

## New Hybrid (Oren- Al-Bayati) Self-Scaling Algorithm With Armijo Line-Search

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

profabbasalbayati@yahoo.com

**Runak M. Abdullah**  
College of Sciences  
University of Suleimani

Received on: 7/11/2006

Accepted on: 24/12/2006

### ABSTRACT

In this paper the development, and numerical testing of a class of self-scaling Quasi-Newton update with adaptive step - size are presented . In our work a new combined (Oren-Al-Bayati) self-scaling algorithm is presented with a modified Armijo line search procedure.

This algorithm has implemented both theoretically and numerically and tested on some well-known test cases. Numerical experiments indicate that this new algorithm is effective and superior to the standard, with respect to the number of functions evaluations (NOF) and number of iterations (NOI).

**Keywords:** self-scaling variable, Oren algorithm, Armijo line search .

خوارزمية هجينة ( اورين-البياتي) جديدة في مجال المتري المتغير ذاتي القياس مع خط بحث (ارميجو)

عباس يونس البياتي  
جامعة الموصل/كلية علوم الحاسوب والرياضيات

روناك محمد عبدالله  
جامعة السليمانية/كلية العلوم

تاريخ قبول البحث : 2006/12/24

تاريخ استلام البحث : 2006/11/7

### الملخص

في هذا البحث تم تطوير واستخدام خوارزمية جديدة في مجال المتغير ذاتي القياسي مع خط بحث متطور. في هذا العمل تم ربط خوارزمية (Oren) [6] مع خوارزمية (Al-Bayati) [1] مع تطوير خط بحث (Armijo) [4]. هذه الخوارزمية الجديدة تم معالجتها من الناحيتين النظرية والعملية مع الحصول على نتائج عددية مشجعة بالاعتماد على مقياسي (NOI), (NOF).

الكلمات المفتاحية: المتغير ذاتي القياسي ، خوارزمية (Oren) ، خط بحث (Armijo) .

### 1. Introduction

The self-scaling update proposed by Oren [7] has some good characteristics. With a self- scaling parameter  $\mu$  , this class of updates can be written as

$$H_{k+1} = \left[ H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi(y_k^T H_k y_k) R_k R_k^T \right] \mu_k + \frac{v_k v_k^T}{v_k^T y_k} \quad \dots(1)$$

Where

$$v_k = x_{k+1} - x_k$$

$$y_k = g_{k+1} - g_k \quad , \quad g_k = \nabla f(x_k)$$

$$H_0 = I$$

$$\mu_k = \frac{v_k^T y_k}{y_k^T H_k y_k} \quad \dots(2)$$

and

$$R_k = \frac{v_k}{v_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \quad \dots(3)$$

Where  $\phi$  is parameter  $\in (0,1)$ , for the BFGS  $\phi = 1$  and for the DFP  $\phi = 0$ . Oren used  $\phi = 1$  in (1) to get

$$H_{k+1} = \mu_k [H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}] + \frac{v_k v_k^T}{v_k^T y_k} + \tau R_k R_k^T, \quad \tau > 0 \quad \dots(4)$$

where  $R_k$  is defined by (3). And  $\mu_k$  is defined by (2). Equation (4) is called 2-parameters VM-update.

Al-Bayati [1] found another interesting family of VM –updates of (1) by further scaling of Oren's family of updates with a scalar  $\mu_k > 0$  such that:

$$\mu_k = \frac{y_k^T H_k y_k}{v_k^T y_k} = \bar{\mu}_k \quad \dots(5)$$

Al-Bayati [1] has a search direction which is identical to the standard CG-direction (see the following theorem):

**Theorem (1.1):**

Assume that  $f(x)$  be the quadratic function defined by

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

and that the line searches are exact: If  $H$  is any symmetric positive definite matrix (of appropriate order) and we define an updating

$$H_{new}^* = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + R_k R_k^T + \bar{\mu}_k \frac{v_k v_k^T}{v_k^T y_k} \quad \dots(6)$$

And  $\bar{\mu}_k$  is defined by (5).,

simply we use \* to denote new values in this theorem.

then the search direction  $d_{new}^* = -H_{new}^* g^*$  is identical to the conjugate-gradient direction [3].  $d_{CG}^*$  define by

$$d_{CG}^* = \begin{cases} -g_k & \text{for } k = 0 \\ -g^* + y_k^T g^* / y_k^T d_k^T d_k & \text{for } k \geq 1 \end{cases}$$

