New search direction of steepest descent method for large-scaled unconstrained optimization problem

Siti Farhana Husin$^{1,*}$, Mustafa Mamat$^2$, Mohd Asrul Hery Ibrahim$^3$ and Mohd Rivaie$^4$

$^1$,$^2$Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu
$^3$Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan, Kelantan
$^4$Department of Computer Sciences and Mathematics, Universiti Teknologi Mara, Terengganu

E-mail: *ctfarhanahusin@gmail.com*

Abstract. The steepest descent (SD) method is well-known as the simplest method in optimization. In this paper, we propose a new SD search direction for solving unconstrained optimization problems. We also prove that the method is globally convergent with exact line search for general objectives functions. This proposed method is motivated by a previous work on SD method by Zubai’ah-Mustafa-Rivaie-Ismail (ZMRI). A comparison was performed with the other SD method and also with the conjugate gradient (CG) method using the conjugate coefficient, Fletcher-Reeves (FR) and Rivaie-Mustafa-Ismail-Leong (RMIL). Based on the numerical results, this new search direction show that this method has substantially outperformed the previous SD method in term of number of iterations and central processing unit (CPU) time for the given standard test problems.

1. Introduction

The steepest descent (SD) method is a Newton type method for solving nonlinear unconstrained optimization problems. It was invented by Cauchy in the nineteenth century and has been widely used since then.

An unconstrained optimization problem is written as
\[ \min_{x \in \mathbb{R}^n} f(x), \quad (1) \]

where $\mathbb{R}^n$ is an $n$-dimensional Euclidean space and $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. The iterative method is used to solve (1) is given by
\[ x_{k+1} = x_k + \alpha_k d_k, \quad (2) \]

where $\alpha_k$ denotes the stepsize and is obtained by using exact line search as follows:
\[ \alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha d_k). \quad (3) \]

Also, $d_k$ denotes the search direction, calculated by
\[ d_k = -g_k, \quad (4) \]

in which $g_k$ is defined as the gradient of $f$. The search direction must satisfy the condition $g_k^T d_k < 0$, in order to guarantee that $d_k$ will be always be a descent direction [1].
This paper is arranged as follows. In section 2, we introduce our new search direction and elaborate on its modification derived from the CG coefficient. The convergence analysis of the proposed method is given in section 3 and numerical result is shown in section 4. The paper ends with a brief closure in section 5.

2. New Search Direction

Zubai’ah et al. [2] introduced a new search direction for SD method and proved that the new method is globally convergent under exact line search. Their search direction is given as follows:

\[
d_k^{ZMRL} = \begin{cases} 
-g_k & \text{if } k = 0 \\
-g_k - \|g_k\|g_{k-1} & \text{if } k \geq 1
\end{cases}
\]

Our objective in this paper is to introduce a new search direction for SD method that has higher efficiency in solving optimization problems compared with current SD methods. This new search direction is motivated from the CG coefficient, \( \beta_k \) of Fletcher and Reeves (FR) [3] and Rivaie et al. (RMIL) [4],

\[
\beta_{FR}^k = \frac{g_k^T g_k}{\|g_{k-1}\|^2},
\]

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

Hence, we have suggested a new search direction for SD method stated as follows

\[
d_k = \begin{cases} 
-g_k & \text{if } k = 0 \\
-g_k - \theta_k g_{k-1} & \text{if } k \geq 1
\end{cases}
\]

where \( \theta_k = \frac{g_k^T g_k}{d_{k-1}^T (d_{k-1} - g_k)} \) (5)

This new search direction is called FMAR, coined on the researchers’ name, Farhana, Mustafa, Asrul and Rivaie. The new algorithm based on (5) is derived as follows:

Algorithm 2.1
Step 1: Given an initial point \( x_0 \) and set \( k = 0 \).
Step 2: If \( \|g_k\| \leq 10^{-6} \), then stop. Else, calculate the search direction, \( d_k \) using (5).
Step 3: Calculate stepsize, \( \alpha_k \) using (3).
Step 4: Update new point, \( x_{k+1} \) using (2).
Step 5: Set \( k := k + 1 \) and go to Step 2.

The efficiency of the above algorithm is tested using some standard test problems.

3. Convergence Analysis

The convergence analysis based on (5) has been discussed carefully in this section. In order to prove that an algorithm will converge, it must possess the sufficient descent and global convergence properties.

