Menemui Matematik (Discovering Mathematics)
Vol. 43, No. 2: 101-110 (2021)

Comparison of the AIM Conjugate Gradient Method Under Exact and Inexact Line Search for Solving Unconstrained Optimization Problems

F N Za'aba[1] and Marjugi S.M[2]
[1,2]*Department of Mathematics & Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang,*
[1]fatinnadhirahzaaba@gmail.com, [2]smahani@upm.edu.my

ABSTRACT

The nonlinear conjugate gradient (CG) method is essential in solving large-scale unconstrained optimization problems due to its simplicity and low memory requirement. Numerous studies and improvements have been made recently to improve this strategy. Hence, this study will create a modified CG method with inexact line search, Strong Wolfe-Powell conditions. The global convergence and sufficient descent properties are established by using an inexact line search. The numerical result demonstrates that the modified method with inexact line search is superior and more efficient when compared to other CG methods.

Keywords: Conjugate gradient, unconstrained optimization, inexact line search, Strong Wolfe-Powell

INTRODUCTION

Unconstrained optimization

Unconstrained optimization problems consider optimizing an objective function that relies on actual variables without any value constraints. According to Fasano (2010), the general problem of unconstrained large-scale optimization can be defined in (1), where the parameter n is huge. He also describes the huge n value claim for appropriate search directions and step length along with the method. The choice of the search directions is responsible for the efficiency of the methods, the convergence rate, while a suitable step length choice guarantees the output. Besides, iterative methods (2) usually reduce the computational burden compared to direct methods when n is large.

One of the methods for solving large-scale unconstrained optimization problems is Newton's method. However, according to Cajori (1911), Newton explained his approximation approach to the real root of the numerical equation in 1669. Besides, another method used is the Quasi-Newton method. Meanwhile, Wikipedia contributor (2020) William C. Davidon, a physicist employed at the Argonne National Laboratory, introduced the first Quasi-Newton algorithm. In 1959, he created the first Quasi-Newton algorithm, the updating method for the Davidon-Fletcher-Powell, which was later popularized by Fletcher and Powell in 1963, but is seldom used today. Unlike Newton's method, in which the technique needs to minimize the gradient and the Hessian matrix of the function's second derivatives, the Quasi-Newton approaches generalise the secant approach to locate the root of multidimensional problems with the first derivative.

Generally, the large-scale unconstrained optimization problem formula can be defined as below:

$$\min_{x \epsilon R^n} f(x) \tag{1}$$

where $f: R^n \rightarrow R$ is a smooth, nonlinear function and the gradient indicated by $g(x)$.

Conjugate Gradient method

The CG method is widely used to solve large-scale unconstrained optimizations because of its convergence properties and low computational cost. Powell (1977) states that the CG technique is beneficial for reducing functions of very many variables as it does not involve any matrices to be stored. However, the algorithm's convergence duration is linear until the iterative process is periodically "restarted". This approach will use only the previous vector P_{k-1} to calculate a new vector P_k . This exceptional property means that the system needs minimal storage and computation (Nocedal and Wright (2006). Continuos research from the CG method includes the methods of Fletcher-Reeves (FR), Polak-Ribi`ere-Polyak (PRP), the method of Hestenes-Stiefel (HS), the method of Liu-Storey (LS), the method of Dai-Yuan (DY) and the method of Conjugate Descent (CD). They are some of the popular parameters, β_k of the CG method.

The CG method is an iterative method given as follows,

$$x_{k+1} = x_k + \alpha_k d_k , \ \ k = 0, 1, 2, 3 \dots \tag{2}$$

where x_k is the current iterate point, a_k is the step length and d_k is the search direction. The search direction is defined by,

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

where $\beta_k \epsilon R$ is a scalar known as a scalar. Various conjugate gradient methods have been suggested, which differ mainly in the choices of the parameter, β_k. There are some popular classical formulas for parameter β_k which given as follows,

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

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

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

$$\beta_k^{LS} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T g_{k-1}} \tag{7}$$

$$\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}} \tag{8}$$

$$\beta_k^{CD} = \frac{g_k^T g_k}{d_{k-1}^T g_{k-1}} \tag{9}$$

where $g_k = g(x_k)$. Accordingly, the above method is named the Fletcher-Reeves (FR) approach by Fletcher (1987), the Polak-Ribi`ere-Polyak (PRP) approach by Polyak (1969), the Hestenes-Stiefel (HS) approach by Hestenes et al. (1952), the Liu-Storey (LS) approach by Liu and Storey (1991), Dai- Yuan (DY) approach by Dai and Yuan (1999) and the Conjugate Descent (CD) approach by Fletcher and Reeves (1964).

