10,000	(50,...,50)
33	FLETCHCR	10	(0,...,0)
34	FLETCHCR	10	(10,...,10)
35	Extended Powel	100	(3,-1,0,1,...,1)
36	Extended Powel	100	(5,...,5)
37	NONSCOMP	2	(3,3)
38	NONSCOMP	2	(10,10)
39	Extended DENSCHNB	10	(1,...,1)
40	Extended DENSCHNB	10	(10,...,10)
41	Extended DENSCHNB	100	(10,...,10)
42	Extended DENSCHNB	100	(-50,...,-50)
43	Extended Penalty	10	(1,2,3,...,10)
44	Extended Penalty	10	(-10,...,-10)
45	Extended Penalty	100	(5,...,5)
46	Extended Penalty	100	(-10,...,-10)
47	Hager	10	(1,...,1)
48	Hager	10	(-10,...,-10)
49	Extended Maratos	10	(1.1,0.1)

(Continued on next page)

Table 1 – *Continued*

Prob.	Test Functions	Dimensions	Initial points
50	Extended Maratos	10	(-1,...,-1)
51	Six Hump Camel	2	(-1,2)
52	Six Hump Camel	2	(-5,10)
53	Three Hump Camel	2	(-1,2)
54	Three Hump Camel	2	(2,-1)
55	Booth	2	(5,5)
56	Booth	2	(10,10)
57	Trecanni	2	(-1,0.5)
58	Trecanni	2	(-5,10)
59	Zettl	2	(-1,2)
60	Zettl	2	(10,10)
61	Shallow	1,000	(0,...,0)
62	Shallow	1,000	(10,...,10)
63	Shallow	10,000	(-1,...,-1)
64	Shallow	10,000	(-10,...,-10)
65	Generalized Quartic	1,000	(1,...,1)
66	Generalized Quartic	1,000	(20,...,20)
67	Quadratic QF2	50	(0.5,...,0.5)
68	Quadratic QF2	50	(30,...,30)
69	Leon	2	(2,2)
70	Leon	2	(8,8)
71	Generalized Tridiagonal 1	10	(2,...,2)
72	Generalized Tridiagonal 1	10	(10,...,10)
73	Generalized Tridiagonal 2	4	(1,1,1,1)
74	Generalized Tridiagonal 2	4	(10,10,10,10)
75	POWER	10	(1,1,1,1)
76	POWER	10	(10,10,10,10)
77	Quadratic QF1	50	(1,...,1)
78	Quadratic QF1	50	(10,...,10)
79	Quadratic QF1	500	(1,...,1)
80	Quadratic QF1	500	(-5,...,-5)
81	Extended Quadratic Penalty QP2	100	(1,...,1)
82	Extended Quadratic Penalty QP2	100	(10,...,10)
83	Extended Quadratic Penalty QP2	500	(10,...,10)
84	Extended Quadratic Penalty QP2	500	(50,...,50)
85	Extended Quadratic Penalty QP1	4	(1,1,1,1)
86	Extended Quadratic Penalty QP1	4	(10,10,10,10)
87	Quartic	4	(10,10,10,10)

(Continued on next page)

Table 1 – *Continued*

Prob.	Test Functions	Dimensions	Initial points
88	Quartic	4	(15,15,15,15)
89	Matyas	2	(1,1)
90	Matyas	2	(20,20)
91	Colville	4	(2,2,2,2)
92	Colville	4	(10,10,10,10)
93	Dixon and Price	3	(1,1,1)
94	Dixon and Price	3	(10,10,10)
95	Sphere	5,000	(1,...,1)
96	Sphere	5,000	(10,...,10)
97	Sum Squares	50	(0,1,...,0,1)
98	Sum Squares	50	(10,...,10)

TABLE 2. Numerical results under the exact line search.

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
1	10	0.30	28	0.89	26	0.86	26	0.98	26	1.02	1251	29.80	20	0.54
2	21	0.56	30	0.82	293	9.74	222	7.54	278	9.69	2227	53.52	44	1.11
3	10	2.54	28	7.28	30	8.01	30	7.93	30	8.04	1180	281.66	19	4.79
4	22	5.58	32	8.14	180	48.63	193	53.46	145	39.89	2923	704.48	43	12.72
5	19	0.10	28	0.14	211	0.85	100	0.41	221	0.88	1283	3.73	19	0.09
6	17	0.10	47	0.20	56	0.23	56	0.25	56	0.35	1157	3.35	44	0.18
7	19	0.34	28	0.51	227	3.87	99	1.66	230	3.91	415	9.13	19	0.37
8	24	0.53	27	1.39	206	3.44	219	3.59	224	3.65	922	17.90	29	0.50
9	9	0.04	9	0.06	15	0.07	15	0.06	15	0.08	560	1.03	9	0.05
10	6	0.04	6	0.03	7	0.03	7	0.03	7	0.03	435	0.81	6	0.03
11	13	0.52	52	1.58	75	2.28	75	2.23	75	2.44	315	8.41	26	0.75
12	13	0.37	45	1.37	81	2.53	81	2.55	81	2.49	263	7.13	25	0.69
13	15	4.08	15	4.17	87	24.01	87	24.07	87	24.03	111	29.97	28	7.54
14	13	3.58	45	12.43	87	23.97	87	23.99	87	23.99	450	121.55	25	6.78
15	34	0.10	495	1.17	fail	fail	fail	fail	fail	fail	990	1.83	374	0.78
16	136	0.35	863	1.79	fail	fail	fail	fail	fail	fail	929	2.27	1387	2.82
17	20	0.07	19	0.08	19	0.04	19	0.05	19	0.05	24	0.07	27	0.09
18	36	0.11	84	0.26	fail	fail	fail	fail	fail	fail	208	0.43	47	0.14
19	74	0.26	107	0.38	99	0.32	90	0.28	96	0.31	249	0.67	97	0.32
20	129	0.42	180	0.59	985	2.85	751	2.07	fail	fail	376	0.97	133	0.38
21	12	0.23	202	2.94	453	7.28	452	7.22	453	7.27	3133	46.12	92	1.34
22	147	2.16	189	2.79	7	0.12	90	1.49	90	1.50	2667	38.50	61	0.90
23	12	0.37	181	4.90	517	15.36	517	15.46	517	20.11	4381	118.20	118	3.16
24	83	2.30	115	3.12	8	0.30	32	1.12	78	2.92	3651	97.61	10	0.31

