A New Globally Convergent Self-Scaling Vm Algorithm for Convex and Nonconvex Optimization

Abbas Y. AL-Bayati *Basim A. Hassan*
Department of mathematics
College of Computers Sciences and Mathematics
University of Mosul

Received: 2009/3/4, Accepted: 2010/5/10

Abstract

In unconstrained optimization, the original quasi-Newton condition $H_{k+1}y_k = v_k$ where y_k is the difference of the gradients at two successive iterations. Li and Fukushima proposed a modified BFGS methods based on a new Quasi –Newton equation $H_{k+1}y_k^* = v_k$ where $y_k^* = y_k + m_k v_k$, where m_k is a small positive constant .In this paper, we first propose the modified version of self-scaling VM-algorithm which was based on Li and Fukushima Quasi–Newton equation, i.e $H_{k+1}y_k^* = \rho_k^* v_k$ where $\rho_k^* = \dfrac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}$. The corresponding AL-Bayati type algorithm is proved to possess the global convergence property in both convex and non-convex optimization problems. Experimental results indicate that the new proposed algorithm was more efficient than the standard BFGS- algorithm.

Introduction

Consider the unconstrained optimization problem:

$$\min\left\{ f(x) \;\middle|\; x \in R^n \right\} \qquad ...(1)$$

Where f is a continuously differentiable function of n variables .Quasi-Newton methods for solving (1) often needs the new search direction d_{k+1} at each iteration given by:

$$d_k = -H_k g_k \qquad ...(2)$$

Where $g_{k+1} = \nabla f(x_{k+1})$ the gradient of is f evaluated at the current iteration x_{k+1} (Storey & HU., 1993). One then computes the next iteration by the formula

$$x_{k+1} = x_k + \alpha_k d_k \qquad ...(3)$$

Where the step size α_k satisfies the Wolfe – conditions

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \qquad ...(4)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \qquad ...(5)$$

Where $0 < \delta_1 < 1/2$ and $\delta_1 < \delta_2 < 1$, and H_{k+1} is an approximation to $\{\nabla^2 f(x_k)\}^{-1}$. The matrix H_{k+1} satisfies the actual quasi-Newton condition

$$H_{k+1} y_k = v_k \qquad ...(6)$$

Where $y_k = g_{k+1} - g_k$, $v_k = x_{k+1} - x_k$, and $B_k = H_k^{-1}$ see **(**Al-Bayati & Al-Salih, 2008).

The standard Broyden - Fletcher-Goldfarb - Shanno (BFGS) update can be separated into two components, $H^{(1)}$ *and* $H^{(2)}$ so that:

$$H_{BFSG} = H^{(1)} + H^{(2)} \qquad ...(7)$$

Where

$$H^{(1)} = H_k - \frac{H_k y_k v_k^T + v_k y_k^T H_k}{y_k^T H_k y_k} + \frac{v_k v_k^T}{v_k^T y_k}\left[\frac{y_k^T H_k y_k}{v_k^T y_k}\right] , \quad H^{(2)} = \frac{v_k v_k^T}{v_k^T y_k}$$

AL-Bayati in (Al-Bayati, 1991) modified the BFGS formula by writing:

$$H_{AL-Bayati} = H^{(1)} + \rho_k H^{(2)} , \quad \rho_k = \frac{y_k^T H_k y_k}{v_k^T y_k} \qquad ...(8)$$

Which will satisfy the quasi-Newton like condition

$$H_{k+1} y_k = \rho_k v_k \qquad ...(9)$$

This relaxation of the Quasi-Newton (QN)-condition is of particular interest in deriving algorithms for non – quadratic objective functions.

Li and Fukushima proposed another Quasi–Newton (QN) like condition (Li. & Fukushima, 2001) defined by:

$$H_{k+1} y_k^* = v_k \qquad ...(10)$$

By using (10), so the modified BFGS update can be writhen as:

$$H_{MBFGS} = H_k - \frac{H_k y_K^* v_k^T + v_k y_k^{*T} H_k}{y_k^{*T} H_k y_k^*} + \frac{v_k v_k^T}{v_k^T y_k^*}\left[1 + \frac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}\right] \qquad ...(11)$$

Where

$$y_k^* = y_k + m_k v_k \qquad ...(12)$$

And m_k is a small positive constant .We chooses m_k to be a very small number, for example $m \le 10^{-6}$. In (Kinsella, 2006) the BFGS update which can be written in the product form:

$$H_{k+1} = (I - \vartheta_k v_k y_k^T) H_k (I - \vartheta_k y_k v_k^T) + \vartheta_k v_k v_k^T \qquad ...(13)$$

