A Modified Hybrid Partitioned VM-Method for Unconstrained Optimization

Abbas Y. AL-Bayati ***Basim A. Hassan***
Department of Mathematics
College of Computers Sciences and Mathematics , University of Mosul
Received:2010/10/19 Accepted:2011/6/19

Abstract

In this paper, we have proposed a modified hybrid partitioned VM-method for minimizing a smooth partially separable nonlinear functions. Numerical results indicate that the proposed (modified hybrid) method with it's two different versions was efficient than the standard BFGS formula of VM-method.

Introduction

Our problem is to minimize a nonlinear function of n variables,

$$\text{Minimize } f(x) \ , \ x \in R^n \qquad \text{..........(1)}$$

where f is smooth, and its gradient g is available. We consider iterations of the form

$$x_{k+1} = x_k + \alpha_k d_k \qquad \text{..........(2)}$$

where d_k is a search direction and α_k is a step length obtained by means of a one-dimensional search, where the step size α_k satisfies the Wolfe – Powell conditions

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \qquad \text{..........(3}a\text{)}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \qquad \text{..........(3}b\text{)}$$

where $\delta_1 < 1/2$ and $\delta_1 < \delta_2 < 1$. In Newton-type methods the search direction is of the form

$$d_{k+1} = -B_{k+1}^{-1} g_{k+1} \qquad \text{..........(4)}$$

where B_{k+1} is a nonsingular symmetric matrix. Important special cases are given by :

$B_{k+1} = I$ Steepest Descent (SD) method

$d_k = \nabla^2 f(x_k)$ Newton method

Variable Metric (VM) methods are also of the form (4), but in this case B_{k+1} is not only a function of x_k, but depends also on B_k and x_k.

All these methods are implemented so that d_k is a descent direction i.e. so that $d_k^T g_k < 0$, which guarantees that the nonlinear function can be decreased by taking a small step along d_k. For the Newton-type methods (4), and optimization problem (1), we can ensure that d_k is a descent direction by defining B_k to be positive definite (Nocedal, 1992).

The Hessian approximation matrix B_k may be updated according to the following formula:

$$B_{k+1} = B_k + U_k \; . \qquad ....(5)$$

where U_k is a nxn correction matrix .There are several strategies for updating U_k. The necessary condition which must be satisfied for this type of correction matrices may be defined by:

$$B_k (x_k - x_{k-1}) = g_k - g_{k-1} \, , \qquad ....(6a)$$

which is known as the secant condition. Two vector definitions follow below, they are introduced for simplicity and are used repeatedly further on,

$$\begin{aligned} v_k &= x_k - x_{k-1} \\ y_k &= g_k - g_{k-1} . \end{aligned} \qquad ....(6b)$$

Combining the secant condition $(6a)$ with the two vector definitions $(6b)$ gives a simplified expression of the secant condition

$$B_{k+1} v_k = y_k \, , \qquad ....(7)$$

See (Mikael, 2006) and (Nash & Sofer, 1996) for more details.

This condition is required to hold for the updated matrix B_{k+1}. Note that it is only possible to fulfill the secant equation if

$$v_k^T y_k > 0 \, , \qquad ....(8)$$

which is known as the curvature condition (Nocedal, 1999). This condition ensures the positive definiteness property of the matrix .

Li et al. (Wei *et al.*, 2006) have proposed a modified BFGS method based on a new Quasi-Newton (QN) equation defined by $H_k y_k^* = v_k$, where y_k^* is the sum of y_k and $A_k v_k$, and A_k is any positive definite matrix.

In the following, we have been given some choices of A_k; for example taking $A_k = m_k I$, yields:

$$H_k y_k^* = v_k \quad \text{where} \quad y_k^* = y_k + m_k v_k \, . \qquad ....(9)$$

where m_k is any positive constant. In fact, Li and Fukushima (Li & Fukushima, 2001) have given another type of modified BFGS update which ensures the global convergence property of the general nonlinear objective function f by using this choice. Unfortunately, their numerical results show that their modification does not outperform the BFGS method even by choosing m_k to be a very small number.

