ISSN 2278-3091

Volume 9, No.1.1, 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1791.12020.pdf

https://doi.org/10.30534/ijatcse/2020/1791.12020

Modification of Search Direction in Steepest Descent Method and Its Application in Regression Analysis

Siti Farhana Husin[1], Mustafa Mamat[2], Mohd Asrul Hery Ibrahim[3], Mohd Rivaie[4], Mohamad A Mohamed[5]

[1,2,5] Faculty of Informatics and Computing, University Sultan Zainal Abidin, Terengganu, Malaysia

[3] Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan, Kelantan, Malaysia

[4] Department of Computer Science and Mathematics, Universiti Technology MARA, Terengganu, Malaysia

ABSTRACT

Steepest descent (SD) method is the basic and simple algorithm for minimizing function of n variables. Although this method is said to lead to very slow convergence, the small change of the search direction for each line can improved the method. Therefore, in this paper, we propose new steepest descent (SD) method which focused on the modification of the direction that possesses sufficient descent conditions and global convergence properties as our first objective. The second approach is that we present the implementation of the proposed modification of SD method into the regression analysis for the real life problems.

Key words: ordinary differential equation, least square method, steepest descent method.

1. INTRODUCTION

One of the simplest and very well-known methods to find the minimum value of function for unconstrained optimization is gradient method. This method is also known as steepest descent (SD) method. Better understanding of this method can lead for a more sophisticated method in order to overcome the drawback of the standard SD due to its slow convergence rate [1].

The standard and oldest SD method was first proposed in 1847 by Cauchy [2]. Unfortunately, this method is not widely used because of the main drawback, quite slow in the rate convergence in most real-world problems. Recently however, several attempts from both theoretical and practical viewpoint have been presented in order to improve the efficiency of SD method [3]–[9]. These modifications have led to a fresh interest in this method and proved that the gradient direction itself is not a bad choice.

Therefore, in this paper, we presented a new modification in the gradient direction of the SD method which possesses both sufficient descent directions and global convergence properties as our main objective and we also proved that this modification can be implemented in the regression analysis to solve the function from the real-world problems.

The overall structure of the study takes the form of six sections, including this introductory section. The remaining part of the paper proceeds as follows: new modification of SD method, the convergence analysis, the numerical results and discussions and the application of the new proposed SD method in the real-world problems. Finally, the conclusion gives a brief summary and a little bit recommendations for future research.

2. NEW MODIFICATION OF STEEPEST DESCENT METHOD

The general minimization problem of a function is given by $min_{x \in R^n} f(x)$ which has the following iterative form

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

where α_k is the step size which in this research we calculate using exact line search procedures and d_k is the search direction. The standard method to determine the search direction is the SD method which defined as

$$d_k = -g_k .$$

Here and throughout, we use g_k to denote the gradient of f at x_k . We will also use f_k as the abbreviation of $f\left(x_k\right)$. The superscript T signifies the transpose.

In this section, we propose new modification on the direction of SD method known as MSD abbreviated of modification of SD. The formula of MSD is given by

$$d_k = -g_k - \theta_k g_{k-1} \qquad (1)$$

where $\theta_k = \dfrac{g_k^T g_{k-1}}{\left\| g_{k-1} \right\|^2}$.

The algorithm is given as follows:

Algorithm: Steepest Descent (SD) Method

Step 1: Initialization. Some initial value is chosen and set $k = 0$

Step 2: Compute the search direction, d_k , by using (1)

Step 3: Compute the step size, α_k , by using exact line search procedure
Step 4: Update new point of iteration, $x_{k+1} = x_k + \alpha_k d_k$
Step 5: Test the convergent and stopping criteria:
If $x_{k+1} < x_k$ and $|g_k| \le \varepsilon$ then stop
Otherwise go to Step 1 with $k = k+1$.

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, let sequence $\{d_k\}$ and $\{x_k\}$ be generated by (5) and (1), then

$$g_k^T d_k \le -\|g_k\|^2 \text{ for } k \ge 0 . \qquad (6)$$

Theorem 1. Consider the three-term SD method with the search direction (5) and the step size determined by exact procedure (2). Then condition (6) holds for all $k \ge 0$.

Proof. Obviously, if $k = 0$, then the conclusion is true.
Now, we need to show that for $k \ge 1$, condition (6) will also hold true.
Multiply (5) by g_k and note that $g_k^T d_{k-1} = 0$ for exact line search, and we get

$$\begin{aligned} g_k^T d_k &= -\|g_k\|^2 - \theta_k g_k^T g_{k-1} \\ &= -\|g_k\|^2 - \frac{\left\|g_k^T g_{k-1}\right\|^2}{\|g_{k-1}\|^2} \\ &\le -\|g_k\|^2 \end{aligned}$$

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 $\Omega = \left\{ x \in R^n \mid f(x) \le f(x_0) \right\}$ 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 $\|g(x) - g(y)\| \le l \|x - y\|$ for any $x, y \in N$.

