Iraqi Journal of Statistical Science (9) 2006
P.P. [40-56]

Two New Approaches for PARTAN Method

Ban A. Metras* **Khary B. Rasheed****

ABSTRACT

In This paper, we suggest two approaches for the parallel tangent (PARTAN) method. First is to combine PARTAN method with Perry algorithm and second is to combine PARTAN method with Al-Bayati-Ahmed, 1996 algorithm. The new suggested methods are tested to solve unconstrained optimization problems by using statistical tests and the results of the new suggested methods are better than the original PARTAN method with respect to time and the accuracy.

أسلوبان جديدان لطريقة الظل المتوازي

الملخص

في هذا البحث تم استحداث أسلوبين جديدين لطريقة الظل المتوازي. الأول هو ربط هذه الطريقة مع خوارزمية Perry ذات الخطوة المنفردة لخزن المتغير كأسلوب لفقدان الذاكرة. والثاني هو ربط نفس الخوارزمية مع خوارزمية Al-Bayati and Ahmed, 1996. اختبرت الطريقتان المقترحتان بتطبيقهما لحل مسائل ألامثلية غير المقيدة وقد أثبتت هاتان الطريقتان كفاءتهما مقارنة مع الخوارزمية القياسية المعروفة في هذا المجال اذ تم فحص هاتين الطريقتين باستخدام اختبارات إحصائية معروفة وكانت النتائج للطريقتين المقترحتين أفضل من النتائج لطريقة الظل المتوازي الأصلية من حيث الوقت والدقة.

*Ass. Pro. / Statistical Dept. / College of Computer Sciences and Mathematics, University of Mosul
**Ass. Lecture/ / Statistical Dept. / College of Computer Sciences and Mathematics, University of Mosul

Received: 30/ 11 /2005 ____________________Accepted: 6/ 2 / 2006

1. Introduction:

This paper is concerned with the unconstrained minimization problem

$$\text{Min}\, f(x):\ R^n \to R, \qquad (1)$$

where f is a reasonably smooth function. Some of the best methods for solving eq.(1) are the quasi-Newton methods (QN), since they rely on matrix computations difficulties with computer storage arise when the dimension of the problem becomes large. A number of attempts has been made to overcome this situation either by modifying the QN- methods themselves or by improving conjugate gradient methods.

The advantage of conjugate gradient methods is of course, that they depend on vector computations only (see Khoda and Storey, 1992).

CG-algorithms are iterative techniques with generating a sequence of approximations to the minimizer x^* (of a scalar function $f(x)$) of the vector variable x. The sequence x_k is defined by

$$x_{k+1} = x_k + \lambda_k d_k \qquad (2)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \qquad (3)$$

where g_k is the gradient of $f(x)$, λ_k is a positive scalar chosen to minimize $f(x)$ along the search direction d_k and β_k is a coefficient, given by one of the following expressions.

$$\beta_k = \frac{y_k^T g_{k+1}}{y_k^T d_k}, \qquad \text{(Hestenes-Stiefel, 1952)} \qquad (4)$$

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \qquad \text{(Fletcher-Reeves, 1964)} \qquad (5)$$

$$\beta_k = \frac{y_k^T g_{k+1}}{g_k^T g_k} \qquad \text{(Polak-Riebere, 1969)} \qquad (6)$$

$$\beta_k = \frac{-g_{k+1}^T g_{k+1}}{g_k^T d_k} \qquad \text{(Dixon, 1975)} \qquad (7)$$

$$\beta_k = \frac{-g_{k+1}^T g_{k+1}}{d_{k+1}^T y_k} \qquad \text{(Al-Assady, Al-Bayati, 1986)} \qquad (8)$$

2. Conjugate Gradient Algorithms as a Memoryless QN-Algorithms:

This type of CG-algorithm was suggested for the first time by Perry (1978) and further analyzed by Shanno (1978a). These algorithms are generating descent directions even if ILS are used since:

$$d_k = -H_k g_k \qquad (9)$$

Multiplying eq.(9) by g_k^T yields

$$d_k^T g_k = -g_k^T H_k g_k < 0$$

Since H_k is positive definite and the second term is positive implies that d_k is a descent direction. H_k is updated through the formula of BFGS update. (see Bazarra et al, 2000).

