Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems

* Sawsan S. Ismael **Basim A. Hassan

* College of Education University of Mosul
*Department of Mathematics, College of Computers Sciences and Mathematics University of Mosul, IRAQ

Received 18/09/2012 Accepted 09/01/2013

Abstract

In this paper, a new hybrid nonlinear conjugate gradient method are presented, which produce sufficient descent search direction at every iteration. This methods showed globally convergent under some assumptions. The numerical results show that all this new hybrid method are efficient for the given test problems.

Introduction

Let us consider the unconstrained optimization problem

$$\min \left\{ f(x) \mid x \in \mathbb{R}^n \right\}$$

(where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, bounded from below. For solving this problem, starting from an initial guess $x_0 \in \mathbb{R}^n$. a nonlinear conjugate gradient method, generates a sequence $\{x_k\}$ as:

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

........(2)
Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems

Journal of Education and Science – Pure Sciences

where $\alpha_k$ is the step-size, and the direction $d_k$ are generated as

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad d_0 = -g_0 \quad \text{.........(3)}$$

where $\beta_k$ is known as the conjugate gradient parameter, $v_k = x_{k+1} - x_k$ and $g_k = \nabla f(x_k) \quad [1]$. The step size $\alpha_k$ is chosen in such a way that $\alpha_k > 0$ and satisfies the strong Wolfe (SW) conditions

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k d_k^T g_k \quad \text{.........(4)}$$

$$g(x_k + \alpha_k d_k)^T d_k \leq -\delta_2 d_k^T g_k \quad \text{.........(5)}$$

with $0 < \delta_1 < \delta_2 < 1$, where $f_k = f(x_k)$, $g_k = g(x_k)$, $g_k$ are the gradient of $f$ evaluated at the current iterate $x_k \quad [7]$. Where $d_k$ is a descent direction. Different conjugate gradient algorithms correspond to different choices for the parameter $\beta_k$. For example Fletcher and Reeves (FR) \([6]\), Dai and Yuan (DY) \([4]\) and Conjugate Descent (CD) \([5]\) :

$$\beta_{k}^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}, \quad \beta_{k}^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k}, \quad \beta_{k}^{CD} = \frac{g_{k+1}^T g_{k+1}}{g_k^T d_k} \quad \text{.........(6)}$$

They have strong convergence properties, but they may have modest practical performance due to jamming. On the other hand, the methods of Polak and Ribiere (PR) \([9]\), Hestenes and Stiefel (HS) \([7]\), or Liu and Storey, (LS) \([8]\) :

$$\beta_{k}^{PR} = \frac{y_k^T y_k}{g_k^T y_k}, \quad \beta_{k}^{HS} = \frac{y_k^T y_k}{y_k^T d_k}, \quad \beta_{k}^{LS} = \frac{g_{k+1}^T g_{k+1}}{y_k^T y_k} \quad \text{.........(7)}$$

in general, may not be convergent, but they often have better computational performances.

Also, under mild assumptions on the objective function, DY method is shown to be globally convergent under a variety of line search conditions. These advantages motivated us to study the hybridizations of HS and DY methods following the effective approach proposed in \([2,3\quad \text{and}\quad 11]\). The formula $\beta_k$ in \([2,3]\), namely $\beta_k^C$, is obtained by a convex combination of $\beta_k^{HS}$ and $\beta_k^{DY}$. That is,

$$\beta_k^C = (1-\theta_k)\beta_k^{HS} + \theta_k \beta_k^{DY} = (1-\theta_k)\frac{g_{k+1}^T y_k}{y_k^T y_k} + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k} \quad \text{.........(8)}$$

where $\theta_k$, namely the hybridization parameter, is a scalar parameter satisfying $0 \leq \theta_k \leq 1$. Therefore, Substituting (8) into (3), we get :

$$d_{k+1} = -g_{k+1} + (1-\theta_k)\frac{g_{k+1}^T y_k}{y_k^T d_k} d_k + \theta_k \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k} d_k, \quad \text{.........(9)}$$
Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems
Journal of Education and Science – Pure Sciences

Or equivalently,
\[ d_{k+1} = -g_{k+1} + (1 - \theta_k) \frac{g_{k+1}^T y_k}{y_k^T y_k} v_k + \theta_k \frac{g_{k+1}^T g_{k+1}^T v_k}{y_k^T y_k} . \] ........(10)

