An Efficient Line Search Algorithm for Large Scale Optimization

Abbas Y. Al-Bayati
College of Computer Sciences and Mathematics
University of Mosul

Ivan S. Latif
College of Scientific Education
University of Salahaddin

profabbasalbayati@yahoo.com
Received on:16/7/2006
Accepted on:4/10/2006

ABSTRACT

In this work we present a new algorithm of gradient descent type, in which the stepsize is computed by means of simple approximation of the Hessian Matrix to solve nonlinear unconstrained optimization function. The new proposed algorithm considers a new approximation of the Hessian based on the function values and its gradients in two successive points along the iterations one of them use Biggs modified formula to locate the new points. The corresponding algorithm belongs to the same class of superlinear convergent descent algorithms and it has been newly programmed to obtain the numerical results for a selected class of nonlinear test functions with various dimensions. Numerical experiments show that the new choice of the step-length required less computation work and greatly speeded up the convergence of the gradient algorithm especially, for large scaled unconstrained optimization problems.

KEYWORDS: Unconstrained optimization, line search, Biggs Variable Metric Update, gradient descent algorithm.

خوارزمية كفؤة لخط البحث في الأمثلية ذات القياس العالي

عباس يونس البياتي
جامعة صلاح الدين/كلية علوم الحاسوب والرياضيات

إيفان صبحي لطيف
جامعة صلاح الدين/كلية التربية

تاريخ قبول البحث : 4/10/2006

الملخص

في هذا البحث تم استخدام خوارزمية جديدة للدورة المتارقة في الأمثلية ذات القياس العالي خط البحث للخوارزمية الجديدة يمكن الحصول عليه بتقريب مبسط لمصفوفة هسي لحل المسائل غير الخطية في الأمثلية غير المغفدة. الخوارزمية الجديدة تتناول ترتيب جديد لمصفوفة هسي
1. Introduction

This Paper considers the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) ,$$

where the objective function $f$ is a twice continuously differentiable function from $\mathbb{R}^n$ into $\mathbb{R}$. For problem (1) Barzilai and Borwein [1] and Fletcher, R. [7] suggested an algorithm which essentially is a gradient one, where the choice of step-size along the negative gradient is derived from a two point approximation to the Hessian of $f$ at $x_k$ considering $D_k = \gamma_k I$ as an approximation to the Hessian of $f$ at $x_k$; they chose the parameter $\gamma_k$ such that

$$D_k = \text{arg min} \| Dv_k - y_k \|_2 ,$$

where $v_k = x_k - x_{k-1}$ and $y_k = -\nabla f(x_k) - \nabla f(x_{k-1})$, yielding

$$\gamma_k^{BB} = \frac{v_k^T y_k}{y_k^T y_k} \quad \text{...}(3)$$

With this parameter the basic iterations of the (BB) method may be given the following iterative scheme:

$$x_{k+1} = x_k - \frac{1}{\gamma_k^{BB}} \nabla f(x_k) \quad \text{...}(4)$$

Mainly, the sequence $\{x_k\}$ generated by the (BB) method uses two initial vectors $x_0$ and $x_1$. Having in view its simplicity and numerical efficiency for well-conditioned problems, the (BB) method has received a great deal of attention. However, like all steepest descent and conjugate gradient methods, the (BB) method becomes slow when the problems happens to be more ill-conditioned [1]. Neculai Andrei (NA) [10] suggested another gradient descent method for unconstrained optimization (its details
give in section 2 of this paper). In contrast with (BB) method a simple interpretation of the secant equation for the step-length is computed. Here, in this paper, we are going to develop numerical techniques described by Dixon [6] and Biggs [2]. The central notion is the estimation of dominant degree of a one dimensional convex function. Previously their estimate have been quite successfully used as the basis of a line search process indeed it is not performing explicit minimizations along every search direction may require more than one function evaluation per iteration to obtain a satisfactory reduction $f$, but if we have some measure of the non-quadratic function $f$ in the directions then we can attempt to improve upon the simple estimate of the second directional derivative and hence update the matrix $H_k$ using more accurate information.