Where

$$\vartheta_k = \frac{1}{v_k^T y_k} \qquad ...(14)$$

A modified self-scaling VM-algorithm

In this section, we will deal with an algorithm which generates the sequence of H_{k+1} matrices that converge to the $\{\nabla^2 f(x_k)\}^{-1} = G^{-1}$ and satisfy the following modified Quasi –Newton equation:

$$H_{k+1}y_k^* = \rho_k^* v_k \qquad ...(15)$$

The latter possibility can be eliminated by Newton's method with line search in which the Newton correction is used to generate a direction by:

$$d_k = -G^{-1}g_k \qquad ...(16)$$

In large group of methods, G^{-1} is to be approximated from information collected in the ($k+1$) th stage, or

$$G^{-1} \cong \rho_k^* H_{k+1} \qquad ...(17)$$

$$G^{-1} \cong \rho_k^*(H_k + \Delta H_k) \qquad ...(18)$$

Where

$$H_{k+1} = H_k + \Delta H_k \qquad \Rightarrow \Delta H_k = H_{k+1} - H_k \qquad ...(19)$$

$$\Delta H_k y_k^* = H_{k+1}y_k^* - H_k y_k^* \qquad ...(20)$$

$$\Delta H_k y_k^* = \rho_k^* v_k - H_k y_k^* \qquad ....(21)$$

$$\Delta H_k = \rho_k^* \frac{v_k z_1^T}{z_1^T y_k^*} - \frac{H_k y_k^* z_2^T}{z_2^T y_k^*} \qquad ...(22)$$

Where z_1 *and* z_2 are arbitrary $n*1$ vectors. Choose the following values:

$$z_1 = z_2 = v_k \qquad ...(23)$$

To get:

$$\Delta H_k = \rho_k^* \frac{v_k v_k^T}{v_k^T y_k^*} - \frac{H_k y_k^* v_k^T}{v_k^T y_k^*} \qquad ...(24)$$

$$H_{k+1}^{new} = H_k + \frac{v_k v_k^T}{v_k^T y_k^*}\left[\rho_k^* + \frac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}\right] - \frac{H_k y_k^* v_k^T + v_k H_k y_k^{*T}}{v_k^T y_k^*} \qquad ...(25)$$

Where $\rho_k^* = \dfrac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}$, y_k^* is a vector defined in (12).

The modified AL-Bayati self-scaling VM-update can be expressed in the following product form:

$$H_{k+1} = (I - \vartheta_k^* v_k y_k^{*T}) H_k (I - \vartheta_k^* y_k^* v_k^T) + \rho_k^* \vartheta_k^* v_k v_k^T \qquad ...(26)$$

Where

$$\rho_k^* = \frac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*} \quad \text{and} \quad \vartheta_k^* = \frac{1}{v_k^T y_k^*} \qquad ...(27)$$

Global convergence property of the new algorithm

The important property of the line-search method is the global convergence defined by the relation:

$$\lim_{k\Rightarrow\infty} \inf \|g_k\| = 0 \qquad ...(28)$$

1. Convex case:

To prove that the new updates (25) generate identical conjugate gradient search directions if the function is quadratic and the exact line searches are used, let us consider the following property:

Property (3.1.1): Let f be given by

$$f(x) = \frac{1}{2} x^T G x + b^T x \qquad ...(29)$$

Where G is a symmetric positive definite matrix. Choose an initial approximation $H_1 = H$ where H is any symmetric positive definite matrix. Obtain H_{new} from H where $d = -Hg$ is the search direction and assuming exact lime searches then

$$H_{k+1} g^* = H g^* \qquad ...(30)$$

Proof: see (Al-Bayati, 1991).

Powell (1983) showed that if the level set $\{x{:}\ f(x) \le f(x_*)\}$ is bounded when α_k is defined so that $\{\, g_{k+1}^T d_k = 0 \,,\, k \ge 1\}$ holds for all k and if $f(x)$ is twice continuously differentiable than conjugate gradient-method achieves the limit:

$$\lim_{k\Rightarrow\infty} \inf \|g_k\| = 0 \qquad ...(31)$$

The following lemma (3.1.1) is given in (Li. & Fukushima, 2001)

Lemma (3.1.1)

Suppose that $(\alpha_k, x_{k+1}, y_{k+1}, d_{k+1})$ generated by the MBFGS algorithm and that G is continuous at x^*.Then we have

$$\lim_{k\Rightarrow\infty} \| m_k v_k \| = 0 \qquad ...(32)$$

Theorem (3.1.1)

Assume that $f(x)$ be a quadratic function defined in (29) and that the line searches are exact: if H is any symmetric positive definite matrix and for the new updating formula,

$$H_{k+1} = H_k - \frac{H_k y_k^* v_k^T + v_k y_k^{*T} H_k}{v_k^T y_k^*} + \frac{v_k v_k^T}{v_k^T y_k^*}\left[\rho_k^* + \frac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}\right] \qquad ...(33)$$