Specifically, the modification of the new parameter, β_k namely, the AIM method formula combines the AMRI method developed by Abashar et al. (2014) and the HRM method developed by Hamoda et al. (2016). The properties of global convergence are proposed with an exact line search, and this reveals that the AIM method holds sufficient conditions for descent as the parameter AIM has always been proved to be less than zero. However, the computation stopped in certain situations due to the line search inability to locate a positive step length. Thus, it is considered a failure. Moreover, the AIM method provides the best results, as it can perfectly solve all the test problems. That has been proven when the AIM method is very competitive with the AMRI method and HRM method. Combinations demonstrate that it outperforms FR methods in terms of iterations and CPU time.

Inexact line search

Line search, referred to as one-dimensional search, is a functional system for univariable functions. There are two types of line search, which are an exact line search (ELS) and an inexact line search (ILS). For our research, we will employ an ILS as we need a step length, a_k to ensure a necessary reduction in the function values that induces global convergence properties of the approach. Our focus is on Strong Wolfe–Powell conditions, designed to approximate the suitable step length. The Strong Wolfe-Powell conditions are introduced as below,

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k \tag{10}$$

$$|g(x_k + a_k d_k)^T d_k| \leq \sigma |g_k^T d_k| \tag{11}$$

where $0 < \delta < \sigma < 1$ and d_k is a search direction. Moreover, the sufficient descent property, namely,

$$g_k^T dk \leq -c||g_k||2 \tag{12}$$

where $c > 0$, it is important to ensure that the nonlinear conjugate gradient approach converges globally with the inexact line search techniques.

Most applications with thousands or millions of variables give rise to unconstrained optimization problems. However, issues of this size can perform productively if the capacity of the optimization algorithms is also maintained at an average level. To achieve this goal, various large-scale optimization methods have been developed, each of which has been particularly successful.

Previously, the Steepest Descent method also referred to as the Gradient Descent method was used to find the nearest local minimum of a function. The convergence properties of the Steepest Descent approach with inexact line searches have been analyzed for the selection of stepsize, a_k under several strategies. However, this method is not widely used in practice due to its slow convergence rate. Knowing the convergence properties of this method contributes to a better understanding of many of the more advanced optimization methods.While the Newton method of finding the roots of the derivative is applied to the derivative f' of a twice-differentiable function f known as the Hessian matrix, the drawback of using the Newton method is that it does not always converge to a minimizer, thus if the starting approximation is too far from the solution, it may diverge.

Since the Newton method is excessively time-intensive, the Quasi-Newton method will be replaced by this approach. It does not need to calculate the inverse Hessian iteratively. Thus, the various forms of the Quasi-Newton method are strongly dependent on what approximation is used. However, the lack of accuracy in the Hessian calculation leads to a slower step by step convergence. Consequently, we address the issue in this work using the CG method. However, the CG method process deflects the direction of the Steepest Descent approach by adding a positive multiple of the direction used in the final step. Therefore, this method is considered one of the best methods currently accessible for general purposes in which it is incredibly successful in addressing the general objective function.

THE NEW MODIFIED CONJUGATE GRADIENT

An ILS is inexpensive and inherits the same advantage as an exact line search. In addition, the ILS will approximate the step length by reducing the function value and direction derivative. The Strong Wolfe-Powell (SWP) line-search is the most popular inexact line search, which is designed to approximate the suitable step length using equations (10) and (11).