2. Neculai Andrei (NA) Algorithm:

Neculai Andrei (NA), 2005 suggested a procedure for computing an approximation of the Hessian of the function $f$ at $x_k$ which can be considered to get the step-size along the negative gradient considered the initial point $x_0$ where $f(x_0)$ and $g_0 = \nabla f(x_0)$ can immediately be computed. Using the backtracking procedure (initialized with $\alpha = 1$) he computed the step-length $\alpha_0$ which the next estimate $x_1 = x_0 - \alpha_0 g_0$ is computed, where a gain he computed $f(x_i)$ and $g_1 = \nabla f(x_1).$ So the first step is computed using the backtracking along the negative gradient. So the point

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

$k = 0, 1, 2, ...$ \hspace{1cm} (5)

and

$$f(x_{k+1}) = f(x_k) - \alpha_k g_k^T g_k + \frac{1}{2} \alpha_k^2 g_k^T \nabla^2 f(z) g_k$$

where $z \in [x_k, x_{k+1}]$. Having in view the local character of the searching procedure and that the distance between $x_k$ and $x_{k+1}$ is small enough he choose $z = x_{k+1}$ and $\gamma(x_{k+1}) I$ as an approximation of the $\nabla^2 f(x_{k+1}),$ where $\gamma(x_{k+1}) \in \mathbb{R}.$ This is computed using the local information from point $x_k$, therefore the parameter

$$\gamma(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{\alpha_k} [f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k]$$

was used to compute the next estimation $x_{k+2} = x_{k+1} + \alpha_{k+1} g_{k+1}.$ To determine the step size $\alpha_{k+1},$ he suggested
\( \Phi_{k+1}(\alpha) = f(x_{k+1}) - \alpha g_k^T g_{k+1} + \frac{1}{2} \alpha^2 \gamma(x_{k+1}) g_k^T g_{k+1} \) \ldots (8)

observing that \( \Phi_{k+1}(0) = f(x_{k+1}) \) and \( \Phi'_{k+1}(0) = -g_k^T g_{k+1} < 0 \). Therefore \( \Phi_{k+1}(\alpha) \) is a convex function for all \( \alpha \geq 0 \). To have a minimum for \( \Phi_{k+1}(\alpha) \) the parameter \( \gamma(x_{k+1}) \) must have a positive value. Considering for the moment that \( \gamma(x_{k+1}) > 0 \), then from \( \Phi_{k+1}(\alpha) = 0 \) yields:

\[ \alpha'_{k+1} = \frac{1}{\gamma(x_{k+1})} \] \ldots (9)

As the minimum of \( \Phi_{k+1} \) yields

\[ \Phi'_{k+1}(\alpha'_{k+1}) = f(x_{k+1}) - \frac{1}{2\gamma(x_{k+1})} \| g_{k+1} \|^2 \] \ldots (10)

which show that if \( \gamma(x_{k+1}) > 0 \), then the value of function \( f \) is reduced. If not the algorithm, will restart. This suggests

\[ \alpha_{k+1} = \arg \min_{\alpha \leq \alpha_{k+1}} f(x_{k+1} - \alpha g_{k+1}) \] \ldots (11)

Using the back tracking procedure to complete the algorithm he considered this situation when \( \gamma(x_{k+1}) > 0 \). If \( f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k < 0 \) then the reduction \( f(x_{k+1}) - f(x_k) \) is greater than \( \alpha_k g_k^T g_k \). In this case, the step-size \( \alpha_k \) will be changed as \( \alpha_k + \eta_k \) in such a manner that

\[ f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k)g_k^T g_k > 0 \] \ldots (12)

To get a value for \( \eta_k \) the parameter \( \delta > 0 \) is chosen small enough, and \( \eta_{k+1} \) may be considered as:

\[ \eta_{k+1} = \frac{2}{g_k^T g_k} \left[ f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k \right] + \delta \] \ldots (13)

and a new value for \( \gamma(x_{k+1}) \) can be computed as:

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k (\alpha_k + \eta_k)^2} \left[ f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k)g_k^T g_k \right] \] \ldots (14)

2.1. Neculai Andrei Algorithm (NA):

Corresponding (NA) gradient descent algorithm may be listed as follows:

Step1: select \( x_0 \in \text{dom} f \) and compute \( f(x_0), g_0 = \nabla f(x_0) \) and \( \alpha_0 = \arg \min_{\alpha \leq \alpha_1} f(x_1) \). Compute \( x_1 = x_0 - \alpha_0 g_0 \), \( f(x_1) \) and \( g_1 = \nabla f(x_1) \) set \( k = 0 \).
Step 2: Test for convergence. If $\|g_{k+1}\| < 1 \times 10^{-5}$ then stop; otherwise continue.