(Continued on next page)

Table 2 – *Continued*

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
25	5	0.03	3	0.01	5	0.03	5	0.03	5	0.03	46	0.14	5	0.04
26	5	0.03	4	0.02	5	0.03	5	0.03	5	0.03	46	0.14	5	0.03
27	5	0.03	3	0.02	5	0.03	5	0.03	5	0.04	46	0.16	5	0.03
28	5	0.05	4	0.02	5	0.04	5	0.04	5	0.04	64	0.21	5	0.03
29	7	0.05	11	0.05	15	0.09	15	0.08	15	0.08	20	0.08	13	0.08
30	6	0.04	6	0.03	8	0.05	8	0.05	8	0.05	10	0.05	8	0.04
31	9	0.19	9	0.18	27	0.53	26	0.54	27	0.52	12	0.22	15	0.29
32	7	0.19	9	0.21	17	0.34	17	0.36	17	0.34	20	0.34	18	0.33
33	56	0.21	72	0.17	1214	3.18	973	2.51	1208	3.09	198	0.39	86	0.22
34	28	0.10	31	0.08	30	0.12	30	0.12	30	0.10	48	0.12	47	0.14
35	102	0.52	fail	fail	5644	87.64	5640	89.26	5653	90.63	fail	fail	fail	fail
36	124	0.67	fail	fail	5338	25.36	5331	25.35	5334	23.55	fail	fail	fail	fail
37	12	0.04	15	0.04	50	0.15	50	0.16	50	0.16	118	0.25	12	0.17
38	18	0.07	16	0.04	915	4.77	241	0.70	1755	4.83	109	0.23	40	0.12
39	8	0.04	5	0.02	9	0.04	9	0.04	9	0.04	10	0.04	10	0.04
40	9	0.04	10	0.03	13	0.06	13	0.06	13	0.06	13	0.04	10	0.04
41	9	0.04	10	0.03	13	0.06	13	0.06	13	0.07	14	0.05	10	0.04
42	11	0.05	9	0.03	77	0.43	77	0.23	77	0.23	14	0.04	11	0.05
43	20	0.07	20	0.05	13	0.06	13	0.06	13	0.06	48	0.13	22	0.09
44	8	0.04	21	0.06	14	0.06	14	0.07	14	0.07	39	0.10	13	0.04
45	10	0.05	15	0.06	13	0.06	13	0.07	14	0.06	fail	fail	fail	fail
46	11	0.06	fail	fail	31	0.10	39	0.16	fail	fail	fail	fail	fail	fail
47	13	0.06	12	0.06	11	0.03	11	0.03	11	0.03	13	0.16	12	0.20
48	19	0.48	18	0.07	97	0.23	99	0.23	97	0.23	21	0.05	19	0.06
49	25	0.08	42	0.12	4820	8.58	1022	1.92	4096	7.57	fail	fail	31	0.10
50	23	0.23	20	0.07	31	0.11	28	0.10	29	0.09	380	1.05	30	0.09
51	7	0.06	8	0.04	24	0.06	24	0.05	24	0.06	7	0.04	10	0.04
52	6	0.04	6	0.04	11	0.04	11	0.03	11	0.04	9	0.03	9	0.04
53	13	0.05	10	0.18	21	0.20	24	0.08	9	0.20	10	0.05	11	0.04
54	10	0.03	11	0.03	fail	fail	39	0.23	fail	fail	8	0.02	14	0.05
55	4	0.02	3	0.02	3	0.01	3	0.02	3	0.02	5	0.02	3	0.02
56	4	0.02	3	0.02	3	0.02	3	0.02	3	0.02	4	0.01	3	0.02
57	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01
58	5	0.03	6	0.03	14	0.05	14	0.06	14	0.06	13	0.04	9	0.04
59	11	0.04	21	0.08	11	0.04	11	0.05	11	0.04	51	0.15	9	0.04
60	11	0.05	19	0.08	10	0.05	10	0.05	10	0.05	73	0.18	12	0.04
61	8	0.05	26	0.14	18	0.10	18	0.12	18	0.09	76	0.29	14	0.08
62	9	0.06	11	0.07	175	0.65	175	0.71	175	0.66	94	0.32	15	0.07
63	10	0.21	37	0.70	47	0.88	47	0.81	47	0.84	73	1.25	26	0.51
64	10	0.38	35	0.65	43	0.76	43	0.73	43	0.73	68	1.17	22	0.43
65	5	0.04	6	0.05	6	0.05	6	0.05	6	0.05	6	0.04	6	0.06

(Continued on next page)