Given some approximation H_k to the inverse Hessian matrix, we compute the search direction $d_k = -H_k g_k$, and we define $v_k = x_{k+1} - x_k$ and

$y_k = g_{k+1} - g_k = G(x_{k+1} - x_k) = Gv_k$.

We now want to construct a matrix

$$H_{k+1} = H_k^{(1)} + H_k^{(2)} \qquad (10)$$

where $H_k^{(2)}$ is some symmetric correction matrix that ensures that $v_1, v_2, \ldots, v_k$ are eigenvectors of $H_{k+1}G$ with unit eigenvalues. Hence

$$H_{k+1} y_k = v_k$$

This condition translates to the requirement that

$$H_{k+1} y_k = v_k - H_k y_k$$

This therefore leads to the rank-two DFP (Dividon; Fletcher and Reeves, 1964) update via the correction term

$$H_k = \frac{v_k v_k^T}{v_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} \equiv H_k^{DFP} \qquad (11)$$

The Broyden updates suggest the use of the correction matrix $H_k = H_k^B$ given by

$$H_k^B = H_k^{DFP} + \frac{\theta \tau_k p_k p_k^T}{v_k^T y_k} \qquad (12)$$

where $p_k = v_k - (\frac{1}{\tau_k}) H_k y_k$ and where τ_k is chosen so that the quasi-Newton condition holds by virtue of $p_k^T y_k$ being zero. Then

$$H_k^{BFGS} = H_k^B(\theta = 1) = \frac{v_k v_k^T}{v_k^T y_k}\left(1 + \frac{y_k^T H_k y_k}{v_k^T y_k}\right) - \left(\frac{H_k y_k v_k^T + v_k y_k^T H_k}{v_k^T y_k}\right) \qquad (13)$$

Also this type of algorithms does not need to update the matrix *H* explicitly (i.e. this matrix reduces a vector of order *n*).

3. Perry's Conjugate Gradient Algorithm:

Among the most efficient CG-algorithms was the Perry-CG algorithm. In eq.(3) the scalar β_k was chosen to make d_k and d_{k+1} conjugate using an exact line search. In general, line searches are not exact, Perry relaxed this requirement and he rewrote eq.(3) where β_k is defined by eq.(4), but assuming inexact line search; thus he obtained

$$d_{k+1} = -[I - \frac{d_k y_k^T}{y_k^T d_k}] g_{k+1} \qquad (14)$$

But this matrix is not of full rank; hence he modified eq.(14) as

$$d_{k+1} = -[I - \frac{v_k y_k^T}{v_k^T y_k} + \frac{v_k v_k^T}{v_k^T y_k}] g_{k+1} \qquad (15)$$

$$= -Q_{k+1} g_{k+1} \qquad (16)$$

the matrix Q_{k+1} satisfies the form

$$Q_{k+1}^T y_k = v_k$$

Algorithm (Perry):

An algorithm based on the search direction given in eq.(14) is as follows:

Step 1: Let $x_{(1)}$ be an estimate of a minimizer x^* of f. and let ε be a tolerance Number.

Step 2: Set $k=1$ and compute $d_k = -g_k / \|g_k\|$.

Step 3: Line search :Compute $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a scalar chosen in such away that $f_{k+1} < f_k$.

Step 4: If $\|g_{k+1}\| < \varepsilon$ take x_{k+1} as x^*, and stop.