Updating formulas for the correction matrix U_k:

The correction matrix U_k can be calculated by different approaches. In fact, one of these approaches is a the class of formulas which will be used in details within this research. The variable metric (VM)-method is given by

$$U_k = -\frac{(B_k v_v)(B_k v_v)^T}{v_k^T B_k v_k} + \frac{y_k y_k^T}{y_k^T v_k} + \vartheta_k (v_k^T B_k v_k) w_k w_k^T \ ; \ y_k^T v_k \neq 0 \qquad ....(10)$$

Where $w_k = \frac{y_k}{y_k^T v_k} - \frac{B_k v_k}{v_k^T B_k v_k}$, ...(11)

and ϑ_k is a scalar. This class is known as the Broyden class, (Nash & Sofer, 1996), or as the one-parameter family of updating formulas (Gill *et al*, 2003).

Mikael in (Mikael, 2006) proposed the following approach: let x_0 be a starting point and B_0 an initial symmetric and positive definite Hessian approximation matrix. Take an update of the Broyden family which for all k satisfies that ϑ_k is larger then the critical value for which the B_{k+1} update turns indefinite. For this type of methods, below there are two different updating formulas for the correction matrix U_k within the Broyden family of VM-updates. The first is the symmetric rank-one updating formula, and the second is the well-known BFGS formula.

The symmetric rank-one updating formula is denoted by (SR1) and belongs to the Broyden family, this will be obtained by setting $\vartheta_k = \frac{v_k^T y_k}{v_k^T y_k - v_k^T B_k v_k}$ provided that $v_k^T y_k \neq v_k^T B_k v_k$. However, it dose not belong to the restricted class, since ϑ_k may be outside the interval $[0,1]$. The SR1 updating formula for the correction matrix U_k takes the following form :

- Symmetric rank one method (SR1)

$$U_k = \frac{(y_k - B_k v_v)(y_k - B_k v_v)^T}{(y_k - B_k v_v)^T v_k}, \text{ where } y_k - B_k v_v \neq 0 \qquad ....(12)$$

The second updating formula is the BFGS method which was named after its developers Broyden, Fletcher, Goldfarb and Shanno discovered it's correction formula in 1970. By setting $\vartheta_k = 0$ in (10), the BFGS formula is obtained. It is believed that it was the most effective updating formula for the Broyden family. The BFGS updating formula for the correction matrix U_k takes the following form :

- Broyden, Fletcher, Goldfarb and Shanno method (BFGS)

$$U_k = -\frac{(B_k v_v)(B_k v_v)^T}{v_k^T B_k v_k} + \frac{y_k y_k^T}{y_k^T v_k}, \qquad ....(13)$$

Where the vectors y,v are defined earlier.

Self-scaling (QN) methods appear in the mid 1970's. The general strategy of self-scaling procedures is to scale the Hessian approximation matrix B_k before it is updated at each iteration. This is to avoid the large difference in the eigen-values of the approximated Hessian of the objective nonlinear function. A new type of Self-Scaling VM-Methods was introduced by Al-Bayati (Al-Bayati, 1991). The Hessian approximated matrix B_k of Al-Bayati's self-scaling VM-method can be updated according to the following updating formula:

$$B_{k+1} = B_k - \frac{B_k v_k v_k^T B_k}{v_k^T B_k v_k} + \rho_k \frac{y_k y_k^T}{y_k^T v_k} \quad , \quad \rho_k = \frac{v_k^T B_k v_k}{y_k^T v_k} \qquad \text{..........(14)}$$

where ρ_k is the self-scaling factor. More details for the convergence analysis of the self-scaling VM-methods may be found in (Nocedal & Yuan, 1993).

A Modified Hybrid Partitioned VM-Method:

Large-scale problems are frequently formulated in such a way that

$$F(x) = \sum_{k=1}^{m} f_k(x), \quad x \in R^n, \, n \geq k \qquad \text{..........(15)}$$

The gradient and the Hessian matrix of the nonlinear function $F: R^n \to R$ can be expressed in the form

$$g(x) = \sum_{k=1}^{m} g_k(x), \qquad \text{..........(16)}$$

$$G(x) = \sum_{k=1}^{m} G_k(x), \qquad \text{..........(17)}$$

where the gradient $g_k(x)$ and the Hessian matrices $G_k(x)$ of the nonlinear function $f: R^n \to R$, $1 \leq k \leq m$, contain non-zero elements, so that they can be stored in a packed form. Partitioned VM methods use approximations $\hat{B}_{k+1}$ of the packed Hessian matrices G_k , $1 \leq k \leq m$. Therefore, the following modified partitioned VM-formulas are investigated :

