International Journal of Engineering & Technology, 7 (4.30) (2018) 458-461

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

The Sufficient Descent Condition of Nonlinear Conjugate Gradient Method

Srimazzura Basri [1, 2]*, Mustafa Mamat [2], Puspa Liza Ghazali [3]

[1]*Faculty of Applied Sciences & Technology, Universiti Tun Hussein Onn Malaysia, Malaysia*
[2]*Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Malaysia*
[3]*Faculty of Economics and Management Sciences, Universiti Sultan Zainal Abidin, Malaysia*
**Corresponding author E-mail: mazzura@uthm.edu.my*

Abstract

Non-linear conjugate gradient methods has been widely used instrumental in solving large scale optimization. These methods has been proved that only required very low memory other than its numerical efficiency. Thus, many studies have been conducted to improve these methods to find the most efficient method. In this paper, we proposed a new non-linear conjugate gradient coefficient that guarantees sufficient descent condition. Numerical tests indicate that the proposed coefficient is better than the three classical conjugate gradient coefficients.

Keywords: *conjugate gradient; decent condition; exact line search; inexact line search; optimization.*

1. Introduction

Optimization has been extensively studied and has been widely applied in many branches of science, engineering, economics, management, industry and other areas. The most common methods for solving optimization problems is a conjugate gradient (CG) methods. These methods has selected because of very low memory requirements and the numerical efficiency, especially in solving large scale optimization problems.
The CG method has been proposed by Hestenes and Stiefel [1] in the 1952. Numerous studies have been done recently to improve these methods in order to find the most efficient method. Some CG have been proposed such as Fletcher and Reeves [2], Polak and Ribiere [3], Fletcher [4], Liu and Storey [5], Dai and Yuan [6], Rivaie et al. [7], Hamoda et al. [8] and Abashar et al. [9].
An unconstrained optimization problem is given as

$$\min\{f(x): x \in R^n\} \tag{1}$$

While continuously differentiable is given by $f: R^n \to R$ where R^n denotes an n-dimensional Euclidean space. CG method will be used to solve (1). This method are generated as

$$x_{k+1} = x_k + \alpha_k d_k \tag{2}$$

for current iteration, $k = 0, 1, 2, \ldots$. We use symbol x_k as current iterate point. While the step size, α_k is a line search. The search direction, d_k defined as below

$$d_k = \begin{cases} -g_k, & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{3}$$

where g_k denotes the gradient of $f(x)$ at x_k and β_k is a CG coefficient.

In this paper, a new class of CG coefficient which guarantee the condition of sufficient descent is proposed. We will test the proposed CG coefficient using exact and inexact line searches. To investigate whether the proposed CG coefficient is most efficient coefficient or not, we will test and compare it with the most well-known classical CG as follows:

$$\beta_k^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{4}$$

$$\beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} \tag{5}$$

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}} \tag{6}$$

where β_k^{FR}, β_k^{PRP} and β_k^{HS} is Fletcher-Reeves (FR) that proposed by Fletcher and Reeves [2], Polak-Ribiere (PRP) that proposed by Polak and Ribiere [3] and Hestenes-Steifel (HS) that proposed by Hestenes-Steifel [3] respectively.
In 1970, Zoutendijk [10] is the first researcher proved that FR converges globally using the exact line search. The global convergence of the non-linear CG method can be insuring by the condition of sufficient descent. Therefore, this condition have been studied by many researchers, such as Abashar et al. [9] and Andrei [11].
The paper is organized as follows: Section 2, we presents the proposed CG coefficient with two different algorithms using exact and inexact line searches. The sufficient descent condition of the proposed CG coefficient are proved in Section 3. Section 4 presents some numerical results to compare the proposed CG coefficient with three classical CG coefficients by the number of iterations and time of central processing unit (CPU) while the short discussion is given in Section 5.

2. Proposed New Conjugate Gradient

In this paper, we are interested to propose a new CG coefficient based on modification of PRP and HS coefficient which is given as below

$$\beta_k^{SM-1} = \frac{g_{k+1}^T(g_{k+1}-g_k)}{d_k^T\left(\frac{\|g_{k+1}\|}{\|g_k\|}g_{k+1}+d_k\right)} \tag{7}$$

The SM denotes Srimazzura and Mustafa. Two different algorithms of CG coefficient using two different line search are generated as follows:

Algorithm 1: Algorithm using exact line.

1st Step Identify an initial point $x_0 \epsilon R^n$. Use $k = 0$.

