A new self –scaling VM-algorithm with an adaptive line search

Abbas. Al-Baiati **Salah G. Shareef**
Hamsa Th. Chilmerane

Department of Mathematics
College of Computers Sciences & Mathematics
Mosul University

Received ٢٥ / 0٦ / 2008 **Accepted** 16 / 02 / 2009

الخلاصة:

في هذا البحث تم اقتراح خوارزمية جديدة للمتري المتغير ذاتي القياس في الامثلية غير الخطية وغير المقيدة مع استخدام خط بحث ملائم وبعد اسـتخدام صـفه التقـارب المحلـي لمصفوفة BFGS للحصول على صفه التقارب الشامل في المصفوفة ألمستخدمة. لقـد تـم مقارنه هذه الخوارزمية مع خوارزمية BFGS القياسية وكانت النتائج مشجعة جداً.

Abstract

In this study, a new self-scaling VM algorithm for solving large-scale nonlinear unconstrained optimization problems is proposed. The new algorithm is a kind of line search procedure .The local convergence theory for the BFGS matrix implementation of this variable metric is applied to achieve a globally convergent property.Numerical experiments have been done with the standard BFGS algorithm with very promising results.

Keywords: Self-scaling Variable Metric, Line search procedure, Large-scale optimization, Global convergence algorithm.

1. Introduction

consider line search descent method for solving the unconstrained optimization problem

$$\min_{x\in R^n} f(x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where f is twice continuously differentiable. Each method generates a sequence of points $\{x_k\}$ for *k=1,2,………*until termination. The initial point x_i is given. If $g_k = \nabla f(x_k) = 0$, for some *k*, then the method terminates with $x_k = x^*$. Otherwise, a search direction d_k is defined for which

$$g_k^T d_k < 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

(the descent property). Then a new iterate is defined by the line search

$$x_{k+1} = x_k + \lambda_k d_k \quad \ldots\ldots\ldots\ldots\ldots(3)$$

where λ_k is a step length chosen to minimize f along d_k for which

$$g_{k+1}^T d_k = 0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

In this case the line search is exact (see [1]). The search direction of Newton's method is defined by

$$d_k = -G_k^{-1} g_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

where $G_k = \nabla^2 f(x_k)$ is the Hessian matrix which, for x_k sufficiently close to x^*, is usually positive definite. The methods that we study are all iterative methods, since in the quadratic case we cannot expect to be able to carry out enough iteration to obtain an exact solution, even if the theory allows this possibility, due to the size of *n* or the build up of round off error. For the quadratic case the conjugate gradient (CG) method itself (see [3]), or some preconditioned conjugate gradient (PCG) method (see[4]), is usually the method of choice, although there are other variants such as the minimum residual (MR) algorithm that are also applicable to the case that G is symmetric and indefinite. We consider the case when a preconditioned version of a nonlinear conjugate gradient method is used to minimize *f*; the matrix $\overline{G}$ will denote the precondition matrix. At the first iteration, the search direction is defined by

$$d_1 = -\overline{G}^{-1} g_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

for $k > 0$,

$$d_k = -\overline{G}_k^{-1} g_k + \beta_{k-1} d_{k-1} \quad \ldots\ldots\ldots\ldots\ldots(7)$$

where

$$\beta_{k+1} = \frac{y_{k-1}^T \overline{G}_k^{-1} g_k}{y_{k-1}^T d_{k-1}} \quad \ldots\ldots\ldots\ldots\ldots\ldots(8)$$

and where

$$y_{k-1} = g_k - g_{k-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(9)$$

(This method is referred to as preconditioned conjugate gradient (PCG), for more details (see[1])). Resetting corresponds to defining d_k for $k>1$ as

$$d_k = -\overline{G}_k^{-1} g_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(10)$$

which in effect starts a new sequence of iteration. For the VM-methods assume that at the k th iteration at approximation point x_k and $n \times n$ matrix H_k are available. Then the methods proceed by generating a sequence of approximation points via the equation:

$$x_{k+1} = x_k + \lambda_k d_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(11)$$

and

$$d_{k+1} = -H_k g_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(12)$$

where H_k is an approximation of G^{-1} which is corrected or updated form iteration to iteration,in general H_k is symmetric and positive definite. There are different choice of H_k (see[5]) we list here some most popular forms (see[6])

