Modified the CG-Algorithm for Unconstrained Non-Linear Optimization by Using Oren’s Update

Abbas Y. Al-Bayati  Abdulghafor M. Al-Rozbayani
abdulghafor_rozbayani@uomosul.edu.iq

College of Computer Sciences and Mathematics, University of Mosul, Iraq

Received on: 20/08/2002 Accepted on: 12/10/2002

ABSTRACT

In this paper we have modified a new extended generalized conjugate gradient steps with self-scaling variable metric updates for unconstrained optimization. The new proposed algorithm is based on the inexact line searches and it is examined by using different non-linear test functions with various dimensions.

Keywords: Conjugate Gradient, Unconstrained Optimization, Nonlinear Optimization, Variable Metric.

1. Introduction:

We try to solve the unconstrained minimization problem

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

when f is twice continuously differentiable function. This problem is usually solved iteratively. Starting with an initial estimate $x_1$ of the minimum point $x^*$, each subsequent point $x_{k+1}$, $k \geq 1$, will be derived by search along a descent direction $d_k$, such that $d_k^T g_k < 0$, where $g_k = \nabla f(x)$, such that

$$x_{k+1} = x_k + \alpha_k d_k, \quad k \geq 1, \quad (2)$$
where $\alpha_k$ is the step-length satisfying line search conditions such that
\[
 f(x_k + \alpha_k d_k) \leq f(x_k) + c_1 \alpha_k g_k^T d_k
\]
and
\[
 |g_{k+1}^T d_k| \leq -c_2 g_k^T d_k
\]
where $0 < c_1 < c_2 < 1$.

We consider that
\[
 v_k = x_{k+1} - x_k \quad (5)
\]
\[
 y_k = g_{k+1} - g_k \quad (6)
\]

Conjugate Gradient (CG) method is one of the few practical methods for solving large dimension problems because it does not require matrix storage and its iteration cost is very low.

Now we begin from the initial direction $d_1$ such that
\[
 d_1 = -g_1. \quad (7)
\]

The search direction for the next iteration has the following form:
\[
 d_{k+1} = -g_{k+1} + \beta_k d_k \quad (8)
\]

Where $\beta_k$ is a constant parameter defined by either
\[
 \beta_k = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} \quad (9a)
\]

Or
\[
 \beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|g_k\|^2} \quad (9b)
\]

The definition of $\beta_k$ in (9a) is due to Fletcher and Reeves [6] and $\beta_k$ in (9b) is due to Polak Ribiere [12].

2. Generalized CG-Methods:

CG-algorithm usually requires more function evaluations than the variable Metric (VM) method to solve small dimensionality problems. Therefore, many extensions and modifications have been proposed in this field. Liu and Storey [8] introduced a generalized PR algorithm. They studied the effect of the inexact line search on conjugacy in unconstrained optimization, and they showed that their algorithm has global convergence for twice continuously differentiable functions with a bounded level set.

2.1 Generalized CG-algorithm (Liu and Storey, 1991)

Step 1: let $x_1$ be an initial point of the minimizer $x^*$ of $f$.
Step 2: set the iteration $k=1$ and the search direction $d_k=-g_k$. 

Step 3: do a line search: set $x_{k+1} = x_k + \alpha_k d_k$, set $k = k+1$, $\alpha$ is a line search parameter.

Step 4: if $\|g_{k+1}\| < \varepsilon$, where $\varepsilon = 5 \times 10^{-5}$, take $x^*$ as $x_{k+1}$ and stop;
Otherwise go to step 5.

Step 5: if $k > n \geq 2$, go to step 9; otherwise go to step 6.

Step 6: let $t_k = d_k^T g_{k+1} d_k$, $s_k = g_{k+1}^T g_{k+1} g_{k+1}$, and $u_k = g_{k+1}^T G_{k+1} d_k$.