2nd Step Find α_k using exact line that defined by

$$f(x_k + \alpha_k d_k) = \min_{\alpha\geq 0} f(x_k + \alpha d_k) \tag{8}$$

3rd Step Calculate d_k based on (3). Stop when $g_k = 0$.

4th Step Compute β_k where β_k is defined by one of CG coefficient that given as (4) to (7).

5th Step Identify new points based on (2).

6th Step Convergent test and stopping criteria. Use $\|g_k\| \leq \varepsilon$ and $f(x_k) > f(x_{k+1})$ as stopping criteria. Then, repeat 1st Step by changing $k = 0$ with $k = k + 1$.

Algorithm 2: Algorithm using inexact line.

1st Step Identify an initial point $x_0 \epsilon R^n$. Use $k = 0$.

2nd Step Find α_k using inexact line. We use the condition of "strong Wolfe" that given by

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta\alpha_k g_k^T d_k \tag{9}$$

And

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k \tag{10}$$

3rd Step Calculate d_k based on (3). Stop when $g_k = 0$.

4th Step Compute β_k where β_k is defined by one of CG coefficient that given as (4) to (7).

5th Step Identify new points based on (2).

6th Step Convergent test and stopping criteria. Use $\|g_k\| \leq \varepsilon$ and $f(x_k) > f(x_{k+1})$ as stopping criteria. Then, repeat 1st Step by changing $k = 0$ with $k = k + 1$.

3. Sufficient Descent Condition

We will studies either the CG method with β_k^{SM-1} will possess sufficient descent condition under both exact and inexact line searches or not.

3.1. Sufficient descent condition under exact line search

CG method is sufficient descent if the sufficient descent condition,

$$g_k^T d_k \leq -C\|g_k\|^2 \tag{11}$$

holds for $k \geq 0$ where $C > 0$. The sufficient descent condition under exact line search needs the following theorem.

Theorem 1: *Suppose that a CG method with the search direction* (3) *and* β_k^{SM-1}. *According to the condition of sufficient descent, for all* $k \geq 0$, (8) *holds true.*

Proof: We use mathematical induction. The condition (11) holds true where $g_0^T d_0 \leq -C\|g_0\|^2$ when $k = 0$. Then, we need to verify that the condition (11) holds true when $k \geq 1$. Therefore, substitute $k + 1$ into (3) and multiply by g_{k+1}^T,

$$\begin{aligned} g_{k+1}^T d_{k+1} &= g_{k+1}^T(-g_{k+1} + \beta_{k+1} d_k) \\ &= -\|g_{k+1}\|^2 + \beta_{k+1} g_{k+1}^T d_k \end{aligned} \tag{12}$$

Since we use the exact line search, then $g_{k+1}^T d_k = 0$. Hence, we have $g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2$. Thus, $g_k^T d_k \leq -C\|g_k\|^2$ holds true. Therefore, the proof is completed.

3.2. Sufficient descent condition under inexact line search

The sufficient descent condition under inexact line search needs the following two assumptions and lemmas.

Assumption 1: *Suppose that the f(x) is bounded below on the level set R^n. In a neighbourhood N of the level set $\ell = \{x \in R^n | f(x) \leq f(x_0)\}$, f(x) is continuous. f(x) also differentiable at the point x_0.*

Assumption 2: *The gradient $g(x)$ is Lipschitz continuous in N, i.e. there exists a constant $\ell > 0$ such that $\|g(x) - g(y)\| \leq \ell\|x - y\|$ for all $x, y \in N$.*

Lemma 1: *If* g_k *and* d_k *are generated by inexact line search* (9) *and* (10) *and* $\sigma < \frac{1}{2}$ *then for all* $k \geq 0$, *we have*

$$\frac{\|g_k\|}{\|d_k\|} \leq \frac{1}{1 - c}$$

where $c \in (0, 1)$.

Lemma 2. *Suppose that these two assumptions holds true and α_k is compute using inexact line search* (9) *and* (10), *then*

$$\alpha_k \leq \frac{1 - c}{l}\frac{\|g_{k+1}\|}{\|g_{k+1}\| - \|g_k\|}$$

Theorem 2: *Suppose that these two assumptions holds,* $g_k^T d_k < 0$, d_{k+1} *is calculate by* (3), β_k by β_k^{SM-1} *and* α_k *by Lemma* 2, *then* (11) *holds.*

Proof: We substitute β_k^{SM-1} into (12), then we obtain

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

Thus (11) is equivalent to

$$\frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T\left(\frac{\|g_{k+1}\|}{\|g_k\|}g_{k+1} + d_k\right)} g_{k+1}^T d_k \leq (1 - C)\|g_{k+1}\|^2$$

We need to consider two conditions. One of the condition is the condition (11) holds true when $g_k^T d_k < 0$.