**Proof:**

The update (6) can be written as :

$$H_{new}^* = H_k - v_k y_k^T H_k / v_k^T y_k - H_k y_k v_k^T / v_k^T y_k + (\bar{\mu}_k + y_k^T H_k y_k / v_k^T y_k) v_k v_k^T / v_k^T y_k$$

Now

$$d_{new}^* = -H_k g^* + y_k^T H_k g^* / v_k^T y_k v_k + v_k^T g^* H_k y_k / v_k^T y_k - 2 y_k^T H_k y_k v_k^T g^* / (y_k^T v_k)^2 v_k$$

$$= -H_k g^* + y_k^T H_k g^* / y_k^T v_k v_k$$

using the property  $v_i^T g^* = 0$ , quoted earlier which holds for line searches. The vector  $g^*$  can be substituted for  $H_k g^*$  by using property

$$H_{i+1} g^* = H_k g^* \quad \text{for } 0 \leq i < k \leq n$$

therefore

$$d_{new}^* = -g^* + (y_k^T g^* / y_k^T v_{new}) v_{new} \quad \dots(7)$$

We also know that  $d_{BFGS}^*$  and  $d_{CG}^*$  are identical (See Nazareth [5]) and  $d_{new}^*$  is identical to  $d_{BFGS}^*$  with exact line searches. Hence equation (7) becomes

$$d_{new}^* = -g^* + (y_k^T g^* / y_k^T d_{CG}) d_{CG} = d_{CG}^*.$$

Hence the proof . #

## 2. Original Armijo Line Search rule [9]:

Given  $s > 0$  and  $\beta, \sigma$  are parameters  $\beta, \sigma \in (0,1)$ . Choose  $\alpha_k$  to be the largest one in  $\{s, s\beta, s\beta^2, \dots\}$  such that  $f(x_k + \alpha d_k) - f_k \leq \sigma \alpha g_k^T d_k$ .

Obviously, Armijo's rule is easy to implement and useful in practice. The most important advantage of the Armijo line search rule is that it enable use to estimate an initial test step-size  $s$ .

Good estimation for  $s$  can make us cut down the function evaluation at each iteration.

How to choosing the parameters ( such as  $s, \sigma, \beta$  ) is very important for practical problems.

Several choosing techniques have been appearing in many literatures. In this paper, we propose a new inexact line search rule for Armijo for Quasi-Newton method and establish some global convergent results of this method.

Theses results are useful in designing new Quasi-Newton methods with the new line search rule.

## 3. Inexact Line Search rule

We first assume that

( $H_1$ ): The function  $f(x)$  has a lower bound on the level set

$$L_0 = \{x \in R^n / f(x) \leq f(x_0)\} \text{ where } x_0 \text{ is given.}$$

( $H_2$ ): The gradient  $g(x)$  of  $f(x)$  is Lipschitz in an open convex set  $B$  that contains

$L_0$ , i.e., there exists a constant  $L > 0$  such that

$$\|g(x) - g(y)\| \leq L \|x - y\|, \quad \forall x, y \in B.$$

Some times we require that  $f(x)$  is twice continuously differentiable. In what follows, we first describe the Quasi-Newton Method.

### Original Armijo Algorithm (I):

Step(1): Given  $x_1 \in R^n$  and  $B_1$  is a symmetric definite matrix,  $k = 1$ .

Step(2): If  $\|g_k\| = 0$  then stop else go to step(3).

Step(3):  $x_{k+1} = x_k + \alpha_k d_k$ , where  $d_k = -B_k^{-1} g_k$  and  $\alpha_k$  is determined by

For the next step use Given  $\beta \in (0,1)$  and  $\sigma \in (0,1/2)$ . Choose  $\alpha_k$  to be the largest one in  $\{1, \beta, \beta^2, \dots\}$  such that

$$f(x_k + \alpha d_k) - f_k \leq \sigma \alpha [g_k^T d_k + \frac{1}{2} \alpha d_k^T B_k d_k]$$

Step(4): Let  $v_k = x_{k+1} - x_k$ ,  $y_k = g_{k+1} - g_k$ , modify  $B_k$  as  $B_{k+1}$  by using BFGS or DFP formula or other Quasi-Newton formulae.

