Applied Mathematical Sciences, Vol. 9, 2015, no. 96, 4799 - 4812
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.53243

A New Conjugate Gradient Method with Exact Line Search

Syazni Shoid[1], Mohd Rivaie[2], Mustafa Mamat[1] and Zabidin Salleh[3]

[1]Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin (Unisza), Terengganu, Malaysia

[2]Department of Computer Sciences and Mathematics
Universiti Teknologi MARA (UiTM), Terengganu, Malaysia

[3]School of Informatics and Applied Mathematics
Universiti Malaysia Terengganu (UMT), Terengganu, Malaysia

Abstract

Conjugate gradient (CG) methods have been practically used to solve large-scale unconstrained optimization problems due to their simplicity and low memory storage. In this paper, we proposed a new type of CG coefficients (β_k). The β_k is computed as an average between two different types of method which are Polak and Ribiere (PR) and Norrlaili et al. (NRMI). Numerical comparisons are made with the five others β_k proposed by the early researches. A set of eight unconstrained optimization problems with several different variables are used in this paper. It is shown that, the new proposed β_k with an exact line search is possessed global convergence properties. Numerical results also show that this new β_k outperforms some of these CG methods.

Keywords: Conjugate gradient method, conjugate gradient coefficient, global convergence, exact line searches

1 Introduction

Conjugate gradient (CG) method is one of methods for solving large-scale problems where it does not require matrix storage and its iteration cost is low compared with others method. Consider the optimization problem which is to be minimize as a function of *n* variables as,

$$\min \quad f(x) \quad \text{subject to } x \in X \tag{1}$$

where $f(x)$ is a real-valued function called the objective function. The $x \in R^n$ is denoted as decision variable and $X \in R^n$ is a constraint set or feasible set. If $x = R^n$, then optimization problem (1) can be express as an unconstrained optimization problem,

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

This problem is solved iteratively, which is $x_0 \in R^n$ become an initial point. By using the recurrence formula, the CG method has the following form,

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

where x_k is the current iteration point and the $\alpha_k > 0$ is a stepsize which is obtained by some line search method. Most line searches used in practice is inexact line searches and also known as approximate line search. In this procedure, the value of α_k is estimated that will give sufficient decreased of the objective function. One of the most common and popular of an inexact line search is Wolfe line search [12]. This line search introduced two conditions as follow,

$$f(x_k + \alpha_k d_k) \le f(x_k) + \mu \alpha_k \, g_k{}^T d_k \tag{3}$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k \tag{4}$$

where $0 < \mu < \sigma < 1$. By replacing (4) with condition (5), then this is called as strong Wolfe line search

$$\left| g_{k+1}^T d_k \right| \le -\sigma g_g^T d_k. \tag{5}$$

Another well-known line searches procedure is an exact line search. Lately, an exact line search has been a significant increase used in the number of research due to new generation of computer processors. In this procedure, the value of α_k is determine, such that the objective function with the d_k direction is exactly minimized. The formula is,

$$f(x_k + \alpha_k d_k) = \min_{\alpha_k \ge 0} f(x_k + \alpha_k d_k). \tag{6}$$

The advantage of using this line search is, it will minimizes the complexity of the algorithm. An exact line search also gives an accurate descent of the objective function. Furthermore, convergence proving for exact line search became much easier as compared to inexact line search [14]. Therefore, in this paper an exact line search is used. The d_k is the search direction defined by

$$d_k = \begin{cases} -g_k & \text{if} \quad k = 0 \\ -g_k + \beta_k d_{k-1} & \text{if} \quad k \geq 1 \end{cases} \tag{7}$$

where $g_k = \nabla f(x_k)$, and $\beta_k \in R$ is a coefficients which determines the different conjugate gradient methods. The following are the most common β_k proposed by the early researches,

$$\beta_k^{FR} = \frac{g_k^T g_k}{\|g_{k-1}\|^2}, \tag{8}$$

$$\beta_k^{PR} = \frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2}, \tag{9}$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}}, \tag{10}$$

$$\beta_k^{RMIL} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}, \tag{11}$$

where $g_{k-1} = \nabla f(x_{k-1})$ and $\|.\|$ denotes the Euclidian norm of vectors. The above corresponding methods are known as Fletcher and Reeves (FR) method [9], Polak and Ribiere (PR) method [1], Hestenes and Stiefel (HS) method [4], and Rivaie et al. (RMIL) method [5]. Recently, [16-18], and [21, 23] also gave a new β_k in order to improve this CG method.