As known, if the point \( x_{k+1} \) is close enough to a local minimizer \( x^* \), then a good direction to follow is the Newton direction, that is,
\[ d_{k+1} = -G_{k+1}^{-1} g_{k+1} . \] ........(11)

Motivated by this, Andrei [2,3] rewrite (10) as follows :
\[ -G_{k+1}^{-1} g_{k+1} = -g_{k+1} + (1 - \theta_k) \frac{g_{k+1}^T y_k}{y_k^T y_k} v_k + \theta_k \frac{g_{k+1}^T g_{k+1}^T v_k}{y_k^T y_k} v_k . \] ........(12)

After some algebraic manipulations one obtains :
\[ \theta_k = \frac{\frac{g_{k+1}^T g_{k+1}^T v_k}{y_k^T y_k} v_k^T G_{k+1} v_k}{\frac{g_{k+1}^T g_{k+1}^T v_k}{y_k^T y_k} v_k^T G_{k+1} v_k} . \] ........(13)

In quasi-Newton methods, an approximation matrix \( B_k \) for the Hessian \( G_k \) is used and updated so that the new matrix \( B_{k+1} \) satisfies a version of the secant equation. In [2], \( B_{k+1} \) is determined to satisfy the standard secant equation, that is, \( B_{k+1} v_k = y_k \). Therefore, \( \theta_k \) is computed by :
\[ \theta_k = \frac{v_k^T g_{k+1}}{g_{k+1}^T g_{k+1}} . \] ........(14)

In [3], \( B_{k+1} \) is determined to satisfy the modified secant equation proposed by Li et al. [10],
\[ B_{k+1} v_k = y_k + \eta_k v_k / \| v_k \|^2 , \] ........(15)

where
\[ \eta_k = 2(f_k - f_{k+1}) + (g_{k+1} + g_k)^T v_k , \] ........(16)

and so \( \theta_k \) is computed by
\[ \theta_k = \left[ \frac{\eta_k}{\| v_k \|^2} - 1 \right] \frac{v_k^T g_{k+1}}{g_{k+1}^T g_{k+1} - \eta_k} - \eta_k \frac{g_{k+1}^T g_{k+1}^T v_k}{y_k^T y_k} . \] ........(17)

Now, using (17) in (10) we get :
\[ d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{y_k^T v_k + \eta_k} v_k - \left[ \frac{1}{\|v_k\|^2} \right] \frac{v_k^T g_{k+1}}{y_k^T v_k + \eta_k} v_k. \] 

However, using exact line searches \((v_k^T g_{k+1} = 0)\) in (18), the direction \(d_{k+1}\) reduced to

\[ d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{y_k^T v_k + \eta_k} v_k. \] 

It was shown in [3] that the hybrid CG method with \(\theta_k\) as in (17) incorporated with an acceleration scheme is more efficient than the HS and DY method, and the hybrid CG methods proposed Andrei [2].

The structure of the paper is as follows. In section 2, we present the new hybrid conjugate gradient algorithm. Section 3 presents a new Algorithm and Convergence analysis. Section 4 numerical results are presented and In section 5 discuss the we give brief conclusions and discussions.

2. A new hybrid conjugate gradient algorithm

We develop the secant equation based on the modified BFGS method proposed by Li et al. [10]. For this purpose, in order to unify both approaches, we consider a slight modification of the modified secant condition (15) as

\[ B_{k+1} y_k = z_k \quad \text{where} \quad z_k = u_k y_k + (1-u_k) \eta_k v_k / \|v_k\|^2, \] 

This leads us to development a hybrid conjugate gradient algorithm (10) where

\[ \theta_k = \frac{\eta_k y_k^T g_{k+1} - g_{k+1}^T y_k (1-u_k) \eta_k}{g_{k+1}^T g_k + \frac{g_{k+1}^T g_k (1-u_k) \eta_k}{u_k y_k^T v_k}}. \] 

Now, using (21) in (10) we get :

\[ d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k) \eta_k} v_k - \left[ \frac{1}{\|v_k\|^2} \right] \frac{v_k^T g_{k+1}}{u_k y_k^T v_k + (1-u_k) \eta_k} v_k. \] 