Step 3: Compute the (scalar) approximation $\gamma(x_{k+1})$ of the Hessian of function $f$ at $x_{k+1}$ as

$$\gamma(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{\alpha_k^2} [f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k]$$

Step 4: If $\gamma(x_{k+1}) > 0$ then select $\delta > 0$ and compute a new value for $\gamma(x_{k+1})$ as equation (14) where $\eta_k$ is given by equation (13) where $\delta$ is select $\delta > 0$ enough small.

Step 5: Compute the initial step-size as equation (9) with which a backtracking procedure is performed in the next step.

Step 6: Using a backtracking procedure, determine the step-length $\alpha_{k+1}$ as equation (11).

Step 7: Update the variables:

$$x_{k+2} = x_k + \alpha_{k+1} g_{k+1} \quad , \quad d_{k+2} = -g_{k+1} + \eta_k \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} d_{k+1}$$

set $k = k + 1$ and go to step 2.

3. A new proposed Algorithm for solving problem (1)

In this section we are going to suggest another procedure for computing an approximation of the Hessian of the function $f$ at $x_k$ which can be considered to get the step-size along the negative gradient for (NA) algorithm for equation (10) which shows that if $\gamma(x_{k+1}) > 0$ then the value of function $f$ is reduced. This enable us to determine a step-size $\alpha_{k+1}$ as defined in equation (11) using the new way to backtracking procedure. To complete the algorithm we must consider the situation when $\gamma(x_{k+1}) > 0$ if $f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k < 0$ then the reduction $f(x_{k+1}) - f(x_k)$ is greater that $\alpha_k g_k^T g_k$. In this case we use Biggs VM-Update as backtracking procedure to make the step-size $\alpha_k$ as $\alpha_k + \eta_k$ in such a manner that this suggests a value for $\eta_k$, as

$$\eta_k' = \frac{v_k^T y_k}{4v_k^T g_{k+1} + 2v_k^T g_k - 6(f(x_{k+1}) - f(x_k))} \quad \ldots(15)$$
and

\[ \eta_k = \begin{cases} 
\frac{1}{\eta_k'} & \text{if } \eta_k' > 0.5 \\
\eta_k & \text{otherwise} 
\end{cases} \quad \text{...(16)} \]

where \( \gamma_{k+1} = g_{k+1} - g_k \), \( k = 0, 1, 2, \ldots \) and the new value of \( \gamma(x_{k+1}) > 0 \) can be computed as:

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k (\alpha_k + \eta_k)^2} [f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k] \]

### 3.1. The new suggested Algorithm (New):

In order to increase the efficiency of algorithm (NA), cubic line search rule is used to find the best value of the step-size used Biggs VM parameter [2] is used as backtracking procedure in order to locate the new hybrid line search to \( f \) as shown below:

Step1: Select \( x_0 \in \text{dom} f \) and compute \( f_0, g_0 \) and \( \alpha_0 = \arg \min_{\alpha \in \mathbb{R}} (x_0 - \alpha g_0) \).

Now compute \( x_i, f_i \) and \( g_i \) set \( k = 0 \).

Step2: Test for convergence, i.e if \( \| g_{k+1} \| < 1 \times 10^{-5} \) then stop; otherwise continue.