$$\beta_k^{NRMI} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}{}^T (g_k - d_{k-1})}, \tag{12}$$

$$\beta_k^{RAMI} = \frac{g_{k+1}^T \left(g_{k+1} - \frac{\|g_{k+1}\|}{\|g_k\|} g_k \right)}{d_k{}^T (d_k - g_{k+1})}, \tag{13}$$

$$\beta_k^{NMR} = \frac{\left(\frac{g_k^T (g_k - g_{k-1})}{\|g_{k-1}\|^2} + \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}} \right)}{2}, \tag{14}$$

$$\beta_k^{AMRI} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|g_{k-1}\|} |g_k^T g_{k-1}|}{\|d_{k-1}\|^2}. \tag{15}$$

The FR has a strong convergence properties but it may has average practical performance due to jamming. For the methods of PR, HS and RMIL they may not always be convergent but often have better computational performance [10]. In theoretically, if $f(x)$ is a strongly convex quadratic, all these methods are equivalent with the use of an exact line search. For non quadratic functions, different choice of d_k will leads to different performance [19, 20].

The global convergences of CG methods have been studied by many researchers such as the first global convergence result for the FR method was given by Zoutendijk [2] in 1970. He proved that, the FR method poses globally convergence when the line search is exact. But in 1977, Powell [7] has proven the poor performance of the FR method due to jamming phenomenon. In [7], the global convergence of the PR method is established when the functions is strongly convex under the exact line search. However, Powell [6] later showed that the PR and HS methods could cycle infinitely without converging to minimizer when using an exact line search.

The stepsize α_k is define along the search direction after the d_k is calculated at each iteration. Progress toward minimum has been made if

$$f(x_{k+1}) < f(x_k).$$

The structure of this paper is organized as follows. In the section 2 a new type of CG coefficient is presented. In section 3, we presented the sufficient descent condition and global convergence proof of general CG methods, while in Section 4, some numerical results corresponding to the to this new β_k are given. Lastly, our discussion and conclusion are presented in Section 5 and Section 6 respectively.

2 New Type CG Coefficient

In this paper, we develop a new β_k formula named as β^{SRMI} [22]. Where, SRMI denotes the researchers name Syazni, Rivaie, Mustafa and Ismail. This new

β_k is resulted by average between Polak and Ribiere (PR) method in (3) and Norrlaili et al. (NRMI) method in (12). Hence,

$$\beta_k^{SRMI} = \frac{\dfrac{g_k^T(g_k - g_{k-1})}{\|g_{k-1}\|^2} + \dfrac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T(g_k - d_{k-1})}}{2} = \frac{PR + NRMI}{2}. \tag{16}$$

The following algorithm is the general algorithm of CG method used in this study.

Algorithm 1. The basic of Conjugate Gradient algorithm

Step 1: Given an initial point x_0 and set $k = 0$
Step 2: Computing conjugate gradient coefficient
Compute β_k based on (8) until (15)
Step 3: Computing search direction
$d_k = \begin{cases} -g_k & \text{if} \quad k = 0 \\ -g_k + \beta_k d_{k-1} & \text{if} \quad k \geq 1 \end{cases}$. If $g_k = 0$, terminate the execution of the algorithm .
Step 4: Computing step size α_k by exact line search rule
Solve $\alpha_k = \min_{\alpha > 0} f(x_k + \alpha d_k)$,
Step 5: Updating new point
Let $x_{k+1} = x_k + \alpha_k d_k$
Step 6: Convergent test and stopping criteria.
If $f(x_{k+1}) < f(x_k)$ and $\|g_k\| < \varepsilon$, then terminate.
Otherwise go to Step 1 with $k = k + 1$.