Using exact line searches \((v_k^T g_{k+1} = 0)\) in (22), the direction \(d_{k+1}\) defined in (22) reduced to

\[ d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k) \eta_k} v_k. \]
Where \( u \in [0,1] \) is a constant. Our motivation to get a good algorithm for solving (1) is to choose the parameter \( u \) in (12) in such a way so that for every \( k \geq 1 \) the direction \( d_{k+1} \) given by (23) is the Newton direction. This is motivated by the fact that when the initial point \( x_0 \) is near the solution of (1) and the Hessian is a nonsingular matrix then the Newton direction is the best line search direction. Therefore, from the equation

\[
-G^{-1}g_{k+1} = -g_{k+1} + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k)\eta_k} v_k .
\] ........(24)

Multiplying (24) by \( y_k^T \), we have

\[
-G^{-1}y_k^T g_{k+1} = -y_k^T g_{k+1} + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k)\eta_k} y_k^T v_k .
\] ........(25)

Since \( G^{-1}y_k = v_k \) then we have

\[
-v_k^T g_{k+1} = -y_k^T g_{k+1} + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k)\eta_k} y_k^T v_k .
\] ........(26)

from (26) we get :

\[
\frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k)\eta_k} = -v_k^T g_{k+1} + y_k^T g_{k+1}
\]  ....(27)

\[
(y_k^T g_{k+1})y_k^Tv = (-y_k^T g_{k+1}) - u_k \eta_k + u_k y_k^Tv
\]  ....(28)

\[
(y_k^T g_{k+1})y_k^Tv = (-y_k^T g_{k+1})(1-u_k)\eta_k + (-y_k^T g_{k+1})u_k y_k^Tv + (y_k^T g_{k+1})(1-u_k)\eta_k + (y_k^T g_{k+1})u_k y_k^Tv
\]  ....(29)

\[
(y_k^T g_{k+1})y_k^Tv + \eta_k (y_k^T g_{k+1}) - \eta_k (y_k^T g_{k+1}) =
\]  ....(30)

and from (30) we get :

\[
u_k = \frac{(y_k^T g_{k+1})y_k^Tv + \eta_k (y_k^T g_{k+1}) - \eta_k (y_k^T g_{k+1})}{-\eta_k (y_k^T g_{k+1}) + (-y_k^T g_{k+1})(1-u_k)\eta_k - \eta_k (y_k^T g_{k+1}) + (y_k^T g_{k+1})(1-u_k)\eta_k}
\]  ....(31)

### 3. New Algorithm and Convergence

We analyze the convergence property of the hybrid CG-method using our newly proposed formal as in (23). Throughout this section, we assume \( g_{k+1} \neq 0 \), for \( k \geq 1 \), otherwise, a stationary point is at hand. We make the following basic assumptions on the objective function.

**Assumptions**
The level set \( l = \{ x \in \mathbb{R}^n | f(x) \leq f(x_0) \} \) is bounded, there exists a constant \( B > 0 \) such that
\[
\|x\| \leq B, \quad \forall x \in l.
\]

ii- In some neighborhood \( U \) of \( L \), \( f(x) \) is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant \( \mu > 0 \) such that
\[
\|g(x_{k+1}) - g(x_k)\| \leq \mu \|x_{k+1} - x_k\|, \quad \forall x_{k+1}, x_k \in U.
\]

3.1. The Algorithm has the Following Steps:

Step 0: Given parameters \( \varepsilon = 1 \times 10^{-5} \), \( \delta_1 \in (0,1) \), \( \delta_2 \in (0,1/2) \)
choose initial point \( x_0 \in \mathbb{R}^n \), \( k = 1 \), \( d_k = -g_k \).

Step 1: Computing \( g_k \); if \( \|g_k\| \leq \varepsilon \) then stop; else continue.

Step 2: Set \( x_{k+1} = x_k + \alpha_k d_k \), (Use strong Wolfe line search technique to compute the parameter \( \alpha_k \)).

Step 3: Compute \( u_k \) is defined by (31). If \( u_k < 0 \) then \( u_k = 0 \)
and if \( u_k > 1 \) then \( u_k = 1 \)

Step 4: Set \( \beta_k = \frac{g_{k+1}^T y_k}{u_k y_k d_k + (1-u_k) \eta_k} \).

Step 5: Compute \( d_{k+1} = -g_{k+1} + \beta_k d_k \).

Step 6: If \( k = n \) go to Step 1 with new values of \( x_{k+1} \) and \( g_{k+1} \).

If not continue.

Theorem (3.1)

Assume that \( f \) is a convex function and \( \alpha_k \) in algorithm (2) and (23), where \( \theta_k \) is given by (31), is determined by the strong Wolfe conditions (4)–(5). If \( 0 < \theta_k < 1 \) then the direction \( d_{k+1} \) given by (23) is a sufficient descent direction.

