

## 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

Received on:16/7/2006

Accepted on:4/10/2006

### الملخص

في هذا البحث تم استحداث خوارزمية جديدة للتدرج المترافق في الأمثلية ذات القياس العالي خط البحث للخوارزمية الجديدة يمكن الحصول عليه بتقريب مبسط لمصفوفة هسي لحل المسائل غير الخطية في الأمثلية غير المقيدة. الخوارزمية الجديدة تتناول ترتيب جديد لمصفوفة هسي. الفكرة الأساسية تعتمد على قيمة الدالة ومشتقاتها في نقطتين ناجحتين (ذكيّتين) على طول خطوط البحث. إحدى هاتين النقطتين تستخدم صيغة Biggs لأيجاد الخطوات الخوارزمية الجديدة التي تعتبر ضمن خوارزميات التدرج المترافق في حل المسائل غير المقيدة. الخوارزمية الجديدة تمتلك خاصية التقارب فوق الخطي وقد أثبتت النتائج العددية كفاءة الخوارزمية الجديدة وحاجتها الى وقت أقل وسرعتها أكبر في حل المسائل غير المقيدة ولأبعاد مختلفة .

### 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.

## 1. Introduction

This Paper considers the unconstrained minimization problem

$$\min_{x \in \mathfrak{R}^n} f(x) \quad , \quad \dots(1)$$

where the objective function  $f$  is a twice continuously differentiable function from  $\mathfrak{R}^n$  into  $\mathfrak{R}$ . 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  $D_k = g_k I$  as an approximation to the Hessian of  $f$  at  $x_k$ ; they chose the parameter  $g_k$  such that

$$D_k = \arg \min \|Dv_k - y_k\|_2, \quad \dots(2)$$

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

$$g_k^{BB} = \frac{v_k^T y_k}{y_k^T y_k} \quad \dots(3)$$

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

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

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  $a = 1$  he computed the step-length  $a_0$  which the next estimate  $x_1 = x_0 - a_0 g_0$  is computed, where a gain he computed  $f(x_1)$  and  $g_1 = \nabla f(x_1)$ . So the first step is computed using the backtracking along the negative gradient. So the point

$$x_{k+1} = x_k + a_k g_k, \quad k = 0, 1, 2, \dots \quad \dots(5)$$

and

$$f(x_{k+1}) = f(x_k) - a_k g_k^T g_k + \frac{1}{2} a_k^2 g_k^T \nabla^2 f(z) g_k \quad \dots(6)$$

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  $g(x_{k+1})I$  as an approximation of the  $\nabla^2 f(x_{k+1})$ , where  $g(x_{k+1}) \in \Re$ . This is computed using the local information from point  $x_k$ , therefore the parameter

$$g(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{a_k^2} [f(x_{k+1}) - f(x_k) + a_k g_k^T g_k] \quad \dots(7)$$

was used to compute the next estimation  $x_{k+2} = x_{k+1} + a_{k+1} g_{k+1}$ . To determine the step size  $a_{k+1}$ , he suggested

$$\Phi_{k+1}(a) = f(x_{k+1}) - a g_{k+1}^T g_{k+1} + \frac{1}{2} a^2 g(x_{k+1}) g_{k+1}^T g_{k+1} \quad \dots(8)$$

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

$$a'_{k+1} = \frac{1}{g(x_{k+1})} \quad \dots(9)$$

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

$$\Phi'_{k+1}(a'_{k+1}) = f(x_{k+1}) - \frac{1}{2g(x_{k+1})} \|g_{k+1}\|_2^2 \quad \dots(10)$$

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

$$a_{k+1} = \arg \min_{a \leq a'_{k+1}} f(x_{k+1} - a g_{k+1}) \quad \dots(11)$$

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

$$f(x_{k+1}) - f(x_k) + (a_k + h_k) g_k^T g_k > 0 \quad \dots(12)$$

To get a value for  $h_k$  the parameter  $d > 0$  is chosen small enough, and  $h_{k+1}$  may be considered as:

$$h_{k+1} = \frac{2}{g_k^T g_k} [f(x_{k+1}) - f(x_k) + a_k g_k^T g_k] + d \quad \dots(13)$$

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

$$g(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{(a_k + h_k)^2} [f(x_{k+1}) - f(x_k) + (a_k + h_k) g_k^T g_k] \quad \dots(14)$$

### 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  $a_0 = \arg \min_{a < 1} f(x_1)$ . Compute  $x_1 = x_0 - a_0 g_0$ ,  $f(x_1)$  and  $g_1 = \nabla f(x_1)$  set  $k = 0$ .