3 Convergent Analysis

The convergent properties of β^{SRMI} will also be studied. We only show the result of convergence for the general CG method. To prove the convergence, we assumed that every search direction d_k should satisfy descent condition

$$g_k^T d_k < 0 \tag{17}$$

for all $k \geq 0$. If there exists a constant $C > 0$ for all $k \geq 0$ then, the search directions satisfy following sufficiently descent condition.

$$g_k^T d_k \leq -C\|g_k\|^2 \tag{18}$$

Theorem 1

Consider a CG method with the line search direction (7) and β^{SRMI} given as (16), then condition (14) holds for all $k \geq 0$. ■

Proof:

If $k = 0$, then it is clear that $g_0^T d_0 = -C\|g_0\|^2$. Hence, condition (14) holds true. We also need to show that for $k \geq 1$, condition (14), will also hold true.
From (7), multiply by g_{k+1}^T then,

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

For exact line search, we know that $g_{k+1}^T d_k = 0$. Thus,

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 ,$$

which implies that d_{k+1} is a sufficient descent direction. Hence, $g_k^T d_k \leq -C\|g_k\|^2$ holds true. The proof is completed. ■

We also need the following assumption [11].

Assumptions 1

(i) $f(x)$ has lower bound, on the level set $\ell = \{x | f(x) \leq f(x_0)\}$ where x_0 is the starting point.
(ii) In a neighbourhood N *of* ℓ, the function $f(x)$ is continuously differentiable, and its gradient is Lipschitz
continuous; then, there exists a constant $L > 0$ such that
$\|g(x) - g(y)\| \leq L\|x - y\|$, for all $x, y \in N$.

Lemma 1

Assume that conditions (i) and (ii) hold and $x, x + d \in N$. Then,

$$f(x+d) - f(x) = \int_0^1 g(x+td)^T d dt$$
$$f(x+d) - f(x) \leq g(x)^T d + \tfrac{1}{2} L\|d\|^2$$

Let $x = x_k$ and $d = \alpha d_k$, then become

$$f(x_k + \alpha d_k) - f(x_k) = \alpha \int_0^1 g(x_k + \alpha t d_k)^T d_k dt$$
$$f(x_k + \alpha d_k) - f(x_k) \leq \alpha g(x_k)^T d_k + \tfrac{1}{2} \alpha^2 L\|d_k\|^2 \quad \text{(see [15]).} \ ■$$

Noted that, the convergence properties are the same either by using the exact or inexact.

Theorem 2

If condition (13) and Assumption 1 holds, then for any line search rule the following convergence properties holds,

$$\lim_{k\to\infty}\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2 = 0 .$$

Proof:

By exact line search, mean value theorem, Cauchy-Schwartz inequality, Assumption 1 (ii), set $\alpha = -\frac{g_k^T d_k}{L\|d_k\|^2}$, then we obtained

$$\begin{aligned} f(x_k) - f(x_{k-1}) \geq f(x_k) - f(x_k + \alpha d_k) &= -\alpha\int_0^1 g(x_k + \alpha t d_k)^T d_k dt \\ &= -\alpha g_k^T d_k - \alpha\int_0^1 (g(x_k + \alpha t d_k) - g_k)^T d_k dt \\ &\geq -\alpha g_k^T d_k - \alpha\int_0^1 \|g(x_k + \alpha t d_k) - g_k\| \cdot \|d_k\| dt \\ &\geq -\alpha g_k^T d_k - \frac{\alpha^2}{2} L\|d_k\|^2 = \frac{1}{2L}\left(\frac{-g_k^T d_k}{\|d_k\|}\right) \end{aligned}$$

By Assumption 1 (i) and (13), it follows that $\{f(x_k)\}$ is a monotone decreasing number sequence and has bound below, thus $\{f(x_k)\}$ has a limit, and therefore convergence properties holds [13].⦿

4 Numerical Results

We analyze the efficiency of the newly proposed CG coefficient SRMI, as compared to other classical CG methods such as FR, PR, HS and RMIL. The comparisons are based on the number of iterations and Central Processing Unit (CPU) time per second to reach minimizer. All these methods have been tested using eight different standard functions problems. Besides that, these test problems was tested several times for selected ranges number of variable n=2, 4 and 10. On the other hand, all the comparisons are done with four different initial points, starting from a point that is closer to the solution point, to the point further away from the solution point.