Table 2 – *Continued*

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
66	6	0.05	10	0.06	12	0.07	12	0.07	13	0.10	9	0.05	10	0.06
67	67	0.18	77	0.25	116	0.31	117	0.31	116	0.32	151	0.32	85	0.24
68	68	0.19	77	0.23	125	0.41	124	0.32	126	0.33	153	0.35	109	0.27
69	15	0.06	15	0.05	37	0.11	40	0.15	40	0.12	1046	4.18	26	0.08
70	18	0.07	30	0.12	504	1.12	474	2.93	256	1.81	755	2.64	18	0.06
71	22	0.09	22	0.10	27	0.13	27	0.12	27	0.11	28	0.09	23	0.10
72	28	0.12	27	0.11	43	0.17	43	0.17	43	0.17	41	0.11	29	0.10
73	4	0.02	4	0.03	5	0.02	5	0.02	5	0.02	5	0.03	5	0.03
74	9	0.05	7	0.04	fail	fail	fail	fail	fail	fail	20	0.21	23	0.07
75	47	0.18	123	0.31	20	0.05	20	0.05	21	0.05	146	0.33	64	0.17
76	50	0.15	139	0.34	24	0.05	24	0.07	23	0.07	157	0.31	88	0.24
77	55	0.17	69	0.31	38	0.08	38	0.08	38	0.09	126	0.27	88	0.23
78	57	0.31	78	0.22	41	0.10	41	0.10	41	0.09	129	0.27	101	0.25
79	208	1.43	422	3.23	131	0.58	131	0.55	131	0.59	532	3.08	698	4.54
80	264	1.71	538	3.32	137	0.56	150	0.84	137	0.60	525	1.89	777	4.65
81	24	0.18	33	0.15	189	0.90	231	0.77	141	0.75	598	1.53	44	0.19
82	30	0.14	31	0.35	2758	7.67	101	0.45	fail	fail	567	1.48	44	0.18
83	31	0.30	58	0.63	1034	24.57	149	2.06	fail	fail	8007	49.45	88	0.71
84	35	0.37	41	0.54	fail	fail	140	1.76	fail	fail	5972	38.48	112	0.88
85	10	0.05	14	0.06	20	0.06	20	0.05	20	0.05	17	0.05	20	0.08
86	11	0.05	9	0.05	19	0.06	19	0.06	19	0.06	21	0.07	26	0.08
87	37	0.15	807	5.76	271	1.61	271	1.57	271	1.22	1016	4.28	1241	2.77
88	39	0.17	807	2.96	272	1.09	272	1.07	272	1.00	577	3.68	1216	2.65
89	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01
90	1	0.01	1	0.01	1	0.00	1	0.00	1	0.01	1	0.01	1	0.01
91	106	0.65	1035	4.00	fail	fail	fail	fail	fail	fail	1392	2.91	1591	3.03
92	28	0.09	392	0.87	34	0.10	34	0.11	34	0.11	1468	2.92	578	1.07
93	9	0.03	35	0.11	15	0.05	15	0.03	15	0.03	74	0.16	14	0.06
94	18	0.06	56	0.17	29	0.07	29	0.07	29	0.07	76	0.17	36	0.10
95	1	0.02	1	0.02	1	0.02	1	0.01	1	0.02	1	0.02	1	0.03
96	1	0.02	1	0.02	1	0.02	1	0.02	1	0.02	1	0.01	1	0.02
97	40	0.12	46	0.14	26	0.08	26	0.09	26	0.08	73	0.20	33	0.11
98	65	0.18	81	0.22	42	0.12	42	0.13	41	0.12	183	0.44	102	0.27

TABLE 3. Numerical results under the strong Wolfe line search.

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
1	15	0.05	22	0.06	61	0.12	43	0.10	49	0.10	2861	3.43	17	0.06
2	38	0.14	35	0.10	218	1.08	229	0.95	210	0.92	5763	8.88	55	0.20

(Continued on next page)

Table 3 – *Continued*

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
3	16	0.38	22	0.44	67	1.00	49	0.76	50	0.81	2129	27.16	17	0.41
4	34	0.96	43	0.97	272	6.30	109	4.10	178	5.25	4134	70.62	42	1.10
5	18	0.03	28	0.04	83	0.13	105	0.14	59	0.11	773	0.50	35	0.06
6	26	0.04	50	0.06	173	0.22	159	0.22	178	0.22	1300	0.93	25	0.04
7	18	0.18	28	0.31	88	0.98	106	1.06	59	0.82	811	4.64	36	0.38
8	19	0.22	46	0.38	244	3.22	201	2.99	196	3.04	2277	15.89	23	0.30
9	8	0.00	15	0.00	30	0.01	fail	fail	fail	fail	657	0.02	9	0.01
10	11	0.01	9	0.01	21	0.00	21	0.00	21	0.00	fail	fail	fail	fail
11	14	0.04	52	0.10	75	0.13	75	0.13	75	0.13	281	0.39	26	0.06
12	13	0.04	45	0.09	81	0.14	81	0.14	81	0.13	166	0.24	25	0.06
13	14	0.29	15	0.27	87	1.16	87	1.18	87	1.18	290	3.83	28	0.47
14	14	0.26	45	0.67	87	1.16	87	1.15	87	1.13	257	3.31	25	0.43
15	44	0.00	500	0.02	fail	fail	fail	fail	fail	fail	784	0.02	281	0.01
16	120	0.01	1085	0.03	fail	fail	fail	fail	fail	fail	1637	0.05	1427	0.05
17	20	0.00	19	0.00	19	0.00	19	0.00	19	0.00	22	0.00	27	0.00
18	35	0.00	fail	fail	2670	0.09	3215	0.11	2613	0.09	65	0.00	29	0.00
19	75	0.03	109	0.03	95	0.03	95	0.03	95	0.03	192	0.05	97	0.03
20	113	0.03	180	0.05	857	0.15	868	0.14	fail	fail	364	0.07	134	0.04
21	12	0.02	28	0.05	452	0.36	452	0.37	452	0.36	720	0.59	22	0.05
22	8	0.02	8	0.03	9	0.03	9	0.02	9	0.02	376	0.34	8	0.02
23	12	0.04	28	0.07	517	0.73	517	0.73	517	0.80	82	0.14	22	0.07
24	8	0.03	5	0.02	9	0.04	9	0.04	9	0.03	151	0.24	8	0.04
25	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	5	0.01	2	0.00
26	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	9	0.01	2	0.00
27	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	5	0.01	2	0.00
28	2	0.01	2	0.00	2	0.00	2	0.00	2	0.00	11	0.01	2	0.00
29	7	0.01	11	0.02	15	0.02	15	0.03	15	0.03	20	0.03	13	0.02
30	6	0.01	7	0.03	9	0.03	9	0.02	9	0.02	11	0.02	10	0.02
31	9	0.10	9	0.10	17	0.23	20	0.25	17	0.22	12	0.17	15	0.13
32	10	0.12	11	0.12	13	0.13	13	0.12	13	0.12	14	0.15	10	0.10
33	51	0.01	72	0.00	1208	0.06	1239	0.06	1208	0.06	124	0.01	85	0.01
34	51	0.01	111	0.01	285	0.02	303	0.02	299	0.03	183	0.01	134	0.01
35	44	0.03	fail	fail	5492	0.96	5607	1.02	5584	1.00	4345	0.80	fail	fail
36	69	0.04	fail	fail	5934	1.08	5980	1.09	6019	1.15	6556	1.27	fail	fail
37	15	0.00	54	0.00	166	0.01	155	0.01	156	0.01	148	0.01	15	0.00
38	13	0.00	21	0.00	93	0.01	93	0.00	93	0.01	446	0.02	15	0.00
39	8	0.00	6	0.00	9	0.00	9	0.00	9	0.00	10	0.00	10	0.00
40	10	0.00	7	0.00	11	0.00	11	0.00	11	0.00	12	0.00	9	0.00
41	11	0.00	7	0.00	11	0.00	11	0.01	11	0.00	12	0.00	9	0.00
42	7	0.00	9	0.00	63	0.02	63	0.02	63	0.02	8	0.00	13	0.01
43	14	0.00	24	0.01	11	0.00	11	0.00	11	0.00	25	0.00	14	0.00