Step2: Test for convergence. If  $\|g_{k+1}\| < 1 \times 10^{-5}$  then stop; otherwise continue.

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

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

Step4: If  $g(x_{k+1}) > 0$  then select  $d > 0$  and compute a new value for  $g(x_{k+1})$  as equation (14) where  $h_k$  is given by equation (13) where  $d$  is select  $d > 0$  enough small.

Step5: Compute the initial step-size as equation (9) with which a backtracking procedure is performed in the next step

Step6: Using a backtracking procedure, determine the step-length  $a_{k+1}$  as equation (11).

Step7: Update the variables:

$$x_{k+2} = x_{k+1} + a_{k+1} g_{k+1}, \quad d_{k+2} = -g_{k+1} + h_{k+1} \frac{g_{k+1}^T g_{k+1}}{g_{k+1}^T y_{k+1}} d_{k+1}$$

set  $k = k+1$  and go to step2.

### 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  $g(x_{k+1}) > 0$  then the value of function  $f$  is reduced. This enable us to determine a step-size  $a_{k+1}$  as defined in equation(11) using the new way to backtracking procedure. To complete the algorithm we must consider the situation when  $g(x_{k+1}) > 0$  if  $f(x_{k+1}) - f(x_k) + a_k g_k^T g_k < 0$  then the reduction  $f(x_{k+1}) - f(x_k)$  is greater than  $a_k g_k^T g_k$ . In this case we use Biggs VM-Update as backtracking procedure to make the step-size  $a_k$  as  $a_k + h_k$  in such a manner that :this suggests a

$$f(x_{k+1}) - f(x_k) + (a_k + h_k) g_k^T g_k > 0$$

value for  $h_k$ , as

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

and

$$h_k = \begin{cases} \frac{1}{h'_k} & \text{if } h'_k > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad \dots(16)$$

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

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

### 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  $a_0 = \arg \min_{a < 1} (x_0 - a_0 g_0)$ .

Now compute  $x_1, f_1$  and  $g_1$  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  $g(x_{k+1})$  of the Hessian of function  $f$  at  $x_{k+1}$  as :

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

Step4: if  $g(x_{k+1}) < 0$  then compute a new value for  $g(x_{k+1})$  as

$$g(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{(a_k + h_k)^2} [f(x_{k+1}) - f(x_k) + (a_k + h_k) g_k^T g_k] \quad \text{Where } h_k \text{ is given by}$$

$$h'_k = \frac{v_k^T y_k}{4v_k^T g_{k+1} + 2v_k^T g_k - 6(f(x_{k+1}) - f(x_k))}.$$

Moreover, if  $h'_k > 0.5$  then  $h_k$  is given by

$$h_k = \begin{cases} \frac{1}{h'_k} & \text{if } h'_k > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

If available storage is exceeded then employ a restart option either with  $k = N$  or  $g_{k+1}^T g_{k+1} > g_{k+1}^T g_k$ ,

Step5: Update the variables:

$$x_{k+2} = x_{k+1} + a_{k+1} g_{k+1}, \quad d_{k+2} = -g_{k+1} + h_{k+1} d_{k+1}$$

Set  $k = k + 1$ , go to step2

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 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  $f^*$  as

$$f(x) - \frac{1}{2m} \|\nabla f(x)\|_2^2 \leq f^* \leq f(x) - \frac{1}{2M} \|\nabla f(x)\|_2^2 \quad \dots(17)$$

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

### 3.2.1. Theorem

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

$$f(x_k) - f^* \leq \left(\prod_{i=0}^{k-1} C_i\right) (f(x_0) - f^*) \quad \dots(18)$$

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

Proof:

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

$$f(x_{k+1}) = f(x_k) - \left(a - \frac{1}{2} a^2 g(x_{k+1})\right) \|g_k\|_2^2, \quad \dots(19)$$

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

$a - a^2 g(x_{k+1})/2 \geq a/2$ . Hence

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

The backtracking procedure terminates either with  $a = 1$  or with  $a = s^{p_k}$ , where  $p_k$  is an integer.