We considered $\varepsilon = 10^{-6}$ and all these methods terminate when the stopping criteria $\|g_k\| < 10^{-6}$ is fulfilled. All the problems mention below are solved by *Maple13* subroutine program using the exact line search. We record the number of iteration and CPU time in purpose of our comparisons. The results will be shown in Table 1 and Table 2 respectively. There are two conditions where are the iteration is considered as failed. The first condition is when the routines is stopped since it is fail to find the positive value of stepsize and the second condition is when iteration is exceed 1000. In the Table 1, the word “Fail” is represent the first condition while the word “Fail*” is represent the second condition. In the Table 2, the symbol “NA” which denote not available represented the result for both condition. We further simplifies Table 1 and Table 2 and shown the percentage performance of SRMI as compared to the other method in Table 3 and Table 4 respectively.

The following test functions are based on Andrei [8] and Molga and Smutnicki [3].

TABLE 1: Performance comparison of different CG method based on number of Iterations

No.	Function	Initial Point	SRMI	FR	PR	HS	RMIL
1	Rosenbrock (*n*=2)	(10,10)	18	442	20	20	15
		(15,15)	22	404	18	18	16
		(20,20)	26	Fail*	18	18	15
		(50,50)	30	Fail*	29	29	23
2	Rosenbrock (*n*=10)	(10,...,10)	33	Fail	176	177	Fail*
		(15,...,15)	336	Fail	382	403	Fail*
		(20,...,20)	318	Fail	246	205	Fail*
		(50,...,50)	374	Fail*	493	529	Fail*
3	Shalow (*n*=4)	(15,15, 15,15)	10	261	11	11	18
		(20,20, 20,20)	11	277	10	10	14
		(50,50, 50,50)	13	889	15	15	17
		(80,80, 80,80)	14	Fail*	18	18	20
4	Shalow (*n*=10)	(15,...,15)	11	Fail*	12	12	22
		(20,...,20)	12	Fail*	11	11	15
		(50,...,50)	14	Fail*	16	16	18
		(80,...,80)	15	Fail	19	19	21
5	Cube (*n*=4)	(7,7, 7,7)	24	305	23	Fail*	28
		(14,14,14,14)	33	Fail*	22	Fail*	24
		(21,21,21,21)	37	Fail*	28	Fail*	28
		(28,28,28,28)	52	Fail*	31	Fail*	21
6	Wood (*n*=4)	(3,3,3,3)	64	31	68	68	230
		(5,5,5,5)	77	30	50	50	222
		(10,10,10,10)	94	33	63	63	178
		(13,13,13,13)	66	41	132	198	749

TABLE 1: (Continued): Performance comparison of different CG method based on number of Iterations

7	Liarwhd (*n*=2)	(10,10) (25,25) (50,50) (100,100)	10 10 10 10	Fail* Fail* 992 939	12 11 11 11	12 11 11 11	14 13 13 13
8	Three Hump(*n*=2)	(4,-4) (8,-8) (16,-16) (32,-32)	6 5 4 4	10 7 5 5	5 4 4 4	5 4 4 4	7 7 6 5
9	Six Hump (*n*=2)	(-10,-10) (-15,-15) (10,10) (15,15)	6 6 6 6	7 7 7 7	6 6 6 6	6 6 6 6	6 6 6 6
10	White and Holst (n=4)	(4,4,4,4) (8,8,8,8) (12,12,12,12) (16,16,16,16)	321 413 257 413	Fail* Fail* Fail* Fail*	403 151 442 151	448 678 255 742	Fail* Fail* Fail* Fail*