(Continued on next page)

Table 3 – *Continued*

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
44	9	0.01	26	0.01	19	0.00	19	0.00	19	0.01	35	0.25	14	0.00
45	9	0.00	fail	fail	28	0.01	30	0.01	fail	fail	fail	fail	10	0.00
46	22	0.01	fail	fail	1919	1.42	860	0.25	fail	fail	197	0.05	16	0.01
47	13	0.00	12	0.00	11	0.00	11	0.00	11	0.00	13	0.00	12	0.00
48	19	0.00	18	0.00	97	0.01	97	0.01	97	0.01	21	0.00	19	0.00
49	27	0.00	42	0.00	3001	0.23	9042	0.59	fail	fail	610	0.03	48	0.01
50	28	0.00	37	0.00	128	0.02	122	0.01	128	0.02	173	0.02	37	0.17
51	6	0.00	fail	fail	10	0.00	11	0.00	11	0.00	7	0.00	10	0.00
52	9	0.00	11	0.00	15	0.00	17	0.00	17	0.00	13	0.00	10	0.00
53	11	0.01	11	0.00	fail	fail	14	0.01	fail	fail	11	0.16	12	0.00
54	12	0.01	fail	fail	fail	fail	22	0.01	17	0.01	11	0.00	8	0.00
55	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	5	0.00	2	0.00
56	2	0.00	2	0.00	2	0.00	2	0.00	2	0.00	4	0.00	2	0.00
57	2	0.00	fail	fail	fail	fail	fail	fail	2	0.00	2	0.00	1	0.00
58	7	0.00	7	0.00	10	0.00	10	0.00	10	0.00	15	0.00	7	0.00
59	11	0.00	21	0.00	11	0.00	11	0.00	11	0.00	34	0.00	9	0.00
60	12	0.00	21	0.00	16	0.00	16	0.00	16	0.00	76	0.00	13	0.00
61	7	0.01	26	0.02	18	0.02	18	0.02	18	0.02	86	0.06	14	0.02
62	14	0.02	17	0.02	98	0.09	89	0.07	96	0.07	70	0.05	35	0.05
63	10	0.08	37	0.23	47	0.29	47	0.27	47	0.38	123	0.73	26	0.23
64	10	0.08	35	0.28	10	0.09	10	0.09	9	0.08	131	0.87	12	0.11
65	5	0.03	fail	fail	7	0.04	7	0.05	7	0.04	7	0.04	6	0.05
66	10	0.04	11	0.05	38	0.19	23	0.10	40	0.19	23	0.08	14	0.08
67	71	0.01	78	0.01	116	0.01	116	0.01	116	0.01	184	0.02	86	0.01
68	66	0.01	77	0.01	1271	0.09	1331	0.09	1176	0.09	149	0.01	109	0.02
69	18	0.00	32	0.00	196	0.01	222	0.02	194	0.01	807	0.03	26	0.00
70	36	0.00	33	0.00	297	0.04	211	0.02	736	0.03	3688	0.10	57	0.00
71	25	0.00	22	0.00	27	0.00	27	0.00	27	0.00	29	0.00	23	0.00
72	28	0.00	28	0.00	43	0.01	43	0.00	43	0.00	41	0.00	29	0.00
73	4	0.00	4	0.00	5	0.00	5	0.00	5	0.00	5	0.01	5	0.00
74	11	0.00	7	0.00	6347	0.20	fail	fail	6384	0.18	28	0.00	14	0.00
75	10	0.00	123	0.01	10	0.00	10	0.00	10	0.00	269	0.02	10	0.00
76	11	0.00	139	0.01	10	0.00	10	0.00	10	0.00	196	0.01	10	0.00
77	38	0.00	69	0.01	38	0.01	38	0.01	38	0.01	107	0.01	38	0.00
78	40	0.01	78	0.01	40	0.01	40	0.01	40	0.01	128	0.02	40	0.01
79	186	0.09	447	0.18	131	0.07	131	0.07	131	0.06	670	0.25	639	0.24
80	194	0.08	500	0.22	137	0.08	137	0.06	137	0.07	649	0.23	716	0.27
81	29	0.03	34	0.02	204	0.14	235	0.15	368	0.17	2030	0.27	50	0.02
82	32	0.03	31	0.02	336	0.18	1320	0.37	398	0.19	1577	0.28	50	0.03
83	49	0.09	57	0.12	1137	3.68	1202	3.55	1188	3.50	7631	3.99	75	0.13
84	50	0.10	fail	fail	1142	3.46	1461	4.13	1216	3.45	7044	5.30	76	0.14

