Modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm for unconstrained optimization problem

Muna M. M. Ali
Department of Mathematics, College of Computers Sciences and Mathematics, Mosul University, Iraq

ABSTRACT
The use of the self-scaling Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is very efficient for the resolution of large-scale optimization problems, in this paper, we present a new algorithm and modified the self-scaling BFGS algorithm. Also, based on noticeable non-monotone line search properties, we discovered and employed a new non-monotone idea. Thereafter first, an updated formula is exhorted to the convergent Hessian matrix and we have achieved the secant condition, second, we established the global convergence properties of the algorithm under some mild conditions and the objective function is not convexity hypothesis. A promising behavior is achieved and the numerical results are also reported of the new algorithm.

Keywords: BFGS algorithm, global convergence property, Nonmonotone line search, Self-scaling, Unconstrained optimization

1. INTRODUCTION
Consider the unconstrained optimization problem:

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

(1)

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, to solve problem (1) one uses an algorithm that generates a sequence of iterates $x_k$ according to:

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

(2)

for $k \geq 0$, where $d_k$ is a search direction, $\alpha_k > 0$ is step length and $x_0$ is given the initial point. Basic steps in these algorithms are choosing suitable direction and timely step size. To satisfy the descent condition $\nabla f(x_k)^T d_k \leq 0$, generally, in order to securities a sufficient reduction to value of function we required the search direction $d_k$ and $\alpha_k$ is specified, there are various examples for procedures to choose the search direction $d_k$, conjugate gradient (CG), steepest descent (SD), Newton, quasi-Newton, and trust-region methods see [1]. Newton has the highest rate of convergence and the direction is accounted by solving the system $G_k d_k = -g_k$ where $G_k = \nabla^2 f(x_k)$ and $g_k = \nabla f(x_k)$.

Quasi-Newton criterion methods convention the following secant equation: $B_{k+1} s_k = y_k$ where $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$, at the first iteration, $B_0$ is an arbitrary nonsingular positive definite
matrix and $B_{k+1}$ is an approximation of $G_k$. The most efficient of Quas-Newton methods are perhaps to self-scaling BFGS method which was updated suggested by [2], [3] and this method is overall numerical computation than the other method. The matrix $B_{k+1}$ in the self-scaling BFGS method can be updated by the following formula:

$$B_{k+1} = B_k - \frac{b_k s_k s_k^T b_k}{s_k^T b_k s_k} \mu_k + \frac{y_k y_k^T}{s_k^T y_k} \delta_k$$

(3)

where:

$$\mu_k = \frac{s_k^T y_k}{y_k^T y_k} \delta_k$$

(4)

If the curvature condition $s_k^T y_k > 0$ holds, the method of self-scaling BFGS maintains the positiveness of the matrices $\{B_k\}$. For this reason, the descent direction of $f$ at $x_k$ is satisfy in the direction of the self-scaling BFGS not problem if $G_k$ is positive definite or not. Many modifications have been proposed made to afflicted the global convergence property of the (Broyden-Fletcher-Goldfarb-Shanno) BFGS method, for instance, some modulations in the criterion BFGS method are made, and submitted a modified BFGS (MBFGS) algorithms [4]-[6]. The superlinear convergence and the global of their methods have been proved under appropriate conditions for non-convex problems.

A sufficient reduction produces from suitable line search is another making a good iterative process in function value, as we say. A public situation to accept a step length mentioned Armijo rule as (5):

$$f(x_k + \alpha_k d_k) \leq f_k + \sigma \alpha_k g_k^T$$

(5)

and the largest member $\alpha_k$ in $\{1, \rho, \rho^2, ..., \}$ satisfying (4) such that $\rho \in (0,1)$ and $\sigma \in (0,1)$.