Step 5: If k=n or $|g_{k+1}^T g_k| \geq 0.2 |g_{k+1}^T g_{k+1}|$. Then compute the new search direction defined by

$d_{k+1} = -g_{k+1}(\frac{\lambda_k d_k^T d_k}{g_{k+1}^T g_{k+1}})$. Set k=1 and go to step 3. Else k=k+1

Step 6: Compute the new search direction defined by

$d_{k+1} = -g_{k+1} - (\frac{v_k^T g_{k+1}}{v_k^T y_k} - \frac{y_k^T g_{k+1}}{v_k^T v_k})v_k$, go to step 2

4. Single-Step Variable- Storage Conjugate Gradient Algorithm:

Al-Bayati and Ahmed in 1996, developed a variable – storage CG-algorithm as follows:

$$H_{k+1} = H_k + \left[\frac{2 y_k^T H_k y_k}{(v_k^T y_k)^2}\right] v_k v_k^T - \frac{H_k y_k v_k^T + v_k y_k^T H_k}{v_k^T y_k} \qquad (17)$$

the above formula generates positive definite matrices. Now since

$$d_{k+1} = -H_{k+1} g_{k+1} \qquad (18)$$

hence

$$d_{k+1} = -g_{k+1} - \left[\frac{2 y_k^T y_k}{v_k^T v_k} \frac{v_k^T g_{k+1}}{v_k^T v_k} - \frac{y_k^T g_{k+1}}{v_k^T y_k}\right] v_k + \frac{v_k^T g_{k+1}}{v_k^T y_k} y_k \qquad (19a)$$

It is clear that if $v_k^T g_{k+1} = 0$ and by using exact line search, then eq. (19a) becomes

$$d_{k+1} = -g_{k+1} + (\frac{y_k^T g_{k+1}}{v_k^T y_k}) v_k \qquad (19b)$$

which is the standard HS-CG-algorithm and therefore has *n*-step convergence to the minimum of a quadratic function. Thus this CG-algorithm as defined precisely by the new VM-update eq.(3), where the approximate of inverse Hessian is reset to the identity matrix at every step.

Algorithm (Al-Bayati-Ahmed, 1996):

Step 1: Let initial point x_1.

Step 2: Set k=1, $d_k = -g_k / \|g_k\|$

Step 3: Set $x_{k+1} = x_k + \alpha_k d_k$ where α_k is a scalar chosen in such a way that

$f_{k+1} < f_k$.

Step 4: Check for convergence i.e. if $\|g_{k+1}\| < \varepsilon$ where ε is small positive tolerance, stop.

Step 5: Otherwise. If $k=n$ or $|g_{k+1}^T g_k| \geq 0.2 |g_{k+1}^T g_{k+1}|$ compute the new search direction defined by

$d_{k+1} = -g_k \times (\frac{\lambda_k d_k^T d_k}{g_{k+1}^T g_{k+1}})$, set $k=1$ and go to step (3). Else set $k=k+1$.

Step 6: Compute the new search direction defined by

$$d_{k+1} = -g_{k+1} - \left[\frac{2 y_k^T y_k}{v_k^T y_k} \frac{v_k^T g_{k+1}}{v_k^T v_k} - \frac{y_k^T g_{k+1}}{v_k^T y_k} \right] v_k + \frac{v_k^T g_{k+1}}{v_k^T y_k} v_k$$

and go to step (2).

5. The Parallel Tangent Method (PARTAN):

This procedure proceeds to the minimum of differentiable objective function f on successive straight lines. The path directions are alternately determined by positions of points already reached or by certain gradient directions. This method does not involve the explicit construction of mutually conjugate direction vectors although vectors can be constructed from the direction vectors that are mutually conjugate. This property underlies the convergence of the (PARTAN) method.

6. A General Outlines of the PARTAN Algorithm:

Starting procedure: For the first step,

Let, $d_0=-g_0$ (20)

So that

$$x_1 = x_0 + \lambda_0 d_0 \quad (21)$$

Next, choose

$$d_2=-g_2 \quad (22)$$

Then, the fourth point is generated by moving in direction that is collinear with (x_3-x_1) so that

$$d_3=-(x_3-x_0) \quad (23)$$

This is referred to as an acceleration step. Continuing the procedure:

After determining x_4, the procedure is continued by successively alternating gradient and acceleration steps.