Where $\rho_k = \dfrac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}$

Then the search direction

$$d_{new} = -H_{new} g_{k+1} \qquad ...(34)$$

is identical to the conjugate gradient (CG) direction [H/S direction]; d_{CG} defined by

$$d_{CG} = -g_{k+1} + \frac{y^{*T} g_{k+1}}{y^{*T} d} d \quad \text{for } k \geq 1 \qquad ...(35)$$

Proof:

$$H_{k+1} = H_k - \frac{H_k y_k^* v_k^T}{v^T y_k^*} - \frac{v_k y_k^{*T} H_k}{v^T y_k^*} + \frac{v_k v_k^T}{v_k^T y_k^*}\left[\rho_k + \frac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}\right]$$

Now

$$d_{new} = -H_k g_{k+1} + \frac{H_k y_k^* v_k^T g_{k+1}}{v_k^T y_k^*} + \frac{y_k^{*T} H_k g_{k+1}}{v_k^T y_k^*} v_k - 2\frac{y_k^{*T} H_k y_k^* v_k^T g_{k+1}}{(v_k^T y_k^*)^2} v_k \qquad ...(36)$$

$$= -H_k g_{k+1} + \frac{y_k^{*T} H_k g_{k+1}}{v_k^T y_k^*} v_k \qquad ...(37)$$

Using the property $v_k^T g_{k+1} = 0$, quoted earlier which holds for line searches. The vector g_{k+1} can be substituted for $H_k g_{k+1}$ by using property (3.1.1). Therefore:

$$d_{new} = -g_{k+1} + \frac{y_k^{*T} g_{k+1}}{v_{k\,new}^T y_k^*} v_{k\,new} \qquad ...(38)$$

New by using lemma (3.1.1) we have

$$d_{new} = -g_{k+1} + \frac{y_k^T g_{k+1}}{v_{k\,new}^T y_k} v_{k\,new} \qquad ...(39)$$

It is also known that d_{BFGS} and d_{CG} are identical (Nazareth, 1979), and d_{new} is identical to d_{BFGS} with exact line searches, hence equation (38) becomes

$$d_{new} = -g_{k+1} + \frac{y_k^T g_{k+1}}{d_{kCG}^T y_k} d_{kCG} \qquad ...(40)$$

In the following section we will list outline of the new proposed self-scaling VM algorithm.

2. New Algorithm:

The outline of the modified algorithm is:

Step 0: Choose an initial point $x_1 \in R^n$, set $k = 1$.

Step 1: If the hybrid stopping criterion is satisfied stop:

ITERM=1- if $|x_{k+1} - x_k|$ was less than or equal to $1.0D-16$,

ITERM=2- if $|f_{k+1} - f_k|$ was less than or equal to $1.0D-14$,

ITERM=3- if f_{k+1} is less than or equal to $1.0D-16$,

ITERM= 4- if $\|g_k\|$ is less than or equal to $1.0D-6$,

ITERM= 6 if termination criterion was not satisfied, but

Solution is probably acceptable,
ITERM=12- if NOF exceeded 1000,
ITERM <0 - if the method failed.

Step 2: Compute the search direction $d_k = -H_k g_k$.

Step 3: Find a step size α_k which satisfy the rule (4) and (5).

Step 4: Generate a new iteration point by $x_{k+1} = x_k + \alpha_k d_k$ and calculate The new updating formula (26) and (27).

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

3. Non-convex case:

To guarantee the global convergence property of the line-search method consists in turning the search direction towards the negative gradient when necessary, i.e. when $\frac{-g_k^T d_k}{\|g_k\|\|d_k\|} > 0$ is not satisfied. This idea is realized if (2) is replaced by the formula $d_k = -\bar{H}_k g$ with $\bar{H}_k = H + \sigma\|H_k g_k\| I$, where $\sigma > 0$ is a small number (Luksan & Spedicato, 2000)

Theorem (3.3.1): Let the objective function $f(x)$ be bounded from below and has bounded up to at least second-order derivatives .Consider the line-search method satisfying (4) and (5).If

$$\sum_{k=1}^{\infty} \frac{\left(g_k^T d_k\right)^2}{g_k^T g_k d_k^T d_k} \to \infty \quad \text{and} \quad g_k^T d_k < 0 \qquad \text{...(41)}$$

Then (28) holds **(**Luksan & Vlcek, 2005**).**

Theorem (3.3.2)

Let the objective function $f : R^n \to R$ be bounded from below and has bounded up to at least second-order derivatives .Consider the line-search method satisfying (4) and (5) with

$$d_k = -H_k g_k - \sigma\|H_k g_k\| g_k \qquad \text{...(42)}$$

Where H_k is symmetric positive and $k \geq 0$. If $\sigma \geq 0$ then (28) holds.

Proof: Assume, for contradiction purposes, that (31) does not hold. Then **we** can suppose

$$\|g_k\| \geq \varepsilon_*$$

For some $\varepsilon_* > 0$ and $g_k H g_k > 0$, $k \geq 1$, by positive definiteness of H_k .

