

A New Dogleg Method for Solving the Trust-Region Subproblem

B.I. Oruh¹, and O.M. Bamigbola²

¹(Department of Mathematics, Michael Okpara University of Agriculture, Umudike, Nigeria)

²(Department of Mathematics, University of Ilorin, Ilorin, Nigeria)

Abstract: In this paper a new dogleg method for solving the trust region subproblem where convergence is based on constructing two paths is presented. The condition on the paths is incorporated into an algorithm to determine the optimum point of a smooth function. Numerical experiments with some classical problems showed that the new dogleg method is robust and efficient.

Keywords: Trust region, Trust region subproblem, Dogleg paths, Convergence

I. Introduction

Trust region methods for optimization problems have become very popular over the last decade. One possible explanation of this popularity is the fact that the method has wide application in many fields, such as science, engineering and economy, due to its strong global convergence and robustness ([1], [2], [3], [4]). Consider an unconstrained optimization problem.

$$\min f(w): w \in \mathfrak{R}^n \quad (1.1)$$

where \mathfrak{R}^n is an n-dimensional Euclidean space and $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ is a twice continuously differentiable function on \mathfrak{R}^n . Trust region strategy for solving problem (1.1) is based on the following outline:

- Given a bound Δ_k , called the trust region radius, and a current iterate $w_k \in \mathfrak{R}^n$ to the solution of (1.1), define a model $\varphi_k: \mathfrak{R}^n \rightarrow \mathfrak{R}$ of the objective function $f(w)$, in the region $\Omega_k[w_k, \Delta_k] := [w: \|w - w_k\| \leq \Delta_k]$, called the trust region, surrounding the current iterate w_k where the model is trusted to be accurate representation of $f(w)$. The model is often assumed to be the quadratic

$$\varphi_k(s) = s^T g_k + \frac{1}{2} s^T B_k s$$

for some symmetric matrix B_k , where $g_k = \nabla f(w_k)$ is the gradient of the objective function at the current approximate solution w_k .

- Compute a step s_k that sufficiently reduces the model φ_k and such that

$w_k + s_k \in \Omega_k$. This is achieved by solving the subproblem:

$$\min_{s \in \mathfrak{R}^n} \varphi_k(s) = s^T g_k + \frac{1}{2} s^T B_k s \quad (1.3)$$

$$\text{such that } \|s\| \leq \Delta_k \quad (1.4)$$

- Then compute the ratio

$$R_k = \frac{f(w_k) - f(w_k + s_k)}{\varphi_k(w_k) - \varphi_k(w_k + s_k)}$$

Then the trust region radius Δ_k and the iterate w_k are updated according to the value of R_k :

$$w_{k+1} = \begin{cases} w_k + s_k & \text{if } R_k \geq \eta_1 \\ w_k & \text{otherwise} \end{cases}$$

while

$$\Delta_{k+1} \in \begin{cases} (\Delta_k, \infty) & \text{if } R_k \geq \eta_2 \\ (\gamma_2 \Delta_k, \Delta_k) & \text{if } R_k \in (\eta_1, \eta_2) \\ (\gamma_1 \Delta_k, \gamma_2 \Delta_k) & \text{if } R_k < \eta_1 \end{cases}$$

where the parameters η_1, η_2, γ_1 and γ_2 satisfy

$$0 \leq \eta_1 < \eta_2 < 1 \quad \text{and} \quad 0 \leq \gamma_1 < \gamma_2 < 1 \quad (1.2)$$

Under some assumptions, this algorithm has been proved to be globally convergent to a local optimum ([1],[4],[5]).

Problem (1.3)-(1.4) which involves the computation of the step s_k such that $\|s\| \leq \Delta_k$ is called trust region subproblem. The problem is a very significant optimization problem(which involves solving a constrained optimization problem). As a result a lot of research efforts are still being directed towards finding better algorithms of solving the trust region subproblem.