Proof.

Since \( d_0 = -g_0 \), we have \( g_0^T d_0 \leq -\|g_0\|^2 < 0 \). Assume by induction that
Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems
Journal of Education and Science – Pure Sciences

\[ g_k^T d_k \leq -c \|g_k\|^2 < 0 \text{ where } 0 < c \leq 1 \] ...........(33)

which is a sufficient descent direction. To complete the proof, we have to show that the theorem is true for \( k + 1 \)

\[ g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + \beta_k g_{k+1}^T v_k \]

\[ = -\|g_{k+1}\|^2 + \frac{g_{k+1}^T y_k}{u_k y_k^T v_k + (1-u_k)\eta_k} g_{k+1}^T v_k \] ...........(34)

The second term in (34) can be written

\[ \frac{g_{k+1}^T y_k (g_{k+1}^T v_k)}{u_k y_k^T v_k + (1-u_k)\eta_k} = \frac{(g_{k+1}^T y_k) (u_k y_k^T v_k + (1-u_k)\eta_k) (g_{k+1}^T v_k)}{(u_k y_k^T v_k + (1-u_k)\eta_k)^2} \]

\[ = \left[ \frac{(u_k y_k^T v_k + (1-u_k)\eta_k) g_{k+1}^T (g_{k+1}^T v_k) y_k}{(u_k y_k^T v_k + (1-u_k)\eta_k)^2} \right] \]

\[ \leq \frac{(u_k y_k^T v_k + (1-u_k)\eta_k)^2 \|g_{k+1}\|^2 + (g_{k+1}^T v_k)^2 \|y_k\|^2}{2 (u_k y_k^T v_k + (1-u_k)\eta_k)^2} \] ....(35)

\[ = \frac{1}{2} \|g_{k+1}\|^2 + \frac{(g_{k+1}^T v_k)^2 \|y_k\|^2}{2 (u_k y_k^T v_k + (1-u_k)\eta_k)^2}. \]

Now, using (35) in (34) we get

\[ g_{k+1}^T d_{k+1} \leq -\|g_{k+1}\|^2 + \frac{1}{2} \|g_{k+1}\|^2 + \frac{(g_{k+1}^T v_k)^2 \|y_k\|^2}{2 (u_k y_k^T v_k + (1-u_k)\eta_k)^2} \] ...........(36)

Observe that the last term in (36) tends to zero very fast. Therefore, the direction \( d_{k+1} \) satisfies the sufficient descent condition:

\[ \frac{g_{k+1}^T d_{k+1}}{\|g_{k+1}\|} \leq -c \] ...........(37)

where \( c \) is a positive constant, and \( c \approx 1/2 \).

Dai et al. [4] proved that for any conjugate gradient method with strong Wolfe line search the following general result holds:

Lemma (3.1)
Suppose that the assumptions (i) and (ii) hold and consider any conjugate gradient method (2) and (3), where $d_{k+1}$ is a descent direction and $\alpha_k$ is obtained by the strong Wolfe line search (3) and (4). If
\[
\sum_{k=0}^{\infty} \frac{1}{\|d_{k+1}\|^2} = \infty ,
\]
then
\[
\liminf_{k \to \infty} \|g_{k+1}\| = 0 .
\]

As we know, if $f$ is a uniformly convex functions, then there exists a constant $\mu > 0$ such that
\[
(\nabla f(x) - \nabla f(y))^T (x-y) \geq \mu\|x-y\|^2 , \text{ for any } x, y \in S. \quad \text{(40)}
\]
Equivalently, this can be expressed as
\[
f(x) \geq f(y) + \nabla f(y)^T (x-y) + \frac{\mu}{2}\|x-y\|^2 , \text{ for any } x, y \in S. \quad \text{(41)}
\]
From (40) and (41) it follows that
\[
y_k^T v_k \geq \mu\|v_k\|^2 ,
\]
and
\[
f_k - f_{k+1} \geq -g_{k+1}^T v_k + \frac{\mu}{2}\|v_k\|^2 .
\]
Obviously, from (38) we get :
\[
\mu\|v_k\|^2 \leq y_k^T v_k \leq L\|v_k\|^2 , \text{ i.e. } \mu \leq L .
\]
More details can be found in [2,3].