Thus

$$d_i=-g_i \quad \text{for } i=0,2,\dots,2n-2 \quad (24)$$

$$d_i=-(x_i-x_{i-2}) \quad \text{for } i=3,5,\dots,2n-1 \quad (25)$$

This method will reach the minimum of an n dimensional quadratic surface in no more than $2n$ steps. The d_i that are generated are not mutually conjugate but the following properties are true:

1- The search direction are descent i.e. $d_i^T g_i<0$.

2- The vectors (x_2-x_0), (x_4-x_2), …, $(x_{2n}-x_{2n-2})$ are mutually conjugate.

3- The points $x_4, x_6, \ldots, x_{2n}$ are the minimum for the space spanned respectively by; d_1 and d_2, g_2 and g_4; $\ldots(d_1, d_2, \ldots, d_{2n-2})$

4- The gradient vectors $g_0, g_2, \ldots, g_{2n}$ are orthogonal.

(Wilde, 1967).

PARTAN Algorithm stops when $\|g_{k+1}\|$ is sufficiently small and in perfect arithmetic should terminate in at most n iterations, whatever the choice of x_0. In particular, the algorithm will converge in k $(<n)$ iterations if the Hessian matrix of the function f has only k distinct eigenvalues. These properies follow because the recurrence relation of d_i is designed to ensure that the search directions are conjugate with respect to the Hessian matrix of f. Scalar products appear in the expressions for d_i and the step length q.

The behavior of PARTAN algorithm in finite precision arithmetic will depend on how accurately these scalar products are computed.

7. The Outlines of the Modified (PARTAN) Algorithm (1):

Step (1): Set the initial point x_0

Step (2): Let $d_0 = -g_0 / \|g_0\|$

Step (3): Compute $x_1 = x_0 + \lambda_0 d_0$, next, choose

$d_2 = -g_2$

Then, the fourth point $d_3 = -(x_3 - x_0)$

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

Step (5): Compute:

$d_k = -g_k$ if k is even

$d_k = -(x_k - x_{k-3})$ if k is odd,

$d_{k+1} = -g_{k+1} + \beta_k d_k$, where β_k is the conjugancy coefficient.

Step 6: If k=n or $\left|g_{k+1}^{T} g_{k}\right| \geq 0.2\left|g_{k+1}^{T} g_{k+1}\right|$. Then compute the new search direction defined by $d_{k+1} = -g_{k+1}(\frac{\lambda_k d_k^T d_k}{g_{k+1}^T g_{k+1}})$.(AL-Bayati & Ahmad 1996) Set k=1 and go to step 2. Else k=k+1

Step 7: Compute the new search direction defined by

$d_{k+1} = -g_{k+1} - (\frac{v_k^T g_{k+1}}{v_k^T y_k} - \frac{y_k^T g_{k+1}}{v_k^T v_k})v_k$ and go to step 3

Computational cost appears at each iteration of the new algorithm with accurate scalar products is approximately ten times as expensive as one without, i.e. if the cost of a "normal" iteration is sn^2 the cost of one with accurate scalar products is about $10sn^2$. This penalty should be set against the fact that accurate scalar products will sometimes allow less iteration to be taken.

8. The Outlines of the Modified (PARTAN) Algorithm (2):

Step (1): Set the initial point x_0

Step (2): Let $d_0 = -g_0 / \|g_0\|$

Step (3): Compute $x_1 = x_0 + \lambda_0 d_0$, next, choose

$d_2 = -g_2$

Then, the fourth point $d_3 = -(x_3 - x_0)$

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

Step (5): Compute:

$d_k = -g_k$ if k is even

$d_k = -(x_k - x_{k-3})$ if k is odd,

$d_{k+1} = -g_{k+1} + \beta_k d_k$, where β_k is the conjugancy coefficient.