(Continued on next page)

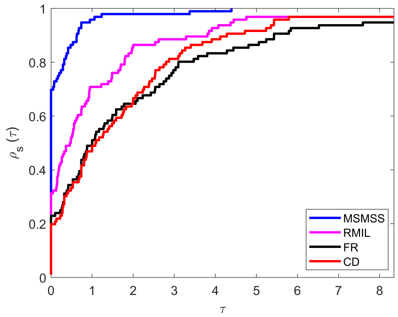
Table 3 – *Continued*

P	MSMSS		RMIL		FR		CD		DY		WYL		NPRP	
	N	C	N	C	N	C	N	C	N	C	N	C	N	C
85	9	0.00	14	0.00	20	0.00	20	0.00	20	0.00	15	0.00	19	0.00
86	9	0.00	15	0.00	51	0.00	51	0.00	51	0.00	17	0.00	10	0.01
87	39	0.00	802	0.04	272	0.02	272	0.01	272	0.02	608	0.03	1234	0.06
88	38	0.01	806	0.05	273	0.02	273	0.02	273	0.02	736	0.04	1198	0.06
89	1	0.00	1	0.00	1	0.00	1	0.00	1	0.00	1	0.00	1	0.00
90	1	0.00	1	0.00	1	0.00	1	0.00	1	0.00	1	0.00	1	0.01
91	128	0.02	1032	0.06	fail	fail	fail	fail	fail	fail	1276	0.04	1293	0.04
92	25	0.00	669	0.04	34	0.01	34	0.00	33	0.00	1448	0.05	578	0.04
93	9	0.01	35	0.00	16	0.00	16	0.00	16	0.00	53	0.00	14	0.00
94	21	0.03	46	0.01	23	0.00	25	0.00	25	0.00	66	0.01	47	0.07
95	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01
96	1	0.01	1	0.01	1	0.01	1	0.01	1	0.01	1	0.02	1	0.18
97	25	0.00	46	0.01	25	0.01	25	0.00	25	0.01	73	0.01	25	0.01
98	42	0.01	81	0.01	41	0.01	41	0.01	41	0.01	149	0.02	41	0.83

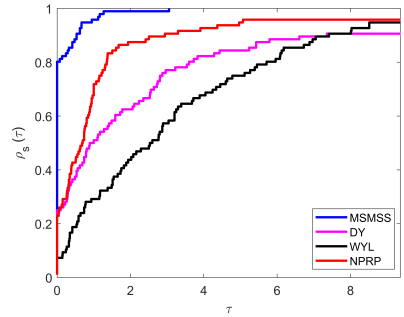
Table 4. Summary of the numerical results.

Method	Exact Line Search			Strong Wolfe Line Search		
	NOI	CPU	FAIL	NOI	CPU	FAIL
MSMSS	2,968	37.0108	0	2,542	4.0639493	0
RMIL	8,510	85.4049	3	8,419	5.568807	10
FR	28,449	330.0861	7	37,293	30.1217298	6
CD	19,978	295.7988	5	37,863	27.067427	6
DY	23,582	295.5019	11	32107	26.5428355	9
WYL	55,755	1706.7404	5	69,392	157.0444284	2
NPRP	10,586	73.4837	4	9,631	7.1375309	3

The numerical results are summarized in Table 4, that is, the total number of iterations (NOI), CPU time, and fails when using the exact and strong Wolfe line searches. Under the exact line search, the MSMSS method reach the solution point for all problems with 100% success, compared to RMIL 96%, FR 92%, CD 94%, DY 88%, WYL 94%, and NPRP 95%. On the other hand, for the results under the strong Wolfe line search, the MSMSS method reaches solution points of all the test problems with 100% success compared to RMIL method with 89%, the FR and CD with 93%, the DY with 90%, the with WYL 97%, and the NPRP with 96%. In this regard, we can say the MSMSS method is the best. Besides that, the performance of these methods can further be analyzed based on the performance profile curve.

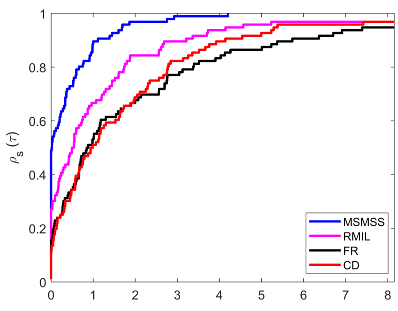


(A) Comparison of MSMSS, RMIL, FR, and CD methods

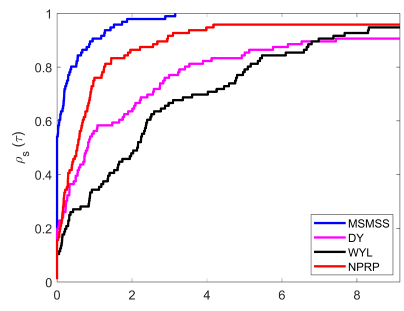


(B) Comparison of MSMSS, DY, WYL, and NPRP methods

FIGURE 1. Performance profile based on NOI using exact line search



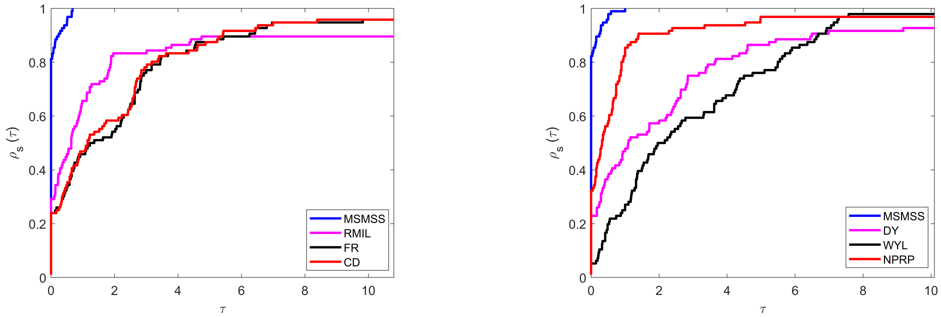
(A) Comparison of MSMSS, RMIL, FR, and CD methods



