International Journal of Engineering & Technology, 7 (3.28) (2018) 72-75

International Journal of Engineering & Technology

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

Research paper

A modification of steepest descent method for solving large-scaled unconstrained optimization problems

Siti Farhana Husin[1], Mustafa Mamat[1*], Mohd Asrul Hery Ibrahim[2], Mohd Rivaie[3]

[1]*Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UNISZA), Malaysia*
[2]*Faculty of Entrepreneurships and Business, Universiti Malaysia Kelantan (UMK), Malaysia*
[3]*Department of Computer Sciences and Mathematics, Universiti Teknologi MARA (UiTM), Terengganu, Malaysia*
**Corresponding author E-mail: must@unisza.edu.my*

Abstract

In this paper, we develop a new search direction for Steepest Descent (SD) method by replacing previous search direction from Conjugate Gradient (CG) method, d_{k-1}, with gradient from the previous step, g_{k-1} for solving large-scale optimization problem. We also used one of the conjugate coefficient as a coefficient for matrix g_{k-1}. Under some reasonable assumptions, we prove that the proposed method with exact line search satisfies descent property and possesses the globally convergent. Further, the numerical results on some unconstrained optimization problem show that the proposed algorithm is promising.

Keywords: *Steepest Descent method, Conjugate Gradient method; exact line search; global convergence*.

1. Introduction

Let $f : R^n \to R$ be continuously differentiable. Consider the unconstrained optimization problem:

$$\min f(x), \quad x \in R^n. \tag{1}$$

In general, let x_k is the current iterate point, α_k is the step length which is determined by some line search, d_k is a search direction, the iterative formula for solving (1) take the form:

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

In the steepest descent method [1], the search direction, d_k, is defined as the negative gradient direction

$$d_k = -g_k \tag{2}$$

and the most common line search for determining the step length is the exact line search, which is

$$\alpha_k = \min_{\alpha > 0} f(x_k + \alpha_k d_k). \tag{3}$$

In recent years, we have discovered an enlarging number of studies adopting the exact line search due to the new era of faster computer processors such as [2-4]. Different search directions correspond to different iterative methods. We concerned with the steepest descent (SD) method for solving (1).
SD method is efficient for medium-scale problems, but cannot be used to solve large-scale problems because it performs poorly and is badly affected by ill-conditioning [5]. Large-scale optimization is one of the vital research areas in both theory of optimization and algorithm structure. Hence, in this research, we focused on solving large-scale problems using SD method with dimensions up to 1,000.
This paper is systematized as follows. In the following section, we present new search direction of SD method and the steps of our new algorithm is generalized. In section 3, we prove its sufficient descent condition and global convergence of the corresponding algorithm. The numerical result are reported in section 4 and we complete this paper with some explanations in the last section.

2. New Search Direction

For simplicity, we denote $\nabla f(x_k)$ by g_k. The common form of a nonlinear conjugate gradient method is given by

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

where β_k is a scalar. The CG method was extended by Fletcher-Reeves (FR) and marks the start of the area of large-scale unconstrained optimization since it only needs storage of several vectors and is more rapid than SD method.
In [6] derived a modest sufficient descent method for solving unconstrained optimization problems. This method is modified from [7], Polak–Ribière–Polyak (PRP) conjugate gradient method and they substitute d_{k-1} as the gradient of the previous step in which the search direction is given by

$$d_k = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k + \left(I - \dfrac{g_k g_k^T}{\|g_k\|^2} \right) g_{k-1} & \text{if } k \geq 1 \end{cases}$$

In [8] introduced new search direction for SD method. In their research, they proved that the new search direction are globally convergent under the exact line search [4]. Their search direction are given as follows.

$$d_k^{ZMRI} = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k - \|g_k\| g_{k-1} & \text{if } k \geq 1 \end{cases}, \tag{4}$$

Inspired by those literatures, our objective for this paper is to introduce the modification of SD method for finding the search direction whose form is similar to (1) but with different coefficient of g_{k-1} which motivated by CG coefficient and it's called as FMAR1 given by the researcher's name, Farhana, Mustafa, Asrul and Rivaie.

$$d_k = -g_k - \theta_k g_{k-1}$$
$$\text{where } \theta_k = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \tag{5}$$

where g_k is the abbreviation of $g(x_k)$ and $\|.\|$ positions for the Euclidean norm of vectors.
Comparing to (1), we use θ_k as a coefficient matrix of g_{k-1} and this is motivated from CG coeffiecient which is Fletcher-Reeves (FR) coeffiecient

$$d_k = -g_k - \beta_k d_{k-1}$$
$$\text{where } \beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}$$

In this section, the steps of the proposed algorithm is given as follows.