3.1 Sufficient descent condition

For the sufficient descent condition to hold,
Theorem 1. Consider an SD method with the search direction (5) and the stepsize determined by exact procedure (3). Then condition (6) holds for all $k \geq 0$.

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

Multiply (5) by $g_k$ and note that $g_0^T d_{k-1} = 0$ for exact line search, and we get

$$g_k^T d_k = -\|g_k\|^2 - \frac{\|g_k\|^2}{\|d_{k-1}\|} g_k^T g_{k-1}.$$  \hfill (7)

From [5], $g_k^T g_{k-1} \geq \varepsilon \|g_k\|^2$ where $\varepsilon = (0,1)$, which implies that

$$g_k^T d_k \leq -\left(1 + \frac{\varepsilon \|g_k\|^2}{\|d_{k-1}\|^2}\right)\|g_k\|^2$$ with $\varepsilon = (0,1]$.

We can see that $c = 1 + \frac{\varepsilon \|g_k\|^2}{\|d_{k-1}\|^2}$ and $c > 0$. Hence condition (6) holds and the proof is complete, which implies that $d_k$ is a sufficient descent direction.

3.2 Global convergence

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

Assumption 1.

(i) The level set $\Omega = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded where $x_0$ is the initial point.

(ii) In some neighborhood $N$ of $\Omega$, 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 $x, y \in N$.

These assumptions yield the following Lemma 1, which was proven by Zoutendijk [6].

Lemma 1. Suppose that Assumption 1 holds true. Let $x_k$ be generated by Algorithm 2.1 and $d_k$ satisfies (6), then the following condition, known as Zoutendijk condition, holds.

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{\|d_k\|^2} < \infty.$$  

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

$$\lim_{k \to \infty} \|g_k\| = 0 \text{ or } \sum_{k=0}^{\infty} \frac{\|g_k^T d_k\|^2}{\|d_k\|^2} < \infty.$$
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 \( \delta \), such that \( \| g_k \| \geq \delta \).

From (5), using Cauchy-Schwarz and taking into account that \( g_k^T d_{k-1} = 0 \), we get
\[ \| d_k \| \leq \| g_k \| + \frac{\| g_k \|}{\| d_{k-1} \|} \| g_{k-1} \|. \]
Dividing both sides by \( \| g_k \|^2 \) yields,
\[ \frac{\| d_k \|}{\| g_k \|^2} \leq \frac{1}{\| g_k \|^2} + \frac{\| g_{k-1} \|}{\| d_{k-1} \|}. \]
Using (6), we get
\[ \frac{\| d_k \|}{\| g_k \|^2} \leq \frac{1}{\| g_k \|^2} + \frac{1}{c^2 \| g_{k-1} \|^2} \]
Squaring both sides of the equations imply,
\[ \frac{\| d_k \|^2}{\| g_k \|^4} \leq \frac{1}{\| g_k \|^4} + \frac{1}{c^4 \| g_{k-1} \|^4} \]
From [5], \( g_k^T g_{k-1} \geq \epsilon \| g_k \|^2 \) where \( \epsilon = (0,1) \). Hence,
\[ \frac{\| d_k \|^2}{\| g_k \|^4} \leq \frac{c^4 \epsilon^2 + c^2 \epsilon + 1}{c^4 \epsilon^2 \| g_k \|^2} \]
\[ \frac{\| g_k \|^2}{\| d_k \|^2} \leq \frac{c^4 \epsilon^2 \| g_k \|^2}{c^4 \epsilon^2 + c^2 \epsilon + 1} \]
By using the assumption that \( \| g_k \| \geq \delta \) where \( \delta > 0 \), we now have
\[ \| g_k \|^2 \geq \frac{c^4 \epsilon^2 \delta^2}{c^4 \epsilon^2 + c^2 \epsilon + 1}, \]
which implies,
\[ \sum_{k=0}^{\infty} \frac{\| g_k \|^2}{\| d_k \|^2} \geq \infty. \]  \( \tag{8} \)
This contradicts Zoutendijk condition in Lemma 1.

Therefore from (8), it follows that,
\[ \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < \infty \]
Hence, the proof is completed.