Step 7: if $t_k > 0$, $s_k > 0$, $1 - u_k^2 / (t_k s_k) \geq 1 / (4r_k)$, and

$$\frac{(s_k / g_{k+1}^T g_{k+1}) / (t_k / d_k^T d_k)} \leq r_k, \ n > 0$$
then go to step 8;
Otherwise go to step 9.

Step 8: let

$$d_{k+1} = [(u_k g_{k+1}^T d_k - t_k g_{k+1}^T g_{k+1}) g_{k+1} + (u_k g_{k+1}^T g_{k+1} g_{k+1} - s_k g_{k+1}^T d_k) d_k] / w_k,$$

where $w_k = t_k s_k - u_k^2$, go to step 3.

Step 9: set $x_{k+1} = x_1$ and go to step 2.

This algorithm was compared with currently available standard routines and their results demonstrate a general efficient GPR algorithm.

Usually CG-algorithms are implemented with restarts, in order to avoid the effects of an accumulation error. Fletcher [5] in his standard method suggested to restart his algorithm with the steepest descent direction every $n$ or $n+1$ iteration, where $n$ is a dimension of the problem, another restarting direction was suggested by Powell [13]. He developed a new procedure for starting CG-methods. Powell checked that the new search direction $d_{k+1}$ will be sufficiently downhill if their inequalities

$$\left| g_{k+1}^T g_k \right| > 0.2 \left| g_{k+1}^T g_{k+1} \right|$$

is satisfied.

3. Self–Scaling VM methods:

The practical and theoretical merits of the quasi Newton (QN) family of methods for unconstrained optimization have been systematically explored since the classic paper of Fletcher and Powell [7] analyzing Davidon’s VM method.

In the family of VM updating, it is useful to multiply each $M_k$ by some scale factor $\rho_k > 0$ before using the update formula. With exact line searches, this can be shown to present the conjugacy property in the quadratic case, although we may no longer have $H_{k+1} = G^{-1}$. However, the focus here is to improve the single-step rather than the $n$-step convergence behavior of the algorithm. Methods that automatically prescribe scale factor in a manner such that, if the function is quadratic then the eigenvalues of
Thus in 1970’s the self-scaling VM algorithms were introduced showing significant improvement in efficiency over standard VM-methods. In particular, in a series of papers by Oren [11], Al-Bayati [1], Nocedal [10] and Al-Bayati and Al-Salih [2] algorithms for minimizing an unconstrained nonlinear function $f(x)$ were developed.

Now we summarize the scaled BFGS algorithms due to Oren [11].

3.1 Oren’s algorithm (1974)
Step 1: Start with an initial point $x_1$, Set $k=1$ and choose $H_1=I$, where I is the identity matrix.
Step 2: Determine the step-size $\alpha_k$ that minimizes $f(x_k+\alpha_k d_k)$ where $d_k=-H_k g_k$ and obtain $x_{k+1}=x_k+\alpha_k d_k$, $\alpha_k$ is determined by inexact line search.
Step 3: If $\|g_{k+1}\| < \varepsilon$, where $\varepsilon=5\times10^{-5}$, take $x^*$ as $x_{k+1}$ and stop;
Otherwise continue.
Step 4: Update $H_k$ by $H_{k+1}$ by using Oren’s update as follows

$$H_{k+1}^{\text{Oren}} = H_k - \frac{H_k y_k y_k^T}{y_k^T H_k y_k} + w_k w_k^T \left[ \frac{\psi_k + v_k v_k^T}{v_k^T y_k} \right]$$

where

$$\psi_k = \frac{v_k^T y_k}{y_k^T H_k y_k}$$

and $w_k$ is vector defined by

$$w_k = (y_k^T H_k y_k)^{1/2} \left[ \frac{v_k}{y_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right]$$

Step 5: Set $k=k+1$ and go to step 1.