We have from (42) by the Schwarz inequality

$$d^T d \leq \|Hg\|^2 + 2\sigma\|Hg\|^2\|g\| + \sigma^2\|Hg\|^2\|g\|^2 = (1+\sigma\|g\|\ )^2\|Hg\|^2 \qquad \text{...(43)}$$

$$-g^T d > \sigma\|Hg\|\ \|g\|^2 \qquad \text{...(44)}$$

$$\frac{-g^T d_k}{\|g_k\|\ \|d_k\|} > \frac{\sigma\|Hg\|\ \|g\|^2}{(1+\sigma\|g\|\ )\|Hg\|\ \|g_k\|} \qquad \text{...(45)}$$

$$\frac{\sigma\|Hg\|\,\|g\|^2}{(1+\sigma\|g\|\,)\|Hg\|\,\|g_k\|} > 0 \qquad ...(46)$$

Using Theorem (3.3.1), we get a contradiction.

4. Numerical Results:

In this section, we reported the numerical results for the new algorithm (3.2). We tested, using the collection of problems, for general sparse and separable unconstrained optimization test problems from Luksan & Vlcek, 1999). The following values for the dimension N of the problem have been used: N=10, 100, 500, 1000. For each problem, we choose the initial matrix $H_{k+1} = \rho_k \frac{v_k v_k^T}{v_k^T y_k}$. This algorithms use a hybrid line search technique which satisfy Wolfe- conditions see (Luksan & Spedicato, 2000) The following two VM-methods hare been tested.

BFGS: BFGS method with the Wolfe- conditions as

In which $\delta_1 = 0.0001, \delta_2 = 0.2$

New algorithm (3.2): The Algorithm with the Wolfe- conditions as

In which $\delta_1 = 0.0001$, $\delta_2 = 0.2$.

Tables of numerical results shows the computational results, where the columns have the following meanings:

Problem : the name of the test problem.

NIT : number of iterations.

NFV : number of function evaluations.

f : value of the objective function at the point x.

g : gradient of the objective function at the point x.

ITERM : hybrid stopping criterion

From Table **(1)**, we observed that the average performances of the modified AL-Bayati algorithm are more efficient and better than the standard BFGS for unconstrained minimization problems.

Table (1): Comparison results of all algorithms as a total of (15) test functions.

	Standard BFGS			**New algorithm (3.2)**		
N	**NOI**	**NOF**	**TIME**	**NOI**	**NOF**	**TIME**
10	3885	7703	0:00:00.08	3340	7709	0:00:00.06
100	5463	10332	0:00:00.67	4407	10043	0:00:00.57
500	4091	9164	0:00:02.20	3565	9276	0:00:02.22
1000	4065	9003	0:00:04.63	3375	9102	0:00:04.84
Total	17504	36202	0:00:07.58	14684	36130	0:00:07.69

The details of these results are fully described in the subsequent tables**.**

Table (2)

Algorithm	**Standard BFGS with N=10**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	398	1001	22.3407562	0.163E+01	12
2	654	1001	0.121328883E-03	0.125E-01	12
3	656	1000	0.921077850	0.620E-02	12
4	106	153	0.799804276E-10	0.702E-05	2
5	139	179	0.160886197E-09	0.788E-05	2
6	69	112	3.01929454	0.965E-06	4
7	423	1000	11.4354732	0.217E+01	12
8	57	92	-133.510600	0.233E-05	2
9	374	1002	1.05358706	0.316E-01	12
10	13	23	0.944550269E-13	0.428E-06	4
11	500	1001	0.105143334E-03	0.167E-02	12
12	262	698	1.92460901	0.698E-04	2
13	74	123	-8.05139211	0.877E-06	4
14	74	145	-0.385263183E-01	0.836E-06	4
15	86	173	-0.251419625E-01	0.968E-06	4
Total	3885	7703			

TIME= 0:00:00.08

Table (3)

Algorithm	**Standard BFGS with N=100**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	357	1000	375.772751	0.416E+01	12
2	675	1000	0.417037296E-03	0.229E-01	12
3	642	1000	25.2065330	0.110E-01	12
4	94	126	0.736274182E-10	0.347E-05	2
5	195	259	0.353606890E-08	0.233E-04	2
6	84	145	33.3754297	0.800E-06	4
7	490	725	1039.41617	0.274E-04	2
8	56	57	-98.8560279	0.562E-06	4
9	350	1001	18.9030005	0.500E+00	12
10	7	14	0.512511155E-13	0.667E-07	4
11	500	1001	0.110666145E-05	0.118E-04	12
12	334	1002	2.39784602	0.109E+00	12
13	578	1000	-49.9997833	0.285E-04	12
14	501	1001	-0.133766122E-02	0.207E-01	12
15	600	1001	0.908528629E-02	0.334E-01	12
Total	5463	10332			

TIME= 0:00:00.67

Table (4)