Step3: Compute the (scalar) approximation \( \gamma(x_{k+1}) \) of the Hessian of function \( f \) at \( x_{k+1} \) as:

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k (\alpha_k + \eta_k)^2} [f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k] \]

Step4: If \( \gamma(x_{k+1}) < 0 \) then compute a new value for \( \gamma(x_{k+1}) \) as

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k (\alpha_k + \eta_k)^2} [f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k] \]

Where \( \eta_k \) is given by

\[ \eta_k = \frac{v_i^T y_k}{4v_i^T g_{k+1} + 2v_i^T g_k - 6(f(x_{k+1}) - f(x_k))} \]

Moreover, if \( \eta_k' > 0.5 \) then \( \eta_k \) is given by

\[ \eta_k = \begin{cases} 
\frac{1}{\eta_k'} & \text{if } \eta_k' > 0.5 \\
0 & \text{otherwise} 
\end{cases} \]
If available storage is exceeded then employ a restart option either with $k = N$ or $g_{k+1}^T g_{k+1} > g_{k+1}^T g_k$.

Step 5: Update the variables:

$$x_{k+2} = x_{k+1} + \alpha_{k+1} g_{k+1}, \quad d_{k+2} = -g_{k+1} + \eta_{k+1} d_{k+1}$$

Set $k = k + 1$, go to step 2.

Now theoretically, to ensure that the new algorithm has a super-linear convergence let us consider the following theorems in the next section.

### 3.2. The convergence analysis of the new suggested Algorithm:

In the following section let us consider the convergence analysis of this proposed algorithm. Assume that $f$ is strongly convex and the sublevel set $\{x \in \text{Dom}_f : f(x) \leq f(x_0)\}$ is closed. Strong convexity of $f$ on $S$ involves the existence of the constants $m$ and $M$ such that $mI \leq \nabla^2 f(x) \leq MI$ for all $x \in S$. A consequence of strong convexity of $f$ on $S$ is that we can bound $f^\star$ as

$$f(x) - \frac{1}{2m} \|\nabla f(x)\|_2^2 \leq f^\star \leq f(x) - \frac{1}{2M} \|\nabla f(x)\|_2^2$$

...(17)

For more details see [11],[12].

#### 3.2.1. Theorem

For strongly convex function the new algorithm with backtracking has a superlinear, convergence and

$$f(x_k) - f^\star \leq (\prod_{i=0}^{k-1} C_i)(f(x_0) - f^\star)$$

...(18)

where $0 < \alpha' < 0.5$ and $0 < s < 1$, $C_i = 1 - \min\{2m\alpha', 2m\alpha's^{p_i}\} < 1$ and $p_i \geq 1$ is an integer ($p_i = 1, 2, \ldots$) given by the backtracking procedure.

Proof:

First we can write $f(x_{k+1})$ as:

$$f(x_{k+1}) = f(x_k) - (\alpha - \frac{1}{2} \alpha^2 \gamma(x_{k+1})) \|g_k\|_2^2,$$

...(19)

with $\alpha - \alpha^2 \gamma(x_{k+1}) / 2$ is a concave function, and for all $0 \leq \alpha \leq 1 / \gamma(x_{k+1})$, $\alpha - \alpha^2 \gamma(x_{k+1}) / 2 \geq \alpha / 2$. Hence

$$f(x_{k+1}) \leq f(x_k) - \frac{\alpha}{2} \|g_k\|_2^2 \leq f(x_k) - \alpha' \alpha \|g_k\|_2^2$$

The backtracking procedure terminates either with $\alpha = 1$ or with $\alpha = s^{p_k}$, where $p_k$ is an integer. Therefore
\[ f(x_{k+1}) \leq f(x_k) - \min\{\alpha', \alpha's^p\}\|g_k\|^2 \]

Having in view that for strongly convex function
\[ \|g_k\|^2 \geq 2m(f(x_k) - f^*) \] it follows that where \( C_k = 1 - \min\{2m\alpha, 2m\alpha's^p\} \).
Since \( C_k < 1 \) the sequence \( f(x_k) \) with Biggs VM-parameter has a superlinear convergent, like a geometric series to \( f^* \).

3.2.2. Theorem:

For every \( k = 0, 1, \ldots \), \( \gamma(x_{k+1}) \) generated by the New Algorithm, is bounded away from zero.