Since each iteration of a trust region algorithm requires to solve(exactly or in exactly) the trust region subproblem (1.3)-(1.4), finding efficient solver for the trust region subproblem is very important. The trust region subproblem (1.3)-(1.4) has been studied by many authors. The following lemma is well known(for example see [6] and [7]):

Lemma 1.1

A vector $s^* \in \mathfrak{R}^n$ is a minimizer of

$$\min_{s \in \mathfrak{R}^n} \varphi_k(s) = s^T g_k + \frac{1}{2} s^T B_k s \quad (1.5)$$

$$\text{such that} \quad \|s\| \leq \Delta_k \quad (1.6)$$

if and only if there exists $\alpha^* \geq 0$ such that

$$(B + \alpha^* I) s^* = -g$$

$$\alpha^* (\Delta - \|s^*\|) = 0$$

$$(B + \alpha^* I) \text{ is positive semi-definite}$$

If $(B + \alpha^* I)$ is positive definite, s_k is unique.

However, there is no finite method for determining such an α . Most algorithms find an approximate solution of problem (1.3)-(1.4). Gould et al([8]) states that although in principle we are seeking the optimal solution to the subproblem, it is enough for global convergence purposes to find an approximate solution s_k that lies within the trust region and gives a sufficient reduction in the model. Research efforts are still being directed to finding ways of computing approximate solutions that are good enough to guarantee global convergence of the overall trust region algorithm. Toint ([9]) states that whatever method chosen for computing the approximate minimizer, s_{k+1} must compare favorably to a specific benchmark, the Cauchy point method. That is, the sufficient reduction can be quantified in terms of the Cauchy point method.

The Cauchy point, s_k^{cp} is defined [18] as $\ell_k s_k^p$ where

$$s_k^p = \min_{s \in \mathfrak{R}^n} f(w_k) + \nabla f(w)^T s \quad \text{such that} \quad \|s\| \leq \Delta_k \quad (2.10)$$

and

$$\ell_k = \arg \min_{\ell > 0} m_k(\ell s_k^p) + \nabla f(w)^T s \quad \text{s.t.} \quad \|s_k^p\| \leq \Delta_k \quad (2.11)$$

The solutions [11] to these problems are

$$s_k^p = - \frac{\Delta_k}{\|f(x_k)\|} \nabla f(x_k) \quad (2.12)$$

and

$$\ell_k = \begin{cases} 1 & \text{if } \Delta_k (w_k)^T B_k \nabla f(x_k) \leq 0 \\ \min \left[1, \frac{\|\Delta f(x_k)\|^3}{\Delta_k f(x_k)^T B_k \nabla f(x_k)} \right] & \text{otherwise} \end{cases}$$

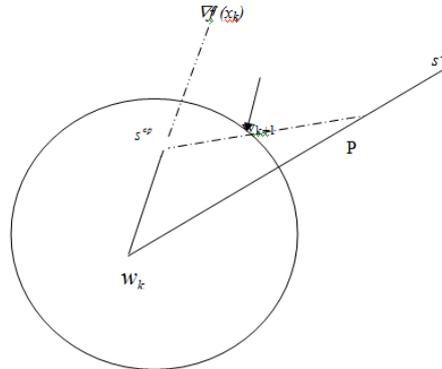
The problem with using the Cauchy-point at each iteration is that it is always in the steepest descent direction and therefore, performs poorly in terms of speed. Another method that is currently being used as an improvement to the Cauchy point approach is the dogleg method.

II. The Dogleg Method

The dogleg approach to solving the trust region subproblem originated from Dennis and Schnabel ([10]): The idea is as follows:

Rather than finding a point $w_{k+1} = w_k + s_k(\alpha)$ on the $s(\alpha)$ curve such that $w_{k+1} - w_k = \Delta_k$ as required by Lemma 1.1, the dogleg method approximates the curve by a piecewise linear function connecting the origin w_k , the Cauchy point s^{cp} , the Newton direction s^N as indicated in the figure below

Fig. 1



Then the method chooses w_{k+1} to be the point on this polygonal arc such that $\|w_{k+1} - w_k\| = \Delta_k$, unless $\|B_k^{-1} \nabla f(w_k)\| < \Delta_k$, in which case w_{k+1} is the Newton point. Schnabel argued that: If the curve is chosen in such a way that the distance from the origin w_k to s^{cp} increases monotonically as one moves along the piecewise linear curve, then for any $\Delta \leq \|B_k^{-1} \nabla f(w_k)\|$, there is a unique point w_{k+1} on the curve such that $\|w_{k+1} - w_k\| = \Delta_k$. This makes the process well defined. Thus the process will be reasonable if the value of the model $\varphi_k(w_k + s)$ decreases monotonically as s goes along the curve from w_k to s^{cp} and to P and to s^N .

