An \(n\)-th section line search in conjugate gradient method for small-scale unconstrained optimization

Imza Fakhri \(^a\),*\), Mohd Rivaie \(^b\),*, Ibrahim Jusoh \(^b\)

\(^a\) Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi Mara Shah Alam, 40450 Shah Alam, Selangor, Malaysia

\(^b\) Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi Mara Terengganu, 21300 Kuala Terengganu, Terengganu, Malaysia

* Corresponding author: ieym_imza@yahoo.com, rivaie75@yahoo.com

Abstract

Conjugate Gradient (CG) methods are well-known methods for solving unconstrained optimization problem and popular for its low memory requirement. A lot of researches and efforts have been done in order to improve the efficiency of this CG method. In this paper, a new inexact line search is proposed based on Bisection line search. Initially, Bisection method is the easiest method to solve root of a function. Thus, it is an ideal method to employ in CG method. This new modification is named \(n\)-th section. In a nutshell, this proposed method is promising and more efficient compared to the original Bisection line search.

Keywords: Conjugate gradient; bisection method; line search; unconstrained optimization

© 2017 Penerbit UTM Press. All rights reserved

INTRODUCTION

Optimization could be defined as the science of finding the best solution to mathematical problems which are the physical modelled from the real world. According to Snyman (2005), optimization includes the study of optimality criteria of problems, the determination of algorithmic methods, the study of structure of methods and its computer experimentation with methods both under trial conditions and also on real life problem.

Mathematically, in this study, we found the minimum solution to a given function which is

\[
\min_{x \in \mathbb{R}^n} f(x).
\]

Using an initial guess or initial point, we try to get a new solution by using an appropriate method. In doing this, we basically return to the same calculation method but, using new initial guess. This returning procedure which is also known as an iterative method forms the basis of optimization that is to minimize \(f(x)\) such that

\[
x_{k+1} = x_k + \alpha_k d_k.
\]

The \(d_k\) is known as the stepsize and the \(d_k\) is known as the search direction. Different \(d_k\) give rise to different method of optimization.

The steepest descent is the earliest and recognized method, dated back in 1847 (Cauchy, 1847). Many other methods arise based on its modification. Conjugate Gradient (CG) method is one of it and dated back as early as 1952. Currently, this method seems to be the attention of many researches lately.

CONJUGATE GRADIENT METHOD

The initial calculation for conjugate gradient method starts with steepest descent direction and after that, the next search direction is calculated by adding the linear combination of the previous direction to the current gradient (Chong and Zak, 2013). The search directions defined is by

\[
d_k = \begin{cases} 
-g_k & \text{if } k = 0; \\
-g_k + \beta_k d_{k-1} & \text{if } k \geq 1
\end{cases}
\]

in which \(\beta_k\) is known as the CG coefficient while \(g_k\) is the gradient of \(f\) at point \(x_k\) (Hamoda et. al, 2015). Many researches have been conducted by researchers to find the best formula for \(\beta_k\) that will yield a better result. Some of the most well-known formulas are Fletcher-Reeves (FR), Polak-Ribiére-Polyak (PRP) and Rivaie, Mustafa, Ismail and Leong (RMIL).

They are written as

\[
\beta_k^{FR} = \frac{\alpha_k^2}{\alpha_k^2 + \alpha_{k-1}^2} \quad \text{(Fletcher–Reeves, 1964)}
\]

\[
\beta_k^{PRP} = \frac{\alpha_k^2 (g_k^2 + g_{k-1}^2)}{\alpha_k^2 (g_{k}^2 + g_{k-1}^2) + \alpha_{k-1}^2 g_k^2} \quad \text{(Polak and Ribiere, 1969;Polyak, 1969)}
\]

\[
\beta_k^{RMIL} = \frac{\alpha_k^2 (g_k^2 - g_{k-1}^2)}{\alpha_k^2 (g_{k}^2 - g_{k-1}^2) + \alpha_{k-1}^2 g_k^2} \quad \text{(Rivaie, Mustafa, Ismail and Leong, 2012)}
\]
BISECTION AND N-th SECTION LINE SEARCH

In finding the step size in CG method, there are two line searches that can be applied which are exact and inexact line search. Exact line search is the exact value of step size, $\alpha_k$, calculated analytically using $f'(\alpha_k) = 0$. While, inexact line search is the approximate value of $\alpha_k$, calculated using numerical analysis such as Bisection, Newton and secant line search.