Proof:

For every \( k = 0, 1, \ldots \) we know that \( f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k > 0 \) (Since \( \gamma(x_{k+1}) \) generated by the new algorithm)
Therefore \( f(x_k) - f(x_{k+1}) < \alpha_k g_k^T g_k \) and hence we have:
\[
\gamma(x_{k+1}) = \frac{2}{\alpha_k} - \frac{2(f(x_k) - f(x_{k+1}))}{\alpha_k^2 (g_k^T g_k)} > \frac{2}{\alpha_k} - \frac{2\alpha_k (g_k^T g_k)}{\alpha_k^2 (g_k^T g_k)} = 0
\]

4. Numerical results:

In this section we report some numerical results obtained by a newly-programmed FORTRAN. Implementation of the above gradient descent algorithms for 24 test functions with different dimensions (specified in the Appendix)[12].

The comparative performances of the algorithms are taken in the usual way by considering both the total number of function evaluations (NOF) and the total number of iterations (NOI).

In each case the convergence criterion is that the value of \( \|g_{k+1}\| < 1 \times 10^{-5} \) the cubic fitting technique, published in its original from by Bunday [3] is used as the common linear search sub program for:

1. The Original algorithm published by Neculai Andrei (NA).
2. The new proposed algorithm (New)

The numerical results in Table (1) give the comparison between the (New) and (NA) algorithms for different dimensions of test functions. Wile Table (2) gives the percentage of improvements of NOI and NOF. The important thing is that the new algorithm is very robust in many situations especially for large-scale unconstrained optimization problems; When the iterative process reaches the same precision.
Table (1): Comparison between the new algorithms and Neculai Andrei (NA) algorithm. Using different value of \( N \).

| N. OF Test | TEST FUNCTION | Neculai Andrei algorithm (NA) | new algorithm (NEW) |
|------------|---------------|-------------------------------|---------------------|
|            |               | NOI(EOF)                      | NOI(EOF)            |
|            |               | 12  36  360  1050  4320  8640 10000 | 12  36  360  1050  4320  8640 10000 |
| 1          | GEN-Center    | 315  391  531  287  989  873  1118  110 | 68  68  68  68  68  68  68  68 |
| 2          | GEN-Shallow   | 26  26  30  30  30  30  7  7 | 26  26  26  26  26  26  26  26 |
| 3          | GEN-Beal     | 58  58  63  63  63  63  7  7 | 27  27  27  27  27  27  27  27 |
| 4          | GEN-Powell    | 80  80  86  90  90  90  26  26 | 58  61  61  61  68  68  68  68 |
| 5          | GEN-Cubic     | 40  43  43  43  43  43  43  43 | 27  27  27  27  27  27  27  27 |
| 6          | EX-penalty    | 21  21  21  16  18  20  20  20 | 22  19  20  16  18  19  20  20 |
| 7          | Non dquart    | 48  74  84  373  133  134  142  32  32 | 31  22  122  53  29  39  41  41 |
| 8          | EX-himmble    | 20  20  20  20  20  20  20  20 | 7  7  7  7  7  7  7  7 |
| 9          | GEN-Osp       | 112  173  320  547  1007  1714  382  1599  374 | 102  123  228  526  905  1474  327  1597  347 |
| 10         | ETETF         | 21  21  21  21  21  21  18  18 | 18  18  18  18  18  18  18  18 |
| 11         | Digonal6      | 8  8  8  8  8  8  8  8 | 8  8  8  8  8  8  8  8 |
| 12         | GEN-strail    | 17  17  17  17  17  17  17  17 | 17  17  17  17  17  17  17  17 |
| 13         | Full Hesian   | 20  20  20  20  20  20  20  20 | 7  7  7  7  7  7  7  7 |
| 14         | Digonal7      | 6  6  6  6  6  6  6  6 | 6  6  6  6  6  6  6  6 |
| 15         | Digonal8      | 2  2  2  2  2  2  2  2 | 2  2  2  2  2  2  2  2 |
| 16         | sincos        | 22  22  25  25  25  25  25  25 | 16  16  16  16  16  16  16  16 |
| 17         | EX-Denschnb   | 20  20  20  20  20  20  20  20 | 14  14  14  14  14  14  14  14 |
| 18         | GEN-PSC1      | 22  22  25  25  25  25  25  25 | 16  16  16  16  16  16  16  16 |
| 19         | EX-BD1        | 75  62  62  62  62  66  66  66 | 55  55  55  55  55  55  55  55 |
| 20         | Sum           | 49  50  153  199  164  340  373  50  49  95  175  124  161  239 | 10  10  10  10  10  10  10  10 |
| 21         | EX-BD2        | 32  32  32  32  32  32  32  32 | 33  35  30  30  30  30  30  30 |
| 22         | EX-Sum2       | 41  54  1941  11459  362  362  368  48  54  100  134  122  160  299 | 7  7  7  7  7  7  7  7 |
| 23         | GEN-penal1    | 11  16  37  55  50  165  175  70  7  11  11  14  24  50  37  44  44 | 11  11  11  11  11  11  11  11 |
| 24         | GEN_penl2     | 29  85  134  434  201  153  49  31  33  35  60  88  83  91  184 | 9  9  9  9  9  9  9  9 |
| General TOTAL of | | 1100  129  3706  13859  3422  4263  4428  700  172  1058  1465  1768  2433  536  2869 | 1100  129  3706  13859  3422  4263  4428  700  172  1058  1465  1768  2433  536  2869 |
Table (2): Percentage performance of the new proposed (NEW) algorithm against Neculai Andrei (NA) algorithm for 100% in both NOI and NOF.