Therefore

$$f(x_{k+1}) \leq f(x_k) - \min\{a', a's^{p_k}\} \|g_k\|_2^2$$

Having in view that for strongly convex function

$\|g_k\|_2^2 \geq 2m(f(x_k) - f^*)$  it follows that where  $C_k = 1 - \min\{2ma, 2ma's^{p_k}\}$ .

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  $k = 0, 1, \dots$ ,  $g(x_{k+1})$  generated by the New Algorithm, is bounded away from zero.

Proof:

For every  $k=0,1,\dots$  we know that  $f(x_{k+1}) - f(x_k) + a_k g_k^T g_k > 0$  (Since  $g(x_{k+1})$  generated by the new algorithm)

Therefore  $f(x_k) - f(x_{k+1}) < a_k g_k^T g_k$  and hence we have:

$$g(x_{k+1}) = \frac{2}{a_k} - \frac{2(f(x_k) - f(x_{k+1}))}{a_k^2 (g_k^T g_k)} > \frac{2}{a_k} - \frac{2a_k (g_k^T g_k)}{a_k^2 (g_k^T g_k)^2} = 0$$

#### 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. While 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.<br>OF<br>Test | TEST<br>FUNCTION | Neculai Andrei algorithm (NA)<br>NOI(NOF) |           |           |           |            |           |             | new algorithm (NEW) NOI(NOF) |          |          |          |          |          |          |
|------------------|------------------|-------------------------------------------|-----------|-----------|-----------|------------|-----------|-------------|------------------------------|----------|----------|----------|----------|----------|----------|
|                  |                  | 12                                        | 36        | 360       | 1080      | 4320       | 8640      | 10000       | 12                           | 36       | 360      | 1080     | 4320     | 8640     | 10000    |
| 1                | GEN-Center       | 315<br>44                                 | 391<br>51 | 531<br>62 | 287<br>42 | 989<br>102 | 873<br>94 | 1118<br>110 | 68<br>13                     | 68<br>13 | 68<br>13 | 82<br>14 | 82<br>14 | 97<br>15 | 97<br>15 |
| 2                | GEN-<br>Shallow  | 26<br>7                                   | 26<br>7   | 30<br>8   | 30<br>8   | 30<br>8    | 30<br>8   | 30<br>8     | 26<br>7                      | 26<br>7  | 26<br>7  | 26<br>7  | 26<br>7  | 26<br>7  | 26<br>7  |
| 3                | GEN-Beal         | 58<br>17                                  | 58<br>17  | 63<br>18  | 63<br>18  | 63<br>18   | 63<br>18  | 63<br>18    | 27<br>7                      | 27<br>7  | 27<br>7  | 27<br>7  | 27<br>7  | 27<br>7  | 27<br>7  |
| 4                | GEN-Powell       | 80<br>23                                  | 80<br>23  | 86<br>25  | 90<br>26  | 90<br>26   | 90<br>26  | 90<br>26    | 58<br>16                     | 61<br>17 | 61<br>17 | 61<br>17 | 68<br>19 | 68<br>19 | 68<br>19 |
| 5                | GEN-Cubic        | 40<br>10                                  | 43<br>11  | 43<br>11  | 43<br>11  | 43<br>11   | 43<br>11  | 43<br>11    | 27<br>6                      | 27<br>6  | 27<br>6  | 27<br>6  | 27<br>6  | 27<br>6  | 27<br>6  |