Using the above relations (42) and (43) we have
\[
\lambda_k y_k^T v_k + (1-\lambda_k)\theta_k = \lambda_k y_k^T v_k + (1-\lambda_k)(2(f_k - f_{k+1}) + (g_{k+1} + g_k)^T v_k) \geq \lambda_k y_k^T v_k + (1-\lambda_k)(2(-g_{k+1}^T v_k + \frac{\mu}{2}\|v_k\|^2) + (g_{k+1} + g_k)^T v_k) \quad \text{(44)}
\]
\[
= \lambda_k y_k^T v_k + (1-\lambda_k)(-2g_{k+1}^T v_k + 2\mu\|v_k\|^2 + g_{k+1}^T v_k + g_k^T v_k) \quad \text{(44)}
\]
\[
= (\lambda_k + \lambda_k -1)y_k^T v_k + (1-\lambda_k)(\mu\|v_k\|^2) \quad \text{(44)}
\]
\[
\lambda_k y_k^T v_k + (1-\lambda_k)\theta_k \geq (2\lambda_k -1)y_k^T v_k + (1-\lambda_k)\frac{\mu}{L} y_k^T v_k \quad \text{(45)}
\]
\[
= (2\lambda_k -1+(1-\lambda_k)\frac{\mu}{L})y_k^T v_k \quad \text{(45)}
\]
Now, if $L \geq \mu$, then the right hand side of (45) is positive, that
Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems

Journal of Education and Science – Pure Sciences

\[ u_k y_k^T v_k + (1-u_k)\eta_k \geq m y_k^T v_k \quad \text{where} \quad m = 2u_k - 1 + (1-u_k) \frac{\mu}{L} \quad \text{.....(46)} \]

**Theorem (3.2)**

Suppose that the assumptions (i) and (ii) hold and descent condition (37) hold. Consider the hybrid CG method in the form of (23) with \( u_k \) defined by (31), where \( \alpha_k \) is computed using the strong Wolfe line search (3) and (4). If the objective function \( f \) is uniformly convex

\[
\lim \inf_{k \rightarrow \infty} \| g_{k+1} \| = 0 .
\]

**Proof:**

Because the descent condition holds, we have \( d_{k+1} \neq 0 \). So, using Lemma 1, it is sufficient to prove that \( \| d_{k+1} \| \) is bounded above. From (23), we have

\[
\| d_{k+1} \| = \left\| -g_{k+1}^T + \frac{g_{k+1}^T y_k}{u_k y_k^T d_k + (1-u_k)\eta_k} v_k \right\| 
\leq \| g_{k+1} \| + \frac{\| g_{k+1}^T y_k \|}{u_k y_k^T d_k + (1-u_k)\eta_k} \| v_k \| 
\text{..........(48)}
\]

Now, from (43) and (46) we have :

\[
\| d_{k+1} \| \leq \| g_{k+1} \| + \frac{L\| v_k \| \| g_{k+1} \|}{mL\| v_k \|} \| v_k \| 
\leq \varepsilon + \frac{\varepsilon L\| v_k \|}{mL\| v_k \|} \| v_k \| 
\leq \varepsilon \left[ 1 + \frac{1}{m} \right] 
\text{..........(49)}
\]

This relation implies

\[
\sum_{k \geq 1} \frac{1}{\| d_{k+1} \|^2} \geq \left[ \frac{\mu}{\mu + L} \right]^2 \frac{1}{\varepsilon^2} \sum_{k \geq 1} 1 = \infty
\]

Therefore, from Lemma (3.1) we have \( \lim \inf_{k \rightarrow \infty} \| g_{k+1} \| = 0 \), which for uniformly convex function is equivalent to \( \lim_{k \rightarrow \infty} \| g_{k+1} \| = 0 \). 66
4. Numerical Results

In this section, we reported some numerical results obtained with the implementation of the new algorithm on a set of unconstrained optimization test problems. We have selected (10) large scale unconstrained optimization problems in extended or generalized form, for each test function, we have considered numerical experiment with the number of variable n=100-1000. Using the standard Wolfe line search conditions (4) and (5) with $\delta_1 = 0.0001$ and $\delta_2 = 0.1$ In the all these cases, the stopping criteria is the $\|g_k\| \leq 10^{-5}$. The programs were written in Fortran 90. The test functions were commonly used for unconstrained test problems with standard starting points and a summary of the results of these test functions was given in Tables (3.1) and (3.2). We tabulate for comparison of these algorithms, the number of function evaluations (NOF) and the number of iterations (NOI).

