A modified Trust-Region Method for Solving Unconstrained Optimization

Mushtaq A.K. Shiker **Zahra Sahib**

Department of Mathematics,

Collage of Education for Pure Science,

University of Babylon.

mmttmmhh@yahoo.com hussainaldeen@yahoo.com

Abstract. In this work, 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.

Keywords: Unconstrained optimization, Trust-region method, global convergence.

Introduction

Consider the following optimization problem

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

where f(x) is the objective function, $F \subseteq R^n$ is a feasible region or conditions set. One of the most important cases is when $F = R^n$, in this case , problem (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 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. Un like 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 paper in this area appeared in (1944) by Levenberg [6] who consider adding a multiple of the identity to the Hessian matrix as a stabilization procedure in the context of the solution of nonlinear least-squares problems. In 1983 A review paper in (TRM) was given by More' [7]. Recently, Conn, Gould and Toint have finished an enormous monograph on (TRM) [3]. Most researches on trust region algorithms are done in the last twenty years. In recent years, the trust region methods are very necessary and active methods in the area of unconstraint optimization, and its algorithms have attracted 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 Δ_0, then the following subproblem must be solved

$$Min\ m_k(x_k + d) = f_k + d^T g_k + \frac{1}{2} d^T B_k d \qquad (2)$$

$$\text{s.t. } \|d\| \leq \Delta_k,$$

where $g_k = \nabla f(\ x_k\ )$ is the gradient at (x_k), B_k is the approximates of the Hessian of $f(x)$ and $\Delta_k > 0$ is the radius.

Many authors used the following subproblem to find the trail step d_k

$$\min_{d \in R^n} g_k^T d + \frac{1}{2} d^T B_k = \emptyset_k d \qquad (3)$$

$$\text{s.t. } \|d\|_2 \leq \Delta_k,$$

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 $f(x_k) - f(x_k + d_k)$ is the actual reduction and $\mathrm{m_k(x_k)} - \mathrm{m_k(x_k + d_k)}$ the predicate reduction, where x_k is the current iterate and d_k is the trial step direction. The next iteration will be decided depending on the value of r_k, either setting $x_{k+1} = x_k$, 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 μ, 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 (2) must be solved again to possibly find an acceptable trial point in the sequel of the process [9].

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

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

where $J(x)$ is the Jacobian matrix of $F(x)$ and $\lambda_k \geq 0$ is a 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 λ_k, in other words, to prevent $\|d_k\|_2$ being too large.

In this paper, 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 Δ_k at each iteration. The ratio $\mathrm{r_k}$, 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 $(\mathrm{r_k})$ 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 $(\mathrm{r_k})$ is closed to 1, then there is a good agreement between (m_k) and (fx_k) on this step, so it is safe to expand the trust region for next iteration. If $(\mathrm{r_k})$ is positive, then we don't alter (TR), but close to zero or negative [2], As well as before the implementation of the trust region algorithms, we must define all the parameters we use in algorithms.

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 (3) must be solved.

The trust region subproblem presents two modulations:

(i) Updating the radius in different way if hard case occurs.

(ii) Using techniques and sensitivity analysis are provided followed by the algorithms on un constrained optimization.

The trust region subproblems 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 $x_0 \in R^n, \Delta' > 0,\ \Delta_0 \in (0, \Delta'), \epsilon \geq 0, B_0 \in R^{n\times n}$ is symmetric, $0 < \mu_1 < \mu_2 < 1$,

$0 \leq \sigma_1 \leq \sigma_2 < 1$.

Step 1: For *k=0, 1, 2, ...*

Use $min_{d\in R^n}\, g_k^T d + \frac{1}{2} d^T B_k = \emptyset_k d$ 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.

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

Step 2: Compute the new radius

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

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 (3) .