Algorithm	Standard BFGS with N=500				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	313	1000	1950.53768	0.506E+01	12
2	654	1000	0.190687684E-03	0.178E-01	12
3	615	1000	133.781367	0.107E-01	12
4	143	179	0.407524183E-10	0.174E-05	2
5	224	298	0.116517131E-08	0.108E-04	2
6	88	151	168.291764	0.134E-05	2
7	83	196	163899.853	0.740E-03	2
8	34	54	90.9672145	0.713E-06	4
9	406	1002	97.5181572	0.102E+01	12
10	6	12	0.315195199E-13	0.251E-06	4
11	132	265	0.101551041E-07	0.100E-05	4
12	334	1002	2.66906251	0.436E-01	12
13	390	1001	-218.394928	0.162E+01	12
14	335	1002	0.311487476	0.209E-01	12
15	334	1002	0.120585472	0.261E-01	12
Total	4091	9164			

TIME= 0:00:02.20

Table (5)

Algorithm	Standard BFGS with N=1000				
Problem	**NOI**	**NOF**	**f**	**g**	**ITERM**
1	318	1002	3920.15325	0.250E+01	12
2	645	1000	0.134675134E-03	0.171E-01	12
3	592	1001	269.500609	0.302E-01	12
4	151	190	0.658806352E-10	0.233E-05	2
5	229	300	0.142991691E-08	0.119E-04	2
6	90	151	336.937181	0.225E-05	2
7	106	265	761774.954	0.242E-02	2
8	34	55	316.436141	0.433E-06	4
9	468	1002	196.256249	0.228E+01	12
10	5	10	0.783883858E-11	0.252E-06	4
11	10	21	0.129032045E-08	0.992E-06	4
12	334	1002	2.68842398	0.300E-01	12
13	414	1000	-423.279033	0.514E+01	12
14	335	1002	0.336253881	0.155E-01	12
15	334	1002	0.128686577	0.194E-01	12
Total	4065	9003			

TIME= 0:00:04.63

Table (6)

Algorithm	**New algorithm (3.2) with N=10**				
Problem	**NOI**	**NOF**	**f**	**g**	**ITERM**
1	319	1001	22.3509232	0.128E+01	12
2	500	1001	0.204896592E-03	0.183E-01	12
3	448	1000	0.921150040	0.787E-02	12
4	105	204	0.162854447E-11	0.902E-06	4
5	141	269	0.351765178E-11	0.740E-06	4
6	64	129	3.01929454	0.965E-06	4
7	373	1002	11.5380087	0.240E+01	12
8	55	117	-133.510600	0.233E-05	2
9	353	1000	1.05358886	0.138E-01	12
10	13	26	0.944262632E-13	0.428E-06	4
11	500	1001	0.105143334E-03	0.167E-02	12
12	243	500	1.92460901	0.106E-04	2
13	67	140	-8.05139211	0.877E-06	4
14	73	146	-0.385263183E-01	0.836E-06	4
15	86	173	-0.251419625E-01	0.968E-06	4
Total	3340	7709			

TIME= 0:00:00.06

Table (7)

Algorithm	**New algorithm (3.2) with N=100**				
Problem	**NOI**	**NOF**	**f**	**g**	**ITERM**
1	276	1002	375.891261	0.440E+01	12
2	500	1001	0.744415616E-03	0.352E-01	12
3	454	1000	25.2068266	0.153E-01	12
4	77	148	0.740969869E-11	0.770E-06	4
5	222	433	0.284088442E-10	0.989E-06	4
6	80	167	33.3754297	0.799E-06	4
7	248	518	1039.41624	0.324E-01	2
8	37	63	-98.8560279	0.234E-03	2
9	340	1002	18.9030390	0.500E+00	12
10	7	14	0.512705437E-13	0.667E-07	4
11	500	1001	0.110666145E-05	0.118E-04	12
12	334	1002	2.39784602	0.109E+00	12
13	332	691	-49.9993495	0.429E-01	2
14	500	1000	-0.118051218E-02	0.223E-01	12
15	500	1001	0.179335699E-01	0.375E-01	12
Total	4407	10043			

TIME= 0:00:00.57

Table (8)

Algorithm	**New algorithm (3.2) with N=500**				
Problem	**NOI**	**NOF**	**F**	**g**	**ITERM**
1	319	1001	1950.79424	0.553E+01	12
2	500	1001	0.367590874E-03	0.872E-02	12
3	438	1000	133.782356	0.340E-01	12
4	87	167	0.101233922E-10	0.794E-06	4
5	232	452	0.232191798E-10	0.982E-06	4
6	43	107	168.291793	0.105E-01	2
7	69	191	163899.853	0.379E-02	2
8	27	74	90.9672145	0.752E-06	4
9	364	1002	97.6385394	0.943E+00	12
10	6	12	0.315201668E-13	0.251E-06	4
11	132	265	0.101551041E-07	0.100E-05	4
12	334	1002	2.66906251	0.436E-01	12
13	346	1000	-218.195126	0.172E+01	12
14	334	1000	0.311551615	0.240E-01	12
15	334	1002	0.120585472	0.261E-01	12
Total	3565	9276			