Algorithm 1.
Step 1: Given an initial point x_0 and set $k = 0$
Step 2: Calculate search direction, d_k using (5).
Step 3: Calculate step size, α_k using (3).
Step 4: Update new point, x_{k+1} using (2).
If $\|g_k\| = 0$, then stop.
Step 5: Set $k = k+1$ and go to Step 2.

3. Convergence Analysis

The remainder of this section is dedicated to consider the sufficient descent condition and global convergence of Algorithm 1. The search direction in this suggested method always fulfils a sufficient descent condition and under some reasonable condition, we proof that the algorithm converges globally by using exact line search.

3.1. Sufficient Descent Condition

For the sufficient condition to hold

$$g_k^T d_k \leq -\|g_k\|^2 \text{ for } k \geq 0 \tag{6}$$

Theorem 1. Consider a SD method with the search direction (5) and condition (6) holds for all $k \geq 0$.

Proof. If $k = 0$, then $g_0^T d_0 = -\|g_0\|^2$. Hence, condition (6) holds true. We also need to show that for $k \geq 1$ condition (6) will also hold true.

From (5), multiply by g_k and using exact line search,

$$g_k^T d_k = -\|g_k\|^2 - \frac{\|g_k\|^2}{\|g_{k-1}\|^2} g_k^T g_{k-1} \tag{7}$$

From [9]

$$g_k^T d_k \leq -\|g_k\|^2 - \frac{\varepsilon \|g_k\|^4}{\|g_{k-1}\|^2} \text{ with } \varepsilon = [0,1]$$
$$\leq -\|g_k\|^2$$

which implies d_k is a sufficient descent direction. Hence, condition (6) holds and the proof is completed.

3.2. Global convergence properties

The following assumptions and lemma are always needed in the analysis of global convergence of SD methods.

Assumption 1.
(i) The level set $\Omega = \{x \in R^n \mid f(x) \leq f(x_0)\}$ is bounded where x_0 is the initial point.
(ii) In some neighborhood N of Ω , the objective function is continuously differentiable and its gradient is Lipchitz continuous namely there exists a constant $l > 0$ such that $\|g(x) - g(y)\| \leq l \|x - y\|$ for any $x, y \in N$.

Theorem 2. Suppose that Assumption 1 holds true. Consider x_k generated by Algorithm 2.1, α_k is obtained by using exact line search and the sufficient descent condition hold true. Then either

$$\lim_{k \to \infty} \|g_k\| = 0 \text{ or } \sum_{k=0}^{\infty} \frac{\left(g_k^T d_k\right)^2}{\|d_k\|^2} < \infty$$

Proof. The proof is done by using contradiction. Assume that Theorem 2 is not true that is $\lim_{k \to \infty} \|g_k\| \neq 0$, then there exists a positive constant δ , $\delta > 0$, such that $\|g_k\| \geq \delta$.
From (5) using Cauchy-Schwarz inequality,

$$\|d_k\| \leq \|g_k\| + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \|g_{k-1}\|$$

Dividing both sides by $\|g_k\|^2$ yields

$$\frac{\|d_k\|}{\|g_k\|^2} \leq \frac{1}{\|g_k\|} + \frac{1}{\|g_{k-1}\|}$$

Squaring both sides of the equations imply

$$\frac{\|d_k\|^2}{\|g_k\|^4} \leq \frac{1}{\|g_k\|^2} + \frac{1}{\|g_k\|\|g_{k-1}\|} + \frac{1}{\|g_{k-1}\|^2}$$

