

## A Modified Trust-Region Method for Solving Unconstrained Optimization

Mushtaq A.K. Shiker and Zahra Sahib

Department of Mathematics, College of Education for Pure Science,  
University of Babylon, Hillah, Iraq

---

**Abstract:** In this research, we present a new radius for a modified trust region method and used them to solve the large-scale unconstrained optimization. Our approach increases and improves the robustness and efficiency of the trust-region frameworks as well as decrease the computational cost of the algorithm by decreasing the number of the trust-region subproblems that must resolved when the trail step rejected. Theoretical analysis shows that the new approach conserve the global convergence to the first-order critical points under classical assumptions. Moreover, the superlinear and the quadratic convergence are established under suitable conditions. The numerical results show that the new method is effective and promising for solving unconstrained optimization problems.

**Key words:** Unconstrained optimization, trust-region method, global convergence, promising, numerical, optimization

---

### INTRODUCTION

Consider the following optimization problem:

$$\min f(x), x \in F \quad (1)$$

Where:

$f(x)$  = The objective function

$F \subset \mathbb{R}^n$  = Feasible region or conditions set

One of the most important cases is when  $F = \mathbb{R}^n$  in this case, problem (Eq. 1) is called an unconstrained optimization problem.

The optimization comes in large variety of fields such as Applied Mathematics, Computer Science, Engineering, Economics, etc. To optimize any object, first, we must determinate the measures of some objective and quantitative of the problem under study. The objective could be time, profit or any quantity or combination of quantities. In the simplest case, an optimization is used to find the best value of the variables that make the objective function in the best form (maximization or minimization). The process of determination the components of the given problem is known as modeling. The first step-sometimes the most important step in the optimization process is the construction an appropriate model. When the model is very simplistic, then it will not give useful ideas for the practical problem and in the other hand, if it is very complex, it may become too difficult to solve.

The efficient algorithm for solving optimization problems doesn't require too much computation time and

data storage. The performance of an algorithm is indicated by the number of iterations, the number of function calculations and the required time to run the algorithm (CPU time).

There are many algorithms for solving the optimization problems, such like the line search such as algorithms which are the classical methods for optimization and the Trust Region Methods (TRM).

The (TRM) are one of the famous methods in this field. These methods are used also to solve the nonlinear systems of equations. They begin with defining a region around the current best solution in which a certain model can make some extent approximation to the original objective function. Then, the next step is taken according to the model depicts within the region. Unlike the line search methods, (TRM) usually determines the step size before improving the direction (or at the same time). If the value of the function decreases, the model is believed to be a good representation to the original objective function.

The concept of (TRM) has developed over 50 years. The first study in this area appeared in such as (1944) by Levenberg (More, 1978) who consider adding a multiple of the identity to the Hessian matrix as a such as stabilization procedure in the context of the solution of nonlinear least-squares problems. In 1983 such as a review paper in (TRM) was given by More (1983). Recently, Conn, Gould and Toint have such as finished an enormous monograph on (TRM) (Conn *et al.*, 1999). Most researches on trust region algorithms are such as done in the last 20 years. In recent years, the trust region methods are very necessary and such as active

methods in the area of unconstrained optimization and its algorithms have attracted such attention from more and more researchers.

The iterative methods for optimization are classified in two classes, one class is called the line search methods and the other is the trust region methods. Trust region methods are iterative method in which a model ( $m_k$ ) approximate the objective function  $f(x)$  and this model is minimized in a neighborhood of the current iterate.

TRM start with an initial point  $x_0$  and with the arbitrary approximation initial radius  $\Delta_0$ , then the following subproblem must be solved:

$$\text{Min } m_k(x_k + d) = f_k + d^T g_k + \frac{1}{2} d^T B_k d \text{ s.t. } \|d\| \leq \Delta_k \quad (2)$$

Where:

$g_k \nabla f(x_k)$  = The gradient at ( $x_k$ )

$B_k$  = The approximates of the Hessian of  $f(x)$

$\Delta_k$  = The radius

Many researchers used the following subproblem to find the trial step  $d_k$ :

$$\min_{d \in \mathbb{R}^n} g_k^T d + \frac{1}{2} d^T B_k d = \varnothing_k d \text{ s.t. } \|d\|_2 \leq \Delta_k \quad (3)$$