It is clear that $f_k$ denotes $f(x_k)$ and $f_{k+1} < f_k$ for every descent direction, and called monotone line search. The first non-monotone line search technique were proposed by [7], Newton's method using the Armijo condition was defined by (6):

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq m(k)} \{f_{k-j}\} + \sigma \alpha_k g_k^T d_k$$

(6)

where $0 \leq m(k) \leq \min \{m(k + 1) + 1, N\}$. $N$ is a non-negative integer constant, Many kinds of researchers, for example [8]-[12]. A non-monotone schema can promote of finding a global optimum and also developed a speed of convergence. One of the efficient non-monotone line search methods have been proposed by [13] to overcome some drawbacks in the non-monotone in (6) though have features and well work for many situations [14], and have the same general planner while the statement "max" is substitute average weights for values of function with sequential iterations.

2. MODIFIED A NEW NON-MONOTONE SELF-SCALING BFGS METHOD

A non-monotone BFGS methods were proposed for solving (1) in [15]-[17]. These algorithms were proved the convergence analysis under the convex hypothesis on the objective function. In this work, a new non-monotone modified self-scaling BFGS method is inserted and evidence the global convergence of the method without convexity assumption. This work is arranged as follows. The New1 non-monotone proposed and defined in line search (7)-(9) and we note that the numerical results of the New1 non-monotone line search (7)-(9) have been more effective than the [18]. The New2 method is expressed in this part. Also, we remember the properties convergence of the new algorithm in part 3. Numerical experiences show that the new method is very favorable and investigated both theatrically and numerically against some well-known algorithms. In the last part, some conclusions are list.

Now we explain the new non-monotone line search method (New1) which is described as follows:

$$f(x_k + \alpha_k d_k) - \max_{0 \leq j \leq m(k)} \{f_{k-j}\} \leq E_k - \sigma t_k \|g_k\|^2$$

(7)

where:

$$E_k = \delta_k t_k \delta_k^T g_k$$

(8)
Modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm for ...

(Muna M. M. Ali)

\[ t_k = \frac{\delta_{k-1}}{\delta_{k-1}} \delta_{k-1} y_{k-1} \]

\[ \delta_1 = 0.0001, \quad k \geq 1, \quad \text{with } \sigma \in (0,1) \]

Two reasons made the BFGS algorithm had important disadvantages despite this method is a successful algorithm for unconstrained nonlinear optimization. Once, the directions of the method may not be descent especially when \( s_k^T y_k > 0 \) is not satisfied and cannot guarantee positive definiteness of the matrix \( B_k \). Second, in general issues, The BFGS method may not be convergent for non-convex objective functions, despite established superlinear convergence and the global for convex problems.

A New non-monotone modified self-scaling BFGS algorithm is presented guaranteeing the positive definiteness of the matrix \( B_k \) for non-convex objective functions. In this part, the new method is inserted after describing some inspiration. We defined the modified secant equations:

\[ B_{k+1} s_k = y_k^* \]

where:

\[ y_k^* \triangleq y_k + u_k^* s_k \]

and defined by three forms:

\[ u_k^{*(1)} = 2 \frac{\|y_k^*\|^2}{s_k^T y_k^*} \]

\[ u_k^{*(2)} = 1 + 2 \frac{\|y_k^*\|^2}{y_k^* s_k} \]

\[ u_k^{*(3)} = \|g_k\|^\beta + \max \left\{ \frac{\|y_k^*\|^2}{y_k^* s_k}, 0 \right\} \geq 0 \]

where \( \beta \) is a positive constant, see [19], [20]. Then we have reformed the self-scaling BFGS update formula based on (10) as follows:

\[ B_{k+1} = \left[ B_k - \frac{\theta_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] \mu_k s_k + \frac{y_k^* y_k^T}{s_k^T y_k^*} \]

where:

\[ \mu_k^* = \frac{s_k^T y_k^*}{y_k^* y_k^T} \]

and defined an efficient algorithm that is called modified self-scaling BFGS. It is clear that (17).

\[ \|g_k\|^\beta \|y_k^*\|^2 \geq \|y_k^* s_k \| \quad \text{for all } k \in N \]