From [9], $g_k^T g_{k-1} \geq \varepsilon \|g_k\|^2$ where $\varepsilon = (0,1]$ and using Cauchy-Schwarz inequality

$$\frac{\|g_k\|^4}{\|d_k\|^2} \le \frac{\varepsilon^2 \|g_k\|^2}{\varepsilon^2 + \varepsilon + 1}$$

From the assumption that $\|g_k\| \ge \delta$ where $\delta > 0$ for $k \ge 1$, that is

$$\frac{\|g_k\|^4}{\|d_k\|^2} \ge \frac{\varepsilon^2 \delta^2}{\varepsilon^2 + \varepsilon + 1}$$

Hence, this implies

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \ge \infty$$

This contradicts Zoutendjik condition in Lemma 1.

Therefore from (9), it follows that

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

Hence, the proof is completed.

4. Results and Discussion

In this section, we test the efficiency of Algorithm 1. The standard test problems and their initial point were present in Table 1. The codes of Algorithm 1 were written in MATLAB R2017a and were performed on the computer Lenovo ideapad with an Intel Pentium Core i5 7th Gen. the processor performance was 2.5 GHz with 4.0 GB RAM and 64-bit operating system.

The total number of test functions is 27 with 3 initial point of each function and the dimension is between 2 to 1,000. In the experiments, the termination condition is if the infinity-norm of the final gradient is below 10^{-6} that is $\|g_k\| \le 10^{-6}$ and the iteration stops if the number of iterations exceeds 10,000.

Table 1: A list of standard test functions.

Function	Initial Points
Extended White and Holst	(0,0,…,0), (2,2,…,2), (5,5,…,5)
Extended Rosenbrock	(0,0,…,0), (2,2,…,2), (5,5,…,5)
Extended Freudenstein and Roth	(0.5,0.5,…,0.5), (4,4,…,4), (5,5,…,5)
Extended Beale	(0,0,…,0), (2.5,2.5,…,2.5), (5,5,…,5)
Raydan	(1,1,…,1), (20,20,…,20), (5,5,…,5)
Extended Tridiagonal 1	(2,2,…,2), (3.5,3.5,…,3.5), (7,7,…,7)
Diagonal 4	(1,1,…,1), (5,5,…,5), (10,10,…,10)
Extended Himmelblau	(1,1,…,1), (5,5,…,5), (15,15,…,15)
Fletcher	(0,0,…,0), (2,2,…,2), (7,7,…,7)
Nonscomp	(3,3,…,3), (10,10,…,10), (15,15,…,15)
Extended Denschnb	(1,1,…,1), (5,5,…,5), (15,15,…,15)
Shallow	(-2,-2,…,-2), (0,0,…,0), (5,5,…,5)
Generalized Quartic	(1,1,…,1), (4,4,…,4), (-1,-1,…,-1)
Power	(-3,-3,…,-3), (1,1,…,1), (5,5,…,5)
Quadratic 1	(-3,-3,…,-3), (1,1,…,1), (10,10,…,10)
Extended Sum Squares	(2,2,…,2), (10,10,…,10), (-15,-15,…,-5)
Extended Quadratic Penalty 1	(1,1,…,1), (10,10,…,10), (15,15,…,15)
Extended Penalty	(1,1,…,1), (5,5,…,5), (10,10,…,10)
Hager	(1,1,…,1), (5,5,…,5), (10,10,…,10)
Extended Quadratic Penalty 2	(5,5,…,5), (10,10,…,10), (15,15,…,15)
Maratos	(1.1,1.1,…,1.1), (5,5,…,5), (10,10,…,10)
Generalized Tridiagonal	(0.8,0.8,…,0.8), (15,15,…,15), (20,20,…,20)
Three Hump	(3,3), (20,20), (50,50)
Six Hump	(10,10), (15,15), (20,20)
Booth	(3,3), (20,20), (50,50)
Trecanni	(-5,-5), (20,20), (50,50)
Zettl	(-10,-10), (20,20), (50,50)

For the reason of comparison, the three methods were calculated over the same set of test problems. The result was compared between previous steepest descent methods, which are (1) and (3).