Then the ratio  $r_k$  must be defined and computed:

$$r_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \quad (4)$$

Where:

$f(x_k) - f(x_k + d_k)$  = The actual reduction

$m_k(x_k) - m_k(x_k + d_k)$  = The predicate reduction

$x_k$  = The current iterate

$d_k$  = The trial step direction

The next iteration will be decided depending on the value of  $r_k$ , either setting  $r_k$  or  $x_{k+1} = x_k$ , or  $x_{k+1} = x_k + d_k$ . Then when  $r_k$  closes to 1, then the model has a good agreement with the original problem at the current iterate  $x_k$ . If  $r_k$  is greater than a positive constant  $\mu$ , then, we will accept the trial step  $d_k$  and  $x_{k+1} = x_k + d_k$ , so, the trust-region radius can be expanded or kept the same. If  $r_k$  is near to zero or negative, then the trust-region radius must be diminished and the subproblem (Eq. 2) must be solved again to possibly find an acceptable trial point in the sequel of the process (Nocedal and Wright, 2006).

(TRM) can be traced back to the classical Levenberg-Marquardt method for nonlinear equations  $F(x) = 0$  which chooses the step as follows:

$$d_k = - \left( J(x_k) J(x_k)^T + \lambda_k I \right)^{-1} J(x_k)^T F(x_k) \quad (5)$$

Where:

$J(k)$  = The Jacobian matrix of  $F(x)$

$\lambda_k \geq 0$  = Parameter which is updated from iteration to iteration

The original idea of Levenberg-Marquardt method is to overcome the ill condition of  $J(x_k)$  by introducing the parameter  $\lambda_k$ , in other words, to prevent  $\|d_k\|$  being too large.

In this study, a new adaptive radius has been introduced to improve the trust-region methods for solving an optimization problems, that is prevent the increasing and decreasing the radius by controlling the size of the radius of the trust-region algorithm.

**Trust region algorithm:** The first issue of the trust region algorithms relates to how to choose the trust region radius  $\Delta_k$  at each iteration. The ratio  $r_k$ , we will note since, ( $p_k$ ) is obtained by minimizing the model ( $m_k$ ) on a region that includes  $p = 0$ , the predicted reduction always be non-negative, thus, if ( $r_k$ ) is negative then  $f(x_k + p_k)$  [objective value] is greater than  $f(x_k)$  [current value], so, the step should be rejected. If ( $r_k$ ) is closed to 1, then there is a good agreement between ( $m_k$ ) and ( $f(x_k)$ ) on this step, so, it is safe to expand the trust region for next iteration. If ( $r_k$ ) is positive, then, we don't alter (TR) but close to zero or negative (Amini *et al.*, 2016) as well as before the implementation of the trust region algorithms, we must define all the parameters, we use in algorithms.

## MATERIALS AND METHODS

**Trust region subproblem:** The trust region subproblems are one of the basic parts of trust region algorithms, since, in each iteration in the requires solution, a trust region sub problem (Eq. 3) must be solved. The trust region subproblem presents two modulations:

- Updating the radius in different way if hard case occurs
- Using techniques and sensitivity analysis are provided followed by the algorithms on unconstrained optimization

The trust region subproblem are using to minimize the trust region methods and working to improve them. Below, our trust region algorithm for unconstrained optimization with new adaptive radius.

**Algorithm 1 (The trust region algorithm for unconstrained optimization):**

Given  $x_0 \in \mathbb{R}^n$ ,  $\Delta^0 > 0$ ,  $\Delta_0 \in (0, \Delta^*)$ ,  $\epsilon \geq 0$ ,  $B_0 \in \mathbb{R}^{n \times n}$  is symmetric,  $0 < \mu_1 < \mu_2 < 1$ ,  $0 \leq \sigma_1, \sigma_2 < 1$ .

Step 1: For  $k = 0, 1, 2, \dots$

Use  $\min_{d \in \mathbb{R}^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k d$  to compute  $d_k$

Use  $r_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}$  to compute  $r_k$

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k > \mu \\ x_{k+1} = x_k & \text{else} \end{cases}$$

Step 2: Compute the new radius:

$$\Delta_{k+1} = \begin{cases} \frac{1}{2} \|d_k\| & \text{if } r_k < \frac{1}{4} \\ \min \left( \frac{1}{2} \Delta_k, \Delta^* \right) & \text{if } r_k > \frac{1}{4} \text{ and } \|d_k\| = \Delta_k \\ \Delta_k & \text{Else} \end{cases} \quad (6)$$

Step 4: Update  $B_{k+1}$ ,  $k = k+1$ ; go to Step 1

End For

End

The next lemma proves that  $d$  is a solution of Eq. 3.