To pursue this objective Schnabel ([10]) showed that the Cauchy point s^{cp} is not further away from w_k than the Newton point s^N . Thus he showed that

$$\|s^{cp}\| \leq \gamma \|s^N\| \leq \|s^N\|$$

where $\gamma < 1$. Based on this argument Schnabel proposed the choice of a point P on the Newton direction to have the form

$$p = w_k - \eta B_k^{-1} \nabla f(w_k) \tag{2.14}$$

for some $\eta \in (\gamma, 1)$. Then using heuristic he set,

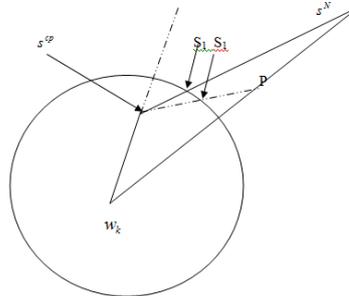
$$\eta = 0.8\gamma + 0.2 \tag{2.15}$$

and called this method dogleg method. The algorithm we propose in this paper is close to the Schnabel's ideals but one that removes heuristic in the choice of the step length η .

III. A New Dogleg Method

In the dogleg approach we observe that the piecewise line connecting the Cauchy point s^{cp} and the Newton point s^N can leave the trust region through two points instead of one point as shown in the diagram below:

Fig.2



The points S_1 and S_2 are the points through which the path could leave the trust region, and if they are such that the model of the objective function decreases as we move along the path from w_k to s^N , then each of the points is a candidate for the next iterate whenever the constrained optimum lies outside the trust region, that is whenever $\|B_k^{-1} g_k\| > \Delta_k$.

Our proposal takes these two paths into consideration. One path involves two points, the points s^{cp} and s^N while the other paths involves three points, the points s^{cp} , $P (= \eta s^N)$ and s^N . Thus, if we imagine the minimizing step $s = s(\Delta)$ as a function of the trust region radius, then $\Delta \rightarrow w + s(\Delta)$ traces a curve in \mathfrak{R}^n that starts from w (when $\Delta = 0$) and stops at $w - H^{-1} \nabla f(w)^T$ when $\Delta = \|H^{-1} \nabla f(w)^T\|$. We can then approximate the curve either by two straight line segments or three line segments with the same end point. We thus define the path by

$$s(t) = \begin{cases} s^{cp} + t (s^N - s^{cp}) & 0 \leq t \leq 1, \eta=1 \\ s^c + t (\eta s^N - s^{cp}) & 0 \leq t \leq 1, \eta < 1 \end{cases} \quad (3.1)$$

We make the same assumption in Schnabel ([10]) that although the distance increases as we proceed along the piecewise linear move, the value of the model decreases all the way.

Theorem 3.1

Assuming that B is positive definite, then

- (i) $\varphi_k(s(t))$ is a decreasing function of t , and
- (ii) $\|s(t)\|$ is an increasing function of t .

Proof: Using (2.1),

- (i) We only need to show that $\frac{d}{dt} \varphi_k(s(t)) < 0$ for $t \in (0, 1)$. Using equation (2.1), we will have

$$\begin{aligned} \frac{d}{dt} \varphi(s(t)) &= \frac{d}{dt} \left[s(t)^T g_k + \frac{1}{2} s(t)^T B_k s(t) \right] \\ &= s'(t)^T g_k + s(t)^T B_k s'(t) + (s^N - s^c)^T g_k + [s^c + t (s^N - s^c)]^T B_k (s^N - s^c) \\ &= (s^N - s^c)^T g_k + (s^c)^T B_k (s^N - s^c) + t (s^N - s^c)^T B_k (s^N - s^c) \\ &= (s^N - s^c)^T (g_k + B_k s^c) + t (s^N - s^c)^T B_k (s^N - s^c) \end{aligned}$$

$$\begin{aligned}
 &< (s^N - s^c) [g_k + B_k s^c + (s^N - s^c) B_k] \\
 &= (s^N - s^c) [g_k + B_k s^c + B_k s^N - B_k s^c] \\
 &= (s^N - s^c) [g_k + B_k s^c + t (s^N - s^c) B_k], \text{ since } 0 < t < 1 \\
 &= (s^N - s^c) [g_k + B_k s^N] \\
 &= (s^N - s^c) [g_k + B_k (-B^{-1} g_k)] \\
 &= 0
 \end{aligned}$$

Thus $\frac{d}{dt}(s(t)) < 0$

i.e, $\varphi_k(s(t))$ is a decreasing function of t.