Step 6: If $k=n$ or $\left|g_{k+1}^{T} g_{k}\right| \geq 0.2\left|g_{k+1}^{T} g_{k+1}\right|$ compute the new search direction defined by

$d_{k+1} = -g_k \times (\frac{\lambda_k d_k^T d_k}{g_{k+1}^T g_{k+1}})$ (AL-Bayati & Ahmad 1996), set $k=1$ and go to step (2).

Else set $k=k+1$.

Step 7: Compute the new search direction defined by

$$d_{k+1} = -g_{k+1} - \left[\frac{2y_k^T y_k}{v_k^T y_k} \frac{v_k^T g_{k+1}}{v_k^T v_k} - \frac{y_k^T g_{k+1}}{v_k^T y_k} \right] v_k + \frac{v_k^T g_{k+1}}{v_k^T y_k} v_k$$

and go to step (3).

9. Duncan Test:

We used Duncan test to compare the difference between the means and depending on the value of Least Significant Range (L.S.R.) (Ronald , 1971), by:

1. Estimate the scalar error value for any coefficient i.e.:

$$S_{\bar{y}i.} = \sqrt{\frac{Mse}{r}}$$

where:
Mse: is the mean of square error.
r: is the number of iterations.

2.Findout SSR from Duncan's table under significant level {o.o5 or o.o1}.

3. Compute L.S.R by:

$$L.S.R = S_{\bar{y}i.} * SSR$$

4. Arrangement efficient means decreasing or increasing.

5. Compared differences means with L.S.R value to discaste it is significant or not. If the difference is less than L.S.R, we say it is significant and the reverse is true.

10. Results and Conclusions:

In order to asses the performance of the new proposed algorithm NEW, three algorithms are tested over 8 generalized selected well-known test functions with different dimensions where $100 \le n \le 1000$

1-CG- algorithm
2- PARTAN algorithm.
3- New algorithm (1).
4- New algorithm (2).

All the algorithms in this paper use the same exact line search strategy which is the cubic fitting technique directly adapted from Bunday (1984).
Also all the algorithms have convergence when $\|g_{k+1}\| < \varepsilon$ where $\varepsilon = 1 \times 10^{-5}$.

The numerical results are presented in the following two tables. In table (1), we have compared all our CG-algorithms by using eight well-known test functions and for dimensions n=100. In table (2) we have compared all our CG-algorithms by using eight well-known test functions and for dimensions n=1000.

Table (1)

Comparisons of all CG-algorithms for test functions with n=100.

Test function	CG algorithm NOI (NOF)	PARTAN algorithm NOI (NOF)	New (1) NOI (NOF)	New (2) NOI (NOF)
Himmel	24 (104)	22 (100)	22 (98)	19 (88)
Powell	93 (201)	87 (122)	83 (130)	77 (110)
Shallow	25 (43)	25 (39)	21 (35)	21 (28)
Tri-diagonal	37 (50)	32 (45)	30 (39)	28 (32)
Dixon	24 (83)	20 (71)	20 (67)	16 (55)
Wood	88 (176)	79 (168)	72 (157)	68 (110)
Rosen	32 (78)	28 (77)	27 (69)	22 (57)
Sum	22 (65)	19 (57)	19 (55)	17 (49)
Total	345(800)	312(679)	294(650)	268(529)

Table (2)
Comparisons of all CG- algorithm for test functions with n=1000.

Test function	CG algorithm NOI (NOF)	PARTAN algorithm NOI (NOF)	New (1) NOI (NOF)	New (2) NOI (NOF)
Himmel	26 (112)	24 (100)	23 (98)	19 (92)
Powell	90 (197)	86 (138)	84 (141)	72 (99)
Shallow	24 (45)	21 (33)	21 (31)	18 (24)
Tri-diagonal	45 (58)	39 (53)	38 (53)	30 (47)
Dixon	27 (99)	25 (78)	21 (66)	18 (57)
Wood	92 (178)	79 (169)	74 (159)	69 (114)
Rosen	32 (82)	30 (78)	30 (69)	25 (60)
Sum	26 (71)	24 (69)	21 (66)	18 (61)
Total	362(842)	328(718)	312(683)	287(554)