1: Modified Partitioned Al-Bayati (Al-Bayati, 1991) VM-Update:

$$\hat{B}_{k+1} = \hat{B}_k + \rho_k^* \frac{\hat{y}_k^* \hat{y}_k^{*T}}{\hat{v}_k^T \hat{y}_k^*} - \frac{\hat{B}_k \hat{v}_k \hat{v}_k^T \hat{B}_k}{\hat{v}_k^T \hat{B}_k \hat{v}_k} \quad , \quad \hat{v}_k^T \hat{y}_k^* > 0 \qquad \text{..........(18)}$$

$$\hat{B}_{k+1} = \hat{B}_k \quad , \quad \hat{v}_k^T \hat{y}_k^* \leq 0 \qquad \text{..........(19)}$$

where $\rho_k^* = \hat{v}_k^T \hat{B}_k \hat{v}_k / \hat{v}_k^T \hat{y}_k^*$.

2: Modified Partitioned Broyden Rank-One Update :

$$\hat{B}_{k+1} = \hat{B}_k + \frac{1}{\hat{b}_k - \hat{c}_k}(\hat{y}_k^* - \hat{B}_k \hat{v}_k)(\hat{y}_k^* - \hat{B}_k \hat{v}_k)^T \quad , \left|\hat{b}_k - \hat{c}_k\right| \neq 0 \qquad ....(20)$$

$$\hat{B}_{k+1} = \hat{B}_k \quad , \left|\hat{b}_k - \hat{c}_k\right| = 0 \qquad ....(21)$$

Which can be used for indefinite matrices.

Where $\hat{c}_k = \hat{v}_k^T \hat{B}_k \hat{v}_k$ and $\hat{b}_k = v_k^T \hat{y}_k^*$. Now, the reduced gradients $\hat{g}_{k+1}$ are computed and the new approximations of reduced Hessian matrices $\hat{B}_{k+1}$, $1 \le k \le m$ are obtained by VM-updates using differences questions, i.e.

$$\hat{v}_k = \hat{x}_{k+1} - \hat{x}_k \ , \quad \hat{y}_k = \hat{g}_{k+1} - \hat{g}_k \ , \quad \hat{y}_k^* = \hat{y}_k + m_k \hat{v}_k \qquad ....(22)$$

Usually, the latter updating formula works worse but can be useful in some pathological cases. Therefore, the self-scaling VM-update may be suggested for this type of method.

However, the idea of the hybrid technique in this proposed method appears when a negative curvature $\hat{b}_k < 0$ appears in some iterations then updating formula of (18)-(19) are switched to (20)-(21) for calculating $\hat{B}_{k+1}$ and will be kept in all subsequent iterations. However, we have suggested this hybrid strategy which was based on the observations that (18)-(19) usually fails in the case when too many elements have indefinite Hessian matrices. Therefore, in that case we have to start with the partitioned Al-Bayati's self-scaling VM-update (18)-(19). If $m_{neg} \ge \theta m$, where m_{neg} is a number of element with a negative curvature and θ is a threshold value, then (20)-(21) is used for all elements in all subsequent iterations (we have recommended $\theta = 1/2$). The modified updating formula for $\hat{B}_{k+1}$ given in $(18-21)$ is identical to the updating formula which was defined in (Luksan and Spedicato, 2000) with used $y_k^* = y_k + m_k v_k$ instead of $y_k = g_k - g_{k-1}$. Finally, as the standard BFGS method, we claim that the modified proposed method has a global convergence property, proceeding as in (Byrd *et al*, 1987).

Modified Hybrid Partitioned VM-Method (MHPVM):

The outline of the Partitioned VM-Methods is as follows :

Step 0 : Choose an initial point $x_1 \in R^n$, set $k = 1$. $\varepsilon_1 = 1.0E-14$, $\varepsilon_2 = 1.0E-16$, $\varepsilon_3 = 1.0E-6$, number of function evaluations

$\xi = 1000$

Step 1 : If the hybrid stopping criterion is a satisfied stop : i.e.