Another condition is for $g_k^T d_k > 0$, noting that $g_k^T d_k < 0$ and from Lemma 2, we have

$$\begin{aligned} g_{k+1}^T d_k &= (g_{k+1} - g_k)^T d_k + g_k^T d_k \\ &\leq (g_{k+1} - g_k)^T d_k \\ &\leq \|g_{k+1} - g_k\| \cdot \|d_k\| \\ &\leq l\alpha_k\|d_k\|^2 \\ &\leq (1 - C)\frac{\|g_{k+1}\|}{\|g_{k+1}\| - \|g_k\|}\|d_k\|^2 \end{aligned}$$

Hence,

$$\frac{g_{k+1}^T(g_{k+1} - g_k)}{d_k^T\left(\frac{\|g_{k+1}\|}{\|g_k\|}g_{k+1} + d_k\right)} g_{k+1}^T d_k \leq (1 - C)\|g_{k+1}\|^2$$

which means that the condition (11) holds for $g_k^T d_k > 0$. Thus, the proof is completed.

4. Numerical Results

The comparison among the proposed CG coefficient with FR, PRP and HS will be shown in this section. Since the proposed CG coefficient possess the condition of sufficient descent under exact and inexact line searches, then we generate two different algorithms for both exact and inexact line searches using MATLAB version 7.10.0 (R2013a) on an Intel Core i7.
We will use fifteen test problem functions considered in Andrei [12] in order to show the efficiency of the proposed CG. According to Hillstrom [13], four initial points will be used where the initial point should be closer to the solution point. Table 1 shows all the test problem functions and initial points that will be used.

Table 1: Fifteen Test Problem Functions

Number	Function	Number of Variables	Initial Points
1	Three Hump	2	(3,3), (24,24), (49,49), (62,62)
2	Six Hump	2	(4,4), (15,15), (39,39), (55,55)
3	Extended Beale	10, 100, 500, 1000	(1,1,…,1), (4,4,…,4), (6,6,…,6), (9,9,…,9)
4	Diagonal 4	10, 100, 500, 1000	(3,3,…,3), (45,45,…,45), (67,67,…,67), (99,99,…,99)
5	Extended Freudenstein & Roth	10, 100, 500, 1000	(4,4,…,4), (14,14,…,14), (23,23,…,23), (30,30,…,30)
6	Extended Himmelblau	10, 100, 500, 1000	(2,2,…,2), (45,45,…,45), (81,81,…,81), (97,97,…,97)
7	Extended Quadratic Penalty QP2	10, 100, 500, 1000	(3,3,…,3), (27,27,…,27), (34,34,…,34), (66,66,…,66)
8	Generalized Quartic	10, 100, 500, 1000	(2,2,…,2), (36,36,…,36), (66,66,…,66), (99,99,…,99)
9	Raydan 2	10, 100, 500, 1000	(1,1,…,1), (2,2,…,2), (4,4,…,4), (8,8,…,8)
10	Extended Rosenbrock	10, 100, 500, 1000	(4,4,…,4), (60,60,…,60), (83,83,…,83), (100,100,…,100)
11	Extended Tridiagonal 1	10, 100, 500, 1000	(7,7,…,7), (43,43,…,43), (81,81,…,81), (97,97,…,97)
12	Extended White & Holst	10, 100, 500, 1000	(2,2,…,2), (7,7,…,7), (11,11,…,11), (13,13,…,13)
13	Extended Denschnb (CUTE)	10, 100, 500, 1000	(2,2,…,2), (34,34,…,34), (63,63,…,63), (91,91,…,91)
14	Fletchcr (CUTE)	10, 100, 500, 1000	(9,9,…,9), (27,27,…,27), (38,38,…,38), (88,88,…,88)
15	Nonscomp (CUTE)	10, 100, 500, 1000	(64,64,…,64), (77,77,…,77), (85,85,…,85), (94,94,…,94)

For each algorithm, we use $\varepsilon = 10^{-6}$ and $\|g_k\| \le 10^{-6}$, the gradient value as the stopping criteria. Each CG coefficient was tested using fifteen test problem functions with different number of variables and initial points. In this paper, we will use the speed of convergence as the performance criteria. Each algorithm will calculate the time of CPU to generate iterations' number. For both algorithms, we will set the number of iterations limited to 10000.
The performances comparison in percentage between the proposed CG coefficient with FR, PRP and HS for exact line search are shown in Table 2. While Table 3 shows performances comparison in percentage between the proposed CG coefficient with FR, PR and HS for inexact line search.