The modification of the new parameter, β_k namely, the AIM method,

$$\beta_k^{AIM} = \frac{\|g_k\|^2 - \frac{||g_k||}{||g_{k-1}||}|g_k^T g_{k-1}|}{\mu\|g_{k-1}\|^2 + (1-\mu)\|d_{k-1}\|^2}. \tag{13}$$

Based on the modification on AMRI method proposed by Abashar et al. (2014),

$$\beta_k^{AMRI} = \frac{\|g_k\|^2 - \frac{||g_k||}{||g_{k-1}||}|g_k^T g_{k-1}|}{\|d_{k-1}\|^2} \tag{14}$$

and HRM method proposed by Hamoda et al. (2016),

$$\beta_k^{HRM} = \frac{g_k^T\left(g_k - \frac{\|g_k\|}{\|g_{k-1}\|}g_{k-1}\right)}{\mu\|g_{k-1}\|^2 + (1-\mu)\|d_{k-1}\|^2}. \tag{15}$$

Meanwhile, the parameter value μ can be set as 0 < μ < 1. For our beta β_k^{AIM} we will take arbitrary value $\mu = 0.4$. With that, we show the algorithm. Then, the numerical has been shown to solve unconstrained, large-scale optimization in terms of number of iterations and CPU time.

- •Step 1: Choose an initial point, $x_0 \in R^n$ and set $k = 0$.
- •Step 2: Compute the parameter β_k based on a predetermined formula.
- •Step 3: Compute d_k based on (3). If $g_k = 0$, then stop.
- •Step 4: Compute a_k by an inexact line search.
- •Step 5: Update a new point based on an iterative formula (2).
- •Step 6: Convergence test and stopping criteria. If $f(x_{k+1}) < f(x_k)$ and then stop. Otherwise set $k = k + 1$ go to Step 1.

Properties and Convergence Analysis

To solve large-scale unconstrained optimization problems, we construct a sufficient descent condition. The descent property is important for the iterative method to be global convergent, especially for the conjugate gradient method. Sufficient decent condition can be expressed by the formula:

$$g_k^T d_k \leq -c||g_k||^2 \quad \text{for } k \geq 0 \tag{16}$$

It shows from this theorem that the AIM method with exact line search has the sufficient descent condition.

Theorem

Suppose x_k and d_k are generated by the method (2) and (3) and (14) as well as by the step size $a_k > 0$ determined by the exact line search, then the condition (16) holds for all k > 0

Proof

By using induction, we prove the theorem. The condition (3.4) is true if $k = 0$, then $g_0^T d_0 = -C\|g_0\|^2$. For condition (16) holds, we need to show that k ⩾ 1. Multiply (3) by g_{k+1}^T then

$$g_{k+1}^T d_{k+1} = g_{k+1}^T(-g_{k+1} + \beta_{k+1} d_k) = -\|g_{k+1}\|^2 + \beta_{k+1} g_{k+1}^T d_k. \tag{17}$$

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

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2. \tag{18}$$

This condition holds for $k + 1$. Hence, the proof is completed.

Global Convergence Properties

To study the new algorithm's properties and convergence, in the following, we assume that $g_k = 0$. For all k, for otherwise, a stationary point has been found. Therefore, the parameter of AIM has to be proven to be not less than zero.

$$\beta_k^{AIM} = \frac{\|g_{k+1}\|^2 - \frac{||g_{k+1}||}{||g_k||}|g_{k+1}^T g_k|}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2}$$

$$\beta_k^{AIM} = \frac{\|g_{k+1}\|^2 - \frac{||g_{k+1}||}{||g_k||}|g_{k+1}^T g_k|}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2} \geq \frac{\|g_k\|^2 - \frac{||g_k||}{||g_{k-1}||}\|g_k^T\|\|g_{k-1}\|}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2} = 0 \tag{19}$$

By exact line search, we know that $g_{k+1}^T d_k = 0$ so $g_{k+1}^T d_k = -\|g_{k+1}\|^2$. Therefore, we obtain $\beta_k^{AIM} \geq 0$,

$$\beta_{k+1}^{AIM} = \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}|g_{k+1}^T g_k|}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2} \leq \frac{\|g_{k+1}\|^2 - \frac{\|g_{k+1}\|}{\|g_k\|}\|g_{k+1}^T\|\|g_k\|}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2} \leq \frac{\|g_{k+1}\|^2}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2}.$$