ITERM=2- if $|f_{k+1} - f_k|$ was less than or equal to ε_1,

ITERM=3- if f_{k+1} is less than or equal to ε_2,

ITERM= 4- if $\|g_k\|$ is less than or equal to ε_3,

ITERM=12- if NOF exceeded ξ.

Step 2 : Compute the search direction $d_{k+1} = -\hat{B}_{k+1}^{-1} g_{k+1}$.

Step 3 : Find a step size α_k which satisfy the rules $(3a)$ and $(3b)$.

Step 4 : Generate a new iteration point by $x_{k+1} = x_k + \alpha_k d_k$ and calculate the modified hybrid updating $\hat{B}_{k+1}$ from $(18-21)$.

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

Numerical Results:

In this section, we have reported the proposed numerical results for the modified hybrid portioned methods MHPVM. We have tested, a number of well-known test problems using the collection of test problems, for general sparse and separable unconstrained optimization test problems from (Luksan and Vlcek, 1999). We have used the dimension of the problem (n), n=10,100,500,1000. For each problem, These Methods use a line search technique which was fully described in (Luksan and Vlcek, 2006) and satisfies the Wolfe conditions as in which $\delta_1 = 0.0001,\ \delta_2 = 0.2$. For the purpose of these comparisons, we have examined the following VM-methods:

1. Separable BFGS method (Original) .
2. Modified (1) stands for the Modified hybrid method defined in 3.1 and denoted by MHPVM with $\rho_k^* = 1$.
3. Modified (2) stands for the modified hybrid method defined in 3.1 and denoted by MHPVM with $\rho_k^* = \hat{v}_k^T \hat{B}_k \hat{v}_k / \hat{v}_k^T \hat{y}_k^*$.

Table (5.1) shows the computational results, where the columns have the following meanings :

Problem : the name of the test problem .

NOI : Number of iterations .

NOF : Number of function evaluations .

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

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

ITERM : the hybrid stopping criterion .

From Table (5.1) ,we have observed that the average performances of the modified (1) and modified (2) are better than the original separable VM-method and especially for our selected test functions.

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

Standard BFGS Method		
N	**NOI**	**NOF**
10	3885	7703
100	5463	10332
500	4091	9164
1000	4065	9003
Total	**17504**	**36202**
Modified (1)		
N	**NOI**	**NOF**
10	381	676
100	694	1592
500	545	952
1000	718	1329
Total	**2338**	**4549**
Modified (2)		
N	**NOI**	**NOF**
10	1407	2663
100	1579	3708
500	1436	3585
1000	1487	3688
Total	**5909**	**13644**

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

Table (2)

Method	**Standard BFGS 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			

Table (3)

Method	Standard BFGS 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**			

Table (4)

Method	Standard BFGS 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**			

Table (5)

Method	**Standard BFGS N=1000**				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	318	1002	3920.15325	0.250E+01	12
2	645	1000	0.1346751E-03	0.171E-01	12
3	592	1001	269.500609	0.302E-01	12
4	151	190	0.6588063E-10	0.233E-05	2
5	229	300	0.1429916E-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	0.7838838E-11	0.252E-06	4
11	10	21	0.1290320E-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**			

Table (6)

Method	**modified (1) N=10**				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	82	123	0.330103577E-16	0.212E-06	3
2	36	71	0.118471352E-13	0.886E-06	4
3	34	64	0.920931745	0.450E-06	4
4	11	21	0.278029262E-12	0.360E-06	4
5	20	40	0.291894533E-11	0.950E-06	4
6	11	16	3.01929454	0.795E-06	4
7	33	65	10.2327785	0.863E-06	4
8	13	15	-133.510600	0.608E-06	4
9	90	180	0.107765879	0.538E-06	4
10	1	2	0.00000000	0.000E+00	3
11	11	18	0.445467779E-13	0.418E-06	4
12	9	13	1.92460901	0.916E-11	4
13	17	27	-8.05139211	0.739E-06	4
14	8	13	-0.385263183E-01	0.114E-07	4
15	5	8	-0.251419625E-01	0.607E-06	4
Total	**381**	**676**			

Table (7)