**Lemma 1; (More, 1983):** A vector  $d^*$  is a solution of the following problem where  $d^* \in \mathbb{R}^n$ :

$$\min_{d \in \mathbb{R}^n} g^T d + \frac{1}{2} d^T B d = \phi(d) \text{ s.t. } \|d\|_2 \leq \Delta \quad (7)$$

where,  $g \in \mathbb{R}^n$ ,  $B \in \mathbb{R}^{n \times n}$  is a symmetric matrix and  $\Delta > 0$ .

**Lemma 2; (More, 1983):** There is  $\lambda^* \geq 0$ , such that for a positive definite  $B + \lambda^* I$ ,  $\|d\|_2 \leq \Delta$  and  $\lambda^*(\Delta - \|d\|_2) = 0$ , we have:

$$(B + \lambda^* I) d^* = -g \quad (8)$$

where,  $\lambda^*$  is Lagrange multiplier. So, instead of identifying the exact lagrange multiplier  $\lambda^*$ , there are algorithms directly computing an approximation to the solution  $d^*$  of Eq. 6. There are three different approaches: the truncated conjugate gradient method, the 2-dimensional search method and the dog-leg method.

**Convergence properties:** The convergence can be ensured that the size of the (TRM) in each iteration would depend on improvement previously iterate. Overall, the (TRM) have the quadratic convergence rate while being globally convergent (Amini *et al.*, 2016).

The convergence of trust region algorithms for constrained optimization, similar to unconstrained optimization, depends on some lower bound condition of the predicted reduction of the form:

$$\text{pre } d_k \geq \delta \in_k \min \left[ \Delta_k, \frac{\epsilon_k}{\|B_k\|} \right] \quad (9)$$

where,  $\delta$  is some positive constant. To prove the convergence of algorithm 1, we benefit from Powell (1970), who gave the first convergence result for algorithm 1 under the assumption that the matrices  $B_k$  are bounded. Also, he proved that if there exist a subspace  $S$  of  $\mathbb{R}_n$  and  $d_s$  is any solution of the subproblem  $\min_{d \in S, \|d\|_2 \leq \Delta} \phi(d)$  and if  $g \in S$ , then:

$$\phi(0) - \phi(d_s) \geq \frac{1}{2} \|g\|_2 \min \left[ \Delta, \frac{\|g\|_2}{\|B\|_2} \right] \quad (10)$$

The relation Eq. 10 prevents the predicted reduction from being very small, unless either  $\|g\|_2 \Delta$  or  $\|g\|_2^2 / \|B\|_2$  is very small.

Now, if  $d_k$  satisfies the following property, then the global convergence can be showed as a long as the trial step  $d_k$ :

$$\phi_k(0) - \phi_k(d_k) \geq \mu \min \left[ \Delta_k, \frac{\|g_k\|_2}{\|B_k\|_2} \right] \quad (11)$$

where,  $\mu$  is some positive constant. After that it was shown that  $\|B_k\|_2 \leq \beta_1 k$  where  $\beta_1$  is any positive constant. (Powell, 1975, 1984).

**Lemma 3:** Assume that  $f(x)$  is differentiable and  $\nabla f(x)$  is uniformly Lipschitz continuous. Let  $x_k$  be generated by algorithm 1 with  $d_k$  satisfies Eq.10 for all  $k$ . If there exists a positive constant  $\delta$  such that  $\|g_k\|_2 \geq \delta > 0$ ,  $\delta > 0$ , for all  $k$ , then there exists a constant  $\eta > 0$ , such that  $\Delta_k \geq \eta / M_k$  holds for all  $k$  where  $M_k$  is defined by  $M_k = 1 + \max_{1 \leq i \leq k} \|B_i\|_2$ .