| N   | Cost 100% | NEW |
|-----|-----------|-----|
| 12  | NOI 36.36 | NOF 33.20 |
| 36  | NOI 45.67 | NOF 40.21 |
| 360 | NOI 71.45 | NOF 36.79 |
| 1080| NOI 89.43 | NOF 40.42 |
| 4320| NOI 48.33 | NOF 39.91 |
| 8640| NOI 42.93 | NOF 38.67 |
| 10000| NOI 35.21 | NOF 20.79 |

It is clear from the numerical results that the new proposed algorithm is very efficient and superior on the standard (NA) algorithm. Namely, there are about (35-89)% improvements of NOI for all dimensions also there are (20-40)% improvement of NOF for all iterations.

5. Conclusions:

In this Paper, a new gradient descent algorithm is proposed in which the step-length is computed by backtracking using a simple approximation of the Hessian based on the function values in two successive points along the iteration using Biggs [2] parameter.

Numerical experiments show that new algorithm converge superlinearly and faster. It is more efficient than Neculai Andrei (NA) algorithm in many situations. The new algorithm is expected to solve ill-conditioned problems and it is clear that any procedure for step-length computation does not change the superlinear convergence property of the new algorithm. The convergence rate depends greatly on the condition number of the Hessian of the minimizing function. For well conditioned convex function both algorithms are doing well, while for ill-conditions problem the new algorithm is doing well. Also, the initial step in backtracking procedure of the new algorithm is lower than the corresponding initial step of (NA) algorithm.

Finally, (NA) has a linear convergence rate while the new algorithm has superlinear rate of convergence.
6. APPENDIX

All the test functions used in this paper are from general literature see[12],[5]:

1. Generalized Central Function:
   \[ f(x) = \sum_{i=1}^{n/3}\left(\exp(x_{3i} - 3) - x_{4i-2}\right)^4 + 100(x_{4i-2} - x_{4i-1})^6 + \left(\arctan(x_{4i-1} - x_{4i})\right)^4 + x_{4i-3}\right), \]
   \[ x_0 = [1..2, 2, 2, ..., 1, 2, 2, 2]. \]
2. Generalized Shallow Function:
   \[ f(x) = \sum_{i=1}^{n/2}(x_{2i-1} - x_{2i})^2 + (1 - x_{2i-1})^2, \]
   \[ x_0 = [-2, -2, ..., -2, -2]. \]
3. Generalized Beale Function:
   \[ f(x) = \sum_{i=1}^{n/3}\left[1.5 - x_{2i} + (1 - x_{2i})\right]^2 + \left[2.25 - x_{2i-1} (1 - x_{2i}^2)\right]^2 + \left[2.625 - x_{2i-1} (1 - x_{2i}^3)\right]^2, \]
   \[ x_0 = [1..0.8, ..., 1..0.8]. \]
4. Generalized Powell function:
   \[ f(x) = \sum_{i=1}^{n/3}\left[3 - \frac{1}{1 + (x_{2i} - x_{3i})} - \sin\left(\frac{\pi x_{3i}}{2}\right) - \exp\left(\frac{(-x_{2i} + x_{3i})^2}{2}\right)\right], \]
   \[ x_0 = [0..1, 2, ..., 0..1, 2]. \]
5. Generalized Cubic function:
   \[ f(x) = \sum_{i=1}^{n/2}\left[100(x_{2i} - x_{3i}^2) + (1 - x_{2i-1})^2\right], \]
   \[ x_0 = [-1.2, 1, ..., -1.2, 1]. \]
6. Extended Penalty Function:
   \[ f(x) = \sum_{i=1}^{n-1}(x_i - 1)^2 + \left(\sum_{j=1}^{n}x_j^2 - 0.25\right)^2, \]
   \[ x_0 = [1, 2, ..., n]. \]
7. Non quart Function (cute):
   \[ f(x) = (x_i - x_j)^2 + \sum_{i=2}^{n-2}(x_i + x_{i+1} + x_n)^4 + (x_{n-1} - x_n)^2, \]
   \[ x_0 = [1, -1, ..., 1, -1]. \]
8. Extended Himmelblau Function:
   \[ f(x) = \sum_{i=1}^{n/2}\left(x_{2i-1}^2 + x_{2i} - 11\right)^2 + \left(x_{2i-1}^2 + x_{2i}^2 - 7\right)^2, \]
   \[ x_0 = [1.1, 1.1, ..., 1.1, 1.1]. \]
9. Generalized OSP (Oren and Spedicato) Function:
\[ f(x) = \sum_{i=1}^{n} ix_i^2 , \]
\[ x_0 = [1, ..., 1]. \]

10. Extended Three Exponential Terms Function:
\[ f(x) = \sum_{i=1}^{n} \left( \exp(x_{2i-1} + 3x_{2i} - 0.1) + \exp(x_{2i-1} - 3x_{2i} - 0.1) + \exp(-x_{2i-1} - 0.1) \right) , \]
\[ x_0 = [0.1, 0.1, ..., 0.1, 0.1]. \]

11. Diagonal 6 Function:
\[ f(x) = \sum_{i=1}^{n} (\exp(x_i) - (1 + x_i)) , \]
\[ x_0 = [1, 1, ..., 1, 1]. \]

12. Generalized Strail Function:
\[ f(x) = \sum_{i=1}^{\lfloor n/2 \rfloor} (x_{2i-1}^2 - x_{2i})^2 + 100(1 - x_{2i-1})^2 , \]
\[ x_0 = [-2, ..., -2]. \]

13. Full Hessian Function:
\[ f(x) = \left( \sum_{i=1}^{n} x_i \right)^2 + \sum_{i=1}^{n} (x_i \exp(x_i) - 2x_i - x_i^2) , \]
\[ x_0 = [1, 1, ..., 1, 1]. \]

14. Diagonal 7 Function:
\[ f(x) = \sum_{i=1}^{n} (\exp(x_i) - 2x_i - x_i^2) , \]
\[ x_0 = [1, 1, ..., 1, 1]. \]

15. Diagonal 8 Function:
\[ f(x) = \sum_{i=1}^{n} x_i \exp(x_i) - 2x_i - x_i^2 , \]
\[ x_0 = [1, 1, ..., 1, 1]. \]

16. SINCOS Function:
\[ f(x) = \sum_{i=2}^{\lfloor n/2 \rfloor} \left( x_{2i-1}^2 + x_{2i}^2 + x_{2i-1}x_{2i} \right)^2 + \sin^2(x_{2i-1}) + \cos^2(x_{2i}) , \]
\[ x_0 = [3, 0.1, ..., 3, 0.1]. \]

17. Extended Denschnb Function:
An Efficient Line Search Algorithm for Large Scale Optimization

\[ f(x) = \sum_{i=1}^{n/2} (x_{2i-1} - 2)^2 + (x_{2i-1} - 2)^2 x_{2i}^2 + (x_{2i} + 1)^2, \]

\[ x_0 = [0.1, 0.1, ... , 0.1, 0.1]. \]