Step(5): Set  $k = k + 1$  and go to step (2).

( $H_3$ ): The matrix  $B_k$  satisfies:  $m \|d\|^2 \leq d^T B_k d \leq M \|d\|^2$ ,  $\forall d \in R^n$ ,  $\forall k$ , where  $0 < m \leq M$ .

**Algorithm (II): (AL-Bayati) [1]**

Start with any initial point  $x_1$

Step(1): Set  $k = 1$  and chose  $H_1$  to be any positive definite matrix (usually  $H_1 = I$ ).

Step(2): Determine the step-size  $\lambda_k$  to minimize  $f(x_k + \lambda d_k)$  where  $d_k = -H_k g_k$ , and obtain  $x_{k+1} = x_k + \lambda d_k$

Step(3): Set  $H_{k+1} = [H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + R_k R_k^T] + \bar{\mu}_k \frac{v_k v_k^T}{v_k^T y_k}$ , where  $\bar{\mu}_k = \frac{y_k^T H_k y_k}{v_k^T y_k}$

Step(4): If not converged increase  $k$  by 1 and return to step(2).

**Algorithm(III): (Oren)[6]**

Start with any initial point  $x_1$

Step(1): Set  $k = 1$  and chose  $H_1$  to be any positive definite matrix (usually  $H_1 = I$ ).

Step(2): Determine the step-size  $\lambda_k$  to minimize  $f(x_k + \lambda d_k)$  where  $d_k = -H_k g_k$ , and obtain  $x_{k+1} = x_k + \lambda d_k$

Step(3): Set  $H_{k+1} = [H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + R_k R_k^T] \mu_k + \frac{v_k v_k^T}{v_k^T y_k}$

Step(4): If not converged increase  $k$  by 1 and return to step(2).

Oren algorithm (III) processes the following properties for a quadratic function:

(a) If  $\lambda_k$  minimizes  $f(x_k - \lambda H_k g_k)$  for all  $k$ , then the vector  $d_k$  are mutually conjugate (with respect to G) and hence the solution is obtained in at most n iterations.

(b) The condition number of the matrix  $R_k = G^{\frac{1}{2}} H_k G^{\frac{1}{2}}$  is strictly monotonically decreasing.

(c) If  $\lambda_k = 1$  for all k, then the algorithm convergent "two- step super linearly", i.e.

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x_{\min}\|}{\|x_k - x_{\min}\|} = 0$$

The proofs of these properties can be found in [2] and [6].

#### 4. A New Hybrid (Oren- Al-Bayati) Self-Scaling VM Algorithm

Our objective is to propose a new class of two parameter updates which will combine the relative merits of different types of QN updates. This class of modified QN updates can be expressed as follows :

$$H_{k+1} = \left[ H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + R_k R_k^T \right] \mu_k + \bar{\mu}_k \frac{v_k v_k^T}{y_k^T v_k}, \text{ where } \mu_k = \frac{1}{\bar{\mu}_k} \quad \dots(8)$$

This formula is invariant under linear transformation provided that  $\mu_k$  and  $\bar{\mu}_k$  are constructed from invariant scalars. Other properties, such as termination and conjugate gradients, are also preserved.

Likewise positive definite matrices are preserved for  $\mu_k > 0$  and sufficiently large  $\bar{\mu}_k$ .

#### 5. New Modified Armijo Line Search

We modify original Armijo algorithm as follows for a smooth differential function  $f$  and for an starting point  $x_0$  with parameters  $\lambda_0, \mu$  follow these steps by considering an stopping criterion  $\varepsilon$  :

Step(1): Input  $\{f, x_0, \lambda_0, \eta, \varepsilon\}$

Step(2): Set  $k = -1$

Step(3): If  $k < \eta$ , replace  $k$  by  $k + 1$ , set  $\lambda = \lambda_0$ ,  $j = 1$  and go to the next step, otherwise, go to step(8)

Step(4): If  $f(x_k - 2\nabla f(x_k)) - f(x_k) \leq -\frac{1}{2} \lambda \|\nabla f(x_k)\|^2$ , go to step(6); otherwise, set  $j = j + 1$  and go to the next step.

Step(5): Set  $\lambda = \frac{\lambda_0}{2^{m-1}}$  and return to step(4)

Step(6): Set  $x_{k+1} = x_k - 2\nabla f(x_k)$

Step(7): If  $\|\nabla f(x_k)\| \leq \varepsilon$ , go to Step(8); other wise go to step(3).