Table (3) represents Duncan test to NOI for Table (1)

	T	Sample size=8	**Subset for alfa=0.5**
			1
Duncan*	(d)	8	69.2500
	(c)	8	85.3750
	(b)	8	89.7500
	(a)	8	105.2500
	Significant		0.149

Where:
(a) NOI for NEW(2) algorithm when n=100
(b) NOI for NEW(1) algorithm when n=100
(c) NOI for PARTAN algorithm when n=100
(d) NOI for CG algorithm when n=100

Table (4) represents Duncan test to NOF for Table (1)

	T	Sample size=8	Subset for alfa=0.05 1
Duncan*	(d)	8	33.6250
	(c)	8	39.0000
	(b)	8	41.0000
	(a)	8	45.2500
	Significant		0.424

Where:

(a) NOF for NEW(2) algorithm when n=100
(b) NOF for NEW(1) algorithm when n=100
(c) NOF for PARTAN algorithm when n=100
(d) NOF for CG algorithm when n=100

Table (5) represents Duncan test to NOI for Table (2)

	T	Sample size=8	Subset for alfa=0.05 1
Duncan*	(d)	8	66.1250
	(c)	8	81.2500
	(b)	8	84.8750
	(a)	8	100.000
	Significant		.184

Where:

(One) NOI for NEW(2) algorithm when n=1000
(b) NOI for NEW(1) algorithm when n=1000
(c) NOI for PARTAN algorithm when n=1000
(d) NOI for CG algorithm when n=1000

Table (6) represents Duncan test to NOF for Table (2)

	T	Sample size=8	**Subset for alfa=0.05**
			1
Duncan*	(d)	8	33.5000
	(c)	8	36.7500
	(b)	8	39.0000
	(a)	8	43.1250
	Significant		.521

Where:

(One) NOF for NEW(2) algorithm when n=1000

(b) NOF for NEW(1) algorithm when n=1000

(c) NOF for PARTAN algorithm when n=1000

(d) NOF for CG algorithm when n=1000

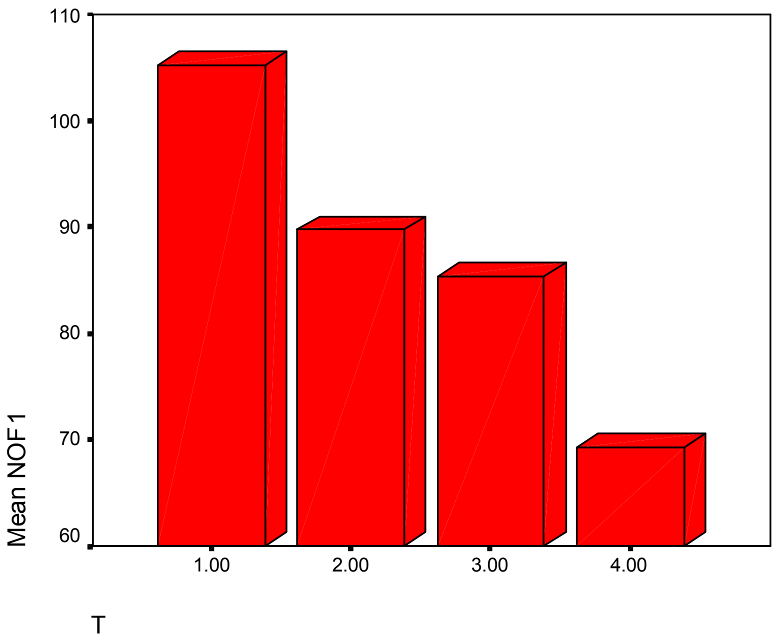

Figure (1) represents Table (3)