Hence, we obtain

$$0 \leq \beta_{k+1}^{AIM} \leq \frac{\|g_{k+1}\|^2}{\mu\|g_k\|^2 + (1-\mu)\|d_k\|^2}.$$

By using exact line search,

$$f(x_k + a_k d_k) = \min_{a \geq 0} f(x_k + a d_k) \tag{20}$$

and (19), we assume μ = 0.4. Then, we simplify β_{k+1}^{AIM},

$$0 \leq \beta_{k+1}^{AIM} \leq \frac{\|g_{k+1}\|^2}{0.4\|g_k\|^2 + (1-0.4)\|d_k\|^2}, \tag{21}$$

$$|\beta_{k+1}^{AIM} g_{k+1}^T d_k| \leq \frac{\|g_{k+1}\|^2}{0.4\|g_k\|^2 + 0.6\|d_k\|^2}\sigma\|g_k^T d_k\|.$$

By (3), we have $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$,

$$\frac{g_{k+1} d_{k+1}}{\|g_{k+1}\|^2} = -1 + \beta_{k+1}\frac{g_{k+1} d_k}{\|g_{k+1}\|^2}. \tag{22}$$

Using induction part, we prove the descent property of d_k. Since $g_0^T d_0 = -\|g_0\|^2 < 0$, if $g_0 \neq 0$.

Suppose that $d_i, i = 1,2,\dots,k,$ are all decent directions, which $g_i^T d_i < 0$. By (21), we get,

$$\frac{\|g_{k+1}\|^2}{0.4\|g_k\|^2 + 0.6\|d_k\|^2}\sigma\|g_k^T d_k\| \leq \beta_{k+1}^{AIM} g_{k+1}^T d_k \leq -\frac{\|g_{k+1}\|^2}{0.4\|g_k\|^2 + 0.6\|d_k\|^2}\sigma\|g_k^T d_k\|. \tag{23}$$

We deduce (22) and (23) to obtain,

$$-1 + \frac{\sigma\|g_k^T d_k\|}{0.4\|g_k\|^2 + 0.6\|d_k\|^2} \leq \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|^2} \leq -1 - \frac{\sigma\|g_k^T d_k\|}{0.4\|g_k\|^2 + 0.6\|d_k\|^2}. \tag{24}$$

From the equation, it implies that the sufficient condition holds. The proof is complete**.**

RESULTS AND DISCUSSION

We selected 15 different functions with different variables and initial points in the present numerical experiment, as shown in Table 4.1. We ran this numerical experiment on MATLAB R2018a, 64-bit (win64), to test the efficiency of our method by using the parameter, β_k^{AIM}. The same PC and CPU processor were used to run all the numerical experiments, an Intel® Core™ i5-8250 CPU @ 1.60GHz 1.80GHz with RAM of 4GB. In order to verify our method's reliability, we performed a comparison with the well-known classical CG method, which includes the Fletcher Reeves (FR) method. We considered ε = 10^{-6} and the gradient values $\|g_k\| \leq \varepsilon$ as the stopping criteria based on work Hillstrom (1977) suggested it.

Hilstrom (1977) recommended four points for each test function problem based on Table 4.1. By using a performance profile introduced by Dolan and Moré (2002), the performance results are shown in Figures 4.1 and 4.2, respectively. From the figure, we denoted AIM with inexact line search as AIM (ILS), AIM with exact line search as AIM (ELS), FR with inexact line search as FR (ILS) and FR with exact line search as FR (ELS). Referring to Figure 4.1, which shows the number of iterations, it is shown that the AIM (ILS) method is very competitive with the AIM (ELS) method at the beginning of the performance profile, but then, at a certain point, the AIM (ILS) method becomes better than the AIM (ELS) method. Compared with the FR (ILS) method and the FR (ELS) method, our proposed method, AIM (ILS), obviously shows a better performance profile because the AIM (ILS) method has a lower number of iterations.