*An Efficient Line Search Algorithm for Large Scale Optimization*

|                               |              |             |             |             |              |             |             |             |            |            |             |             |             |             |             |
|-------------------------------|--------------|-------------|-------------|-------------|--------------|-------------|-------------|-------------|------------|------------|-------------|-------------|-------------|-------------|-------------|
| 6                             | EX-penalty   | 21<br>6     | 22<br>6     | 21<br>5     | 16<br>4      | 18<br>4     | 20<br>4     | 20<br>4     | 22<br>6    | 19<br>5    | 20<br>5     | 16<br>4     | 18<br>4     | 19<br>4     | 20<br>4     |
| 7                             | Non dquart   | 48<br>13    | 74<br>21    | 84<br>21    | 373<br>30    | 133<br>28   | 134<br>31   | 142<br>32   | 31<br>8    | 22<br>6    | 122<br>31   | 53<br>15    | 29<br>8     | 39<br>10    | 41<br>10    |
| 8                             | EX-himmbale  | 20<br>6     | 20<br>6     | 20<br>6     | 20<br>6      | 20<br>6     | 20<br>6     | 20<br>6     | 7<br>2     | 7<br>2     | 7<br>2      | 7<br>2      | 7<br>2      | 7<br>2      | 7<br>2      |
| 9                             | GEN-Osp      | 112<br>17   | 173<br>29   | 320<br>70   | 547<br>129   | 1007<br>227 | 1714<br>382 | 1599<br>374 | 102<br>16  | 123<br>26  | 228<br>54   | 526<br>128  | 905<br>207  | 1474<br>327 | 1597<br>347 |
| 10                            | ETETF        | 21<br>6     | 21<br>6     | 21<br>6     | 21<br>6      | 21<br>6     | 21<br>6     | 21<br>6     | 18<br>5    | 18<br>5    | 18<br>5     | 18<br>5     | 18<br>5     | 18<br>5     | 18<br>5     |
| 11                            | Digonal6     | 8<br>2      | 8<br>2      | 8<br>2      | 8<br>2       | 8<br>2      | 8<br>2      | 8<br>2      | 8<br>2     | 8<br>2     | 8<br>2      | 8<br>2      | 8<br>2      | 8<br>2      | 8<br>2      |
| 12                            | GEN-strail   | 17<br>5     | 17<br>5     | 17<br>5     | 17<br>5      | 17<br>5     | 17<br>5     | 17<br>5     | 17<br>5    | 17<br>5    | 17<br>5     | 17<br>5     | 17<br>5     | 17<br>5     | 17<br>5     |
| 13                            | Full Hessian | 20<br>6     | 20<br>6     | 20<br>6     | 20<br>6      | 20<br>6     | 20<br>6     | 20<br>6     | 7<br>2     | 7<br>2     | 7<br>2      | 7<br>2      | 7<br>2      | 7<br>2      | 7<br>2      |
| 14                            | Digonal7     | 7<br>2      | 7<br>2      | 7<br>2      | 7<br>2       | 16<br>3     | 16<br>3     | 16<br>3     | 6<br>2     | 6<br>2     | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2      |
| 15                            | Digonal8     | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2       | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2     | 6<br>2     | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2      | 6<br>2      |
| 16                            | sincos       | 22<br>7     | 22<br>7     | 25<br>8     | 25<br>8      | 25<br>8     | 25<br>8     | 25<br>8     | 16<br>5    | 16<br>5    | 16<br>5     | 16<br>5     | 16<br>5     | 16<br>5     | 16<br>5     |
| 17                            | EX-Denschnb  | 20<br>6     | 20<br>6     | 20<br>6     | 20<br>6      | 20<br>6     | 20<br>6     | 20<br>6     | 14<br>4    | 14<br>4    | 14<br>4     | 14<br>4     | 14<br>4     | 14<br>4     | 14<br>4     |
| 18                            | GEN-PSC1     | 22<br>7     | 22<br>7     | 25<br>8     | 25<br>8      | 25<br>8     | 25<br>8     | 25<br>8     | 16<br>5    | 16<br>5    | 16<br>5     | 16<br>5     | 16<br>5     | 16<br>5     | 16<br>5     |
| 19                            | EX-BD1       | 75<br>23    | 62<br>19    | 62<br>19    | 62<br>19     | 62<br>19    | 66<br>20    | 66<br>20    | 55<br>16   | 55<br>16   | 55<br>16    | 55<br>16    | 55<br>16    | 55<br>16    | 55<br>16    |
| 20                            | Sum          | 49<br>7     | 50<br>8     | 153<br>25   | 199<br>35    | 164<br>23   | 340<br>66   | 373<br>71   | 50<br>7    | 49<br>8    | 95<br>18    | 175<br>27   | 124<br>19   | 161<br>24   | 239<br>45   |
| 21                            | EX-BD2       | 32<br>10    | 32<br>10    | 32<br>10    | 32<br>10     | 32<br>10    | 32<br>10    | 32<br>10    | 27<br>8    | 27<br>8    | 30<br>9     | 30<br>9     | 30<br>9     | 30<br>9     | 30<br>9     |
| 22                            | EX-Sum2      | 41<br>7     | 54<br>8     | 1941<br>27  | 11459<br>36  | 362<br>68   | 362<br>68   | 368<br>70   | 48<br>7    | 54<br>8    | 100<br>18   | 134<br>24   | 122<br>19   | 160<br>24   | 299<br>50   |
| 23                            | GEN-penal1   | 11<br>3     | 16<br>4     | 37<br>9     | 55<br>13     | 50<br>13    | 165<br>35   | 175<br>37   | 11<br>3    | 14<br>3    | 24<br>6     | 50<br>12    | 57<br>14    | 44<br>11    | 44<br>11    |
| 24                            | GEN_penl2    | 29<br>8     | 85<br>28    | 134<br>44   | 434<br>142   | 201<br>65   | 153<br>49   | 131<br>41   | 33<br>9    | 35<br>10   | 60<br>15    | 88<br>22    | 83<br>22    | 91<br>23    | 184<br>41   |
| General TOTAL of 24 functions |              | 1100<br>244 | 1329<br>291 | 3706<br>405 | 13859<br>574 | 3422<br>674 | 4263<br>874 | 4428<br>784 | 700<br>163 | 722<br>174 | 1058<br>256 | 1465<br>342 | 1768<br>405 | 2433<br>536 | 2869<br>621 |