These assumptions yield the following Lemma 1, which was proven by Zoutendijk [10].

Lemma 1. Suppose that Assumption 1 holds true. Let x_k be generated by Algorithm 1 and d_k satisfies (6), then the exists a positive constant h such that

$$\alpha_k \ge h \frac{\|g_k\|^2}{\|d_k\|^2}$$

and one can also have,

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

This property is known as Zoutendijk condition.

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

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

Proof. The proof is obtained by using contradiction. Assume that Theorem 2 is not true, that is, $\lim_{k \to \infty} \|g_k\| \ne 0$. Then, there exists a positive constant δ_1, such that $\|g_k\| \ge \delta_1$ for all value of k . From (5), we have

$$\begin{aligned} \|d_k\| &\le \|g_k\| + |\theta_k| \|g_{k-1}\| \\ &\le \|g_k\| + \frac{\|g_k\| \|g_{k-1}\|}{\|g_{k-1}\|^2} \left(\|g_{k-1}\|\right) \\ &\le 2\|g_k\| \\ &\le 2\delta_1 \\ &\square M \end{aligned}$$

The above inequality implies

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} \ge \sum_{k=0}^{\infty} \frac{\delta_1^4}{M^2} \qquad (7)$$

This contradicts Zoutendijk condition in Lemma 1.

Therefore from (7), it follows that,

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

Hence, the proof is completed.

4. RESULTS AND DISCUSSION

In this section, we investigate the effectiveness of our new modification of SD by comparing our method with the standard direction of SD and previous modification of direction in SD proposed by [11], [12]. We evaluate the methods using same set of 27 standard test functions with different vales of initial points. We also assumed this as solving large-scale problems as we used up to 1,000 number of variables.

The program was written in the MATLAB 2017a and run on the computer with Intel® Core™ i5 with CPU 2.5GHz and 6.4-bit Operating System. A list of all standard test functions used in this research is given in Table 1.

Table 1: A list of standard test functions

N	Functions	Initial Points
F1	Extended White & Holst	(0,0,…,0), (2,2,…,2), (5,5,…,5)
F2	Extended Rosenbrock	(0,0,…,0), (2,2,…,2), (5,5,…,5)
F3	Extended Freudenstein & Roth	(0.5,0.5,…,0.5), (4,4,…,4), (5,5,…,5)
F4	Extended Beale	(0,0,…,0), (2.5,2.5,…,2.5), (5,5,…,5)
F5	Raydan	(1,1,…,1), (20,20,…,20), (5,5,…,5)
F6	Extended Tridiagonal 1	(2,2,…,2), (3.5,3.5,…,3.5), (7,7,…,7)
F7	Diagonal 4	(1,1,…,1), (5,5,…,5), (10,10,…,10)
F8	Extended Himmelblau	(1,1,…,1), (5,5,…,5), (15,15,…,15)
F9	Fletcher	(0,0,…,0), (2,2,…,2), (7,7,…,7)
F10	Nonscomp	(3,3,…,3), (10,10,…,10), (15,15,…,15)
F11	Extended Denschnb	(1,1,…,1), (5,5,…,5), (15,15,…,15)
F12	Shallow	(-2,-2,…,-2), (0,0,…,0), (5,5,…,5)
F13	Generalized Quartic	(1,1,…,1), (4,4,…,4), (-1,-1,…,-1)
F14	Power	(-3,-3,…,-3), (1,1,…,1), (5,5,…,5)
F15	Quadratic 1	(-3,-3,…,-3), (1,1,…,1), (10,10,…,10)
F16	Extended Sum Squares	(2,2,…,2), (10,10,…,10), (-15,-15,…,-5)
F17	Extended Quadratic Penalty 1	(1,1,…,1), (10,10,…,10), (15,15,…,15)
F18	Extended Penalty	(1,1,…,1), (5,5,…,5), (10,10,…,10)
F19	Hager	(1,1,…,1), (5,5,…,5), (10,10,…,10)
F20	Extended Quadratic Penalty 2	(5,5,…,5), (10,10,…,10), (15,15,…,15)
F21	Maratos	(1.1,1.1,…,1.1), (5,5,…,5), (10,10,…,10)
F22	Three Hump	(3,3), (20,20), (50,50)
F23	Six Hump	(10,10), (15,15), (20,20)
F24	Booth	(3,3), (20,20), (50,50)
F25	Trecanni	(-5,-5), (20,20), (50,50)
F26	Zettl	(-10,-10), (20,20), (50,50)

In the experiments, the termination condition is $\|g_k\|^2 \leq 10^{-5}$. We also forced the routine stopped if the total number of iteration exceeds 10,000.