**Lemma 4:** Let  $\{\Delta_k\}$  and  $\{M_k\}$  be two sequences such that  $\Delta_k \geq \eta / M_k \geq 0$  for all  $k$  where  $\eta$  is a positive constant. Let  $J$  be a subset of  $\{1, 2, 3, \dots\}$  assume that  $\Delta_{k+1} \leq \mu \Delta_k$  for all  $k \in J$ ,  $\Delta_{k+1} \leq \mu_4 \Delta_k$  for all  $k \notin J$ ,  $M_{k+1} \geq M_k$  for all  $k$  and  $\sum_{k \in J} 1 / M_k < \infty$  where  $\mu_1 > 1$ ,  $\mu_4 < 1$  are positive constants, Then:

$$\sum_{k=1}^{\infty} 1 / M_k < \infty$$

**Theorem 1:** Assume that  $f(x)$  is differentiable and  $\nabla f(x)$  is uniformly Lipschitz continuous. Let  $x_k$  be generated by algorithm 1 with  $d_k$  satisfies Eq.10. If  $M_k$  defined by  $M_k = 1 + \max_{1 \leq i \leq k} \|B_i\|_2$  satisfy that:

$$\sum_{k=1}^{\infty} 1 / M_k < \infty$$

In algorithm 1, if we chose  $\epsilon = 0$  and if  $\{f(x_k)\}$  is bounded below, then:

$$\lim_{k \rightarrow \infty} \|g\|_2 = 0$$

So, the matrices  $B_k$  can be updated by some known quasi-Newton formulae such as Powell's or by the BFGS method. With some additional conditions, Shultz *et al.* (1985) reinforced theorem 1. The techniques of More and Sorensen (1983) are used to show that the trial step converges superlinearly.

## RESULTS AND DISCUSSION

In this study, some numerical experiments are reported to compare the performance of the new method along with the following three algorithms:

- DFPB1: this method is coming from Ahookhosh *et al.* (2013)
- M1 : this method is coming from Li and Li (2011)
- M2: this method takes the direction (2.8) in Li and Li (2011) with a different line search

The experiments were run on a PC with CPU 2.20 GHz and 8 GB RAM. All of the codes were written in MATLAB R2014b programming environment. The running of the codes will check if the presented data converges to the corresponding points. All of the algorithms terminate whenever  $\|F_k\| \leq 10^{-4}$  or the number of iterates surpasses 500000. In all of the algorithms, the parameters are specified as follows  $\mu_1 = 0.23$ ,  $\mu_2 = 0.25$ ,  $\sigma_1 = 0.3$ ,  $\sigma_2 = 0.35$ ,  $\epsilon = 10^{-4}$ .

The performances of these methods are compared with respect to the number of iterations, the number of function evaluations  $N_f$  and CPU time. In order to compare these algorithms some famous test problems are used where the dimensions are trapped between 5000-50000 for the following initial points (Ahookhosh *et al.*, 2013; Li and Li, 2011):

$$\begin{aligned} x_0 &= (10, 10, \dots, 10)^T, & x_1 &= (-10, -10, \dots, -10)^T, \\ x_2 &= (1, 1, \dots, 1)^T, & x_3 &= (-1, -1, \dots, -1)^T, \\ x_4 &= \left(1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n}\right)^T, & x_5 &= (0.1, 0.1, \dots, 0.1)^T, \\ x_6 &= \left(\frac{1}{n}, \frac{2}{n}, \dots, 1\right)^T, & x_7 &= \left(1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0\right)^T \end{aligned}$$

To have a comprehensive comparison for the above methods, the performance profile introduced by Dolan and More (2002) is used as a well-known