This property is guarantees positive definiteness of the matrix \( B_k \) and separate on the convexity of f, as such the used line search. The new MBFGS method combined with the new non-monotone line search and satisfies the global convergence. For unconstrained optimization in which \( B_k \) is updated in [21], proposed the relation:

\[ B_{k+1} = B_k - \frac{\theta_k s_k s_k^T g_k}{s_k^T g_k s_k} + \tilde{u}_k \frac{y_k^*(y_k^T)}{s_k^T y_k^*} \]

and:

\[ \tilde{u}_k = 2 \frac{f_k - f_{k+1} + s_k^T g_{k+1}}{s_k^T y_k} \]

so, the local super linear convergence and global properties for convex objective functions preserves in this algorithm too.
Now, the New2 algorithm is suggested which the self-scaling BFGS method update formula using $y_k^*$ in (11), and compute the update formula as follows:

$$B_{k+1} = \left[ B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} \right] \mu_k^* + \frac{y_k^* y_k'^T}{s_k^T y_k'^T}$$

(20)

where:

$$\mu_k^* = s_k^T y_k'^T / y_k'^T y_k'^T$$

(21)

and $B_k$ satisfies the secant condition as follows:

$$B_{k+1} s_k = \mu_k^* y_k^*$$

(22)

outline of the new non-monotone self-scaling MBFGS described in Algorithm 1.

Algorithm 1. New-self-scaling BFGS (new-non-monotone modified self-scaling BFGS)
A start an initial point $x_0 \in \mathbb{R}^n$, a symmetric positive definite matrix $B_0 \in \mathbb{R}^{n \times n}$, $\rho, \sigma \in (0,1)$.

Step 1: set $\delta_k = 0.0001, k = 1$
Step 2: if $\|g_k\| < \varepsilon$, stop
Step 3: compute search direction $d_k$ by solving $B_k d_k = -g_k$
Step 4: set $t_k = \frac{\delta_k}{s_{k+1}^T s_{k+1}}$ where $j/k$ the smallest positive integer and $t_k$ satisfies (7), (8), (9)
Step 5: compute $x_{k+1} = x_k + \alpha t d_k$
Step 6: compute $y_k^*$ in (11) and $\mu_k^*$ in (21). then, update $B_k$ in (20)
Step 7: set $k = k + 1$ and go to step 1.

3. CONVERGENCE ANALYSIS

For the general nonlinear objective function, this part is to explain and prove the properties of the new algorithm. And the following assumptions on the objective function ($f$).

3.1. Assumption (H)

The level set $S = \{x: x \in \mathbb{R}^n, f(x) \leq f(x_1)\}$ is bounded, where $x_1$ is the starting point. In a neighborhood $\Omega$ of $S$, $f$ is continuously differentiable and its gradient $g$ is Lipchitz continuously, namely, there exists a constant $L \geq 0$ such that $\|g(x) - g(x_k)\| \leq L \|x - x_k\|$, $\forall x, x_k \in \Omega$. It is clear that from the assumption (H, i), there exists a positive constant $D$ such that $D = \max \{\|x - x_k\| : \forall x, x_k \in S\}$.

3.2. Some related properties

Some proven mathematical properties to completing the stability study of the theoretical side. Property (1). Let $\{x_k\}$ is the sequence generated by Algorithm 1 new-non-monotone self-scaling MBFGS, then $\{E_k\}$ is a non-increasing sequence and for all $k \in \mathbb{N} \cup \{0\}, \{x_k\} \subset S(x_0)$. Proof: See [22]. Property (2). If the assumptions (H, i) and (H, ii) are contented and $\{x_k\}$ is the sequence produced by the new Algorithm 1 (new-self-scaling MBFGS). If $\|g_k\| \geq \zeta$ holds for all $k \in \mathbb{N}$ with a constant $\zeta > 0$ then there exist positive constants $\theta_1, \theta_2, \theta_3$ such that, for all $k \in \mathbb{N}$, the inequalities:

$$\|B_k s_i\| \leq \theta_1 \|s_i\|, \|B_k s_i\| \leq \theta_2 \|s_i\|^2 \leq s_i^T B_k s_i \leq \theta_3 \|s_i\|^2$$

(23)

contract for fully a half of the indices $i \in \{1, 2, ..., k\}$.