$$H_{k+1}^{SR} = H_k + \frac{(v_k - H_k y_k)(v_k - H_k y_k)^T}{(v_k - H_k y_k)^T y_k} \quad \ldots\ldots\ldots(13)$$

is called rank one correction formula, where $v_k = x_{k+1}-x_k$ and y_k is defined as in equation(9)

$$H_{k+1}^{BFGS} = H_k + \left[1 + \frac{y_k^T H_k y_k}{v_k^T y_k}\right]\frac{v_k v_k^T}{v_k y_k^T} - \left[\frac{v_k y_k^T H_k + H_k y_k v_k^T}{v_k y_k^T}\right] \quad \ldots\ldots\ldots\ldots(14)$$

which are satisfies the Quasi-Newton condition defined by

$$H_{k+1} y_k = v_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(15)$$

and maintains the positive definite matrices if H_0 is positive.

2. Self–scaling VM methods

To improve the performance of VM updates, (see [6]) we are trying to choose H_{k+1} to satisfy the following modified QN equation

$$H_{k+1} y_k = \rho_k v_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(16)$$

where $\rho_k > 0$ is a scaling parameter .The BFGS may be written as:

$$H_{k+1} = H_k - \frac{H_k y_k v_k^T + v_k y_k^T H_k}{v_k^T y_k} + \left[\frac{1}{T_k} + \frac{y_k^T H_k y_k}{v_k^T y_k}\right]\frac{v_k v_k^T}{v_k^T y_k} \quad \ldots\ldots\ldots\ldots(17)$$

where

$$T_k = \frac{1}{\lambda_k} = \frac{6}{v_k^T y_k}(f(x_k) - f(x_{k+1}) + v_k^T g_{k+1}) - 2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(18)$$

in fact the self-scaling update proposed by (Oren [8]) has some good characteristics, with a self –scaling parameter γ_k, this class of updates can be written as

$$H_{k+1} = \left[H_k - \frac{H_k y_k y_k^T H_k}{v_k^T H_k y_k} + \phi_k (y_k H_k y_k) v_k^{-} v^{-T}_{k} \right] \gamma_k + \frac{v_k v_k^T}{v_k^T y_k} \quad \text{……………(19 a)}$$

with $\bar{v} = \dfrac{v_k}{v_k^T y_k} - \dfrac{H_k y_k}{y_k^T H_k y_k}$

(Oren and Luenberger [9]) suggest to use the self- adaptable values for the parameter γ_k

$$\gamma_k = t \frac{g_k^T v_k}{g_k^T H_k y_k} + (1-t) \frac{v_k^T y_k}{y_k^T H_k y_k} \quad \text{……………………………………(19 b)}$$

and usually the value *t=0* is recommended for the updates in the convex class, i.e

$$y_k = \frac{v_k^T y_k}{y_k^T \, H_k \, y_k} \quad and \quad \sigma_k = \frac{1}{\gamma_k} = \frac{y_k^T H_k \, y_k}{v_k^T \, y_k} \text{ (AL-Bayti 1991) ……….(20)}$$

3. Armijo line search procedure

Armijo provided in [10] a modification of the steepest descent method which automatically adapts step size λ of the iterative scheme

$$x_{k+1} = x_k - \lambda_k \nabla f(x_k), k = 1,2,3,........., \text{ ………, ………………………(21)}$$

his method and the corresponding convergence result are as follows:

Theorem (Armijo[10])

suppose that the objective function $f : R^n \to R$ is continues on R^n and bounded below on *R^n*. Assume that for a given $x_0 \in R^n$, the function *f* is continuously differentiable on the bounded level set $\bar{\alpha}(x_0) = \{x : f(x) \le f(x_0\}$ and that there exists a unique point $x^* \in R^n$ which minimize *f*. Suppose further that the equation $\nabla f(x) = 0$ is satisfied for $x_o \in \bar{\alpha}(x_o)$ if and only if $x = x^*$ and that ∇f is Lipschitz continues on $\bar{\alpha}(x_0)$, i.e, there exists a Lipschitz constant $k \neq 0$, such that