(B) Comparison of MSMSS, DY, WYL, and NPRP methods

FIGURE 2. Performance profile based on CPU using exact line search

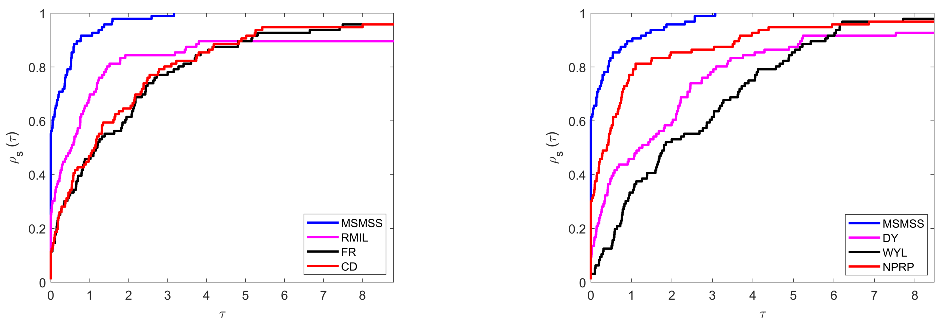
The performance profile is based on the performance tool proposed by Dolan and Moré [8]. This tool aims to analyze which method performs better in terms of NOI or CPU time. The formulas used to describe the performance profile will be explained as follows. Suppose S is set of n_s solvers, P is set of n_p test functions. For each solver $s \in S$ and function $p \in P$, consider $a_{p,s}$ as the number of iterations or CPU time required to solve function $p \in P$ by solver $s \in S$. Then the solver's comparison is based on the performance ratio: $r_{p,s} = \dfrac{a_{p,s}}{\min\{a_{p,s} : p \in P \text{ and } s \in S\}}$. The overall evaluation of the solvers output is then obtained from the output profile feature as follows: $\rho_s(\tau) =$



(A) Comparison of MSMSS, RMIL, FR, and CD methods

(B) Comparison of MSMSS, DY, WYL, and NPRP methods

FIGURE 3. Performance profile based on NOI using exact line search



(A) Comparison of MSMSS, RMIL, FR, and CD methods

(B) Comparison of MSMSS, DY, WYL, and NPRP methods

FIGURE 4. Performance profile based on CPU using strong Wolfe line search

$\dfrac{1}{n_p} size\{p \in P : \log_2 r_{p,s} \leq \tau\}$, with $\rho_s(\tau)$ as the probability for solvers that a performance ratio $\rho_s(\tau)$ is within a factor $\tau \in R$ of the best possible ratio. In general, solvers with high values of $\rho_s(\tau)$ or in the upper right of the image represent the best solver.

The performance profiles are illustrated in Fig. 1, Fig. 2, Fig. 3 and Fig. 4. The performance profiles of MSMSS, RMIL, FR, CD, DY, WYL, and NPRP with regards to the number of iterations methods under the exact and strong Wolfe line searches are shown in Fig. 1 and Fig. 3, respectively. While the performance profile for the CPU time is given in Fig. 2 and Fig. 4. A thorough examination of the left side of these figures shows that the top curve represents the MSMSS method. Hence, this method presents the best performance compared to other existing methods considered.

5. Application in portfolio selection

In this section, we demonstrates the computational performance of the MSMSS method on minimizing risk in portfolio selection. As described in [28], we assume a portfolio consists of a collection of assets $\mathcal{A}_1, ..., \mathcal{A}_i$ in a given proportion. The return R_i on asset $\mathcal{A}_i$ is formulated by equation

$$R_i = \frac{P_i - P_{i-1}}{P_{i-1}},$$

where P_i, P_{i-1} are the price at time t and $t-1$, respectively. We consider the expected value and the variance of the return of asset. The expected return of asset $\mathcal{A}_i$ is defined by

$$\mu_i = E(R_i),$$

and the variance of the return of asset $\mathcal{A}_i$

$$\xi_i^2 = Var(R_i)$$

is called the risk of asset $\mathcal{A}_i$. Now, we will present the return, expected return, and risk of portfolio. The return of portfolio is formulated to be the weighted sum of the returns of each asset

$$R = \sum_{i=1}^{n} w_i R_i$$

where the sum of weights $\sum_{i=1}^{n} w_i = 1$. Meanwhile, the expected return of portfolio is defined as the expected value of the portfolio's return, that is,

$$\mu = E\left(\sum_{i=1}^{n} w_i R_i\right) = \sum_{i=1}^{n} w_i \mu_i, \tag{5.1}$$

and the risk of portfolio is defined as the variance of the portfolio's return, that is,

$$\xi^2 = \mathrm{V}ar\left(\sum_{i=1}^{n} w_i R_i\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \mathrm{Cov}(R_i, R_j), \tag{5.2}$$

where $\mathrm{Cov}(R_i, R_j)$ is the covariance of R_i and R_j.