TABLE 2: Performance comparison of different CG method based on CPU time

No	Function	Initial Point	SRMI	FR	PR	HS	RMIL
1	Rosenbrock (*n*=2)	(10,10) (15,15) (20,20) (50,50)	0.7956 1.0140 1.0140 1.2168	14.4613 14.0557 NA NA	0.7956 0.7176 0.7020 1.0140	0.7956 0.6864 0.6708 1.0608	0.5928 0.5304 0.6396 0.8268
2	Rosenbrock (*n*=10)	(10,...,10) (15,...,15) (20,...,20) (50,...,50)	4.0560 20.9977 21.1069 25.3814	NA NA NA NA	10.8577 23.0102 14.9290 29.7962	10.6549 24.1334 13.6969 34.5542	NA NA NA NA
3	Shalow (*n*=4)	(15,...,15) (20,...,20) (50,...,50) (80,...,80)	0.4212 0.5304 0.5616 0.5928	9.6565 9.0481 31.2938 NA	0.3744 0.4680 0.6552 0.7800	0.3900 0.5772 0.5928 0.7956	0.7020 0.5460 0.7020 0.7800
4	Shalow (*n*=10)	(15,...,15) (20,...,20) (50,...,50) (80,...,80)	0.7800 0.7176 0.9204 0.9672	NA NA NA NA	0.6396 0.5772 0.8580 1.1388	0.6864 0.7800 0.9360 1.1856	0.9048 0.6084 0.8892 0.9360
5	Cube (*n*=4)	(7,...,7) (14,...,14) (21,...,21) (28,...,28)	2.0592 2.6364 3.0732 4.2744	26.7853 NA NA NA	2.2620 1.9812 2.4024 2.6988	NA NA NA NA	2.3400 1.9812 2.3868 1.6692

TABLE 2: (Continued): Performance comparison of different CG method based on CPU time

6	Wood (*n*=4)	(3,...,3) (5,...,5) (10,...,10) (13,...,13)	2.9796 3.4320 4.3680 2.8080	1.2480 1.2948 1.3104 1.6224	2.6208 2.2932 2.5740 5.4288	2.6988 2.6988 4.3212 8.3461	8.6581 8.6737 6.2712 27.2690
7	Liarwhd (*n*=2)	(10,10) (25,25) (50,50) (100,100)	0.4212 0.4524 0.4992 0.4212	NA NA 36.3950 33.1814	0.5616 0.4056 0.5460 0.4836	0.6708 0.5928 0.5928 0.4524	0.5304 0.5616 0.5148 0.5928
8	Three Hump(*n*=2)	(4,-4) (8,-8) (16,-16) (32,-32)	0.5304 0.4836 0.3900 0.4056	0.8112 0.5460 0.4836 0.4680	0.5304 0.3432 0.3900 0.4368	0.4680 0.3900 0.4056 0.3900	0.6552 0.5928 0.5304 0.4212
9	Six Hump (*n*=2)	(-10,-10) (-15,-15) (10,10) (15,15)	0.6396 0.5772 0.4836 0.5460	0.5616 0.5616 0.7176 0.7176	0.4836 0.5148 0.6396 0.6240	0.5616 0.5460 0.5460 0.5928	0.5304 0.6084 0.5460 0.6240
10	White and Holst (n=4)	(4,...,4) (8,...,8) (12,...,12) (16,...,16)	31.1066 39.5151 24.4766 38.2046	NA NA NA NA	37.6586 14.1805 42.6351 13.0885	42.3855 62.6032 23.1506 70.3253	NA NA NA NA

5 Discussion

From Table 1, we see that for all given problems, SRMI and PR successfully reach solution point without exceedingly 1000 iteration. Otherwise, in certain problems, FR HS and RMIL is considered as failed once exceed 1000 or fail to find the positive value of stepsize. Thus, in Table 2 there is no recorded of CPU time for failed problems. Besides that, SRMI also outperformed FR, HS and RMIL in almost all the problems. The words ‘successful’ in Table 3 means that SRMI has achieved the minimizer with less number of iterations compared to FR, PR, HS and RMIL. Besides that, SRMI to other methods get ‘equivalent’ in number of iteration and the world ‘unsuccessful’, means SRMI get worse result compared to others methods. In Table 4, the SRMI is said to be as ‘successful’ when it has achieved the minimizer with the least duration of CPU time compared to others methods. In some problems, SRMI has achieved ‘equivalent’ to others methods in CPU time to reach minimizer. The SRMI is said to be as ‘unsuccessful’, when it needed longer time to reach minimizer compared to others methods.