In Figure 4.2, which shows the CPU time, we can easily recognize that the AIM (ILS) method shows the best performance profile compared to other methods because the AIM (ILS) method has the shortest CPU time. Therefore, we can say that our proposed method, AIM (ILS), has shown the best performance compared to other methods.

We will discuss the performance of our proposed method in this section, which is the AIM (ILS) method with comparison to the AIM (ELS) method and the existing CG method, which are the FR (ILS) and FR (ELS) methods. As indicated by the performance profile, the comparison is demonstrated based on the number of iterations and CPU times in seconds by the values of m and n which depend on the functions' initial point and variables.

The result shown in Figure 4.1 is based on the performance profile in terms of the number of iterations. It is clear that our CG method, the AIM (ILS) method, achieves good performance compared to the AIM (ELS) method and the existing method, which are the FR (ILS) and FR (ELS) methods. We show that our proposed method is better when compared with the FR (ILS) method, which solves 83% of the test problems, and the FR (ELS) method, which solves 85% of the test problems. Although the AIM (ELS) method is competitive with our proposed method, the AIM (ILS) method, at some points, the AIM (ELS) method shows a bit slower performance after that. On the other hand, the AIM (ELS) method can only solve 85% of test problems. Hence, we rate that our CG method, the AIM (ILS) method is the best since it can solve all the test problem functions.

Table 4.1: A List of Problem Function

NO.	FUNCTION	VARIABLE	INITIAL POINT
1	Ex-tridiagonal1 function	2	(1,1),(4,4),(8,8),(10,10)
2	Diagonal 4 function	2	(1,1),(4,4),(8,8),(10,10)
3	Ex-himmelblau function	2	(1,1),(4,4),(8,8),(10,10)
4	Extended Denschnb function	2	(1,1),(4,4),(8,8),(10,10)
5	Extended quadratic penalty	2	(1,1),(4,4),(8,8),(10,10)
6	Ex Penalty	2	(1,1),(4,4),(8,8),(10,10)

7	Hager function	2	(1,1),(4,4),(8,8),(10,10)
8	Booth function	2	(1,1),(4,4),(8,8),(10,10)
9	Shalow function	2	(1,1),(4,4),(8,8),(10,10)
10	Quadrtic QF2	2	(1,1),(4,4),(8,8),(10,10)
11	Generalized tridiagonal 1	2	(1,1),(4,4),(8,8),(10,10)
12	Quadratic QF1	2	(1,1),(4,4),(8,8),(10,10)
13	Matyas function	2	(1,1),(4,4),(8,8),(10,10)
14	Sum Squares function	2	(1,1),(4,4),(8,8),(10,10)
15	Perturbed Quadratic function	2	(1,1),(4,4),(8,8),(10,10)