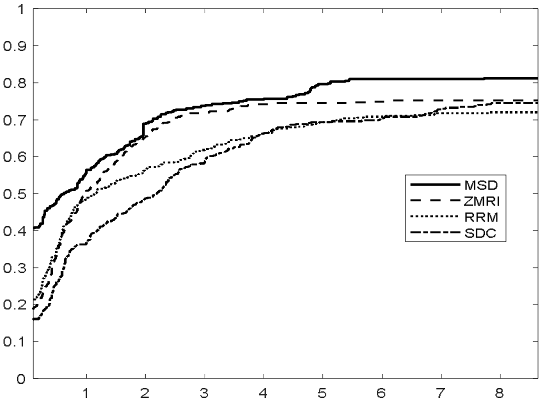

Figure 1: Performance profile for the SD methods used based on the number of iterations

We applied performance profile introduced by [13] in order to emphasize the proposed direction and to make a clear comparison which showing the effectiveness of our proposed method. Figure 1 and 2 show the comparison based on the number of iterations and number of central processing unit (CPU) times of all the methods, respectively. From Figure 1, the left side of the graph showed that MSD is the fastest method on solving all of the test problems and from the right side, MSD gives the highest percentage of the test problems that are successfully solved compared to other methods. Therefore, in numerical experiments we can say that MSD outperforms the other methods.

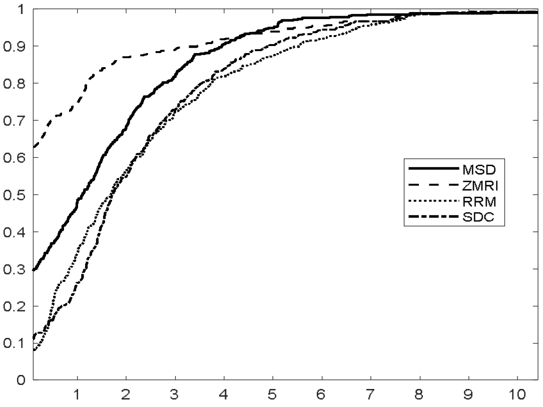

Figure 2: Performance profile for the SD methods used based on the central processing units (CPU) time

Table 2: CPU time per iteration and successful percentage in solving all the functions

Methods	Total number of iterations	Total number of CPU times	CPU time per iteration	Successful functions solved (%)
SDC	329978	11312.45	0.0343	74.45
ZMRI	106316	13811.25	0.1299	75.18
RRM	155271	14053.22	0.0905	72.01
MSD	224744	10282.76	0.0458	81.27

Table 2 shows the results for an average CPU time per single iteration for each method used using an exact line search. Although, MSD took the second place after the standard SD, the total iterations and total CPU times overcome the standard SD. In fact, MSD gives the highest percentage with 81.27% in solving all of the standard test functions followed by other methods.

5. APPLICATION IN REGRESSION ANALYSIS

Recently researchers have examined the application of the optimization method for solving real-data problems that have been transformed into minimization of a function [14]–[16]. Thus in this paper, we test the capability of our proposed method in solving the real-data problems. Data for this study were collected from world data bank by focusing on the data of government expenditure on education in Malaysia from 2006 until 2016.

Table 3: Government expenditure on education based on the total government expenditure for 2006 until 2016.

Number of data, *x*	Years	Government expenditure on education (% of government expenditure), *y*
1	2006	16.74556923
2	2007	16.12466049
3	2008	14.03864002
4	2009	18.46463013
5	2010	18.40623093
6	2011	20.97702026
7	2012	19.92865944
8	2013	19.45430946
9	2014	19.80056
10	2015	19.84806061
11	2016	20.63970947

From the data given in Table 3, we can observe that there is a linear relationship between the year and the expenditure on education made by government with the regression equation define as $y = a_1 + a_2 x$, where a_1 and a_2 are the regression parameters. In order to solve the regression problems, one have to find the parameter values and one of the method to find the parameters is least square method (LSM) that minimized the problem and can be transformed into unconstrained optimization problems define as

$$\min_{a\in\mathbb{R}} f(a) = \sum_{i=1}^{n} \left[y_i - (a_1 + a_2 x_i) \right]^2$$