TABLE 3: Percentage performance of SRMI compared to other CG methods based on number iteration

Method		Comparison			
		FR	PR	HS	RMIL
SRMI	Successful	90.0%	45.0%	57.5%	72.5%
	Equivalent	0.0%	15.0%	15.0%	10.0%
	Unsuccessful	10.0%	40.0%	27.5%	17.5%

From Table 3, it is shown that SRMI is superior when compared to FR, HS, and RMIL. The highest percentage of successful comparison is with FR which is 90.0% and followed by RMIL which is 72.5% and HS which is 57.5%. Though the successful rate comparison for PR is the lowest at 45.0%, their combined rate of successful rate and equivalent rate are equal to 60.0%. Above all, almost all the comparisons showed that the combined rate of successful and equivalent rate exceed 50.0%. Therefore, we considered that, SRMI is superior compared to FR, PR, HS, and RMIL in term number of iteration.

TABLE 4: Percentage performance of SRMI compared to other CG methods based on CPU time.

Method		Comparison			
		FR	PR	HS	RMIL
SRMI	Successful	87.5%	40.0%	60.0%	75.0%
	Equivalent	0.0%	7.5%	2.5%	0.0%
	Unsuccessful	12.5%	52.5%	37.5%	25.0%

From Table 4, it is shown that SRMI is superior whem compared to FR, HS, and RMIL with the least duration of CPU time. The highest percentage of successful comparison is with FR at 87.5%, followed by RMIL which is 75.0% and HS which is 60.0%. However, the successful rate comparison for PR is low at 40.0%. Above all, almost all the comparisons showed that the combined rate of successful and equivalent rate exceed 50.0% except PR. Therefore, we considered that, SRMI is superior compared to FR, HS, and RMIL but inferior when compared to PR in term of CPU time.

6 Conclusion

In this paper, a new β_k named as SRMI has been presented. Based on the result, SRMI shows that it satisfied the sufficiently descent condition. From the above numerical experiments with 8 test problems we have the computational evidence that SRMI is the best when compare to others standard CG methods. Though the successful rate of SRMI is low compare to PR, but it could be an alternative method when the other methods fail by using the exact line search. For further research, we should do more numerical experiment with the other standard test functions with larger scale or variables. We also hope to establish the global convergence properties and the linear convergence rate theoretically.

Acknowledgments. We would like to thank the anonymous referees for valuable suggestions and comments which improved our paper greatly. We also grateful to The Ministry of Higher Education of Malaysia for the funding of this research under the Fundamental Research Grant Scheme (Vot 59256).

References

[1] E. Polak, and G. Ribiere, Note sur la convergence de directions conjugees. *Rev.Francaise Informat Recherche Operationelle, 3E Annee*, **16** (1969), 35-43.

[2] G. Zoutendijk, Nonlinear Programming Computational Methods, in *Integer and Nonlinear Programming*, J. Abadie (editor), (1970), 37-86.

[3] M. Molga and C. Smutnicki, Test Functions for optimization needs, (2005).

[4] M. R. Hestenes and E. Steifel, Method of conjugate gradient for solving linear equations, *J,Res.Nat.Bur.Stand.*, **49** (1952), 409-436.
http://dx.doi.org/10.6028/jres.049.044

[5] M. Rivaie, M. Mamat, W. J. Leong and M. Ismail, A new class of nonlinear Conjugate Gradient Coefficient with global convergence properties, *Applied Mathematics and Computation*, **218** (2012), 11323 – 11332.
http://dx.doi.org/10.1016/j.amc.2012.05.030

[6] M. J. D. Powell, Nonconvex Minimizations Calculations and the Conjugate Gradient Method, *Lecture Notes in Mathematics*, Berlin, Springer, **1066** (1984), 122-141. http://dx.doi.org/10.1007/bfb0099521

[7] M. J. D. Powell, Restart Procedures for the Conjugate Gradient Method, *Mathematical Programming*, **12** (1977), 241-254.
http://dx.doi.org/10.1007/bf01593790

[8] N. Andrei, An Unconstrained Optimization Test Function Collection, *J. Adv. Modeling and Optimization*, **10** (2008), 147-161.