Table (4.1)

<table>
<thead>
<tr>
<th>No.</th>
<th>n</th>
<th>DY-algorithm</th>
<th>New-algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NOF (NOI)</td>
<td>NOF (NOI)</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>252 (33)</td>
<td>160 (24)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>336 (39)</td>
<td>199 (26)</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>299 (121)</td>
<td>263 (110)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>2492 (1001)</td>
<td>2250 (1001)</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>218 (105)</td>
<td>246 (105)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>2025 (1005)</td>
<td>278 (120)</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>277 (94)</td>
<td>60 (21)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>279 (95)</td>
<td>61 (21)</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>216 (103)</td>
<td>40 (16)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>554 (73)</td>
<td>39 (16)</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>209 (102)</td>
<td>22 (8)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>561 (278)</td>
<td>22 (8)</td>
</tr>
</tbody>
</table>
Conclusions and Discussions

In this paper, we have proposed a new hybrid method for solving unconstrained minimization problems. The computational experiments show that the new approaches given in this paper are successful.

Table (4.2) gives a comparison between the new hybrid -algorithm and the DY-algorithm for convex optimization , this table indicates that the new algorithm saves (49.57)% NOI and (49.16)% NOF, overall against the standard DY-algorithm, especially for our selected test problems.

Relative Efficiency of the Different Methods Discussed in the Paper.
Appendix

1. Cantrell function:
\[
f(x) = \sum_{i=1}^{n/4} \left[ \exp(x_{4i-3}) - x_{4i-2} \right]^4 + 100(x_{4i-2} - x_{4i-1})^6 + [\tan^{-1}(x_{4i-1} - x_{4i})]^4 + x_{4i-3}^8
\]

Starting point: \((1, 2, 2, 2, \ldots, 2)\)^T

2. Miele function:
\[
f(x) = \sum_{i=1}^{n/4} \left[ \exp(x_{4i-3}) - x_{4i-2} \right]^2 + 100(x_{4i-2} - x_{4i-1})^6 + [\tan(x_{4i-1} - x_{4i})]^4 + x_{4i-3}^8 + (x_{4i} - 1)^2
\]

Starting point: \((1, 2, 2, 2, \ldots, 2)\)^T

3. Generalized powell function:
\[
f(x) = \sum_{i=1}^{n/4} (x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-2} - x_{4i})^2 + (x_{4i-1} - 2x_{4i})^2 + 10(x_{4i} - x_{4i-1})^4 + (x_{4i-2} - 2x_{4i-1} - x_{4i})^2
\]

Starting point: \((3, -1, 0, 1, \ldots, 1)\)^T

4. Rosenbrock function:
\[
f(x) = \sum_{i=1}^{n/2} (100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2)
\]

Starting point: \((-1, 2, 1, -1, 2, 1, \ldots)\)^T

5. Cubic function:
\[
f(x) = \sum_{i=1}^{n/2} (100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2)
\]

Starting point: \((-1, 2, 1, -1, 2, 1, \ldots)\)^T

6. Penalty 2 function:
\[
f(x) = e^{(x(i) - i)^2} + (x(i)^2 - 0.25)^2
\]

Starting point: \((1, 2, \ldots)\)^T

7. Non-diagonal function:
\[
f(x) = \sum_{i=1}^{n/2} (100(x_i - x_i^3)^2 + (1 - x_i)^2)
\]

Starting point: \((-1, \ldots)\)^T

8. Welfe function:
\[
f(x) = (-x_1(3-x_1/2) + 2x_2 - 1)^2 + \sum_{i=1}^{n/4} (x_{4i-1} - x_i(3-x_i/2) + 2x_{i+1} - 1)^2 + (x_{i+1} - x_i(3x_i/2 - 1)^2
\]

Starting point: \((-1, \ldots)\)^T

9. Shallow function:
\[
f(x) = \sum_{i=1}^{n/2} ((x_{2i-1}^2 - x_{2i}x_{2i-1})^2 + (1 - x_{2i-1})^2)
\]

Starting point: \((-2, \ldots)\)^T
Development a Hybrid Conjugate Gradient Algorithm for Solving Unconstrained Minimization Problems
Journal of Education and Science – Pure Sciences

10. Sum of Quartics function:

\[ f(x) = \sum_{i=1}^{n} (x_i - 1)^4 \]

Starting point: \((2, \ldots)\)^T

References