Step(8): Output  $\{x_k, f(x_k), \nabla f(x_k)\}$

The above algorithm provides an effective and very useful step- size adaptation procedure for various applications, for more details see [8].

#### 6. The Out lines of New Preconditional CG Algorithm With New Armijo Line Search Procedure

Step(1): Set  $x_0, \varepsilon, H_0 = I$

Step(2): For  $k = 1, 2, \dots, n$ , set  $d_1 = -H_1 g_1$

Step(3): Compute  $x_{k+1} = x_k + \lambda_k d_k$  where  $\lambda_k$  is optimal step- size obtained from Armijo line search procedure.

Step(4): Check if  $\|g_{k+1}\| < \varepsilon$  then stop, otherwise go to step (5).

Step(5):  $v_k = x_{k+1} - x_k$ ,  $y_k = g_{k+1} - g_k$ .

Step(6):  $\bar{\mu}_k = \frac{y_k^T H_k y_k}{v_k^T y_k}$ , If  $\bar{\mu}_k > 0.5$  then  $\mu_k = \frac{v_k^T y_k}{y_k^T H_k y_k}$ ,  $\bar{\mu}_k = 1$

$$\text{Step(7): } H_{k+1} = [H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}] \mu_k + y_k^T H_k y_k R_k R_k^T + \bar{\mu}_k \frac{v_k v_k^T}{y_k^T v_k}$$

where  $R_k$  is defined by (3)

$$\text{Step(8): } d_{k+1} = -H_{k+1} g_{k+1} + \beta_k d_k$$

$$\text{where } \beta_k = \frac{g_{k+1}^T H_{k+1} y_k}{d_k^T y_k}$$

Step(9): If  $k = n$  or  $d_{k+1}^T g_{k+1} > 0$  then go to step(2)

Otherwise  $k = k + 1$  and go to step(3)

### 7. Numerical Results

The comparative test involves eleven well-known standard test functions (given in the appendix) with different dimensions. The line search routine is a new modified Armijo line search which uses only function values.

The results are given in the Table (1A) is specifically quoting the number of function evaluations (NOF) and the number of iterations (NOI).

All programs are written in FORTRAN 90 language and for all cases the stopping criterion is taken to be  $\|g_{k+1}\| < 1 \times 10^{-5}$ .

From Table (1B) it is clear that there are a big advantages comparing the new algorithm against the standard BFGS algorithm, namely, there are about %02 NOI and % 62 NOF improvements in the new suggested.

**Table (1A). Comparative Performance of Two Algorithms  
(New Algorithm and Standard BFGS).**

Test Function	N	NEW OPTIMAL STEP SIZE		BFGS METHOD	
		NOI	NOF	NOI	NOF
POWELL	4	53	60	21	88
	60	58	65	68	189
	100	58	65	67	185
	500	58	65	39	118
ROSEN	2	50	62	18	56
	60	51	63	175	563
	100	51	63	237	697
	500	51	63		
CUBIC	2	50	58	14	45
	150	51	59	49	117
	200	51	59	46	119
	500	51	59	79	186
SHALLOW	2	12	15	8	26
	100	12	15	8	26
	200	12	15	8	26
	500	12	15	8	26
DIXON	4	13	15	50	58
	10	15	17	15	33
NON-DIAGONAL	2	41	53	11	45
	12	47	65	41	105
	100	52	66	76	181
	500	42	55	85	210

MIELE	4	33	37	26	90
	52	44	48	31	96
	100	44	48	32	99
	500	48	52	39	120
Gcantrel	4	67	70	9	41
	60	139	142	22	101
	100	164	167	23	107
	500	287	290	23	107
Gwolfe	3	11	14	6	17
	21	42	46	24	50
	99	52	56	72	145
	450	62	66	82	165
Beale	2	18	20	9	24
	80	19	21	10	25
	100	19	21	10	25
	500	19	21	10	25
GWood	4	33	41	37	105
	80	33	41	207	620
	100	33	41	245	757
Total		2007	2251	2040	5818

From Table(1B)

Percentage performance of the new algorithm against 100% BFGS algorithm

Table(1B)

Tools	BFGS	New
NOI	100%	98.38235
NOF	100%	38.69027

### Appendix :

All the test functions used in this paper are from general literature:

1. Powell function ( Generalized form)

$$f = \sum_{i=1}^{n/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4] ,$$

$$x_0 = (3, -1, 0, 1, \dots)^T$$

2. Rosen function

$$f = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 , x_0 = (-1.2, 1.0)^T$$

3. Cubic function

$$f = 100(x_2 - x_1^3)^2 + (1 - x_1)^2 , x_0 = (-1.2, 1.0)^T$$

4. Shallow function

$$f = \sum_{i=1}^{n/2} (x_{2i-1}^2 - x_{2i})^2 + (1 - x_{2i-1})^2 , x_0 = (-2; \dots)^T$$

5. Dixon function

$$f = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{i=2}^n (x_i^2 - x_{i+1})^2, \quad x_0 = (-1; \dots)^T$$

6. Non-diagonal function

$$f = \sum_{i=1}^n [100(x_1 - x_i^2)^2 + (1 - x_i)^2], \quad x_0 = (-1; \dots)^T$$

7. Miele function

$$f = (e^{x_1} - 1)^2 \tan^4(x_3 - x_4) + 100(x_2 - x_3)^6 + x_1^8 + (x_4 - 1)^2, \quad x_0 = (1, 2, 2, 2)^T$$

8. Cantrel function

$$f = (e^{4i-3} - x_{4i-2})^4 + 100(x_{4i-2} - x_{4i-1})^6 + A \tan((x_{4i-1} - x_{4i})^4 + x_{4i-3}^8), \quad x_0 = (1, 2, 2, 2; \dots)^T$$

9. wolfe function

$$f = (-x_1(3 - x_{1/2}) + 2x_2 - 1)^2 \sum_{i=1}^{n-1} (x_{i-1} - x_i(3 - x_{i/2}) + 2x_{i+1} - 1)^2 + (x_{n-1} - x_n(3 - x_{n/2}) - 1)^2$$

$$x_0 = (-1; \dots)^T$$

10. Beale function

$$f = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 + (6.625 - x_1(1 - x_2^3))^2, \quad x_0 = (0, 0)^T$$

11. wood function

$$f = \sum_{i=1}^{n/4} [100(x_{4i-2} - x_{4i-3}^2)^2 + (1 - x_{4i-3})^2 + 90(x_{4i} - x_{4i-1}^2)^2 + (1 - x_{4i-1})^2 + 10.1(x_{4i-2} - 1)^2 + (x_{4i} - 1)^2 + 19.8(x_{4i-2} - 1)(x_{4i} - 1)], \quad x_0 = (-3, -1; -3, -1; \dots)^T$$



**REFERENCES**

- [1] Al-Bayati, A. Y., "A New Family of Self-Scaling Variable Metric Algorithms for Unconstrained Optimization", Journal of Education and Science, Mosul University, 12, PP.25-54, (1991).
- [2] Andrei, N., "A new gradient descent method for unconstrained optimization", ICI Technical Report, (2004).
- [3] Hestenes, H.R. and Stiefel, E., "Methods of Conjugate Gradient for Solving Linear Systems", Journal of Research of the National Bureau of Standards 40, 409-436, (1952).
- [4] Armojo, L., "Minimization of Function having Lipschitz continuous First Partial Derivatives", Pacific J. Math. 16 ,1-3, (1966).
- [5] Nazareth, L., "A Relationship between BFGS and conjugate gradient Algorithm and it's Implementations for New Algorithm", SIAM Journal Numerical Analysis 16 794-800, (1979).
- [6] Oren, S.S. and Luenberger, D.G., "(SSVM) Algorithm, Part I, Criteria and Sufficient Conditions for scaling a class of Algorithms", Management Science 20,845-862, (1974).
- [7] S.S. Oren, "On The Selection of Parameters in Self-Scaling Variable Metric algorithm", Mathematical Programming 3,351-367, (1974).
- [8] Vrahatis, M. N., Androulakis, G. S., Magoulas, G. D. and Lambrinos, J. N., "A class of gradient unconstrained minimization algorithms with adaptive step size", J. Comput. Appl. Math. 114, 367-386, (2000).
- [9] Zhen-Jun Shi, "Convergence of Quasi-Newton Method With New inexact Line Search", J. Math. Anal. Appl. 315, 120-131, (2006).