Proof: To prove that, must offer that there subsist two positive $r$ and $R$ such that:

$$\frac{y_k^T s_k}{\|s_k\|^2} \geq r$$

(24)

and

$$\frac{\|y_k^\prime\|^2}{y_k^\prime s_k} \leq R$$

(25)

From assumption $\|g_k\| \geq \zeta$ and from (17) we have:
\[ y_k^T s_k \geq \|g_k\| \|y_k^*\|^2 \geq \zeta^2 \|y_k^*\|^2 \geq \zeta^2 y^* \|s_k\|^2 \]  \hspace{1cm} (26)

so: \[ \frac{y_k^T s_k}{\|s_k\|^2} \geq r, \] where \( r = \zeta^2 y^* \) is a positive constant. On the other hand, it follows (11), (12) and Cauchy-Schwarz inequality that:

\[ \|y_k^*\| \leq \|y_k\| + \|s_k\| (\|g_k\|^2 + \|y_k\| s_k) \]

and from assumptions (H, i), (H, ii) and the relation in corollary (3.3) there exists \( \bar{R} > 0 \) such that \( \|g_k\| \leq \bar{R} \).

Therefore, it can be seen that:

\[ \|y_k^*\| \leq \|s_k\| (L + \bar{R}^\mu + L) = C \|s_k\| \]  \hspace{1cm} (27)

L is Lipchitz constant from a hypothesis (H, ii), and \( C = L + \bar{R}^\mu + L \). The relation (26) along with (27) for all \( k \in N \), result:

\[ \frac{\|y_k^*\|^2}{y_k^T s_k} \leq R \]

where: \( R = \frac{C^2}{\bar{R}} \). From (24), (25), and theorem (2.1) in [6] we have the rest of the proof.

Property (3). If the assumption (H, i) and (H, ii) exist and \( \{x_k\} \) is the sequence generated by the New1 algorithm. If \( \|g_k\| \geq \zeta \) holds for all \( k \in N \) with a constant \( \zeta > 0 \) then there is a positive constant \( t \) such that \( \tau_k > t \) for all \( k \) belonging to \( J = \{k \in N \text{hold (16)}\} \). Proof: see [22].

Property (4). Suppose that the assumption (H, i) and (H, ii) hold, then:

\[ \sum_{k=0}^{\infty} -\alpha_k g_k^T d_k < \infty \]  \hspace{1cm} (28)

Proof: Using (7), (8), (9) we have:

\[ f_{k+1} - f_k = \sigma \alpha_k g_k^T d_k = -\sigma \alpha_k (\|g_k\|^2 + 1 + \|y_k\|^2) (s_k^T d_k)^2 \leq 0 \]  \hspace{1cm} (29)

therefore, \( \{f_k\} \) is a decreasing sequence. Since \( f \) is bounded below, there exists a constant \( \bar{f} \) such that:

\[ \lim_{k \to \infty} f_k = \bar{f} \]  \hspace{1cm} (30)

3.3. **Theorem**

If the assumption (H, i) and (H, ii) exist and \( \{x_k\} \) is the sequence generated by the New Algorithm 1 (self-scaling NBFGS), then:

\[ \lim_{k \to \infty} f \|g_k\| = 0. \]  \hspace{1cm} (30)

Proof: If we assume that \( \lim_{k \to \infty} f \|g_k\| \neq 0 \), so there exists a constant \( \zeta > 0 \) such that \( \|g_k\| \geq \zeta \). For all \( k \) sufficiently, since \( B_k s_k = \alpha_k B_k d_k = -\alpha_k g_k \), it follows from (28) that:

\[ \sum_{k=0}^{\infty} \frac{1}{\alpha_k} s_k^T B_k s_k = \sum_{k=0}^{\infty} (-\alpha_k g_k^T d_k) < \infty \cdot \|g_k\| \geq \zeta \]. From the property (3) definition of \( J \) is hold, leads us to:

\[ \sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \|g_k\|^2 \geq \zeta^2 \sum_{k=0}^{\infty} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \]

\[ \geq \zeta^2 \sum_{k \in J} \alpha_k \frac{s_k^T B_k s_k}{\|B_k s_k\|^2} \]

from the last inequality which comes from property (4) this leads to:

**Modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm for ...** (Muna M. M. Ali)
\[
\sum_{k \in J} s_k^T B_k s_k < \infty
\]

because the set J is infinite, it is lead to that \(s_k^T B_k s_k \to 0\) for \(k \in J\). This immediately contradicts the fact:

\[
s_k^T B_k s_k \geq \frac{\sigma_k^2 \|s_k\|^2}{\sigma_k^2 \|s_k\|^2} = \frac{\sigma_k^2}{\sigma_k^2}
\]

that is in (31).

4. RESULTS AND DISCUSSION

The main work of this section is to compare the numerical experiments of the New1 non-monotone modified MBFGS algorithm with the MBFGS-XG algorithm proposed by [23]. We present a new algorithm in which the new non-monotone line search to approximate comparison is named (New1 NMBFGS). On the other hand, we compare the numerical experiments of the new self-scaling modified BFGS algorithm named (New2 self-scaling MBFGS) with the standard self-scaling BFGS method straight with Armijo line search [7], [9]. We wrote FORTRAN language and double-precision arithmetic. These results were performed on a PC. Our attempts were performed onset of (50) nonlinear unconstrained problems that have a second derivative available, and the experience problems are contributed in CUTE [24], [25].

We considered numerical experiments with several variable \(n = 2, 4, 6, ..., 1000\). All these methods terminate when the following stopping criterion is met \(\|g_k\|_\infty \leq 10^{-5}\). Our experiences show the parameters \(\rho = 0.46, \sigma = 0.38, \delta_2 = 0.0001\), have the best conclusions for all the algorithms. Tables 1 and 2 compare some numerical experiments for the New1, New2 of algorithms against the BFGS algorithms, and the test problems with different dimensions, \(n = 2, 4, ..., 1000\). In all these tables: \(N = \) Dimension of the problem, NOI = number of iterations, NOF = Number of functions, CPU = Total time required to complete the evaluation process for each test problem.

Figures 1 to 4 compare of the New1 method against MBFGS-XG method due to NOI and it’s clear that New1 have more than 37.89\%, and 66.71\% NOI, and New2 against self-scaling BFGS due to NOI and New2 have more than 44.18\% and 70.76\% NOI respectively. Also Figures 2 and 5 compares the New1 against MBFGS-XG method 38.28\% and 44.27\% due to NOF, and New2 against self-scaling BFGS due to NOF and it’s better than 42.29\% and 69.2\% respectively. Figures 3 and 6 compares of the New1 method against MBFGS-XG and have better results in comparison 70.7\% and 71\%, and New2 against self-scaling BFGS 70.41\% and 70.7\% due to CPU [26]-[28].

Table 1. Comparison of the New1 method against MBFGS-XG and New2 against self-scaling BFGS method with \(n = 2, 4, ..., 100\)