(ii) Here we need to prove that $\frac{d}{dt} \|s(t)\| \geq 0$ for $t \in (0, 1)$

Now,

$$\|s(t)\|^2 = \|s^c\|^2 + 2t(s^c)^T (s^N - s^c) + t^2 \|s^N - s^c\|^2$$

$$\begin{aligned}
 \frac{d}{dt} \|s(t)\| &= \frac{d}{dt} \left(\frac{1}{2} \|s(t)\|^2 \right) \\
 &= (s^c)^T (s^N - s^c) + t \|s^N - s^c\|^2 \\
 &\geq (s^c)^T (s^N - s^c) \\
 &= - \frac{g_k^T g_k}{g_k^T B_k g_k} \left[-B^{-1} g_k + \frac{g_k^T g_k}{g_k^T B_k g_k} \right] \\
 &= - \left(\frac{g_k^T g_k}{g_k^T B_k g_k} \right) (g_k^T B_k^{-1} g_k) \left[-1 + \frac{(g_k^T g_k)^2}{(g_k^T B_k^{-1} g_k)(g_k^T B_k g_k)} \right] \\
 &= \left(\frac{g_k^T g_k}{g_k^T B_k g_k} \right) (g_k^T B_k^{-1} g_k) \left[1 - \frac{(g_k^T g_k)^2}{(g_k^T B_k^{-1} g_k)(g_k^T B_k g_k)} \right] \tag{3.2}
 \end{aligned}$$

Now by Cauchy Schwartz's inequality, $|U^T V| \leq \|U\| \|V\|$ for any vectors U and V.

Thus,

$$\begin{aligned}
 (g_k^T B_k^{-1} g_k)(g_k^T B_k g_k) &= \|(g_k^T B_k^{-1} g_k)\| \|(g_k^T B_k g_k)\| \\
 &= \|g_k\| \|B_k^{-1}\| \|g_k\| \|B_k\| \|g_k\| \\
 &= \|g_k\|^4 \|B_k^{-1}\| \|B_k\| \\
 &\geq \|g_k\|^4 \|B_k^{-1} B_k\| = \|g_k\|^4
 \end{aligned}$$

$$\begin{aligned}
 \text{i.e. } & (\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{g}_k)(\mathbf{g}_k^T \mathbf{B}_k \mathbf{g}_k) \geq (\mathbf{g}_k^T \mathbf{g}_k)^2 \\
 & \Rightarrow 1 \geq \frac{(\mathbf{g}_k^T \mathbf{g}_k)^2}{(\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{g}_k)(\mathbf{g}_k^T \mathbf{B}_k \mathbf{g}_k)} \\
 \text{i.e. } & \frac{(\mathbf{g}_k^T \mathbf{g}_k)^2}{(\mathbf{g}_k^T \mathbf{B}_k^{-1} \mathbf{g}_k)(\mathbf{g}_k^T \mathbf{B}_k \mathbf{g}_k)} \leq 1 \tag{3.5}
 \end{aligned}$$

Using (3.5) in (3.2) we have

$$\frac{d}{dt} \|s(t)\| \geq 0$$

Thus our construction of the dogleg path satisfies the motives behind the Dennis and Schnabel's construction. Thus as d moves steadily away from the current point w_k to the full Newton update $w + s^N$, the model values φ_k steadily decrease along the whole path. It is therefore clear that $s(t)$ will intersect the trust region boundary $\|s\| = \Delta$ at exactly one point if the unconstrained optimal point is outside the trust region. Since φ_k is decreasing along the path, the new direction is either s^N if $\|s^N\| \leq \Delta$ or the point of intersection of the path and the boundary. Putting this together, we present the following algorithm for solving approximately, the trust region subproblem.