TIME= 0:00:02.22

Table (9)

Algorithm	**New algorithm (3.2) with N=1000**				
Problem	**NOI**	**NOF**	**f**	**g**	**ITERM**
1	269	1001	3920.24240	0.474E+01	12
2	500	1001	0.244925354E-03	0.240E-01	12
3	358	910	269.504891	0.851E-01	2
4	95	182	0.986567524E-11	0.619E-06	4
5	237	460	0.209417316E-10	0.958E-06	4
6	44	109	336.937199	0.705E-02	2
7	106	316	761774.954	0.214E-02	2
8	31	85	316.436141	0.523E-06	4
9	374	1002	196.410872	0.191E+01	12
10	5	10	0.783887445E-11	0.252E-06	4
11	10	21	0.129032045E-08	0.992E-06	4
12	334	1002	2.68842398	0.300E-01	12
13	344	1001	-421.874354	0.623E+01	12
14	334	1000	0.336271755	0.177E-01	12
15	334	1002	0.128686577	0.194E-01	12
Total	3375	9102			

TIME= 0:00:04.84

Conclusions and Discussions

In this paper, we have proposed a new self-scaling VM-type for unconstrained minimization based on a modified Quasi-Newton condition. The convergence property and the computational experiments show that the new approach given in this paper is very successful. We claim that the formula (26)-(27) will be more efficient and better than the standard BFGS formula. Namely, there are about 17% improvement in NOI, 1% improvement in NOF, taking overall the calculations and for different dimensions ($10 \le n \le 1000$).

Methods	NOI	NOF	TIME
Standard BFGS	100	100	100
New algorithm (3.2)	83.88	99.80	101.4

Test problems for general sparse and partially separable nconstrained optimization

We seek a minimum of either a general objective function $f(x)$ or a partially separable objective function

$$f(x) = \sum_{k=1}^{n_A} f_k(x) \quad , \quad x \in R^n$$

From the starting point $\bar{x}$. For positive integers k *and* l, we use the notation $div\,(k,l)$ for integer division, i.e., maximum integer not greater than k/l, and $\mathrm{mod}\,(k,l)$ for the remainder after integer division, i.e., $\mathrm{mod}(k,l) = l(k/l - div(k,l))$. The description of individual problems follows.

problem 1: Chained Wood function.

$$f(x) = \sum_{j=1}^{k} [100(x_{i-1}^2 - x_i)^2 + (x_{i-1} - 1)^2 + 90(x_{i+1}^2 - x_{i+2})^2$$

$$+ (x_{i+1} - 1)^2 + 10(x_i + x_{i+2} - 2)^2 + (x_i - x_{i+2})^2 / 10]$$

$$i = 2j \quad , \quad k = (n-2)/2$$

$$\bar{x}_i = -3 \;,\; \mathrm{mod}(i,2) = 1 \;,\; i \le 4 \;, \qquad \bar{x}_i = -2 \;,\; \mathrm{mod}(i,2) = 1 \;,\; i \succ 4$$

$$\bar{x}_i = -1 \;,\; \mathrm{mod}(i,2) = 0 \;,\; i \le 4 \;, \qquad \bar{x}_i = 0 \;,\; \mathrm{mod}(i,2) = 0 \;,\; i \succ 4$$

problem 2: Chained Powel singular function.

$$f(x) = \sum_{j=1}^{k} [(x_{i-1} + x_i)^2 + 5(x_{i+1} - x_{i+2})^2 + (x_i - 2x_{i+1})^4 + 10(x_{i-1} - x_{i+2})^4]$$

$$i = 2j \quad , \quad k = (n-2)/2$$

$$\bar{x}_i = 3 \;,\; \mathrm{mod}(i,4) = 1 \;, \qquad \bar{x}_i = -1 \;,\; \mathrm{mod}(i,4) = 2$$

$$\bar{x}_i = 0 \;,\; \mathrm{mod}(i,2) = 3 \;, \qquad \bar{x}_i = 1 \;,\; \mathrm{mod}(i,4) = 0$$

problem 3 : *Chained Cragg and Levy function*.

$$f(x)=\sum_{j=1}^{k}[(e^{x_{i-1}}-x_i)^4+100(x_i-x_{i+1})^6+\tan^4(x_{i+1}-x_{i+2})+x_{i-1}^8+(x_{i+2}-1)^2]$$

$$i=2j \quad , \quad k=(n-2)/2$$

$$\bar{x}_i=1 \quad , \quad i=1 \quad , \quad \bar{x}_i=2 \quad , \quad i\succ 1$$

problem 4 : *Chained Cragg and Levy function*.