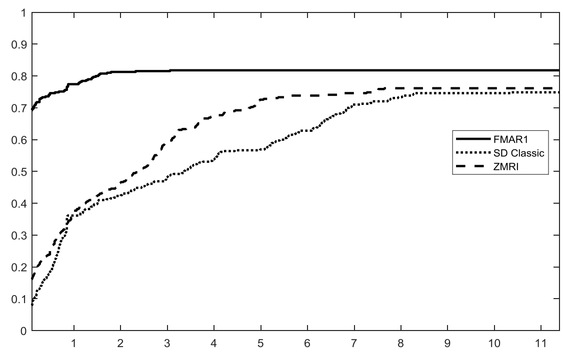

Fig. 1: Comparison between SD methods using Performance Profile based on number of iterations.

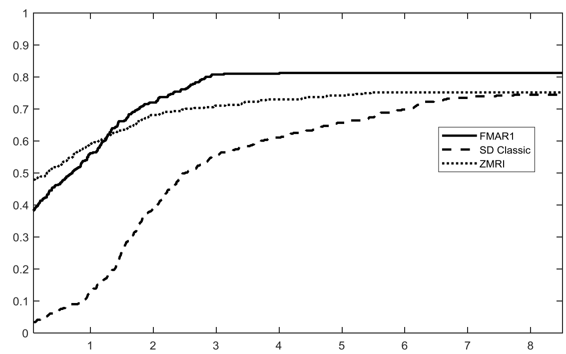

Fig. 2: Comparison between SD methods using Performance Profile based on CPU time.

In order to define the performance method and to identify the greatest method, we adopted the Performance Profile introduced by [10]. Figure 1 and 2 show that FMAR1 has the best performance based on number of iterations and CPU time compared to previous SD method, SD Classic and ZMRI.

Table 2: CPU time per iteration.

Method	Total Iterations, i	Total CPU Time, t	Successful (%)
FMAR1	92581	514.3036	81.27
ZMRI	106316	493.0954	75.18
SD Classic	329978	2638.615	74.45

Table 2 presents that the performance of the average of CPU time for a single iteration for FMAR1 is more effective than the three other methods.

Observing from the above figures, we see that FMAR1 works well, as almost all of the test functions reached an optimal point based on the given stopping criterion.

5. Conclusion

In this paper, inspired from the previous literatures on SD method, we have introduced a new search direction for solving large-scale unconstrained optimization problems. We have shown that this method is globally convergence. The numerical experiments also show that FMAR1 have the best performance compared to previous SD method and this method can possibly use to solve unconstrained optimization problems with higher dimensions.

For future research, we will develop more effective gradient methods with hybridization of conjugate gradient or Quasi-Newton methods.

Acknowledgement

This study was supported by FRGS/1/2017/STG06/UniSZA/01/1.

References

[1] A.-L. Cauchy, “M{é}thode g{é}n{é}rale pour la r{é}solution des syst{è}mes d’{é}quations simultan{é}es [Translated: (2010)],” *Compte Rendu des S’eances L’Acad’emie des Sci.*, 25(2), (1847), 536–538.

[2] 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.

[3] 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.

[4] Z. Z. Abidin, M. Mamat, M. Rivaie,“A New Steepest Descent Method with Global Convergence Properties” *AIP Conf. Proceedings*, (2016).

[5] H. Akaike, “On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method,” *Ann. Inst. Stat. Math.*, 11(1), (1959), 1–16.

[6] Y. H. Xiao, M. L. Zhang, and D. Zhou, “A simple sufficient descent method for unconstrained optimization,” *Math. Probl. Eng.*, (2010).

[7] B. T. Polyak, “The conjugate gradient method in extremal problems,” *USSR Comput. Math. Math. Phys.*, 9(4), (1969), 94–112.

[8] Z. Z. Abidin, M. Mamat, M. Rivaie, and I. Mohd, “A new steepest descent method,” *AIP Conference Proceedings*, (2014), Vol. 1602.

[9] M. J. D. Powell, “Restart procedures for the conjugate gradient method,” *Math. Program.*, 12(1), 241–254, 1977.

[10] E. D. Dolan and J. J. More, “Benchmarking Optimization Software with Performance Pro les,” 213, (2001), 201–213.