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<br>NOF | 89.43<br>40.42 |
| 4320  | NOI<br>NOF | 48.33<br>39.91 |
| 8640  | NOI<br>NOF | 42.93<br>38.67 |
| 10000 | NOI<br>NOF | 35.21<br>20.79 |

It is clear from the two above Tables of the numerical results that the new proposed algorithm is very efficient and superior on the standard (NA) algorithm. Namely there are about (35-89)% improves 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 Cantral Function:

$$f(x) = \sum_{i=1}^{n/4} [(\exp(x_{4i} - 3) - x_{4i-2})^4 + 100(x_{4i-2} - x_{4i-1})^6 + (\arctan(x_{4i-1} - x_{4i}))^4 + x_{4i-3}],$$

$$x_0 = [1., 2., 2., 2., \dots, 1., 2., 2., 2.].$$

2. Generalized Shallow Function:

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

$$x_0 = [-2., -2., \dots, -2., -2.].$$

3. Generalized Beale Function:

$$f(x) = \sum_{i=1}^{n/2} [1.5 - x_{2i} + (1 - x_{2i})]^2 + [2.25 - x_{2i-1}(1 - x_{2i}^2)]^2 + [2.625 - x_{2i-1}(1 - x_{2i}^3)]^2,$$

$$x_0 = [1., 0.8., \dots, 1., 0.8].$$

4. Generalized Powell function:

$$f(x) = \sum_{i=1}^{n/3} \{3 - [\frac{1}{1+(x_i - x_{2i})^2}] - \sin(\frac{px_{2i}x_{3i}}{2}) - \exp[-(\frac{x_i + x_{3i}}{x_{2i}} - 2)^2]\},$$