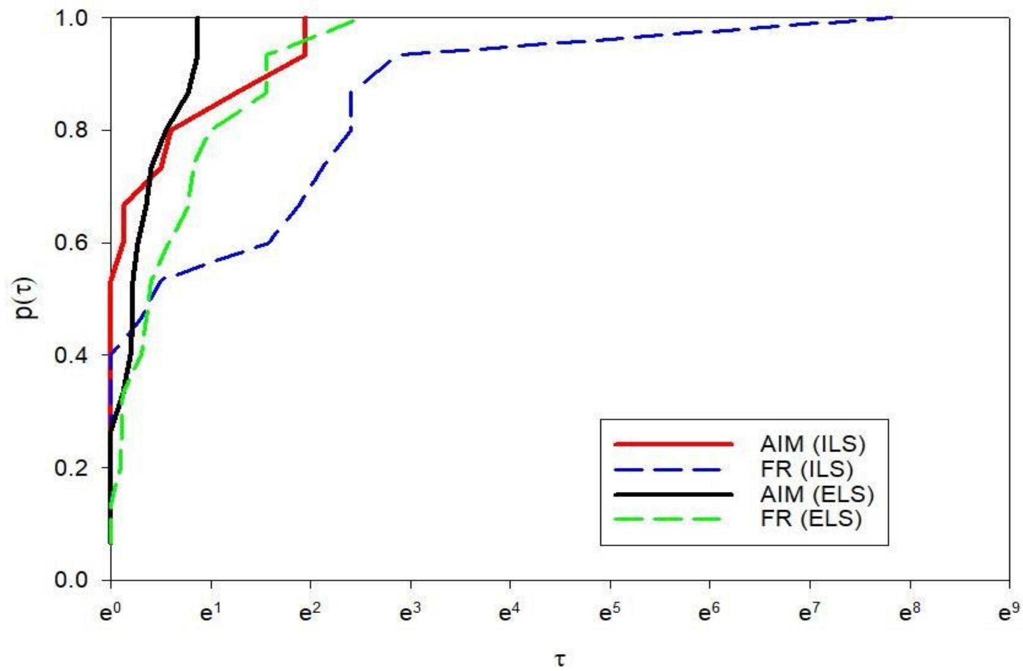

Figure 4.1: Performance profile relative to the iteration time of inexact and exact line search.

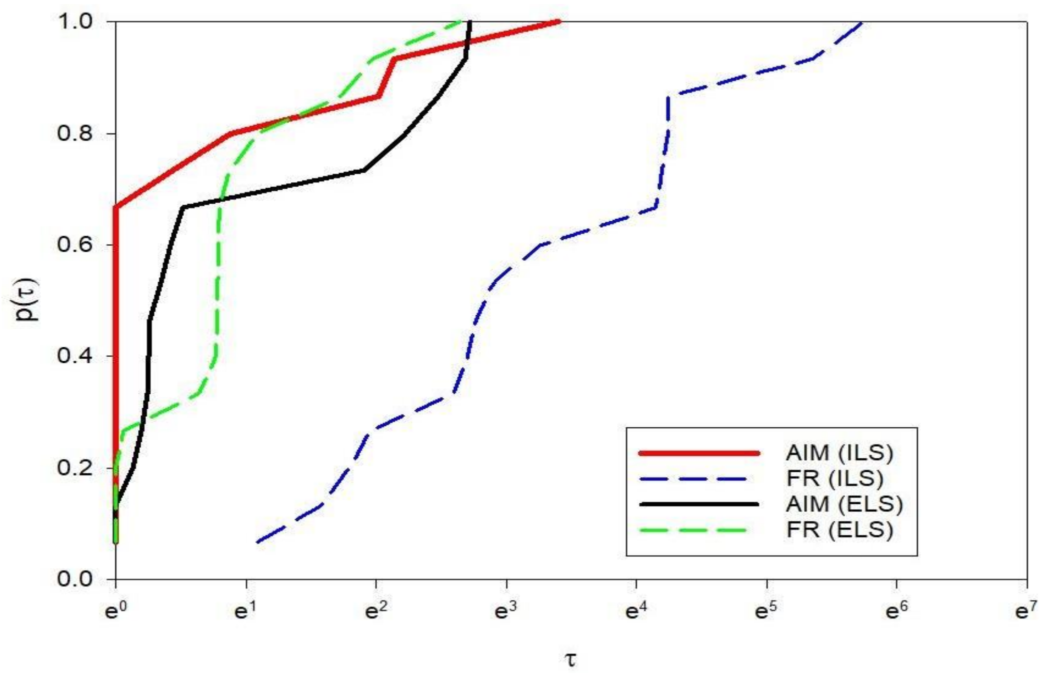

Figure 4.2: Performance profile relative to the CPU time of inexact and exact line search**.**

Meanwhile, in terms of CPU time, the duration includes the time required to generate the search direction to perform inexact and exact line search and convergence tests. Usually, the performance profiles plotted on the top right show the best performance compared to other methods.