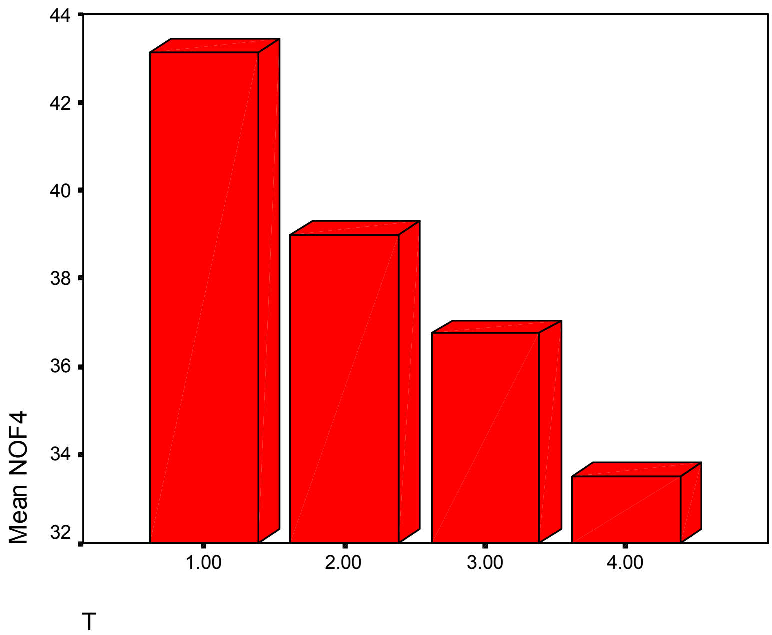

Figure (2) represents Table (4)

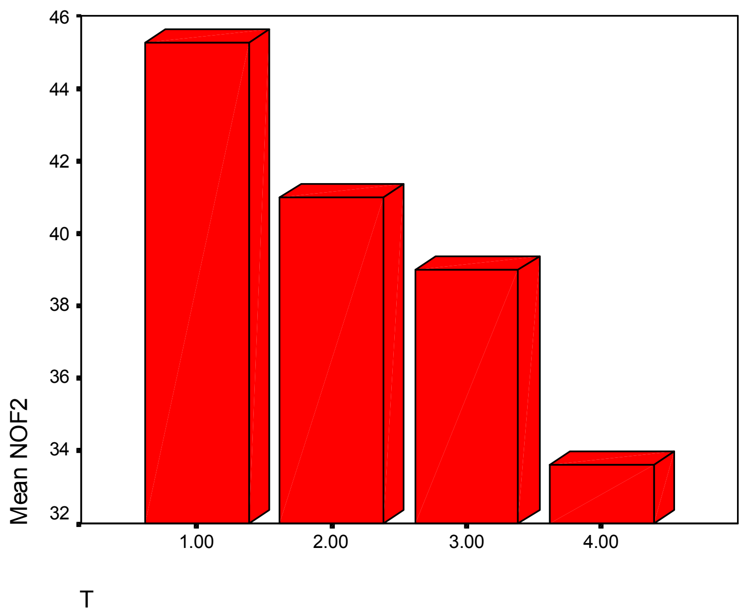

Figure (3) represents Table (5)

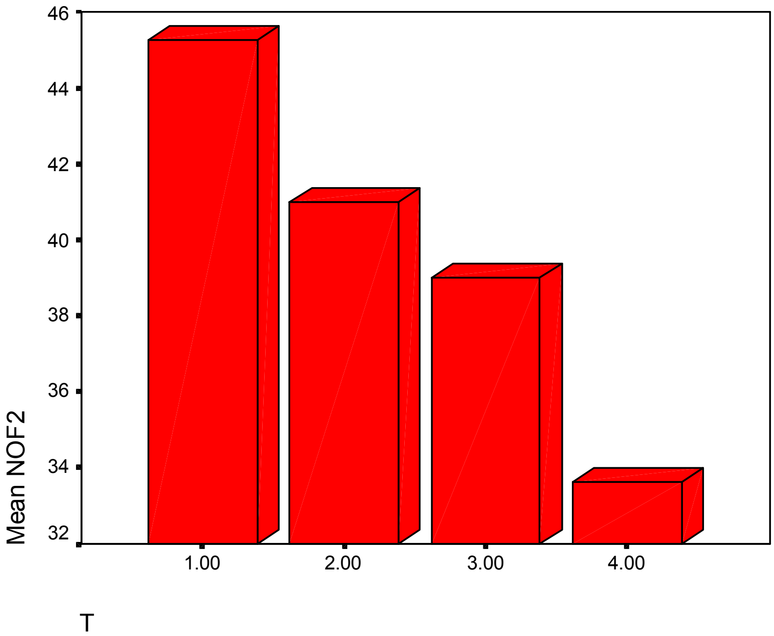

Figure (4) represent Table (6)