Bisection method is the easiest method using an interval which is divided into two or half. Doron (2010) assume that this method has opposite signs at both edges of the intervals where $f(a).f(b) < 0$. Then, it is known that $f(x)$ has at least one root in the interval, $[a, b]$. The existing method proceeds and continues iterating until it converges to a point within the tolerance range and finds the value of $x$ such that $f(x) = 0$ or approximately 0.

The main advantage of Bisection method is the behaviour itself is always convergent since the method brackets the root much more quickly than the incremental search method does. Hence, Bisection method is easy to use, apply and has wide range of applications in other developments.

However, the division of the interval into two sections leads to slow convergence of Bisection method. Hence, a new method is proposed called an n-th section method. This method is compared with the original Bisection method based on number iterations, CPU times and accuracy.

An n-th section method is a modification from classical Bisection method which is fourth and sixth section method. This new scheme divided the interval into four and six section. The root is then identify either in the first, second, third, fourth, fifth or sixth interval. This will lead to faster convergence of the root and provide faster calculation of the roots. For an even number of $n$, the algorithm will be similar to Bisection method. However, the n-th section inherits the problems from the bisection itself and some modifications need to be implemented before fitting it as a line search. Next, computer programming code will be constructed using Maple 16 software. Algorithm 1 shows CG method with an employment of original Bisection and n-th section line search Algorithm 1 and Algorithm 2 shows the proposed CG method under modified line search.

Algorithm 1: CG Method with an employment of original Bisection and n-th section line search algorithm

1. Initialization.
   Given $x_0$, set $k = 0$.

2. Computing search direction.
   
   
   \[
   d_k = \begin{cases} 
   -g_k, & \text{if } k = 0; \\
   -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1
   \end{cases}
   \]

3. Computing step size $\alpha_k$
   Identify two numbers $a$ and $b$ as an interval at which $f'(\alpha_k)$ has different signs.
   Divide into sections, $m = \frac{a+b}{n}$ where $n=2, 4, 6$
   Determine if
   i. $f'(a).f'(a + m) < 0$ then $r \in (a, a + m)$
   ii. $f'(a + m).f'(a + 2m) < 0$ then $r \in (a + m, a + 2m)$
   iii. $f'(a + 2m).f'(a + 3m) < 0$ then $r \in (a + 2m, a + 3m)$
   .
   $f'(a + im).f'(a + (i + 1)m) < 0$ then $r \in (a + im, a + (i + 1)m)$
   where $i = 0, 1, 2, 3, ...$

   Define $C_{mid}$ as midpoint, $C_{mid} = \frac{(a+im) + (a+(i+1)m)}{2}$ and $C_{mid} = \alpha_k$

   Repeated until desired iteration/ accuracy

4. Updating new point.
   $x_{k+1} = x_k + \alpha_k d_k$
Algorithm 2: Proposed CG method with improvement of Bisection and n-th section line search

Step 1: Initialization.
Given $x_0$, set $k = 0$.