The result shown in Figure 4.2 based on the performance profile in terms of CPU times has shown that our proposed method, the AIM (ILS) method, achieves good performance by giving the shortest time taken compared to the AIM (ELS), FR (ILS) and FR (ELS) methods. It is seen that we can conclude from Figure 4.2 that 100% of the test problems can be solved by our AIM (ILS) method compared to the AIM (ELS) method, which can only solve 85% of the test problems. The FR (ILS) method can only solve 83% of test problems, and the FR (ELS) method can solve 85% of test problems. Overall, our proposed AIM (ILS) method successfully solves all the test problems, and it is competitive with the AIM (ELS) method.

From this, we can show that the performance profile of our AIM (ILS) method is improved depending on the number of iterations as well as the CPU times compared to the AIM (ELS), FR (ILS) and FR (ELS) methods.

CONCLUSION

A great deal of study has been done on the CG method, which has resulted in the development of numerous CG methods. As a result, we can conclude that the AIM (ILS) technique is more efficient than the AIM (ELS) method in terms of the number of iterations required and the amount of CPU time required. It is also more efficient when compared to the existing CG approach, the Fletcher Reeves (FR) method, in terms of the number of iterations and the amount of CPU time required to complete the computation. The AIM (ILS) technique, as the last step, is demonstrated to solve the large-scale unconstrained optimization problem for a stable dynamic system with improved convergence qualities.

REFERENCES

Abashar, A., Mamat, M., Rivaie, M., and Mohd, I. (2017). Global convergence properties of a new class of conjugate gradient method for unconstrained optimization.

Abashar, A., Mamat, M., Rivaie, M., Mohd, I., and Omer, O. (2014). The proof of sufficient descent condition for a new type of conjugate gradient methods. In AIP Conference Proceedings, volume 1602, pages 296–303. American Institute of Physics.

Al-Baali, M. (1985). Descent property and global convergence of the fletcher—reeves method with inexact line search. IMA Journal of Numerical Analysis, 5(1):121– 124.

Azeani (2020). A modified parameter β_k in conjugate gradient method using exact line search to solve the unconstrained optimization problems. Universiti Putra Malaysia, pages 2–28.

Cajori, F. (1911). Historical note on the newton-raphson method of approximation. The American Mathematical Monthly, 18(2):29–32.

Dai, Y.-H. and Yuan, Y. (1999). A nonlinear conjugate gradient method with a strong global convergence property. SIAM Journal on optimization, 10(1):177–182.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. Mathematical programming, 91(2):201–213.

Fasano, G. (2010). Methods for large-scale unconstrained optimization. Wiley Encyclopedia of Operations Research and Management Science.

Fletcher, R. (1987). Practical methods of optimization. 1987. Unconstrained Optimization, 1.

Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. The computer journal, 7(2):149–154.

Forst, W. and Hoffmann, D. (2010). Optimization—theory and practice. Springer Science & Business Media.

Hamoda, M., Mamat, M., Rivaie, M., and Salleh, Z. (2016). A conjugate gradient method with strong wolfe-powell line search for unconstrained optimization. Applied Mathematical Sciences, 10(15):721–734.

Hestenes, M. R., Stiefel, E., et al. (1952). Methods of conjugate gradients for solving linear systems. Journal of research of the National Bureau of Standards, 49(6):409– 436.

Hillstrom, K. E. (1977). A simulation test approach to the evaluation of nonlinear optimization algorithms. ACM Transactions on Mathematical Software (TOMS), 3(4):305–315.

Liu, Y. and Storey, C. (1991). Efficient generalized conjugate gradient algorithms, part 1: theory. Journal of optimization theory and applications, 69(1):129–137.

Nocedal, J. and Wright, S. (2006). Numerical optimization. Springer Science & Business Media.

Polyak, B. T. (1969). The conjugate gradient method in extremal problems. USSR Computational Mathematics and Mathematical Physics, 9(4):94–112.

Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. Mathematical programming, 12(1):241–254.

Shewchuk, J. R. et al. (1994). An introduction to the conjugate gradient method without the agonizing pain.

Wikipedia contributors (2020). Quasi-newton method — Wikipedia, the free encyclopedia. [Online; accessed 18-April-2020].

Zoutendijk, G. (1970). Computational methods in nonlinear programming. Studies in Optimization 1, (3):125