In this paper, we consider the portfolio of stock, and we take four blue chip stock in Indonesia, that is, PT Indofood CBP Sukses Makmur Tbk (ICBP), PT Unilever Indonesia Tbk (UNVR), PT Telekomunikasi Indonesia Tbk (TLKM), and PT Perusahaan Gas Negara Tbk (PGAS). Our main goal here is to determine the proportion of stock that must be allocated to provide minimal risk. So, our problem can be formulated as follows:

$$\begin{cases} \text{minimize} : \xi^2 = \text{V}ar\left(\sum_{i=1}^4 w_i R_i\right) = \sum_{i=1}^4 \sum_{j=1}^4 w_i w_j \text{Cov}(R_i, R_j) \\ \text{subject to} : \sum_{t=1}^4 w_t = 1 \end{cases}. \tag{5.3}$$

where w_1, w_2, w_3, w_4 are the proportion of UNVR, TLKM, ICBP, and PGAS stocks, respectively. To adjust to problem (1.1), we change problem (5.3) into unconstrained model by writing $w_4 = 1 - w_1 - w_2 - w_3$. So that, the our model is defined as follows:

$$\min_{\mathbf{w}\in\mathbb{R}^4} \left\{ \sum_{i=1}^{4} \sum_{j=1}^{4} w_i w_j \text{Cov}(R_i, R_j) \right\}, \tag{5.4}$$

where $\mathbf{w} = (w_1,\, w_2,\, w_3,\, 1 - w_1 - w_2 - w_3)$.

The stock price used is the weekly closing price taken from the website http: //finance.yahoo.com, over three years (Jan 1, 2018 - Dec 31, 2020). The movement of stock's closing price is presented in Figure 5, and based on this data, we have the values of mean, variance and covariance for UNVR, TLKM, ICBP, and PGAS stocks in Table 5.

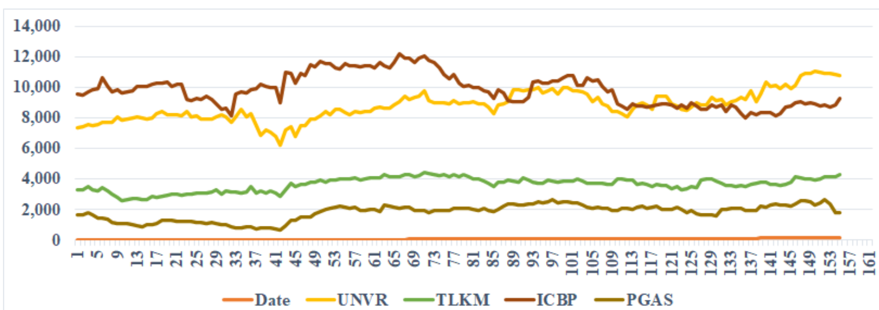


Figure 5. Movement of Stock's Closing Price (IDR)

According to Table 5, we can be excuting the problem (5.4) in MATLAB using the exact and strong Wolfe line searches by choosing some random initial points. The results are $w_1 = 0.39$, $w_2 = 0.33$, $w_3 = 0.33$, $w_4 = -0.05$. By using the value of w_1, w_2, w_3, w_4, (5.1), and (5.2), we get $\mu = 0.02$ and $\xi^2 = 0.0008$. A negative sign in the proportion indicates that investor is short selling. Hence, the selection of stock portfolios for our case with a minimum risk can be done by allocating each stock in the following proportions, that

TABLE 5. Mean, variance, and covariance of return's stock

Stocks	Mean	Variance	Covariance	UNVR	TLKM	ICBP	PGAS
UNVR	0.00311	0.00127	UNVR	0.00127	0.00053	0.00062	0.00105
TLKM	0.00247	0.00166	TLKM	0.00053	0.00166	0.00048	0.001579
ICBP	0.00047	0.00142	ICBP	0.00062	0.00048	0.00142	0.000858
PGAS	0.00359	0.00667	PGAS	0.00105	0.00158	0.00086	0.00667

is, UNVR 39%, TLKM 33%, ICBP 33%, and PGAS −5% with the expected return is 2% and the portfolio risk value is 0.0008.

6. Conclusion

In this paper, we proposed a new parameter of CG method by modifying the parameter of the NPRP method. The proposed method satisfies the sufficient descent condition and global convergence properties under the exact and strong Wolfe line searches with $0 < \sigma < \frac{1}{4}$. The numerical results based on some test functions show that the proposed method is efficient compare to other existing methods considered. Lastly, the proposed method was extended to solve an application problem of minimizing risk problems in portfolio selection.

References

[1] A.B. Abubakar, P. Kumam and A.M. Awwal, *Global convergence via descent modified three-term conjugate gradient projection algorithm with applications to signal recovery*, Results Appl. Math., **4** (2019), p.100069.

[2] A.B. Abubakar, P. Kumam, M. Malik, P. Chaipunya, and A.H. Ibrahim, *A hybrid FR-DY conjugate gradient algorithm for unconstrained optimization with application in portfolio selection*, AIMS Mathematics, **6**(6) (2021), 6506-6527.

[3] M. Al-Baali, *Descent property and global convergence of the Fletcher-Reeves method with inexact line search*, IMA J. Numer. Anal., **5**(1) (1985), 121-124.

[4] N. Andrei, *An Unconstrained Optimization Test Functions Collection*, Adv. Model. Optim., **10**(1) (2008), 147-161.

[5] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., **16**(1) (1966), 1-3.

[6] J. Cao and J. Wu, *A conjugate gradient algorithm and its applications in image restoration*, Appl. Numer. Math., **152** (2020), 243-252.

[7] Y.H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., **10**(1) (1999), 177-182.

[8] E.D. Dolan and J.J. More, *Benchmarking optimization software with performance profiles*, Math. Progr., **91**(2) (2002), 201-213.

[9] R. Fletcher and C.M. Reeves, *Function minimization by conjugate gradients*, The Comput. J., **2** (1964), 149-154.

[10] R. Fletcher, *Practical methods of optimization*, Wiley Interscience John Wiley and Sons, New York, USA, 1987.

[11] J.C. Gilbert and J. Nocedal, *Global convergence properties of conjugate gradient methods for optimization*, SIAM J. Optim., **2**(1) (1992), 21-42.

[12] M. Jamil and X.S. Yang, *A literature survey of benchmark functions for global optimisation problems*, Inter. J. Math. Model. Numer. Optim., **4**(2) (2013), 150-194.