4. Numerical Result
In this section, we have tested the numerical performance of Algorithm 2.1 alongside the SD and CG algorithms. This is to assess the efficiency of our new search direction in comparison to the methods it
is based off. All codes are written in Matlab 2017 and are run using a computer with Intel Core i5 processor. We select a total of 27 test problems from [7] and [8], which are all listed in Table 1 with the dimension range is [2, 1,000]. For all algorithms, the stopping condition is set to be \( \|g_k\| \leq 10^{-6} \) and the maximal number of iteration is set up to 10,000. The numerical results are measured based on the number of iterations and CPU time.

Table 1. A list of test functions.

| Function                     | Initial Points                      |
|------------------------------|-------------------------------------|
| Extended White & Holst       | (0,0,...,0), (2,2,...,2), (5,5,...,5) |
| Extended Rosenbrock          | (0,0,...,0), (2,2,...,2), (5,5,...,5) |
| Extended Freudenstein & 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) |

We apply the performance profile suggested by Dolan and More [9] in order to describe the performance of each tested method. The performance outcomes are shown in figures 1 and 2 for FMAR, ZMRI and SD Classic, while figures 3 and 4 show the performance of FMAR with FR and RMIL.
Figure 1. Performance profile based on the number of iterations.

Figure 2. Performance profile based on the CPU time.

Table 2. Numerical results for FMAR, ZMRI and SD Classic.

| Comparison    | Successful | Unsuccessful | Total iterations | Total CPU time |
|---------------|------------|--------------|------------------|----------------|
| FMAR          | 83.45%     | 16.55%       | 84693            | 497.0183       |
| ZMRI          | 75.18%     | 24.82%       | 106316           | 493.0954       |
| SD Classic    | 74.45%     | 25.55%       | 329978           | 2368.615       |

Figures 1 and 2 show that FMAR has the leading performance since it accumulates the highest percentage of problems solved at 83.45%. Therefore, this method can be considered as the superior method above the other two SD methods tested.
Figure 3. Performance profile based on the number of iterations.

Figure 4. Performance profile based on the CPU time.

Table 3. Numerical results for FMAR, FR and RMIL.

| Comparison | Successful (%) | Unsuccessful (%) | Total iterations | Total CPU time |
|------------|----------------|------------------|------------------|----------------|
| FMAR       | 83.45          | 16.55            | 84693            | 497.0183       |
| FR         | 88.32          | 11.68            | 65882            | 635.8635       |
| RMIL       | 82.24          | 17.76            | 31617            | 323.394        |

Although the performance of the CPU time of FMAR in table 3 is not the fastest, the CPU time is not the main element contributing to the numerical result [10]. The most valuable result should be established on the number of iterations.

5. Conclusion
We have presented a new search direction for steepest descent method for solving unconstrained optimization problems. The numerical results show that our proposed search direction substantially outperform the other tested methods and is very efficient in solving the test problems. Our further interest is to try alternating the search direction and to hybridize the proposed search direction with quasi Newton method.
References

[1] Nocedal J and Wright S J 1999 Numerical optimization 43 doi:10.1002/lsm.21040

[2] Abidin Z Z, Mamat M and Rivaie M 2016 A new steepest descent method with global convergence properties AIP Conf. Proceedings doi:10.1063/1.4952550

[3] Fletcher R and Reeves C M 1964 Function minimization by conjugate gradients Comput J 7 149–154 doi:10.1093/comjnl/7.2.149

[4] Rivaie M, Mamat M, June L W and Mohd I 2012 A new class of nonlinear conjugate gradient coefficients with global convergence properties Appl Math Comput 2012 doi:10.1016/j.amc.2012.05.030

[5] Powell M J D 1977 Restart procedures for the conjugate gradient method Math Program 12 241–254 doi:10.1007/BF01593790

[6] Zoutendijk G 1970 Some algorithms based on the principle of feasible directions Nonlinear Program 93–121 doi:https://doi.org/10.1016/B978-0-12-597050-1.50008-7

[7] Andrei N 2008 An unconstrained optimization test functions collection Adv Model Optim 10 147–161

[8] Moré J J, Garbow B S, Hillstrom K E 1981 Testing unconstrained optimization software ACM Trans Math Softw 7 17–41 doi:10.1145/355934.355936

[9] Dolan E D and More J J 2001 Benchmarking optimization software with performance Prof es 213 201–13

[10] Rivaie M, Mamat M and Abashar A 2015 A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches Appl Math Comput 268 1152–1163 doi:10.1016/j.amc.2015.07.019