4. **Preconditioned CG-Methods:**
Any problem in an unconstrained optimization can be solved by CG or VM methods, each of these methods has particular advantages and disadvantages. In general, a CG-method normally requires more iterations than a QN or VM method to obtain an equally good minimum point but a CG-method requires less storage for implementation per iteration. CG-methods have proved to be valuable where $n$ is large because at each step a
few n vectors have to be stored and hence the computational costs and storage requirements are affordable, even for large problems.

The new class of CG-methods has been developed, termed preconditioned conjugate gradient methods (PCG); the idea of the preconditioning has been extended directly to nonlinear problem [9].

4.1 New Modified Generalized CG-algorithm:
Step 1: Let $x_1$ be an initial point of the minimizer $x^*$ of $f$ and $H_1=I$ where $I$ is the identity matrix.
Step 2: Set $k=1$ and $d_k=-H_k g_k$.
Step 3: Do a line search to compute $\alpha_k$ that minimizes $f(x_k+\alpha_k d_k)$ then compute $x_{k+1}=x_k+\alpha_k d_k$.
Step 4: If $\|g_{k+1}\| < \epsilon$, where $\epsilon=5\times10^{-5}$, take $x^*$ as $x_{k+1}$ and stop; otherwise go to step 5.
Step 5: If $k=n$ or $g_k^T g_k + g_{k+1}^T g_{k+1} > 0.2 |g_k^T g_k|$, go to step 2; otherwise continue.
Step 6: Let $t_k = d_k^T g_{k+1}/d_k$, $s_k = g_k^T g_{k+1}$, and $u_k = g_k^T g_{k+1} d_k$.
Step 7: If $t_k > 0$, $s_k > 0$, $1 - u_k^2/(t_k s_k) \geq 1/(4r_k)$, and $(s_k/g_k^T g_{k+1})/(t_k/d_k) \leq r_k$, $r_k > 0$ then go to step 8; otherwise go to step 10.
Step 8: Let $d_{k+1} = [(u_k g_{k+1} + t_k g_k^T H_k g_{k+1}) H_k g_{k+1} + (u_k g_k^T H_k g_{k+1} - s_k g_{k+1} d_k) d_k]/w_k$ where $w_k = u_k s_k - u_k^2$.
Step 9: Update $H_k$ by $H_{k+1}$ using Oren’s [11] formula as given in (10).
Step 10: Set $x_{k+1} = x_k$, $k=k+1$ and go to step 3.

4.2 Derivation of the Modified Generalized CG Direction:
Let $H$ be any symmetric positive-definite preconditioned matrix, then by Nazareth [9] with Choleski Factorization $H$ can be factorized as:
$$ H=L L^T, $$
where $L$ is a real lower triangular matrix and non-singular matrix. Let $f$ be the strictly quadratic function
$$ f(x)=x^T G x/2 + b^T x + c, $$
then the gradient is
$$ g(x)=f'(x)=G x + b^T. $$
Let $x=Lz$,
where $z$ defines a new vector spaces and is defined as:
$$ h(z)=f(Lz)=(Lz)^T G (Lz)/2+(Lz)^T b+c, $$
(16)
which implies that

\[ h'(z) = f'(Lz) = L^TGLz + L^Tb^T = L^T(GLz + b^T) = L^Tg(x). \]  

Eq. (17) gives a relationship between the gradients in x-space and z-space, i.e.

\[ g = L^Tg, \]  

where \( g \) is the gradient in z-space while \( g \) is the gradient in x-space, so that if \( h(z^*) = 0 \), then \( g(Lz^*) = 0 \).

Set \( z_{k+1} = z_k + \alpha_k d_k \).

Multiplication of Eq. (19) by \( L \) we get,

\[ Lz_{k+1} = Lz_k + \alpha_k Ld_k. \]  

Using Eq. (15), then Eq. (20) becomes

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

therefore,

\[ \tilde{d}_k = L^{-1}d_k. \]  

which implies that

\[ \tilde{d}_k = L^{-1}d_k. \]  

Set

\[ \tilde{y}_k = \tilde{g}_{k+1} - \tilde{g}_k, \]

where \( \tilde{g}_k, \tilde{g}_{k+1} \) are the gradients of \( h(z) \) at the point \( z_k, z_{k+1} \), respectively.