Step 2: Computing search direction.
$$d_k = \begin{cases} -g_k, & \text{if } k = 0; \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 1 \end{cases}$$

Step 3: Computing step size $\alpha_k$
Identify two numbers $a$ and $b$ as an interval at which $f'(a_k)$ has different signs.
Divide into sections, $m = \frac{a+b}{n}$ where $n = 2,4$ and $6$
Determine if
i. $f'(a).f'(a+m) > 0$ then continue
ii. $f'(a+m).f'(a+2m) > 0$ then continue
iii. $f'(a+2m).f'(a+3m) > 0$ then continue
... $f'(a+im).f'(a+(i+1)m) < 0$ then $r \in (a+im, a+(i+1)m)$ then stop
where $i = 0,1,2,3,...$
Define $c_{mid}$ as midpoint, $c_{mid} = \frac{(a+im)+(a+(i+1)m)}{2}$
and $c_{mid} = \alpha_k$
Repeated until desired iteration/ accuracy
Updating new point.
$$x_{k+1} = x_k + \alpha_k d_k$$

Step 5: Convergent test and stopping criteria.
If $f(x_{k+1}) < f(x_k)$ and $\|g_k\| \leq \varepsilon$ then stop
Otherwise go to Step 1 with $k = k + 1$.

The stopping criteria for the standard test problem is $\|g_k\| \leq 10^{-6}$ as suggested by Andrei (2011) is applied. The initial points in a single quadrant are then tested from a point that is closer to the solution point to the one that is furtherest. There is a tendency that an algorithm will found the solution point when we used an initial point which is closer. In this case, four other initial points which is spread further apart is chosen to see the behaviour of the method. Comparison of both method are based on its efficiency in terms of number of iterations. A comparison between existing methods and proposed method based on its efficiency and convergence properties are studied. Performance profile introduced by Dolan & Moré will be also be featured and conducted in this paper. The numerical results is tested on a same computer with Core i5 processor 3337U with 4GB RAM.

STANDARD OPTIMIZATION TEST PROBLEM

The quality of an optimization methods are frequently evaluated using standard optimization test problems. In this research, all methods have been tested using four different functions. In this section, we shall discuss about the main criteria of the selected test problems. We have also plot the test problems with two variables using Maple 16 to show the different shape of the functions which has been selected.

Problem 1: Tridiagonal Function with $n=2$ (Andrei, 2008)
Function: $f(x) = \sum_{i=1}^{n/2} (x_{2i-1} - x_{2i} - 3)^2 + (x_{2i-1} - x_{2i} + 1)^2$
Global minimum is $x^* = (0,0)$ and minimum function values is $f(x^*) = 0$

Problem 2: Booth Function with $n=2$ (Witte and Holst, 1964)
Function: $f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
Global minimum is $x^* = (1,3)$ and minimum function values is $f(x^*) = 0$

Problem 3: Bukin Function with $n=2$ (as cited in Mishra, 2006)
Function: $f(x) = 100(x_1 - 0.01x_2^2 + 1) + 0.01(x_1 + 10)^2$
Global minimum is $x^* = (-10,0)$ and minimum function values is $f(x^*) = 0$, see Fig. 6.
Problem 4: Three-hump Function with \( n=2 \) (Jasson & Knuppel, 1992)

Function:  
\[ f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} - x_1x_2 + x_2^2 \]

Global minimum is \( x^* = (0,0) \) and minimum function values is 0, see Fig. 7.

RESULTS AND DISCUSSION

The results are presented for CG-FR, CG-PRP and CG-RMIL based on the functions that have been discussed in previous section. Table 1, Table 2 and Table 3 shows the number iterations for CG-FR method, CG-PRP method and CG-RMIL method respectively. Fig. 8 shows the performance profile involving the classical Bisection, modified Bisection, modified 4-th section and modified 6-th section. The abbreviation of CB, MB, M4 and M6 are denoted as classical Bisection, modified Bisection, modified 4-th Section and modified 6-th Section line search respectively. In each table, F is denoted as failure to converge in which the method could not reach the solution point.