[13] J.K. Liu, Y.M. Feng and L.M. Zou, *A spectral conjugate gradient method for solving large-scale unconstrained optimization*, Comput. Math. Appl., **77**(3) (2019), 731-739.

[14] M. Malik, S.S. Abas, M. Mamat, Sukono and I.S. Mohammed, *A new hybrid conjugate gradient method with global convergence properties*, Inter. J. Advan. Sci. Techn., **29**(5) (2020), 199-210.

[15] M. Malik, M. Mamat, S.S. Abas and Sukono, *Convergence analysis of a new coefficient conjugate gradient method under exact line search*, Inter. J. Advan. Sci. Techn., **29**(5) (2020), 187-198.

[16] M. Malik, M. Mamat, S.S. Abas, I.M. Sulaiman and Sukono, *A new coefficient of the conjugate gradient method with the sufficient descent condition and global convergence properties*, Engineering Letters, **28**(3) (2020), 704-714.

[17] M. Malik, M. Mamat, S.S. Abas, I.M. Sulaiman and Sukono, *A new spectral conjugate gradient method with descent condition and global convergence property for unconstrained optimization*, J. Math. Comput. Sci., **10**(5) (2020), 2053-2069.

[18] M. Malik, M. Mamat, S.S. Abas, I.M. Sulaiman and Sukono, *A new modification of NPRP conjugate gradient method for unconstrained optimization*, Advan. Math.: Scientific J., **9**(7) (2020), 4955-4970.

[19] M. Malik, M. Mamat, S.S. Abas, I.M. Sulaiman and Sukono, *Performance analysis of new spectral and hybrid conjugate gradient methods for solving unconstrained optimization problems*, IAENG Inter. J. Comput. Sci., **48**(1) (2021), 66-79.

[20] P. Mtagulwa and P. Kaelo, *An efficient modified PRP-FR hybrid conjugate gradient method for solving unconstrained optimization problems*, Appl. Numer. Math.,, **145** (2019), 111-120.

[21] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, 2000.

[22] E. Polak, *Algorithms and Consistent Approximations*, Springer, Berlin, 1997.

[23] E. Polak and G. Ribiere,*Note sur la convergence de méthodes de directions conjuguées*, ESAIM: Math. Model. Numer. Anal.-Modélisation Mathématique et Analyse Numérique, **3** (1969), 35-43.

[24] R. Pytlak, *Conjugate gradient algorithms in nonconvex optimization*, Springer Science & Business Media, **89**, 2008.

[25] B.T. Polyak, *The conjugate gradient method in extremal problems*, USSR Comput. Math. Math. Phy., **9**(4) (1969), 94-112.

[26] M. Rivaie, M. Mamat and A. Abashar, *A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches*, Appl. Math. Comput., **268** (2015), 1152-1163.

[27] M. Rivaie, M. Mamat, L.W. June and I. Mohd, *A new class of nonlinear conjugate gradient coefficients with global convergence properties*, Appl. Math. Comput., **218**(22) (2012), 11323-11332.

[28] S. Roman, *Introduction to the mathematics of finance: from risk management to options pricing*, Springer Science & Business Media, 2004.

[29] Z.J. Shi and J. Shen, *Convergence of PRP method with new nonmonotone line search*, Applied Math. Comput., **181**(1) (2006), 423-431.

[30] I. M. Sulaiman, M. Mamat, M.Y. Waziri, U.A. Yakubu and M. Malik, *The Performance Analysis of a New Modification of Conjugate Gradient Parameter for Unconstrained Optimization Models*, Math. Statistics, **9**(1) (2021), 16-23.

[31] Z. Wang, P. Li, X. Li and H. Pham, *A Modified Three-Term Type CD Conjugate Gradient Algorithm for Unconstrained Optimization Problems*, Math. Prob. Engin., **2020** (2020), Article ID 4381515.

[32] M.Y. Waziri, K. Ahmed and J. Sabi'u, *A family of Hager-Zhang conjugate gradient methods for system of monotone nonlinear equations*, Appl. Math. Comput., **361** (2019), 645-660.

[33] Z. Wei, S. Yao and L. Liu, *The convergence properties of some new conjugate gradient methods*, Appl. Math. Comput., **183**(2) (2006), 1341-1350.

[34] P. Wolfe, *Convergence conditions for ascent methods*, SIAM review, **11**(2) (1969), 226-235.

[35] P. Wolfe, *Convergence conditions for ascent methods. II: Some corrections*, SIAM review, **13**(2) (1971), 185-188.

[36] K. Yang, J. Geng-Hui, Q. Qu, H.F. Peng and X.W. Gao, *A new modified conjugate gradient method to identify thermal conductivity of transient non-homogeneous problems based on radial integration boundary element method*, Inter. J. Heat and Mass Transfer, **133** (2019), 669-676.

[37] S. Yao, Q. Feng, L. Li and J. Xu, *A class of globally convergent three-term Dai-Liao conjugate gradient methods*, Appl. Numer. Math., **151** (2020), 354-366.

[38] O.O.O. Yousif, *The convergence properties of RMIL+ conjugate gradient method under the strong Wolfe line search*, Appl. Math. Comput., **367** (2020), p.124777.

[39] G. Yuan, T. Li and W. Hu, *A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems*, Appl. Numer. Math., **147** (2020), 129-141.

[40] Y. Yuan and W.Y. Sun, *Optimization Theory and Methods*, Science Press, Beijing, 1997.

[41] L. Zhang, *An improved Wei–Yao–Liu nonlinear conjugate gradient method for optimization computation*, Appl. Math. Comput., **215**(6) (2009), 2269-2274.

[42] Z. Zhu, D. Zhang and S. Wang, *Two modified DY conjugate gradient methods for unconstrained optimization problems*, Appl. Math. Comput., **373** (2020), p.125004.

[43] G. Zoutendijk, *Nonlinear programming, computational methods*, Integer and nonlinear programming, 1970.