The steps of the LSM method to solve the minimization problem is given below:

Algorithm: Least Square Method (LSM)

Step 1: Assume the solution is in the cubic form and find the derivative of y' and y''

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

Step 2: Substituting y, y' and y'' into the general ODE

Step 3: Finding the error and compute $\left(E(x)\right)^2$

$$E(x) = \left[\sum_{i=0}^{n} a_i \left[i(i-1)x^{i-2} + ix^{i-1}P(x) + x^i Q(x) \right] - R(x) \right]$$

Step 4: Find partial derivatives with respect to a_1 and a_2 then compute the definite integral from a to b

$$\frac{\partial F}{\partial a_i} = \int_b^a E(x)\,dx = 0$$

Step 5: Solve the system of linear equation $Ax = b$, $\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ using direct inverse.

However, step 5 in the LSM algorithm will lead to solve an inverse matrices and there will be a problem if the matrices involving singular or nearly singular matrices where the inverse does not exist. Hence, to overcome this problem, the SD method is applied to solve the system of linear equation as shown in Table 4.

Table 4: Approximation functions from the numerical experiments used.

Method	Approximation function
Linear least square	$y = 0.5013804455x + 15.57608551$
Standard SD	$y = 0.5013803914x + 15.57608591$
ZMRI	$y = 0.5013803184x + 15.57608646$
RRM	$y = 0.5013804979x + 15.57608504$
MSD	$y = 0.5013804113x + 15.57608576$

Table 5: Approximate values of government expenditure on education based on the numerical experiments.

Years	**Government expenditure on education**	**Approximate Values**				
		LSM	SDC	ZMRI	RRM	MSD
2006	16.74556923	16.07746596	16.07746630	16.07746678	16.07746554	16.07746617
2007	16.12466049	16.57884640	16.57884669	16.57884710	16.57884604	16.57884658
2008	14.03864002	17.08022685	17.08022708	17.08022742	17.08022653	17.08022699
2009	18.46463013	17.58160729	17.58160748	17.58160773	17.58160703	17.58160740
2010	18.40623093	18.08298774	18.08298787	18.08298805	18.08298753	18.08298782
2011	20.97702026	18.58436818	18.58436826	18.58436837	18.58436803	18.58436823
2012	19.92865944	19.08574863	19.08574865	19.08574869	19.08574852	19.08574864
2013	19.45430946	19.58712907	19.58712904	19.58712901	19.58712902	19.58712905
2014	19.80056	20.08850952	20.08850943	20.08850933	20.08850952	20.08850946
2015	19.84806061	20.58988996	20.58988982	20.58988964	20.58989002	20.58988987
2016	20.63970947	21.09127041	21.09127022	21.09126996	21.09127052	21.09127028

Based on the approximation functions evaluated as shown in Table 5, the percentage of relative error is calculated using the formula stated below and tabulated in Table 6.

$$\text{Percentage error } = \frac{|\text{Exact value} - \text{Approximate value}|}{|\text{Exact value}|} \times 100\%$$

Table 6: Error calculations for each methods from the numerical experiments

Years	**Error calculations (%)**				
	LSM	**SDC**	**ZMRI**	**RRM**	**MSD**
2006	3.989731617	3.989729587	3.98972672	3.989734125	3.989730363
2007	2.816716112	2.81671791	2.816720453	2.816713879	2.816717228
2008	21.66582251	21.66582415	21.66582657	21.66582023	21.66582351
2009	4.782239524	4.782238495	4.782237141	4.782240932	4.782238928
2010	1.756161765	1.756161059	1.756160081	1.756162906	1.756161331
2011	11.40606268	11.4060623	11.40606178	11.4060634	11.40606245
2012	4.229641299	4.229641199	4.229640998	4.229641851	4.229641249
2013	0.682725903	0.682725749	0.682725595	0.682725646	0.682725801
2014	1.454249375	1.45424892	1.454248415	1.454249375	1.454249072
2015	3.737540733	3.737540028	3.737539121	3.737541035	3.73754028
2016	2.187826048	2.187825127	2.187823868	2.187826581	2.187825418
Sum of relatives error	**58.70871757**	**58.70871453**	**58.70871074**	**58.70871996**	**58.70871562**

The LSM is the form of mathematical regression analysis that finds the line of best fit for a data set, providing a visual demonstration of the relationship between the data points. However, in order to solve the second order linear ODE using LSM, there will be a problem when one's dealing with the solution that involves inverse matrix.

Taking this problem into a consideration it is possible to implement the optimization method into the LSM especially the SD method as focused in this paper. As can be seen in the Table 6, error calculations tabulated and all of the SD methods tested are compatible.