<table>
<thead>
<tr>
<th>Prob.</th>
<th>MBFGS-XG method</th>
<th>New1 Method</th>
<th>New2 Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOI</td>
<td>NOF</td>
<td>CPU</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>35</td>
<td>2.2</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>19</td>
<td>0.02</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>64</td>
<td>0.3</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>35</td>
<td>0.19</td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>26</td>
<td>0.45</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>28</td>
<td>1.91</td>
</tr>
<tr>
<td>7</td>
<td>39</td>
<td>55</td>
<td>2.5</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>25</td>
<td>1.2</td>
</tr>
<tr>
<td>9</td>
<td>80</td>
<td>113</td>
<td>3.1</td>
</tr>
<tr>
<td>10</td>
<td>25</td>
<td>60</td>
<td>2.5</td>
</tr>
<tr>
<td>11</td>
<td>1310</td>
<td>8600</td>
<td>0.3</td>
</tr>
<tr>
<td>12</td>
<td>101</td>
<td>621</td>
<td>0.32</td>
</tr>
<tr>
<td>13</td>
<td>80</td>
<td>502</td>
<td>0.1</td>
</tr>
<tr>
<td>14</td>
<td>800</td>
<td>3001</td>
<td>1.6</td>
</tr>
<tr>
<td>15</td>
<td>291</td>
<td>1100</td>
<td>0.91</td>
</tr>
<tr>
<td>16</td>
<td>1372</td>
<td>8911</td>
<td>2.9</td>
</tr>
<tr>
<td>17</td>
<td>1050</td>
<td>7000</td>
<td>0.04</td>
</tr>
<tr>
<td>18</td>
<td>180</td>
<td>181</td>
<td>0.41</td>
</tr>
<tr>
<td>19</td>
<td>1340</td>
<td>9101</td>
<td>2.91</td>
</tr>
<tr>
<td>20</td>
<td>1311</td>
<td>8500</td>
<td>0.015</td>
</tr>
<tr>
<td>21</td>
<td>589</td>
<td>479</td>
<td>0.12</td>
</tr>
<tr>
<td>22</td>
<td>220</td>
<td>1601</td>
<td>0.639</td>
</tr>
<tr>
<td>23</td>
<td>180</td>
<td>891</td>
<td>0.46</td>
</tr>
<tr>
<td>24</td>
<td>299</td>
<td>1297</td>
<td>0.15</td>
</tr>
<tr>
<td>25</td>
<td>470</td>
<td>1992</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Total: 9834 | 54237 | 25244 | 3514 | 18488 | 16114 | 5661 | 33242 | 21814 | 1796 | 8825 | 15659
Table 2. Comparison of the New1 method against MBFGS-XG and New2 against self-scaling BFGS method with $n = 110, ..., 1000$