[9] R. Fletcher and C. Reeves, Function minimization by conjugate gradients, *Comput.J.*, **7** (1964), 149-154. http://dx.doi.org/10.1093/comjnl/7.2.149

[10] A. Y. Al-Bayati, and R. Z. Al-Kawaz, A new hybrid WC-FR conjugate gradient algorithm with modified secant condition for unconstrained optimization. *J. Math. Comp. Sci.*, **2** (2012), 937-966.

[11] Y. H. Dai, J. Y. Han, G. H. Liu, D. F. Sun, X. Yin, and Y. Yuan, (Convergence properties of nonlinear conjugate gradient method. *SIAM J. Optim.,* **10** (1999), 348-358. http://dx.doi.org/10.1137/s1052623494268443

[12] P. Wolfe, Convergence conditions for ascent method. II: some corrections. *SIAM Rev*., **13** (2) (1971), 185-188. http://dx.doi.org/10.1137/1013035

[13] Z. J. Shi, Convergence of line search methods for unconstrained optimization. *Applied Mathematics and Computation*., **157** (2004), 393-405. http://dx.doi.org/10.1016/j.amc.2003.08.058

[14] W. Sun, and Y. X. Yuan, *Optimization theory and method (nonlinear programming)*. Springer Science and Business Media, LLC (2006).

[15] Z. J. S. Shi, and J. Shen, On step-size estimation of line search methods. Applied *Mathematics and Computation*., **173** (2006), 360-371. http://dx.doi.org/10.1016/j.amc.2005.04.039

[16] N. Shapiee, R. M. Mamat, and I. Mohd, A new modification of Hestenes-Stiefel method with descent properties, *AIP Conference Proceedings,* **1602** (2014), 520-526. http://dx.doi.org/10.1063/1.4882535

[17] M. Rivaie, A. Abashar, M. Mamat and I. Mohd, The convergence properties of a new type of conjugate gradient methods, *Applied Mathematical Sciences*, **8** (2014), 33-44. http://dx.doi.org/10.12988/ams.2014.310578

[18] N. H. M. Yussoff, M. Mamat, M. Rivaie and I. Mohd, A new conjugate gradient method for unconstrained optimization with sufficient descent, *AIP Conference Proceedings*, **1602** (2014), 514-519. http://dx.doi.org/10.1063/1.4882534

[19] Y. Dai and Y. Yuan, *Nonlinear conjugate gradient method*, Shanghai Scientific and Technical Publisher, Beijing (1998).

[20] Y. Yuan and W. Sun, *Theory and methods of optimization*, Science Press of China, Beijing (1999).

[21] A. Abashar, M. Mamat, M. Rivaie and I. Mohd, Global convergence properties of a new class of conjugate gradient method for unconstrained optimization, *Applied Mathematics Sciences,* **8** (67) (2014), 3307-3319. http://dx.doi.org/10.12988/ams.2014.43246

[22] S. Shoid, M. Rivaie, M. Mamat and I. Mohd, Solving unconstrained optimization with a new type of conjugate gradient method, *AIP Conference Proceedings*,**1602** (2014), 574-579. http://dx.doi.org/10.1063/1.4882542

[23] A. Abashar, M. Mamat, M. Rivaie, I. Mohd and O. Omer, The proof of sufficient descent condition for a new type of conjugate gradient methods, *AIP Conference Proceedings*, **1602** (2014), 296-303.
http://dx.doi.org/10.1063/1.4882502

Received: March 26, 2015; Published: July 8, 2015