$$x_0 = [0., 1., 2., \dots, 0., 1., 2.].$$

5. Generalized Cubic function:

$$f(x) = \sum_{i=1}^{n/2} [100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2],$$

$$x_0 = [-1.2, 1., \dots, -1.2, 1.].$$

6. Extended Penalty Function:

$$f(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 + (\sum_{j=1}^n x_j^2 - 0.25)^2,$$

$$x_0 = [1., 2., \dots, n].$$

7. Non dquart Function (cute):

$$f(x) = (x_1 - x_2)^2 + \sum_{i=2}^{n-2} (x_i + x_{i+1} + x_n)^4 + (x_{n-1} - x_n)^2,$$

$$x_0 = [1., -1., \dots, 1., -1.].$$

8. Extended Himmelblau Function:

$$f(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 + x_{2i} - 11)^2 + (x_{2i-1} + x_{2i}^2 - 7)^2,$$

$$x_0 = [1.1, 1.1, \dots, 1.1, 1.1].$$

9. Generalized OSP (Oren and Spedicato) Function:

$$f(x) = \left[ \sum_{i=1}^n ix_i^2 \right]^2,$$

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

10. Extended Three Exponential Terms Function:

$$f(x) = \sum_{i=1}^{n/2} (\exp(x_{2i-1} + 3x_{2i} - 0.1) + \exp(x_{2i-1} - 3x_{2i} - 0.1) + \exp(-x_{2i-1} - 0.1)),$$

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

11. Diagonal 6 Function:

$$f(x) = \sum_{i=1}^n (\exp(x_i) - (1 + x_i)),$$

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

12. Generalized Strail Function:

$$f(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 - x_{2i})^2 + 100(1 - x_{2i-1})^2,$$

$$x_0 = [-2., \dots, -2.].$$

13. Full Hessian Function:

$$f(x) = \left( \sum_{i=1}^n x_i \right)^2 + \sum_{i=1}^n (x_i \exp(x_i) - 2x_i - x_i^2),$$

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

14. Diagonal 7 Function:

$$f(x) = \sum_{i=1}^n (\exp(x_i) - 2x_i - x_i^2),$$

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

15. Diagonal 8 Function:

$$f(x) = \sum_{i=1}^n x_i \exp(x_i) - 2x_i - x_i^2,$$

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

16. SINCOS Function:

$$f(x) = \sum_{i=2}^{n/2} (x_{2i-1}^2 + x_{2i}^2 + x_{2i-1}x_{2i})^2 + \sin^2(x_{2i-1}) + \cos^2(x_{2i}),$$

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

17. Extended Denschnb Function :

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

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

18. Generalized pscl Function:

$$f(x) = \sum_{i=2}^{n-1} (x_i^2 + x_{i+1}^2 + x_i x_{i+1})^2 + \sin^2(x_i) + \cos^2(x_i) ,$$

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

19. Extended Diagonal BDI Function:

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

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

20. Generalized Sum of Quatrics (SUM) Function:

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

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

21. Extended Block-Diagonal BD2 Function:

$$f(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 + x_{2i}^2 - 2.)^2 + (\exp(x_{2i-1} - 1.) + x_{2i}^3 - 2.)^2 ,$$

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

22. Sum of Quatrics (SUM) Function:

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

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

23. Generalized Penal1 Function:

$$f(x) = \sum_{i=1}^n (x_i - 1)^2 + \text{eps}(x_i^2 - 0.25)^2 ,$$

$$x_0 = [1., 2., \dots, n] , \text{ eps} = 1.E-5 .$$

24. Generalized Penal2 Function:

$$f(x) = \sum_{i=1}^n \text{eps}(x_i - 1)^2 + (x_i^2 - 0.25)^2 ,$$

$$x_0 = [1., 2., \dots, n] , \text{ eps} = 1.E-5 .$$

**REFERENCES**

- [1] Barzilai, J. & Borwein, 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] Dai and,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.(9), (1970).
- [7] Fletcher,R. "**On the Barzilai-Borwein method**", Numerical analysis Report. No.(207), (2001).
- [8] Friedlandr,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 d'Informatique et Recherche Operationnelle, 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).