Table 2: Performances comparison in percentage based on iterations' number and time of CPU (in second) for exact line search

Coefficient	Total Iteration	Total CPU Time	Successful	Unsuccessful
FR	240474	6354.669	82.41%	17.59%
PRP	34361	309.1615	98.61%	1.39%
HS	3743	54.7167	82.41%	17.59%
SM-1	9220	156.0576	100%	0%

Table 3: Performances comparison in percentage based iterations' number and time of CPU (in second) for inexact line search

Coefficient	Total Iteration	Total CPU Time	Successful	Unsuccessful
FR	189033	1616.32	82.41%	17.59%
PRP	44461	299.3697	98.15%	1.85%
HS	4473	9.8226	95.83%	4.17%
SM-1	9226	16.1277	100%	0%

From Table 2 and 3, we can see that three classical CG coefficients may fail either get the number of iterations more than 1000 or the algorithm fail to compute the positive step size.
All the numerical results can be summarized in performance profiles introduced by Dolan and More [14]. He suggested that all the results represent using cumulative. Figure 1 and 2 show performance profile relative to the iterations' number for exact and inexact line search respectively. Figure 3 and 4 show performance profile relative to the time of CPU for exact line and inexact line respectively.

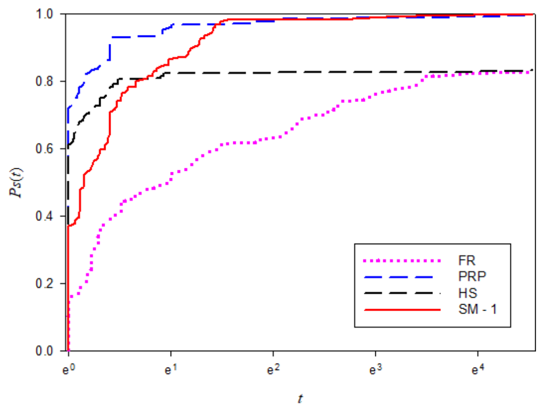

Fig 1: Performance profile for the iterations' number (exact line)

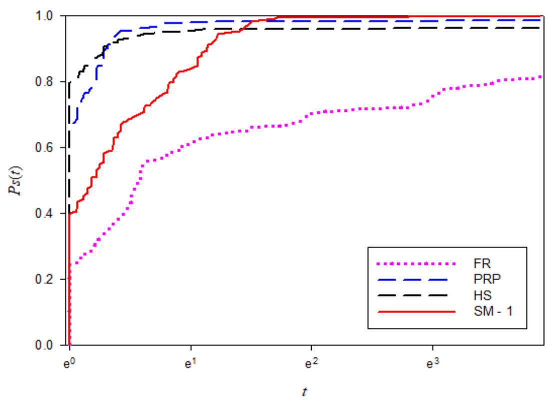

Fig 2: Performance profile for the iterations' number of (inexact line)

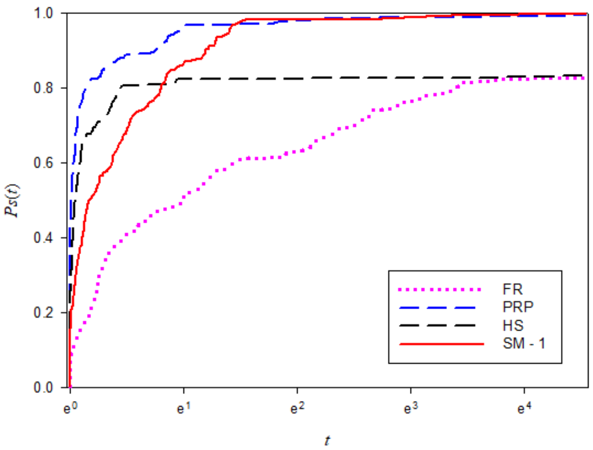

Fig 3: Performance profile for the time of CPU (exact line)