$$f(x)=\sum_{i=1}^{n}\left\|(3-2x_i)x_i-x_{i-1}-x_{i+1}+1\right\|^p$$

$$p=7/3 \quad , \quad x_0=x_{n+1}=0$$

$$\bar{x}_i=-1 \quad , \quad i\ge 1$$

problem 5 : *Generalized Broyden banded function*.

$$f(x)=\sum_{i=1}^{n}\left\|(2+5x_i^2)x_i+1+\sum_{j\in J_j}x_j(1+x_j)\right\|^p$$

$$p=7/3 \quad , \quad J_j=\{\,j:\max\,(1,i-5)\le\min\,(n,i+1)\}$$

$$\bar{x}_i=-1 \quad , \quad i\ge 1$$

problem 6 : *Seven – diagonal generaliization of the Broyden tridiagonal function*.

$$f(x)=\sum_{i=1}^{n}\left\|(3-2x_i)x_i-x_{i-1}-x_{i+1}+1\right\|^p+\sum_{i=1}^{n/2}\left\|x_i+x_{i+n/2}\right\|$$

$$p=7/3 \quad , \quad x_0=x_{n+1}=0$$

$$\bar{x}_i=-1 \quad , \quad i\ge 1$$

problem 7 : *Sparse modification of the Nazareth trigonometric function*.

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\left(n+i-\sum_{j\in J_I}(a_{ij}\sin x_j+b_{ij}\cos x_j)\right)^2$$

$$a_{ij}=5[1+\mathrm{mod}\,(i,5)+\mathrm{mod}\,(j,5)] \;,\quad b_{ij}=(i+j)/10$$

$$J_i=\{j:\max(1,i-2)\le\min(n,i+2)\}\cup\{j:\|j-i\|=n/2\}$$

$$\bar{x}_i=1/n \quad , \quad i\ge 1$$

problem 8 : *Another trigonometric function*.

$$f(x)=\frac{1}{n}\sum_{i=1}^{n}\left(i(1-\cos x_i)-\sum_{j\in J_I}(a_{ij}\sin x_j+b_{ij}\cos x_j)\right)$$

$$a_{ij}=5[1+\mathrm{mod}\,(i,5)+\mathrm{mod}\,(j,5)] \;,\quad b_{ij}=(i+j)/10$$

$$J_i=\{j:\max(1,i-2)\le\min(n,i+2)\}\cup\{j:\|j-i\|=n/2\}$$

$$\bar{x}_i=1/n \quad , \quad i\ge 1$$

problem 9 : *Chained Wood function*.

$$f(x)=\sum_{j\in J}\left\{\exp P\left(\prod_{j=1}^{5}x_{i+1-j}\right)+10\left(\sum_{j=1}^{5}x_{i+1-j}^{2}-10-\lambda_1\right)^2 +10\left(x_{i-3}x_{i-2}-5x_{i-1}x_i-\lambda_2\right)^2+10\left(x_{i-4}^3+x_{i-3}^3+1-\lambda_3\right)^2\right\}$$

$$\lambda_1=-0.002008\quad,\quad\lambda_2=-0.001900\quad,\quad\lambda_3=-0.000261$$

$$j=\{i,\ \mathrm{mod}(i,5)=0\}$$

$$\bar{x}_i=-2\ ,\ \mathrm{mod}(i,5)=1\ ,\ i\le 2\ ,\qquad \bar{x}_i=-1\ ,\ \mathrm{mod}(i,5)=1\ ,\ i\succ 2$$

$$\bar{x}_i=2\ ,\ \mathrm{mod}(i,5)=2\ ,\ i\le 2\ ,\qquad \bar{x}_i=-1\ ,\ \mathrm{mod}(i,5)=2\ ,\ i\succ 2$$

$$\bar{x}_i=2\ ,\ \mathrm{mod}(i,5)=3\ ,\qquad \bar{x}_i=-1\ ,\ \mathrm{mod}(i,5)=4$$

$$\bar{x}_i=-1\ ,\ \mathrm{mod}(i,5)=0$$

problem 10 : *Generalization of the Brown* 2 *function*.

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

$$\bar{x}_i=-1.0\ ,\ \mathrm{mod}(i,2)=1\ ,\ \bar{x}_i=1.0\ ,\ \mathrm{mod}(i,2)=0$$

problem 11 : *Discrete boundary value problem*.

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

$$h=1/(n+1)\quad,\quad x_0=x_{n+1}=0$$

$$\bar{x}_i=ih\ (1-ih)\quad,\quad i\ge 1$$

problem 12 : *Discretization of a variational problem*.

$$f(x)=2\sum_{i=1}^{n}\left[x_i(x_i-x_{i+1})/h+2h\sum_{i=0}^{n}\left[(e^{x_{i+1}}-e^{x_i})/(x_{i+1}-x_i)\right]\right]$$

$$h=1/(n+1)\quad,\quad x_0=x_{n+1}=0$$

$$\bar{x}_i=ih\ (1-ih)\quad,\quad i\ge 1$$

problem 13 : *Banded trigonometric problem*.