11. Appendix:

The unconstrained problems used are the following:

1- Generalized Edgar of Himmel function:

$$f=\sum_{i=1}^{n}[(x_{2i-1}-2)^4+(x_{2i}-2)^2.x_{2i}^2+(x_{2i}+1)^2]$$

$x_0=(1,0;\ldots)^T$.

2-Generalized Powell function:

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

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

3- Generalized Shallow function:

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

$x_0=(-2;\ldots)^T$

4- Generalized Tri-diagonal function:

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

$x_0=(1;\ldots)^T$

5- Generalized Wood function:

$$f=\sum_{i=2}^{n/4}[100(x_{4i-2}-x_{4i-3}^2)^2+(1-x_{4i-3})^2+90(x_{4i}-x_{4i-1}^2)^2+(1-x_{4i-1}^2)^2+10.1$$

$x_0=(-3,-1,-3,-1;\ldots)^T$

6- Generalized Dixon function:

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

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

7- Generalized Rosenbrock function:

$$f=\sum_{i=2}^{n/2}[100(x_{2i}-x_{2i-1}^2)^2+(1+x_{2i-1})^2$$

$x_0=(-1.2,1;\ldots)^T$

8- Sum of quartics function:

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

$x_0=(1;\ldots)^T$

12. References:

(1) Al-Assady, N.H. (1991), "New QN and PCG Algorithms Based on Non-Quadratic Properties" J.Educ. and Sci. Vol. (12), pp. 13-23.

(2) Al-Assady, N.H. and Al-Bayati, A.Y. (1986), "Conjugate Gradient Methods" Technical Research Report No.(1) School of Computer Studies, Leeds University, U.K.

(3) Al-Bayati, A.Y. and Al-Salih, M.S. (1994), "New VM-Methods for Nonlinear Unconstrained Optimization" to appear in the Education and Science Magazine, University of Mosul, Iraq.

(4) Al-Bayati, A.Y. and Ahmed, H. E. (1996), " On Single-Step Variable-Storage Conjugate Gradient Algorithm" Journal of Mu'uta, No. 2 ,PP. 183-192.

(5) Bazarra, S. and Shetty, C.M., (2000), " Nonlinear Programming: Theory and Algorithms", Newyork, Chichester, Brisbane, Toronto.

(6) Bunday, B. (1984), "Basic Optimization Methods" Edward Arnold Bedford Square, London.

(7) Dixon, L.C.W. (1975), "Conjugate Direction Without Linear Search" Journal of Institute Mathematics, 11, PP.317-328.

(8) Fletcher, R. and Reeves. C.M. (1964) " Function Minimization by Conjugate Gradient" Computer Journal, 7, PP. 149-154.

(9) Hestenes, M.R. and Stiefel, E. (1952) " Methods of Conjugate Gradients for Solving Linear System" Journal of Research of the National Burean of Standards, 49, PP.409-436.

(10) Khoda, K.M. ; Liu, Y. and Storey, C. (1992) " Generalized Polak-Ribiere Algorithm" Journal of Publishing Cosporation, 22, PP. 3239- 3343.

(11) Perry, A. (1978) "A Modified Conjugate Gradient Algorithm" Journal Of Operation Research, 26, PP.1073-1078.

(12) Polak, E. and Riebere, (1969) " Computational Methods in Optimization a Unified Approach" Academic Press, New York.

(13) Ronald, F.,(1971) "The Design of Experiments" Hafner press, New York.