Algorithm 3.1

At iteration k ,

1. Compute $s_k^c = -\frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_k^T \mathbf{B}_k \mathbf{g}_k}$
 If $\|s_k^c\| \geq \Delta_k$ then stop with $s_k = -\frac{\Delta_k}{\|\mathbf{g}_k\|} \mathbf{g}_k$
2. Compute $s_k^N = -\mathbf{B}_k^{-1} \mathbf{g}_k$
 If $\|s_k^N\| \leq \Delta_k$ then stop with $s_k = s_k^N$;
 else
3. find t^* such that
 $\|s_k^c + t(s_k^N - s_k^c)\| = \Delta_k$

IV. Numerical Examples

We tested our algorithms on a set of test problems from CUTE collection established by Bongartz, Conn, Gould and Toint ([11]). We choose the test problems and their initial points from the literature ([11]) with some of them highly nonlinear. The stopping criterion is $\|\mathbf{g}_k\| \leq 10^{-8}$. Our computational results for the test problems are summarized in Table 4.1 – 4.3, where we compared our method with the standard dogleg method. In table 4.3 we give particularly the results of three test problems that, according to Verdi([12]), are often used to test minimization algorithms. There in table 4.3, comparisons of our method, standard dogleg, with Dennis-Mei's MINOP, Powell's MINFA, Davidon's OCOPT, Fletcher's VM01 and Verdi's FMIN are made.

Example 4.1 Wood Function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1), \quad x_0 = [-3, -1, -3, 1]$$

Example 4.2

$$f(x) = 100(x_2 - x_1)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 \quad x_0 = [-1.2, 1, -1.2]$$

Example 4.3 Powell's Function

$$\text{Minimize } f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4, x_0 = [3, -1, 0, 1]^T$$

Example 4.4 $f(x) = 20 + x_1^3 + x_2^3 - 10(\cos 2\pi_1 x_1 + \cos 2\pi_2 x_2), x_0 = [0.1, 0.1]$

Example 4.5 Rosenbrock function

$$\text{Minimize } f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, x_0 = [-1.2, 1]$$

Numerical results are reported in tables 4.1 – 4.2

4.1 PRESENTATION OF NUMERICAL RESULTS

The first column in table 4.1 is the name of the test problem, and the second column is the number of variables n. For the NEW DOGLEG method and the standard dogleg method (SDM) we record, among other things, the number of iterations. The first column in table 4.2 is also the name of the test problem, and the second column is the number of variables n. For the NEW DOGLEG and the standard dogleg method we also record the number of iterations, the number of function evaluations and the number of gradient evaluations.

Table 4.1 COMPARISON WITH STANDARD DOGLEG METHOD

Ex	x ₀	New Dogleg method			Standard Dogleg method			Exact Solution	
		x*	f*	No. of Itn.	x*	f*	No. of Itn.	x*	f*
1	$\begin{pmatrix} -3 \\ -1 \\ -3 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 1.000000 \\ 1.000000 \\ 1.000000 \\ 1.000000 \end{pmatrix}$	0	36	*	*	*	0	0
2	$\begin{pmatrix} -1.2 \\ 1 \\ -1.2 \end{pmatrix}$	$\begin{pmatrix} 0.999697 \\ 0.999997 \\ 0.999993 \end{pmatrix}$	0	69	$\begin{pmatrix} 0.999619 \\ 0.999630 \\ 0.999253 \end{pmatrix}$	0	434	$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$	0
3	$\begin{pmatrix} -3 \\ -1 \\ 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0.001697 \\ 0.000170 \\ 0.000725 \\ 0.000725 \end{pmatrix}$	0	85	$\begin{pmatrix} 0.007062 \\ 0.000706 \\ 0.003519 \\ 0.003519 \end{pmatrix}$	0	500	$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	0
4	$\begin{pmatrix} 0.1 \\ 0.1 \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	0	3	$\begin{pmatrix} 0.00000 \\ 0.00000 \end{pmatrix}$	0	3	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	0
5	$\begin{pmatrix} -1.2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1.00000 \\ 1.00000 \end{pmatrix}$	0	22	$\begin{pmatrix} 1.00000 \\ 1.00000 \end{pmatrix}$	0	85	$\begin{pmatrix} 1 \\ 1 \end{pmatrix}$	0

* Not available

Table 4.2 Comparisons with Dennis-Mei's MINOP, Powell's MINFA, Davidon's OCOPT, Fletcher's VM01 and Verdi's FMIN are made.