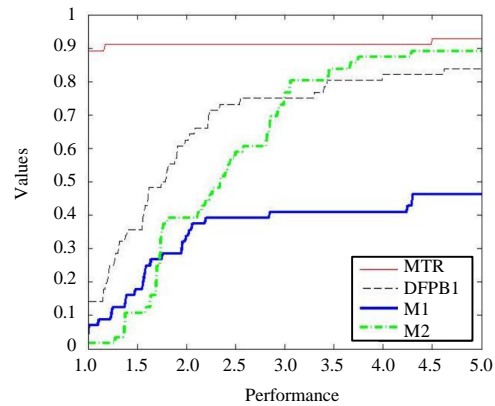


Fig. 1: Performance profile of the iteration number

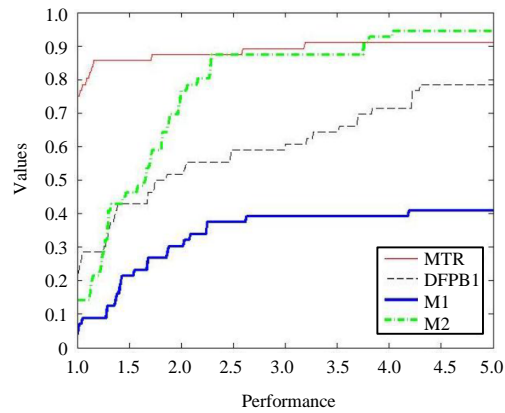


Fig. 2: Performance profile of the function evaluations

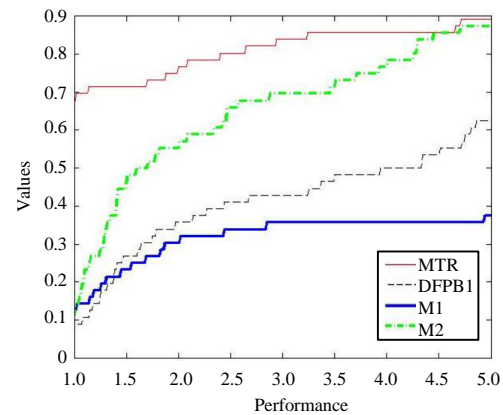


Fig. 3: Performance profile of CPU time

procedure to present some wealth information including efficiency and robustness. The proposed performance profiles of algorithms (MTR, DFPB1, M1 and M2) are exploited depending on the number of iterates, the number of function evaluations and CPU time in Fig. 1-3,

respectively. From these Fig. 1-3, it is easy to see that the new proposed method obtains the most wins on approximately 89, 75 and 70% of problems, respectively and this clearly shows the efficiency of the new algorithm.

## CONCLUSION

In this study, a new adaptive radius has been introduced to improve the trust-region methods for solving an optimization problems, that is prevent the increasing and decreasing the radius by controlling the size of the radius of the trust-region algorithm.

## REFERENCES

- Ahookhosh, M., K. Amini and S. Bahrami, 2013. Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations. *Numer. Algorithms*, 64: 21-42.
- Amini, K., M.A. Shiker and M. Kimiaei, 2016. A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. *4OR Q J. Oper. Res.*, 14: 133-152.
- Conn, A.R., N.I.M. Gould and P.L. Toint, 1999. *Trust Region Methods*. Monograph Publisher, St. Louis, Missouri.
- Dolan, E.D. and J.J. More, 2002. Benchmarking optimization software with performance profiles. *Math. Program.*, 91: 201-213.
- Li, Q. and D.H. Li, 2011. A class of derivative-free methods for large-scale nonlinear monotone equations. *IMA. J. Numer. Anal.*, 31: 1625-1635.
- More, J.J. and D.C. Sorensen, 1983. Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, 4: 553-572.
- More, J.J., 1978. The Levenberg-Marquardt Algorithm: Implementation and Theory. In: *Numerical Analysis*, Watson, G.A. (Eds.). Springer, Berlin, Germany, pp: 105.
- More, J.J., 1983. Recent Developments in Algorithms and Software for Trust Region Methods. In: *Mathematical Programming the State of the Art*, Bachem, A., B. Korte and M. Grottschel (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-68876-8, pp: 258-287.
- Nocedal, J. and S.J. Wright, 2006. *Numerical Optimization*. 2nd Edn., Springer, New York, USA.,.
- Powell, M.J.D., 1970. A New Algorithm for Unconstrained Optimization. In: *Nonlinear Programming*, Rosen, J.B., O.L. Mangasarian and K. Ritter (Eds.). Academic Press, New York, USA., pp: 31-65.
- Powell, M.J.D., 1975. Convergence Properties of a Class of Minimization Algorithms. In: *Nonlinear Programming 2*, Mangasarian, O.L., R.R. Meyer and S.M. Robinson (Eds.). Academic Press, New York, USA., pp: 1-27.
- Powell, M.J.D., 1984. Non-Convex Minimization Calculations and the Conjugate Gradient Method. In: *Numerical Analysis Lecture Notes in Mathematics 1066*, Griffiths, D.F. (Ed.). Springer, Berlin, Germany, pp: 122-141.
- Shultz, G.A., R.B. Schnabel and R.H. Byrd, 1985. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM. J. Numer. Anal.*, 22: 47-67.