18. Generalized pscl Function:
\[ f(x) = \sum_{i=2}^{n-1} (x_i^2 + x_{i+1}^2 + x_i x_{i+1})^2 + \sin^2(x_i) + \cos^2(x_i), \]

\[ x_0 = [3, 0.1, ..., 3, 0.1]. \]

19. Extended Diagonal BDI Function:
\[ f(x) = i = 1 \sum_{i=2}^{n/2} (x_{2i-1}^2 + x_{2i}^2 - 2)^2 + (\exp(x_{2i-1} - 1) - x_{2i})^2, \]

\[ x_0 = [0.1, 0.1, ... , 0.1, 0.1]. \]

20. Generalized Sum of Quatrics (SUM) Function:
\[ f(x) = \sum_{i=1}^{n} (x_i - i)^4, \]

\[ x_0 = [2, ..., 2]. \]

21. Extended Block-Diagonal BD2 Function:
\[ f(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 + x_{2i}^2 - 2)^2 + (\exp(x_{2i-1} - 1) + x_{2i}^3 - 2)^2, \]

\[ x_0 = [1.5, 2, ..., 1.5, 2]. \]

22. Sum of Quatrics (SUM) Function:
\[ f(x) = \sum_{i=1}^{n} (x_i - i)^4, \]

\[ x_0 = [1, 1, ..., 1]. \]

23. Generalized Penal1 Function:
\[ f(x) = \sum_{i=1}^{n} (x_i - 1)^2 + \text{eps}(x_i^2 - 0.25)^2, \]

\[ x_0 = [1, 2, ..., n], \text{ eps } = 1.0 \times 10^{-5}. \]

24. Generalized Penal2 Function:
\[ f(x) = \sum_{i=1}^{n} \text{eps}(x_i - 1)^2 + (x_i^2 - 0.25)^2, \]

\[ x_0 = [1, 2, ..., n], \text{ eps } = 1.0 \times 10^{-5}. \]
REFERENCES

[1] Barzilii, J. & Borwin, J. "Two point step size gradient method" IMAJ .Numer .Anal.Vol.(8),pp.141-148, (1988).

[2] Biggs,M."A new method of linear minimisation", Numerical Optimisation centre.Tech.Report.No.(1), (1991).

[3] Bunday, B. "Basic Optimization Methods " Edward Arnold. Bedford square, London, UK.(1984).

[4] Cauchy, A. "Methods generates pourla resolution des systemes dequations simultances", C.R.Acad.Sci-par., Vol.25, pp.536-538, (1847).

[5] Daiand,Y. & Liao,L."R-linear convergence of the Barzilai and Borwein gradient method", IMAJ.Numer.Anal, Vol.22, pp.1-10, (2002).

[6] Dixon, L., Biggs, M. "Meander-A Newton based procedure for N-dimensional function minimization", Numerica Optimisation. center.Report .No.(10), (1970).

[7] Fletcher,R. "On the Barzilai-Borwein method", Numerical analysis Report. No.(207), (2001).

[8] Friedlanr,A., Martinez, J. and Raydan, M. "A new method for large-scale box constrained convex quadratic minimization problems", Optimization Method and Software,Vol.57, PP.55-74, (1995).

[9] Gippo, L. and Sciadrone,M. "Nonmonotone globalization techniques for Barzilai-Borwein gradient method", Computational Optimization and Applications, Vol.23, pp.143-169, (2002).

[10] Neculai,A., "A new gradient descent method for unconstrained optimization", Research institute for informatics, 8-10 Averscu Avenu, Bucharest, Romania, (2005).

[11] Nocedal, J. "Updating quasi-Newton matrices with limited strategy", Math.of comput.,Vol.(35), pp.773-782,(1980).

[12] Nocedal, J. "Unconstrained Optimization Test function Research Institute for Informatics", Center for advanced Modeling and Optimization, Bucharest1, Romania. April 28, (2005).
[13] Polka, E. and Ribiere, G. "Note sur la convergence de methods de directions conjugate ", Revue Francaise Ainformatique et Recherche Operational , Vol.35, pp.35-43, (1969).

[14] Raydan, M. "On the Barzilai-Borwein choice of the step length for the gradient method", IMAJ.Numer.Anal., Vol.13, pp.618-622, (1993).