<table>
<thead>
<tr>
<th>Prob.</th>
<th>MBFGS-XG method NOI</th>
<th>NOF</th>
<th>CPU</th>
<th>NOI</th>
<th>NOF</th>
<th>Self-scaling BFGS method NOI</th>
<th>NOF</th>
<th>CPU</th>
<th>NOI</th>
<th>NOF</th>
<th>New2 Method NOI</th>
<th>NOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>181</td>
<td>391</td>
<td>1.19</td>
<td>47</td>
<td>200</td>
<td>0.1</td>
<td>50</td>
<td>241</td>
<td>1.19</td>
<td>21</td>
<td>149</td>
<td>0.1</td>
</tr>
<tr>
<td>27</td>
<td>200</td>
<td>500</td>
<td>1.19</td>
<td>65</td>
<td>210</td>
<td>0.1</td>
<td>68</td>
<td>250</td>
<td>1.19</td>
<td>51</td>
<td>190</td>
<td>0.1</td>
</tr>
<tr>
<td>28</td>
<td>245</td>
<td>821</td>
<td>0.46</td>
<td>37</td>
<td>126</td>
<td>0.46</td>
<td>28</td>
<td>105</td>
<td>0.46</td>
<td>19</td>
<td>73</td>
<td>0.46</td>
</tr>
<tr>
<td>29</td>
<td>151</td>
<td>391</td>
<td>0.05</td>
<td>10</td>
<td>59</td>
<td>0.05</td>
<td>13</td>
<td>61</td>
<td>0.05</td>
<td>7</td>
<td>50</td>
<td>0.05</td>
</tr>
<tr>
<td>30</td>
<td>681</td>
<td>1501</td>
<td>0.21</td>
<td>520</td>
<td>1101</td>
<td>0.21</td>
<td>579</td>
<td>1391</td>
<td>0.21</td>
<td>434</td>
<td>1031</td>
<td>0.21</td>
</tr>
<tr>
<td>31</td>
<td>780</td>
<td>8001</td>
<td>0.0</td>
<td>121</td>
<td>781</td>
<td>0.1</td>
<td>124</td>
<td>690</td>
<td>0.0</td>
<td>112</td>
<td>583</td>
<td>0.1</td>
</tr>
<tr>
<td>32</td>
<td>981</td>
<td>6351</td>
<td>0.25</td>
<td>392</td>
<td>4381</td>
<td>0.25</td>
<td>428</td>
<td>4221</td>
<td>0.15</td>
<td>219</td>
<td>4001</td>
<td>0.25</td>
</tr>
<tr>
<td>33</td>
<td>100</td>
<td>520</td>
<td>1.24</td>
<td>15</td>
<td>40</td>
<td>0.91</td>
<td>15</td>
<td>40</td>
<td>1.9</td>
<td>15</td>
<td>40</td>
<td>0.91</td>
</tr>
<tr>
<td>34</td>
<td>180</td>
<td>690</td>
<td>0.0</td>
<td>20</td>
<td>69</td>
<td>0.0</td>
<td>29</td>
<td>119</td>
<td>0.0</td>
<td>18</td>
<td>59</td>
<td>0.0</td>
</tr>
<tr>
<td>35</td>
<td>1050</td>
<td>7211</td>
<td>1.1</td>
<td>189</td>
<td>1481</td>
<td>1.19</td>
<td>190</td>
<td>1591</td>
<td>1.1</td>
<td>99</td>
<td>790</td>
<td>1.23</td>
</tr>
<tr>
<td>36</td>
<td>520</td>
<td>1300</td>
<td>0.9</td>
<td>79</td>
<td>245</td>
<td>1.31</td>
<td>43</td>
<td>159</td>
<td>0.5</td>
<td>25</td>
<td>80</td>
<td>1.51</td>
</tr>
<tr>
<td>37</td>
<td>325</td>
<td>981</td>
<td>0.05</td>
<td>56</td>
<td>105</td>
<td>0.01</td>
<td>28</td>
<td>160</td>
<td>0.01</td>
<td>9</td>
<td>49</td>
<td>0.01</td>
</tr>
<tr>
<td>38</td>
<td>260</td>
<td>1231</td>
<td>0.35</td>
<td>117</td>
<td>963</td>
<td>0.35</td>
<td>187</td>
<td>1141</td>
<td>0.15</td>
<td>133</td>
<td>1121</td>
<td>0.35</td>
</tr>
<tr>
<td>39</td>
<td>17</td>
<td>18</td>
<td>0.15</td>
<td>17</td>
<td>18</td>
<td>0.18</td>
<td>17</td>
<td>18</td>
<td>0.18</td>
<td>17</td>
<td>18</td>
<td>0.18</td>
</tr>
<tr>
<td>40</td>
<td>19</td>
<td>31</td>
<td>0.01</td>
<td>19</td>
<td>31</td>
<td>0.01</td>
<td>19</td>
<td>31</td>
<td>0.01</td>
<td>19</td>
<td>31</td>
<td>0.01</td>
</tr>
<tr>
<td>41</td>
<td>85</td>
<td>791</td>
<td>0.9</td>
<td>60</td>
<td>572</td>
<td>0.95</td>
<td>71</td>
<td>613</td>
<td>0.01</td>
<td>50</td>
<td>495</td>
<td>0.95</td>
</tr>
<tr>
<td>42</td>
<td>35</td>
<td>69</td>
<td>0.8</td>
<td>35</td>
<td>77</td>
<td>0.0</td>
<td>32</td>
<td>64</td>
<td>0.02</td>
<td>35</td>
<td>77</td>
<td>0.0</td>
</tr>
<tr>
<td>43</td>
<td>35</td>
<td>78</td>
<td>1.9</td>
<td>20</td>
<td>70</td>
<td>1.9</td>
<td>30</td>
<td>70</td>
<td>1.9</td>
<td>15</td>
<td>60</td>
<td>1.9</td>
</tr>
<tr>
<td>44</td>
<td>172</td>
<td>4852</td>
<td>0.9</td>
<td>60</td>
<td>840</td>
<td>0.0</td>
<td>121</td>
<td>1960</td>
<td>0.5</td>
<td>45</td>
<td>840</td>
<td>0.0</td>
</tr>
<tr>
<td>45</td>
<td>101</td>
<td>200</td>
<td>0.63</td>
<td>139</td>
<td>215</td>
<td>1.5</td>
<td>101</td>
<td>200</td>
<td>0.61</td>
<td>139</td>
<td>215</td>
<td>1.5</td>
</tr>
<tr>
<td>46</td>
<td>19</td>
<td>30</td>
<td>1.5</td>
<td>11</td>
<td>27</td>
<td>1.6</td>
<td>16</td>
<td>27</td>
<td>0.39</td>
<td>10</td>
<td>20</td>
<td>1.6</td>
</tr>
<tr>
<td>47</td>
<td>176</td>
<td>1981</td>
<td>0.01</td>
<td>142</td>
<td>1210</td>
<td>0.01</td>
<td>170</td>
<td>1900</td>
<td>0.01</td>
<td>89</td>
<td>982</td>
<td>0.01</td>
</tr>
<tr>
<td>48</td>
<td>69</td>
<td>121</td>
<td>0.81</td>
<td>41</td>
<td>121</td>
<td>0.6</td>
<td>50</td>
<td>129</td>
<td>0.05</td>
<td>39</td>
<td>117</td>
<td>0.6</td>
</tr>
<tr>
<td>49</td>
<td>37</td>
<td>77</td>
<td>0.15</td>
<td>21</td>
<td>70</td>
<td>0.12</td>
<td>30</td>
<td>70</td>
<td>0.01</td>
<td>21</td>
<td>67</td>
<td>0.12</td>
</tr>
<tr>
<td>50</td>
<td>30</td>
<td>50</td>
<td>0.01</td>
<td>29</td>
<td>50</td>
<td>0.01</td>
<td>25</td>
<td>44</td>
<td>0.01</td>
<td>26</td>
<td>49</td>
<td>0.01</td>
</tr>
<tr>
<td>Total</td>
<td>6389</td>
<td>38167</td>
<td>14.76</td>
<td>2262</td>
<td>13062</td>
<td>11.92</td>
<td>2464</td>
<td>15295</td>
<td>10.58</td>
<td>1667</td>
<td>11187</td>
<td>12.16</td>
</tr>
</tbody>
</table>