Problem Name	Starting Point	New Dogleg method	MINOP	MINFA	OCOPR	VMM01	PMIN
--------------	----------------	-------------------	-------	-------	-------	-------	------

WOOD FUNCTION	(-3, -1, -3, -1)	36(151)	71(61)	117(117)	72(50)	129(129)	66(44)
ROSENBROCK FUNCTION	(-1.2,1)	22(74)	28(25)	37(37)	54(43)	28(28)	27(18)
POWELL'S FUNCTION	(3,-1,0,1)	37(116)	37(35)	**	**	**	33(28)

** (Not available)

4.2 Remarks on Numerical Results

Table 4.1 Presents the main results obtained by the new dogleg algorithm while tables 4-2 and 4.3 compares the new results with recently reported results. A general observation in these tables is that the performance of the new dogleg method compares favorably with the standard dogleg method. An important observation is that the new dogleg method requires far less number of iterations to obtain the optimum than some recently reported results and this translates into smaller computer times. Chh-Chen Lin and Jorge J. More [13] noted that fewer number of iterations often translates into smaller computing times. Example 5 is regarded in literature as a tricky problem because it is often difficult to minimize. But the new dogleg method solved this problem with remarkably few numbers of iterations.

V. Conclusion

In this paper, we presented a new dogleg method for solving trust-region subproblem. The results show that the new method is effective, accurate and converges in fewer iterations.

References

- [1]. Y.H. Dai and D.C. Xu, A new family of trust region algorithm for unconstrained optimization, J. Computational. Mathematics 21, 2003, 221-228
- [2]. E.M. Gertz, A quasi-Newton trust-region method. Mathematical Programming Ser. A, 100, 2004, 447 – 470
- [3]. J.E. Dennis(JR), M. El-Alem and M.C. Maciel, A global convergence theory for general trust-region based algorithms for equality constrained optimization, SIAM Journal on Optimization, 7(1), 1997, 177 - 207.
- [4]. Shultz G.A., Schnabel R.B. and R.H. Byrd, A family of trust-region-based algorithms for unconstrained minimization with strong global convergence, SIAM J. Number. Analy. 22, 1985, 47 – 67.
- [5]. Ph.L. Toint, Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space, IMA J. Numer. Analy., 8 , 1986, 231 – 252.
- [6]. N.I.M. Gould, D. Orban, A. Sartenaer and Ph.L. Toint, Sensitivity of trust-region algorithm to their parameters 4 OR 3 (2005) 227 – 241.
- [7]. J.H. Fu and W.Y. Sun, Nonmonotone adaptive trust-region method for unconstrained optimization problems. Applied Mathematics and Computation 163, 2005, 489 – 504.
- [8]. Y.H. Dai and D.C. Xu, A new family of trust region algorithm for unconstrained optimization, J. Computational. Mathematics 21, 2003, 221-228
- [9]. N.I.M. Gould, D. Orban, and Ph.L. Toint, Numerical methods for large-scale nonlinear optimization, Acta Number, 14, 2005, 299 – 361.
- [10]. J.E. Dennis (JR) and Schnabel R.B, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.
- [11]. I. Bongartz, A.R. Conn, N.I.M. Gould, and Ph.L. Toint, (2003) CUTE: Constrained and Unconstrained Testing Environment. Revisit ACM Transactions on Mathematical Software 29(4): 373 – 394.
- [12]. A. Verdi, A Trust Region Algorithm for unconstrained. Technical Report, No. 80-85, 1980, Institute for Computer Applications in Science and Engineering, NASA Langley Research Centre, Hampton, Virginia.
- [13]. Lin. Chin-Jen and J. J. More, Newton's Method for Large Bound-Constrained Optimization Problems. SIAM Journal on Optimization vol. 9, No.4, 1999, 1100-1127.