For CG-FR method using the Tridiagonal function, the number of iterations is equivalent for all line searches using any initial points. Both Tridiagonal and Bukin function possess global convergence properties since all the tested initial points converges to the solution point. However, some of the initial points failed to converge for Bukin and Threehump function. In this results, the application of modified 6th Section line search in CG-FR method is quite promising since the convergence is unlikely to fail when compared to the classical Bisection.

For CG-PRP method, only booth function possess global convergence properties. For Tridiagonal function, only the nearest point tend to fail. Some of the initial points are also fail to converge for Bukin and Threehump function. In bukin function, it is obvious that the application of \( n^{th} \) Section line search can be a problem solver when using modified 6th Section line search.

### Table 1 Number of iterations of CG-FR method

<table>
<thead>
<tr>
<th>Initial Points</th>
<th>CB</th>
<th>MB</th>
<th>M4</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tridiagonal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(12,12)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>(50,50)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(100,100)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(500,500)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Booth</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(12,12)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(50,50)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(100,100)</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(500,500)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Bukin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7,7)</td>
<td>F</td>
<td>38</td>
<td>28</td>
<td>F</td>
</tr>
<tr>
<td>(15,15)</td>
<td>F</td>
<td>F</td>
<td>39</td>
<td>220</td>
</tr>
<tr>
<td>(30,30)</td>
<td>F</td>
<td>F</td>
<td>80</td>
<td>24</td>
</tr>
<tr>
<td>(50,50)</td>
<td>F</td>
<td>F</td>
<td>172</td>
<td>290</td>
</tr>
<tr>
<td>Threehump</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1,-1)</td>
<td>F</td>
<td>12</td>
<td>12</td>
<td>F</td>
</tr>
<tr>
<td>(-10,10)</td>
<td>8</td>
<td>8</td>
<td>F</td>
<td>6</td>
</tr>
<tr>
<td>(-30,30)</td>
<td>F</td>
<td>F</td>
<td>10</td>
<td>F</td>
</tr>
<tr>
<td>(-100,100)</td>
<td>8</td>
<td>8</td>
<td>F</td>
<td>33</td>
</tr>
</tbody>
</table>

### Table 2 Number of iterations of CG-PRP method

<table>
<thead>
<tr>
<th>Initial Points</th>
<th>CB</th>
<th>MB</th>
<th>M4</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tridiagonal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(12,12)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>(50,50)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(100,100)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(500,500)</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Booth</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(12,12)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(50,50)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(100,100)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>(500,500)</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Bukin</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(7,7)</td>
<td>F</td>
<td>F</td>
<td>38</td>
<td>22</td>
</tr>
<tr>
<td>(15,15)</td>
<td>F</td>
<td>F</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>(30,30)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>36</td>
</tr>
<tr>
<td>(50,50)</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>111</td>
</tr>
<tr>
<td>Threehump</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(-1,-1)</td>
<td>F</td>
<td>F</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>(-10,10)</td>
<td>4</td>
<td>4</td>
<td>F</td>
<td>4</td>
</tr>
<tr>
<td>(-30,30)</td>
<td>5</td>
<td>5</td>
<td>F</td>
<td>4</td>
</tr>
<tr>
<td>(-100,100)</td>
<td>7</td>
<td>7</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Based on this performance profile, the best method is determined based on the right side of figure which has the highest curve value. In this case, it shows that the best line search is modified 6th section with 0.8. This means that this method is able to solve 80% of the given problems. Based on the left side, it shows that the classical Bisection, 4th section and 6th section possess the top curve which means that this method is competitively fast. However, the classical Bisection and 4th section are the lowest curve on the right side which implies the least problem solver.

CONCLUSION

In this paper, a new line search has been proposed which is the modified n-th section method that is applied in CG method. It shows that this method could be used as a line search for CG method with superior performance. It is believe that some improvement could also be done in future research to enhance the performance of these line searches.

ACKNOWLEDGEMENT

This research is conducted under the fundamental research grant scheme (FRGS) grant number : 600-RMI/FRGS 5/3 (64/2015).

REFERENCES