Lemma 1 [8]: A vector d^* is a solution of the following problem, where $d^* \in R^n$

$$\min_{d\in R^n} g^T d + \frac{1}{2} d^T B d = \emptyset(d) \tag{7}$$

$$\text{s.t. } \|d\|_2 \leq \Delta,$$

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

Lemma 2 [8]: 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

$$(B + \lambda^* I)\, d^* = -g, \tag{8}$$

where λ^* is Lagrange multiplier.

So, Instead of identifying the exact Lagrange multiplier λ^*, there are algorithms directly computing an approximation to the solution d^* of (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 [2].

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

$$pre\ d_k \geq \delta \epsilon_k \min\left[\Delta_k, \frac{\epsilon_k}{\|B_k\|}\right]. \tag{9}$$

where δ is some positive constant.

To prove the convergence of algorithm 1, we benefit from Powell [10], 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 R^n, and d_s is any solution of the subproblem $\min_{d\in S, \|d\|_2 \leq \Delta} \emptyset(d)$, and if g ∈ S, then

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

The relation (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,

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

where μ is some positive constant.

After that, it was shown that $\|B_k\|_2 \leq \beta_1 k,\ \forall\ k$, where β_1 is any positive constant [11], [12].
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 (10) for all *k*. If there exists a positive constant δ 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 $M_k = 1 + \max_{1\leq i\leq k}\|B_k\|_2$. .

Lemma 4 Let $\{\Delta_k\}$ and $\{M_k\}$ be two sequences such that $\Delta_k \geq \upsilon/M_k \geq 0$ for all *k*, where υ is a positive constant. Let J be a subset of {1, 2 ,3 , ... }, Assume that $\Delta_{k+1} \leq \mu_1\Delta_k,\ for\ all\ k \in J$), $\Delta_{k+1} \leq \mu_4\ \Delta_k$, 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 (10). If M_k defined by $M_k = 1 + \max_{1\le i\le k}\|B_k\|_2$ 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

$$\lim_{k\to\infty} \|g_k\|_2 = 0.$$

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 [13] reinforced theorem 1.
The techniques of Dennis and More' [8] are used to show that the trial step d_k converges superlinearly.

Numerical results

In this section, 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 [1].
M1 : This method is coming from Li and Li [5].
M2: This method takes the direction (2.8) in [5] 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 $\|\mathrm{F_k}\| \le 10^{-4}$, 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 N_i, 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 [1] [5]

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

To have a comprehensive comparison for the above methods, the performance profile introduced by Dolan and More´[4] is used as a well-known 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 Figures 1-3 respectively. From these Figures, 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.

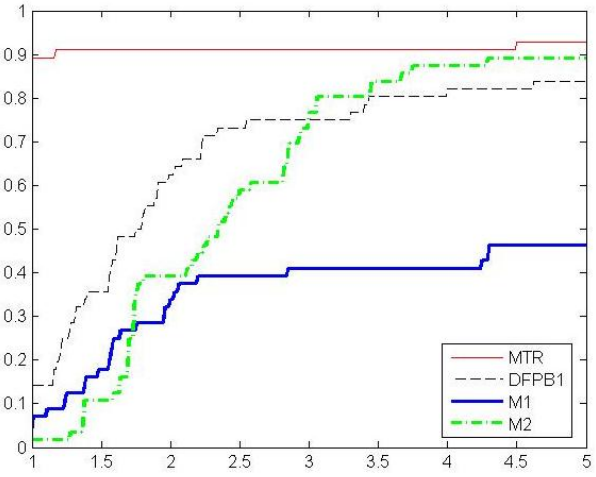

Fig. 1. Performance profile of the iteration number.