Method	modified (1) N=100				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	199	678	7.14632070	0.386E-06	4
2	39	73	0.218549203E-13	0.487E-06	4
3	34	53	25.2061295	0.106E-06	4
4	14	26	0.534905949E-12	0.458E-06	4
5	30	56	0.176827215E-11	0.611E-06	4
6	23	37	31.6908719	0.119E-06	4
7	202	405	1059.36872	0.713E-05	2
8	16	30	-98.8560279	0.790E-06	4
9	68	113	1.07765879	0.535E-06	4
10	1	2	0.00000000	0.000E+00	3
11	23	43	0.156657278E-13	0.251E-06	4
12	8	13	1.92402295	0.178E-08	4
13	18	31	-49.9997833	0.393E-06	4
14	13	21	-0.379985307E-01	0.199E-06	4
15	6	11	-0.245807867E-01	0.858E-07	4
Total	**694**	**1592**			

Table (8)

Method	modified (1) N=500				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	106	171	38.3383609	0.146E-06	4
2	35	65	0.224048441E-13	0.970E-06	4
3	32	51	133.780980	0.956E-06	4
4	13	22	0.942820192E-12	0.723E-06	4
5	27	53	0.602713456E-11	0.823E-06	4
6	20	31	166.607206	0.947E-06	4
7	25	37	163899.853	0.258E-04	2
8	88	178	90.9672145	0.796E-07	4
9	122	211	5.38829394	0.568E-06	4
10	1	2	0.00000000	0.000E+00	3
11	24	45	0.356423455E-08	0.374E-06	4
12	8	15	1.92401620	0.495E-06	4
13	21	36	-218.910580	0.283E-06	4
14	15	21	-0.379923068E-01	0.382E-06	4
15	8	14	-0.245743245E-01	0.627E-08	4
Total	**545**	**952**			

Table (9)

Method	**modified (1) N=1000**				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	100	167	89.5345927	0.178E-06	4
2	40	75	0.447527768E-13	0.783E-06	4
3	33	56	269.499543	0.442E-06	4
4	17	30	0.122409430E-11	0.962E-06	4
5	27	52	0.105565975E-10	0.764E-06	4
6	20	32	335.252624	0.443E-06	4
7	26	40	761774.954	0.276E-04	2
8	298	597	316.436141	0.207E-05	2
9	71	120	11.1589533	0.450E-06	4
10	1	2	0.00000000	0.000E+00	3
11	27	53	0.806589238E-09	0.697E-06	4
12	9	15	1.92401599	0.811E-06	4
13	26	50	-427.404476	0.761E-06	4
14	14	22	-0.379921091E-01	0.308E-06	4
15	9	18	-0.245741193E-01	0.140E-07	4
Total	**718**	**1329**			

Table (10)

Method	**modified (2) N=10**				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	89	140	0.168569223E-15	0.456E-06	4
2	36	51	0.182521334E-13	0.985E-06	4
3	29	46	0.920931745	0.915E-06	4
4	10	20	0.396688126E-12	0.409E-06	4
5	16	29	0.547361777E-12	0.268E-06	4
6	11	16	3.01929454	0.586E-06	4
7	515	1000	10.2699784	0.413E+00	12
8	11	14	-133.510600	0.664E-07	4
9	60	105	0.107765879	0.677E-06	4
10	1	2	0.00000000	0.000E+00	3
11	8	13	0.683259631E-14	0.149E-06	4
12	500	1001	1.95538685	0.124E+01	12
13	21	36	-8.05139211	0.300E-06	4
14	9	12	-0.385263183E-01	0.424E-07	4
15	91	178	-0.251419625E-01	0.949E-06	4
Total	**1407**	**2663**			

Table (11)

Method	modified (2) N=100				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	275	1000	129.808266	0.215E+02	12
2	36	48	0.189528096E-13	0.743E-06	4
3	29	46	25.2061295	0.649E-06	4
4	10	18	0.507411689E-12	0.343E-06	4
5	18	32	0.180055648E-13	0.582E-07	4
6	21	34	31.6908719	0.330E-06	4
7	162	328	1057.85018	0.228E-05	2
8	13	16	-98.8560279	0.843E-06	4
9	64	104	1.07765879	0.448E-06	4
10	1	2	0.00000000	0.000E+00	3
11	14	25	0.402468685E-14	0.764E-07	4
12	400	1000	2.53403514	0.249E+01	12
13	24	38	-49.9997833	0.781E-06	4
14	11	16	-0.379985307E-01	0.486E-06	4
15	501	1001	-0.322322042E-02	0.245E-01	12
Total	**1579**	**3708**			