$$\|\nabla f(x) - \nabla f(y)\| \le k\|x - y\|,$$

for every pair $x, y \in s(x_0)$. Let λ$_0$ be an arbitrary assigned positive number,and consider the sequence $\lambda^m = \lambda^m / 2^{m-1}$, *m=1,2,……….* . Then for the sequence of p*oints* $\{x_k\}_{k=0}^{\infty}$

defined by

$x_{k+1} = x_k - \lambda_k^T \nabla f(x)$, k=0,1, ……………………….…………….(22)

where m is the smallest positive integer for which

$$f(x_k - \lambda_k^m \nabla f(x_k)) - f(x_k) \le -\frac{1}{2}\lambda_k^m \|\nabla f(x_k)\|^2$$

it holds that $\lim_{k\to\infty} x = x^*$

Next, a high–level description of the Armijo's algorithm, in which the corresponding parameters indicate x_0 initial point, λ_0 an arbitrary large initial step size, NIT (number of iterations) the maximum number of iterations required and $\in$ the predetermined desired accuracy.

Algorithm1

```
1- Input {f : x_0:λ_0 ; NIT,ε}
2- Set k=1
3- If k<NIT, replace k by k+1,set λ=λ_0,m=1 and go to next step: otherwise, go to step(8)
4- If $f(x_k - \lambda\nabla f(x_k)) - f(x_k) \le -\frac{1}{2}\lambda\|\nabla f(x_k)\|^2$, go to step (6); otherwise, set m=m+1 and go to the next step.
5- Set $\lambda = \lambda_0 / 2^{m-1}$* and return to step (4).
6- Set $x_{k+1} = x_k - \lambda\nabla f(x_k)$
7- If $\|\nabla f(x_k)\| \le \varepsilon$ go to step(8); otherwise go to step (3).
8- Output $\{x_k; f(x_k); \nabla f(x_k)\}$
```

4. A new self-scaling VM- update

In this section a new self-scaling VM algorithm is proposed, with modified Armijo line search, and so we develop a Preconditioned CG-algorithm. This new approach finds the minimum of an n dimensional non quadratic function. Now let the g_n is denoted as the gradient of the function $f(x_k)$. For simplicity f_k and G_k denote $f(x_k)$ and $\nabla^2 f(x_k)$, respectively.

Using the Taylor expansion to the third-order terms f and g^T can be written as respective

$$f_k = f_{k+1} - g_{k+1}^T v + \frac{1}{2!} v^T{}_k G_{k+1} v_k - \frac{1}{3!} v_k^T (T_{k+1} v_k) v_k + o\|v_k\|^4$$

$$g_k^T v_k = g_{k+1}^T v_k - v_k^T G_{k+1} v_k + \frac{1}{2!} v_k^T (T_{k+1} v_k) v_k + o(\|v_k\|^4)$$

where $T_{k+1} \in R^{n \times n \times n}$ is the tensor of f at x_{k+1}. We obtain the following relation by canceling the terms which induce the tensor in above expressions

$$v_k^T G_{k+1} v_k = v_k^T y_k + 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T v_k + o\|v_k\|^4$$

Then using a new approximation B_{k+1}, we have, after putting $o\|v_k\|^4$ (small value),

$$v_k^T B_{k+1} v_k = v_k^T y_k + \theta_k \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(23)$$

where

$$\theta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T v_k$$

equation (23) givens a class of modified secant condition in the form

$$y_k^{-} = y_k + \frac{\theta_k}{v_k^T u_k} u_k \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(24)$$

where $u_k \in R^n$ is any vector such that $v_k^T u_k \neq 0$.

Now from (17),replacing (24) instead of y_k then we get

$$H_{k+1} = H_k - \frac{H_k \bar{y}_k v_k^T + v_k \bar{y}_k^T H_k}{v_k^T \bar{y}_k} + \left(\frac{1}{T} + \frac{\bar{y}_k H_k \bar{y}_k}{v_k^T \bar{y}_k}\right) \frac{v_k v_k^T}{v_k^T y_k} \qquad \ldots\ldots\ldots(25)$$

And by Armijo line search with λ_k is the largest number in $\{s, s/2, s/2^2, \ldots\ldots\ldots\}$ such that

$f_k - f(x_k + \lambda d_k) \geq -\eta \lambda g_k^T d_k$ Where $\eta \in (0,1)$, and $s \neq 0$