5. CONCLUSION AND RECOMMENDATIONS

In this study, the aim was to introduce the new modification of the direction of SD method that possesses the sufficient descent conditions and global convergence properties. As a result of experiment, among all the competitors MSD method has demonstrated the best efficiency. And also we have proved that the proposed method is compatible to be implemented into the real-world problems that have been transformed to unconstrained optimization problems.

ACKNOWLEDGEMENT

The authors thank the referees for their contributions and also grateful to the Malaysian government and Universiti Sultan Zainal Abidin for funding this research under the Fundamental Research Grant Scheme (FRGS/1/2017/STG06/Unisza/01/1).

REFERENCES

1. J. C. Meza. **Stepeest Descent**, Wiley Interdisciplinary Reviews: Computational Statatistis, vol. 2, no. 6, pp. 719–722, 2010. https://doi.org/10.1002/wics.117
2. A.-L. Cauchy. **Méthode générale pour la résolution des systèmes d'équations simultanées** [Translated: (2010)], *Compte Rendu des S'eances L'Acad'emie des Sci*., vol. 25, no. 2, pp. 536–538, 1847.
3. Y. Dai, J. Yuan, and Y. X. Yuan. **Modified two-point stepsize gradient methods for unconstrained optimization**, *Comput. Optim. Appl*., vol 22, no. 1, pp. 103–109, 2002. https://doi.org/10.1023/A:1014838419611
4. Z. A. Zubai'ah et al. **Hybrid BFGS-ZMRI Methods with Global Convergence Properties**, *AIP Conf. Proc*., vol 1974, no. 1, 2018.
5. R. De Asmundis, D. di Serafino, W. W. Hager, G. Toraldo, and H. Zhang. **An efficient gradient method using the Yuan steplength**, *Comput. Optim. Appl*., vol. 59, no. 3, pp. 541-563, 2014.
6. J. Barzilai and J. M. Borwein. **Two-point step size gradient methods**, IMA Journal of Numerical Analysis, vol. 8, no. 1, pp. 141–148, 1998.
7. G. C. Bento, O. P. Ferreira, and P. R. Oliveira. **Unconstrained Steepest Descent Method for Multicriteria Optimization on Riemannian Manifolds**, J. Optim. Theory Appl., vol. 154, no. 1, pp. 88–107, 2012. https://doi.org/10.1007/s10957-011-9984-2
8. Z. Z. Abidin, M. Mamat, M. Rivaie, and I. Mohd. **A new steepest descent method**, in AIP Conference Proceedings, vol. 1602, pp. 273-278, 2014.
9. C.-S. Liu, J.-R. Chang, and Y.-W. Chen. A modified algorithm of steepest descent method for solving unconstrained nonlinear optimization problems, J. Mar. Sci. Technol., vol. 23, no. 1, pp. 88–97, 2015.
10. G. Zoutendijk. **Some Algorithms Based on the Principle of Feasible Directions**, *in Nonlinear Programming*, pp. 93–121, 1970.
11. Z. Zainal Abidin, M. Mamat, and M. Rivaie. **A New Steepest Descent Method with Global Convergence Properties**, *AIP Conf. Proc*., vol. 1739, pp. 020070-020078, 2016.
12. R. Johari, M. Rivaie, M. Mamat, and Z. Salleh. **A New Scaled Steepest Descent Method for Unconstrained Optimization with Global Convergence Properties**, *J. Eng. Appl. Sci.*, vol. 13, no. 6 SI, pp. 5442-5445, 2018.
13. E. D. Dolan and J. J. More. **Benchmarking Optimization Software with Performance Pro les**, *Math. Program*, vol. 213, no. 1, pp. 201–213, 2001. https://doi.org/10.1007/s101070100263
14. X. Wang. **Method of Steepest Descent and its Applications**, *IEEE Microw. Wirel. Components Lett*., 37996, pp. 1–3, 2018.
15. N. Shapiee, M. Rivaie, M. Mamat, and P. L. Ghazali. **A new family of conjugate gradient coefficient with application**, *International Journal of Engineering and Technology(UAE*), vol. 7, no. 3.28, pp. 36–43, 2018. https://doi.org/10.14419/ijet.v7i3.28.20962
16. W. Khadijah, M. Rivaie, M. Mamat, N. Hajar, N. Aini, and Z. Abidin. **An application of conjugate gradient method under strong Wolfe line search for solving unconstrained optimization**, *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 3.28, pp. 12–16, 2018. https://doi.org/10.14419/ijet.v7i3.28.20956