Table (12)

Method	modified (2) N=500				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	353	1000	371.225742	0.220E+02	12
2	37	59	0.537046032E-12	0.817E-06	4
3	24	38	133.780980	0.450E-06	4
4	11	20	0.336619161E-12	0.238E-06	4
5	18	31	0.738398667E-13	0.173E-06	4
6	27	42	166.607206	0.147E-06	4
7	11	15	163899.853	0.769E-07	4
8	95	169	90.9672145	0.426E-05	2
9	66	107	5.38829394	0.428E-06	4
10	1	2	0.00000000	0.000E+00	3
11	27	49	0.725609352E-09	0.853E-06	4
12	334	1000	2.67937417	0.163E+01	12
13	20	35	-218.910580	0.654E-06	4
14	11	17	-0.379923068E-01	0.785E-06	4
15	401	1001	0.115914327	0.172E+00	12
Total	**1436**	**3585**			

Table (13)

Method	**modified (2) N=1000**				
Problem	**NOI**	**NOF**	***f***	***g***	**ITERM**
1	420	1000	3928.83427	0.167E+02	12
2	39	55	0.381233140E-14	0.588E-06	4
3	24	35	269.499543	0.219E-06	4
4	12	21	0.660759524E-12	0.354E-06	4
5	19	31	0.245209052E-12	0.333E-06	4
6	25	40	335.252624	0.695E-06	4
7	12	16	761774.954	0.437E-06	4
8	158	310	316.436141	0.251E-05	2
9	65	106	10.7765879	0.278E-06	4
10	1	2	0.00000000	0.000E+00	3
11	6	11	0.126662417E-08	0.670E-06	4
12	334	1001	2.69763968	0.159E+01	12
13	24	40	-427.404476	0.594E-06	4
14	13	18	-0.379921091E-01	0.154E-06	4
15	335	1002	0.126959008	0.113E+00	12
Total	**1487**	**3688**			

Conclusions and Discussions:

In this paper, we have proposed two versions of a modified hybrid VM-method, denoted by MHPVM, for solving unconstrained minimization nonlinear problems. The computational experiments show that the modified approaches given in this paper are successful. We claim that the two modified (1) and (2) are better than the original formula. Namely, for the modified (1) and (2) there are about $(13-33)\%$ improvements in NOI and there are about $(12-37)\%$ improvement in NOF, overall, the calculations and for different dimensions.

Table (14): Relative efficiency of the different methods discussed in the paper.

Methods	**NOI**	**NOF**
Standard BFGS	100	100
Modified (1)	87	67
Modified (2)	88	63

Test Problems for General Sparse & Partially Separable Unconstrained Optimization

In (Luksan & Vlcek, 1999), Test problems for general sparse and partially separable unconstrained 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 & l, we use the notation $div\,(k,l)$ for integer division, i.e., maximum integer not greater than k/l, and $\bmod(k,l)$ for the remainder after integer division, i.e., $\bmod(k,l) = l(k/l - div(k,l))$. The description of individual problems is as follows:

Problem 1: Chained Wood function.

$$f(x) = \sum_{j=1}^{k} [100(x_{i-1}^2 - x_i)^2 + (x_{i-1} - 1)^2 + 90(x_{i+1}^2 - x_{i+2})^2$$
$$+ (x_{i+1} - 1)^2 + 10(x_i + x_{i+2} - 2)^2 + (x_i - x_{i+2})^2 / 10]$$

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

$\bar{x}_i = -3$, $\bmod(i,2) = 1$, $i \le 4$, $\bar{x}_i = -2$, $\bmod(i,2) = 1$, $i \succ 4$

$\bar{x}_i = -1$, $\bmod(i,2) = 0$, $i \le 4$, $\bar{x}_i = 0$, $\bmod(i,2) = 0$, $i \succ 4$

Problem 2: Chained Powell singular function.