Note 1 : In fact, we can use the generalized Armijo's line search λ_k is the largest number in $\{s, s\beta, s\beta^2, s\beta^3, \ldots\ldots\}$ such that

$f_k - f(x_k + \lambda_k d_k) \geq -\eta_1 \lambda g_k^T d_k$ where $\eta_1 \in (0,1)$, $\beta \in (0,1)$, and $s \neq 0$

Note 2: we should choose a suitable step size at each iteration so that the Armijo, type line search on $\bar{\theta}$ is then carried out until the acceptance condition

$$f(x_k + d_k) \leq \max_{\max(k-m) \leq j \leq k} f(j) - \eta g_k^T d \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(26)$$

is met, where $d = -\bar{\theta} g_k$ is the displace element along the steepest direction where

$$\eta = 10, \eta \in (0,1), \bar{\theta} = 1, \frac{1}{10}, \frac{1}{100}$$

5. The Global convergence property of the new proposed algorithm

Let s be a positive parameter and let $x_s=x_k-sd_k$ where

$d_k=-H_k g_k$

Armijo's rule for determining λ is the following evaluate $\|g(x_s)\|$ at $s=1,\frac{1}{2},\frac{1}{4},\ldots\ldots\ldots\ \ldots\ldots\ldots\ \ldots\ldots\ldots\ ..$

Stopping when

$$\|g(x_s)\|\leq(1-\frac{s}{2})\|g(x_k)\| \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(27)$$

if λ denotes the first *s* which satisfies (27), then x_{k+1} is given by

$$x_{k+1}=x_k-\lambda H_k g_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(28)$$

usually, the Euclidean norm is used in (27) and the sum of the squares of g's components are reduced at each iteration.

The problem (1), at the minimum when *g(x)=0*, the partial derivative respect to each of the unknown vanishes:

$$\sum_{i=1}^{n}\frac{\partial f_i}{\partial x_j}(x)f_i(x)=0 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(29)$$

For *j=1,2,…………………,n*

Using matrix-vector notation (29) is written as $j(x)^T g(x)=0$. If j(x) is invertible the equation $j(x)^T g(x)=0$ implies that any minimizer for (1) satisfies *g(x)=0.*

6. Outline of new self-scaling VM algorithm with Armijo line search

Step (1): set the initial point x_0,the scalar ε and $H_0=I, \beta=1,k_1=1$

Step (2): for k=1 set $d_1=-H_1 g_1$ where $g_1=\nabla f(x_1)$

Step (3): compute $g_k^T g_k$

Step(4): if ($k\neq 1$) go to step (5)

B=λ

go to step (6)

Step(5): if (k=2)

$\overline{\beta}=0.2$

$\lambda=\overline{\beta}$

goto step (6)

step(6): compute $x_{k+1} = x_k + \lambda_k d_k$ where λ_k is obtained from Armijo line search procedure.

Step(7): if $f_{k+1} - f_k \leq 0.1\lambda(g_{k+1} d_k - 0.5\lambda g_k d_k)$ go to step (8).

Otherwise set

$k = k + 1$ and go to step (4)

step(8): set $v_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$

step (9): compute θ_k where

$$\theta_k = 6(f_k - f_{k+1}) + 3(g_k - g_{k+1})^T v_k$$

$$\bar{y} = y_k + \frac{\theta_k}{v_k^T y_k} y_k$$

step (10): compute T_k from (20)

$$\bar{\tau}_k = \frac{\bar{y}_k^T H_k \bar{y}}{v_k^T \bar{y}_k}$$

step(11): if $(\bar{\tau}_k \neq 0.5.OR.\bar{\tau}_k \neq \frac{1}{\tau_k})$

then

$$\phi = \frac{1}{\tau_k}$$

$$\bar{\tau}_k = 1$$

step(12): compute H_{k+1}

$$H_{k+1} = H_k - \frac{H_k \bar{y}_k y_k^T H_k}{\bar{y}_k^T H_k \bar{y}_k} + \phi_k (y_k H_k y_k) \bar{v}_k \bar{v}_k^T + \bar{\tau}_k \frac{v_k v_k^T}{v_k^T y_k}$$

where $\bar{v}_k = \frac{v_k}{v_k^T \bar{y}_k} - \frac{H_k \bar{y}_k}{\bar{y}_k^T H_k \bar{y}_k}$

step (13): $d_{k+1} = -H_{k+1} g_{k+1} + \frac{\bar{y}_k H_{k+1} g_{k+1}}{d_k^T \bar{y}_k} d_k$

step(14): if $k=N$ or $g_k^T g_k \leq g_{k+1}^T g_{k+1}$ then go to step (2)

else go to step (3)