Figure 1. Performance due to NOI NOF CPU

Figure 2. Performance due to NOI NOF CPU

Figure 3. Performance due to NOI NOF CPU

Figure 4. Performance due to NOI NOF CPU

*Modified limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm for ... (Muna M. M. Ali)*
5. CONCLUSION

In this paper, we have proposed a new non-monotone BFGS algorithm and combined it with a new modified self-scaling BFGS update to a sacrificial Hessian matrix with a known line search planning for non-convex optimization problems. It is clear that a new non-monotone can progress the probability of finding a global optimum and also promote speed of convergence especially in presence of a narrow-curved valley and sufficient descent property of algorithm convergence. Thus, in our algorithms, we are enjoyable to get benefits from their properties. Lastly, our numerical results show that our new algorithms have competitive with the standard self-scaling BFGS method and have robust numerical results as compared to the non-monotone (self-scaling BFGS) algorithm had proposed.

ACKNOWLEDGEMENTS

The research is supported by the College of Computer Sciences and Mathematics, University of Mosul, Republic of Iraq. The author declares that there are no conflicts of interest regarding this work.

REFERENCES


**BIOGRAPHY OF AUTHOR**

Muna M. M. Ali, Teaching in the Department of Mathematics, College of Computers Sciences and Mathematics, Mosul University, Al-Majmooa Street, Mosul, Iraq. I completed my PhD in Numerical optimization. I have 12 national and international published joint and single research papers. Email: munamoh74@uomosul.edu.iq