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

$$x_0=x_{n+1}=0\ ,\ \bar{x}_i=1/n\ ,\quad i\ge 1$$

problem 14 : *Variational problem* 1.

This problem is a finite difference analogue of a variational problem defined as a minimization of the functional :

$$f(x)=\int_0^1\left[\frac{1}{2}x^2(t)+e^{x(t)}-1\right]dt$$

where $x(0)=0$ & $x(1)=0$.*We use the trapeziodal rule together with* $3-$*point finite differences on a uniforrm grid having* $n+1$ *internal nodes*. *The starting point is given by the formula*

$\bar{x}_i=x(t_i)=ih\ (1-ih)$, *where* $h=1/(n+1)$.

problem 15 : *Variational problem* 2.

This problem is a finite difference analogue of a variational problem defined as a minimization of the functional :

$$f(x)=\int_0^1\left[x^2(t)-x^2(t)-2t\ x(t)\right]dt$$

where $x(0)=0$ & $x(1)=0$.*We use the trapeziodal rule together with* $3-$*point finite differences on a uniforrm grid having* $n+1$ *internal nodes*. *The starting point is given by the formula* $\bar{x}_i=x(t_i)=ih\ (1-ih)$, *where* $h=1/(n+1)$.

References

- Al-Bayati, A.Y., (1991): A New Family of Self-Scaling Variable Metric Algorithms for Unconstrained Optimization, Journal of Educ. and Sci., Iraq, Vol.12, pp.25-54.
- Al-Bayati, A.Y. and Al-Salih, M.S., (2008): New Self-Scaling Vm-Algorithm for Non – Convex Optimization, 2nd international conference Aleppo- Syria, pp.225-237.
- Kinsella, J., (2006): Course Notes for MS4327 Optimization.
- Nazareth L., (1979): A Relationship Between BFGS and Conjugate Gradient Algorithm and its Implementation for New Algorithms, SIAM Journal Numerical Analysis, Vol.61, pp.409-436.
- Li. D and Fukushima M., (2001): Modified BFGS-Method and Global Convergence in Non-Convex Minimization, Journal of Computational and Applied Math.,Vol.129, pp. 152-172.
- Luksan L. and Vlcek J., (2005): Shifted Limited-Memory VM Methods for Unconstrained Optimization ,J. of Computational and Applied Mathe, pp.1-26 .

- Luksan L. and Spedicato E., (2000): VM Methods for Unconstrained Optimization and Non Linear Least Squares, J. of Computational and Applied Mathe. , pp. 61-95.
- Luksan L.and Vlcek J., (1999): Sparse and Separable Test Problems for Unconstrained and Equality Constrained Optimization,Report V-767 Prague, ICS AS CR.
- Powell M. J., (1983): Non-Convex Minimization Calculations and the Conjugate Gradient Method , Report No.Damtp ,Department of Applied Math. And Theoretical Physics, University of Cambridge, Cambridge, England.
- Storey C. and HU .Y. F., (1993): Preconditioned Low-Order Newton Methods, J. of Optimization Theory and Application, Vol.79, No.2, pp.311-331.

خوارزمية جديدة للمتري المتغير ذاتي القياسي ذات التقارب الشامل في الامثلية المحدبة وغير المحدبة

عباس يونس البياتي   باسم عباس حسن
قسم الرياضيات
كلية علوم الحاسبات والرياضيات ــ جامعة الموصل

تاريخ الاستلام: ٢٠٠٩/٣/٤، تاريخ القبول:٢٠١٠/٥/١٠

الخلاصة

إن الشرط الأساسي لخوارزمية شبيه نيوتن هو $H_{k+1}y_k = v_k$ في الامثلية غير المقيدة، عندما y_k تعـرف بأنها فرق المشتقات المتعاقبة . استطاع Li و Fukushima من تطوير هذا الشرط وبالشكل $H_{k+1}y_k^* = v_k$ عندما $y_k^* = y_k + m_k v_k$ ، m_k عدد موجب صغير. في هذا البحث سوف نتطرق للتقصي عن خوارزمية جديدة للمتري المتغير ذاتي القياسي عندما تستخدم الشرط التالي $H_{k+1}y_k^* = \rho_k^* v_k$ علماً بان $\rho_k^* = \dfrac{y_k^{*T} H_k y_k^*}{v_k^T y_k^*}$. لقد تم دراسة الخوارزمية المقترحة وذلك ببرهان صفة التقارب الشامل في حالتي الامثلية المحدبة وغير المحدبـــة. النتائج العددية أشارت إلى ان الخوارزمية الجديدة أكثر كفاءة من خوارزمية BFGS القياسية .