7. Numerical Results

The following table, gives the comparison between the results of the new self-scaling VM algorithm with Armijo line search and standard BFGS algorithm. In the all cases the stopping criterion is $\|g_{k+1}\| \le 1\times 10^{5}$. Also we run the program on a set of test functions with five versions of variables, that is *n=4,20,100,500* and *1000*.

The results are given in the table is specifically quoting the number of function evaluations NOF and the number of iterations NOI. Experimental results confirm that the new algorithm is superior to the standard BFGS algorithm .From the Table, taking the standard BFGS algorithm as 100 % NOI, and NOF respectively, there are about 47 % NOI and 55 % NOF improvement overall selected group of test functions.

Comparative performance of the two algorithms (classical BFGS with cubic line search and new self-scaling VM with Armijo line search)

Test function	Classical BFGS	New self-scaling VM
	NOF (NOI)	NOF (NOI)
Powell(4)	22(71)	32(44)
Powell(20)	40(117)	39(69)
Powell(100)	71(197)	64(155)
Powell(500)	50(148)	49(114)
Powell(1000)	62(155)	137(329)
Wood(4)	55(145)	24(46)
Wood(20)	130(358)	49(114)
Wood(100)	262(746)	105(278)
Wood(500)	583()	148(371)
Wood(1000)	692(1792)	234(422)
Cubie(4)	19(59)	11(25)
Cubie(20)	27(66)	24(73)
Cubie(100)	70(167)	30(94)
Cubie(500)	53(124)	31(100)
Cubie(1000)	44(112)	28(88)
Rosen (4)	35(94)	18(36)
Rosen (20)	70(180)	34(89)
Rosen (100)	139(353)	49(161)
Rosen (500)	88(192)	66(214)
Rosen (1000)	92(252)	64(208)
Dixon (4)	11(21)	11(21)
Dixon (20)	11(21)	30(56)
Dixon (100)	11(21)	37(75)
Dixon (500)	11(21)	38(77)
Dixon (1000)	11(21)	39(7)
Shallow (4)	8(26)	8(12)
Shallow (20)	8(26)	8(12)
Shallow (100)	8(26)	8(12)
Shallow (500)	8(26)	8(12)
Shallow (1000)	8(26)	8(12)
Total	2699(7246)	1431(3326)

References

1) AL-Bayati; A.4. and lahif, I. S. (Anew 3-term PCG algorithm for non linear optimization problems) The American J. of Math. and Statist., USA, vol (4), no(2) pp81-87, (2008).

2) R. Fletcher, Practical Methods of optimization, second Edition, Wiley, chichester, England,1987.

3) M. R. Hestenes, and Stiefel E. “Methods of Conjugate Gradients for Solving Linear Systems” Journal of Research of the National Burean of Standards, Vol. 49, pp:409-436,1952

4) G. H. GoluBandc. and C. F. Vanloan. "Matrix computation", Johns Hopkins University Press, Baltimore, Maryland, 3nd edition,1996.

5) R. Fletcher, Practical Methods of optimization. Chichester Wiley, second Edition., 1987.

6) P. E. Frandsen, K. Jonasson, H. Nielsen and O. Tingleff, "Informatics and Mathematical Modelling", Technical University of Denmark, 3rd Edition, March 2004.

7) M. C. Biggs, "A note of minimization Algorithms which make use of non-Quadratic properties of the objective Function" journal of Institute of Mathematics and its Application vol.12, pp.337-338, 1973.

8) S. S. Oren, "self –scaling variable metric algorithms", Part II, Management Science, vol. 20, pp. 863-874, 1974.

9) S. S. Oren and D. G. Luenberger, "self-scaling variable metric (SSVM) Algorithms", part I, criteria and Sufficient conditions for scaling a class of Algorithms, Management science vol. 20, pp. 845-865, 1974.

10) L. Armijo, "Minimization of functions having Lipchitz continuous first partial derivative", Pacific Journal of Mathematics, vol. 6, pp.1-3, 1966.