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

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

$\bar{x}_i = 3$, $\bmod(i,4) = 1$, $\bar{x}_i = -1$, $\bmod(i,4) = 2$

$\bar{x}_i = 0$, $\bmod(i,2) = 3$, $\bar{x}_i = 1$, $\bmod(i,4) = 0$

Problem 3: Chained Cragg and Levy function.

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

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

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

Problem 4: Chained Cragg and Levy function.

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

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

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

Problem 5: Generalized Broyden banded function.

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

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

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

Problem 6: Seven – diagonal generalization of the Broyden tridiagonal function.

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

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

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

Problem 7: Sparse modification of the Nazareth trigonometric function.

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

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

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

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

Problem 8: Another trigonometric function.

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

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

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

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

Problem 9: Chained Wood function.

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

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

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

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

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

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

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

Problem 10: Generalization of the Brown2 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.

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

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

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

Problem 12: Discretization of a variational problem.

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

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

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

Problem 13: Banded trigonometric problem.

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

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

Problem 14: Variational problem1.

This problem is a finite 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 $x(0)=0$ and $x(1)=0$. We use the trapezoidal rule together with 3-points finite differences on a uniform grid having $n+1$ interval 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 problem2.

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 $x(0)=0$ and $x(1)=0$. We use the trapezoidal rule together with 3-points finite differences on a uniform grid having $n+1$ interval 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 algorithm for unconstrained optimization, Journal of Education and Science, University of Mosul, Iraq, vol.12, pp.25-54.
- Byrd, R. H.; Nocedal, J. and Yuan, Y.X., (1987): Global convergence of a class of quasi-Newton methods on convex problems, SIAM J. Numerical. Analysis, vol.24, pp.1171–1190.
- Gill, P.; Murray, W., and Wright, M., (2003): Practical optimization. Academic Press, London.
- Li. D and Fukushima M., (2001): A modified BFGS-method and global convergence in non-convex minimization, Journal of Computational and Applied Mathematics, vol.129, pp.15-35.
- Luksan, L. and Spedicato, E., (2000): VM methods for unconstrained Optimization and nonlinear least squares, J. of Computational and Applied Math., vol.124, pp.61-95.
- Luksan, L. and Vlcek, J., (1999): Sparse and separable test problems for unconstrained and equality constrained optimization, Technical Report No. 767 Prague, ICS AS CR.
- Luksan, L. and Vlcek, J., (2006): Shifted limited-memory VM methods for large-scale unconstrained optimization, J. of Computational and Applied Math., vol.186, pp.365-390.
- Mikael, F., (2006): On the robustness of conjugate gradient methods and quasi-Newton methods, Master Thesis, 5B1022, Department of Mathematics, Royal Institute of Technology, pp.1-35.
- Nash, G. S. and Sofer, A., (1996): Linear and nonlinear programming, McGraw- Hill, New York.
- Nocedal, J., (1992): Theory of algorithms for unconstraint optimization. In Acta Numerical, Cambridge University Press, Cambridge, U.K.
- Nocedal, J., (1999): Numerical Optimization, Springer, New York.
- Nocedal, J. and Yuan, Y. X., (1993): Analysis of a self-scaling quasi-Newton method, Math. Programming, vol.61, pp.19–37.
- Wei, Z. Li, G. and Qi, L., (2006): New quasi-Newton methods for unconstrained optimization problems, Applied Mathematics and Computation, vol.175, pp.1156-1188.

خوارزمية هجينة مطورة مجزأة للمتري المتغير في الأمثلية غير المقيدة

عباس يونس البياتي **باسم عباس حسن**

قسم الرياضيات

كلية علوم الحاسوب والرياضيات –جامعة موصل

تاريخ الاستلام: 2010/10/19 تاريخ القبول:2011/6/19

الخلاصة

في هذا البحث تم اقتراح طريقة هجينة مطورة ومجزأة للمتري المتغير في تصغير دوال لاخطية مجزأة جزئياً وقابلة للاشتقاق. النتائج العددية تشير إلى أن الطريقة المقترحة (الهجينة المطورة) مع قسميها المختلفين كانت كفوءة عند مقارنتها بصيغة BFGS القياسية في طريقة المتري المتغير.