By using Eq. (18), then Eq. (24) becomes

\[ \tilde{y}_k = L^T\tilde{g}_{k+1} - L^T\tilde{g}_k = L^Ty_k. \]  

Now consider applying the conjugate gradient method,

\[ \tilde{d}_{k+1} = [(u_k \tilde{g}_{k+1}^T \tilde{d}_k - t_k \tilde{g}_{k+1}^T \tilde{g}_{k+1}) \tilde{g}_{k+1} + (u_k \tilde{g}_{k+1}^T \tilde{g}_k - s_k \tilde{g}_{k+1}^T \tilde{d}_k) \tilde{d}_k] / w_k \]

By using Eqs. (15), (18), (23) and (25) we get

\[ L^{-1}d_{k+1} = [(u_k \tilde{g}_{k+1}^T LL^{-1}d_k - t_k \tilde{g}_{k+1}^T LL^T\tilde{g}_{k+1}) L^T\tilde{g}_{k+1} + (u_k \tilde{g}_{k+1}^T LL^T\tilde{g}_{k+1} - s_k \tilde{g}_{k+1}^T LL^{-1}d_k) L^{-1}d_k] / w_k, \]

multiply (26) by \( L \) and using (12) and \( LL^{-1} = I \) where I is the identity matrix we get:
\[ d_{k+1} = \left[ (u_k g_k^T d_k - t_k g_k^T H g_{k+1}) H g_{k+1} + (u_k g_k^T H g_{k+1} - s_k g_k^T d_k) d_k \right] / w_k \] (27)

Equation (27) is our proposed CG Method with Metric update H, where H is any positive-definite symmetric matrix, or the preconditioned CG Method (PCG). Thus the search direction defined in Eq. (27) is a new direction for the Generalized Polak-Ribiere CG-direction.

5. **Numerical Results:**

In this section we are going to employ the new proposed algorithm with a certain effective CG-method; namely Liu and Storey [8].

We will show that our algorithm is better than the Generalized CG-of Liu and Storey’s [8] algorithm. Using Oren [11] self-scaling VM update Oren [11] as an acceleration tool to the Generalized CG-algorithm to decrease the number of iterations (NOI) and number of function evaluations (NOF). The comparison tests involve ten well-known test functions with different dimensions. All the results were obtained using double precision on the (Pentium II Computer) using programs written in Fortran Language.

The terminate criteria is \((5 \times 10^{-5})\) with Powell’s restarting criterion \( \| g_{k+1} \| > 0.2 \| g_k \| \) and the algorithms use the cubic line search strategy, with function and gradient values and it is an adaptation from that published by Bunday [4].

The comparative performances of the algorithms are evaluated by considering both the total numbers of iterations (NOI) and total number function evaluations (NOF).

Thus all our numerical results are presented in table (1). In this table we have compared our modified proposed algorithm with Liu and Storey [8] algorithm by using (Ten) cases and for dimension \( 2 \leq n \leq 1000 \).

It is clear that the modified algorithm outperform the standard Liu and Storey’s algorithm as results of this comparison.
We see from the above table that taking the cubic routine as 100% NOI and NOF yields:

<table>
<thead>
<tr>
<th>Tools</th>
<th>Liu and Storey’s Algorithm</th>
<th>Modified GCG algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>100</td>
<td>44.8</td>
</tr>
<tr>
<td>NOF</td>
<td>100</td>
<td>37.8</td>
</tr>
</tbody>
</table>

It is clear from the above table that the modified algorithm is more efficient than Generalized Liu and Storey’s CG-algorithm.

6. **Conclusions:**

In the field of unconstrained optimization, a new search direction is implemented for modified extended generalized CG-VM steps. The technique uses inexact line search with promising numerical results.
REFERENCES