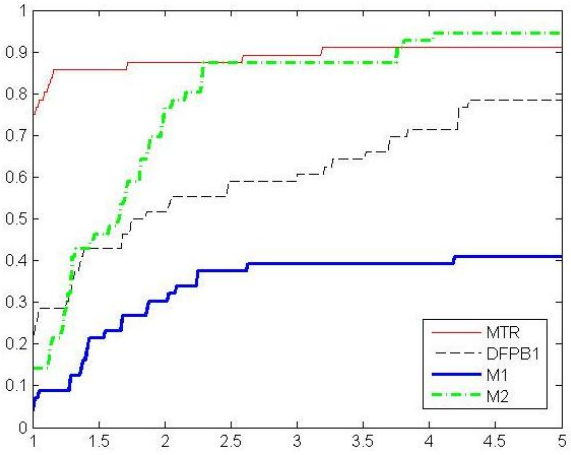

Fig. 2. Performance profile of the function evaluations

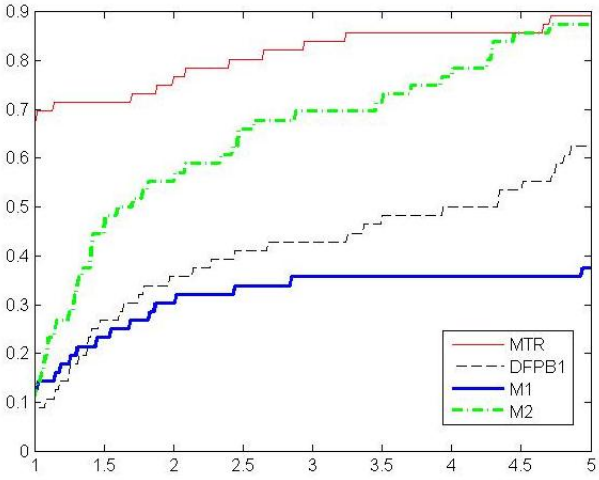

Fig. 3. Performance profile of CPU time

Reference

1. M. Ahookhosh, K., Amini, and S. Bahrami "Two derivative-free projection approaches for systems of large-scale nonlinear monotone equations" Numerical Algorithms, 2013, 64, pp: 21–42.

2. K. Amini, M. A. K. Shiker and M. Kimiaei "A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations" 4 OR- Journal of operation research, 2016, 14(2) , pp: 133-152.

3. A. R. Conn, N. I. M. Gould and Ph. L. Toint "Trust Region Methods" Monograph, to be published, 1999.

4. E. D. Dolan, and J. J. More´" Benchmarking optimization software with performance profiles" Mathematical Programming, 2002, 91, pp: 201–213.

5. Q. Li and D. H. Li "A class of derivative-free methods for large-scale nonlinear monotone equations" IMA Journal of Numerical Analysis, 2011, 31, pp: 1625–1635.

6. J. J. More' "The Levenberg-Marquardt algorithm: implementation and theory" in: G.A. Watson, ed. , Lecture Notes in Mathematics 630: Numerical Analysis (Springer-Verlag, Berlin), 1978, pp: 105-116.

7. J. J. More' " Recent developments in algorithms and software for trust region methods" in: A. Bachem, M. GrÄotschel and B. Korte, eds., Mathematical Programming: The State of the Art (Springer-Verlag, Berlin) , 1983, pp: 258-287.

8. J. J. More' and D. C. Sorensen "Computing a trust region step" SIAM Journal of Scientific Computer, 1983, 4, pp: 553-572.

9. J. Nocedal , S. J. Wright " Numerical Optimization " Springer, New York (2006).

10. M. J. D. Powell "A new algorithm for unconstrained optimization" in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., Nonlinear Programming (Academic Press, New York) , 1970, pp: 31-66.

11. M. J. D. Powell "Convergence properties of a class of minimization algorithms" in: O.L. Mangasarian, R.R. Meyer and S.M. Robinson, eds., Nonlinear Programming 2 (Academic Press, New York) , 1975, pp: 1- 27.

12. M. J. D. Powell " Non-convex minimization calculations and the conjugate gradient method" in: D.F. Griffiths, ed., Numerical Analysis Lecture Notes in Mathematics 1066 (Springer-Verlag, Berlin) , 1984, pp: 122-141.

13. G.A. Shultz, R.B. Schnabel and R.H. Byrd "A family of trust-region based algorithms for unconstrained minimization with strong global convergence" SIAM Journal of Numerical Analysis,1985, 22, pp: 47-67.