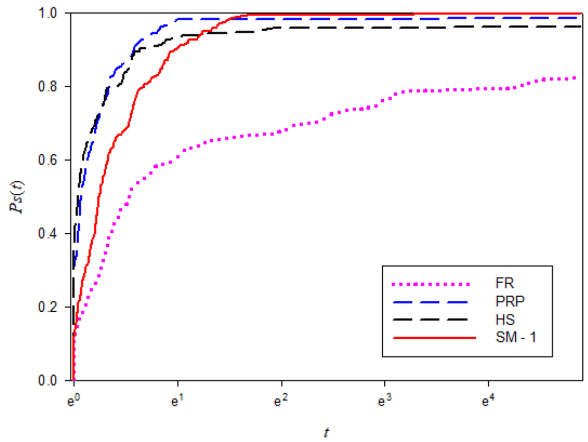

Fig 4:. Performance profile for the time of CPU (inexact line)

From Table 2, for the proposed CG that using exact line search, the total iterations' number is 9220 with total time of CPU is 156.0576 seconds without any failure. While for inexact line search, the proposed CG get 9226 number of iterations with 16.1277 seconds as shown in Table 3.

In Figure 1 – 4, we can see that the proposed CG coefficient always converges since it achieve the top of the curve in the performance profile compared the others.

5. Conclusion

From Table 2 and 3, we can conclude that performance of the proposed CG that using inexact line search is better than for the exact line search since it take a few second to converge. While from the Figure 1 – 4, by comparing the proposed CG with FR, PRP and HS, the numerical results suggest that the Srimazzura-Mustafa (SM) provide better performance compared to others. For further research, we should study this proposed CG either convergence globally under exact and inexact line since it possesses the sufficient descent condition.

Acknowledgement

The authors would like to thank the Universiti Sultan Zainal Abidin and Universiti Tun Hussein Onn Malaysia for the financial support.

References

[1] Stiefel E & Hestenes MR, Method of conjugate gradient for solving linear equation, *Journal of Research of the National Bureau of Standards* 49, (1952), pp. 409-436.

[2] Fletcher R & Reeves C, Function minimization by conjugate gradients, *The Computer Journal*, Vol.7, (1964), pp. 149-154.

[3] Polyak BT, The conjugate gradient method in extreme problems, *USSR Computational Mathematics and Mathematical Physics* 9, (1969), pp. 94-112.

[4] Fletcher R, *Practical method of optimization Vol. 1: unconstrained optimization*, John Wiley & Sons, New York, (1987).

[5] Liu Y & Storey C, Efficient generalized conjugate gradient algorithms, *Part 1: Theory, Journal of Optimization Theory and Applications* 69, (1991), pp. 129-137.

[6] Dai YH & Yuan Y, A nonlinear conjugate gradient method with a strong global convergence property, *SIAM Journal on Optimization* 10, (1999), pp. 177-182.

[7] Rivaie M, Mamat M, Leong WJ & Ismail M, A new conjugate gradient coefficient for large scale nonlinear unconstrained optimization, *International Journal of Mathematics Analysis* 6 (23), (2012), pp. 1131-1146.

[8] Hamoda M, M. Rivaie, M. Mamat and Z. Salleh (2015), A new nonlinear conjugate gradient coefficients for unconstrained optimization, *Applied Mathematical Sciences* 9 (37), pp. 1813-1822.

[9] Abashar A, Mamat M, Rivaie M, Mohd I & Omer O, The proof of sufficient descent condition for a new type of conjugate gradient methods, *AIP Conference Proceedings* 1602 (1), (2014), pp. 296-303.

[10] Zoutendijk G, *Nonlinear programming computational methods*, Chapter in Integer and Nonlinear Programming, North-Holland, Amsterdam, (1970), pp. 37-86.

[11] Andrei N, Dai–Yuan A, Conjugate gradient algorithm with sufficient descent and conjugacy conditions for unconstrained optimization, *Applied Mathematics Letters* 21, (2008), pp. 165-171.

[12] Andrei N, An unconstrained optimization test functions collection, *Advanced Modeling and Optimization* 10, (2008), pp. 147-161.

[13] Hillstrom KE, A simulation test approach to the evaluation of nonlinear optimization algorithms, *ACM Transactions on Mathematical Software* 3, (1977), pp. 305-315.

[14] Dolan DE & Mor JJ, Benchmarking optimization software with performance profiles, *Mathematical Programming* 91, (2002), pp. 201-213