1

A review of trust region algorithms for optimization

Ya-xiang Yuan
State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics and Scientific/Engineering Computing,
Chinese Academy of Sciences, P. O. Box 2719, Beijing 100080, China [1]

Abstract

Iterative methods for optimization can be classified into two categories: line search methods and trust region methods. In this paper we give a review on trust region algorithms for nonlinear optimization. Trust region methods are robust, and can be applied to ill-conditioned problems. A model trust region algorithm is presented to demonstrate the trust region approaches. Various trust region subproblems and their properties are presented. Convergence properties of trust region algorithms are given. Techniques such as backtracking, non-monotone and second order correction are also briefly discussed.

1 Introduction

Nonlinear optimization problems have the form

$$\min_{x\in\Re^n} \quad f(x) \tag{1.1}$$
$$s.t. \quad c_i(x) = 0, \qquad i = 1, 2, \ldots, m_e; \tag{1.2}$$
$$c_i(x) \geq 0, \qquad i = m_e + 1, \ldots, m, \tag{1.3}$$

where $f(x)$ and $c_i(x)$ $(i = 1, \ldots, m)$ are real functions defined in $\Re^n$, at least one of these functions is nonlinear, and $m \geq m_e$ are two non-negative integers. If $m = m_e = 0$, problem (1.1) is an unconstrained optimization problem, otherwise it is a constrained problem.

Numerical methods for nonlinear optimization problems are iterative. At the $k-$th iteration, a current approximate solution x_k is available. A new point x_{k+1} is computed by certain techniques, and this process is repeated until a point can be accepted as a solution.

The classical methods for optimization are line search algorithms. Such an algorithm obtains a search direction in each iteration, and searches along this direction to obtain a better point. The search direction is a descent direction, normally computed by solving a subproblem that approximates the original op-

[1] The author would like to thank the Chinese NSF for supports

timization problem near the current iterate. Therefore, unless a stationary point is reached, there always exist better points along the search direction.

Trust region algorithms are a class of relatively new algorithms. The trust region approach is strongly associated with approximation. Assume that we have a current guess of the solution of the optimization problem, an approximate model can be constructed near the current point. A solution of the approximate model can be taken as the next iterate point. In fact, most line search algorithms also solve approximate models to obtain search directions. However, in a trust region algorithm, the approximate model is only "trusted" in a region near the current iterate. This seems reasonable, because for general nonlinear functions local approximate models (such as linear approximation and quadratic approximation) can only fit the original function locally. The region that the approximate model is trusted is called the trust region. A trust region is normally a neighbourhood centered at the current iterate. The trust region is adjusted from iteration to iteration. Roughly speaking, if the computations indicate the approximate model fit the original problem well, the trust region can be enlarged. Otherwise when the approximate model works not good enough (for example, a solution of the approximate model turns out to be a "bad" point), the trust region should be reduced.

The key contents of a trust region algorithm are how to compute the trust region trial step and how to decide whether a trial step should be accepted. An iteration of a trust region algorithm has the following form. A trust region is available at the beginning. Then an approximate model is constructed, and it is solved within the trust region, giving a solution s_k which is called the trial step. A merit function is chosen, which is used for updating the next trust region and for choosing the new iterate point.

Because of the boundedness of the trust region, trust region algorithms can use non-convex approximate models. This is one of the advantages of trust region algorithms comparing with line search algorithms. Trust region algorithms are reliable and robust, they can be applied to ill-conditioned problems, and they have very strong convergence properties.

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

$$d_k = -(J(x_k)J(x_k)^T + \lambda_k I)^{-1} J(x_k)F(x_k) \tag{1.4}$$

where $J(x)$ is the Jacobi matrix of $F(x)$ and $\lambda_k \geq 0$ is a parameter which is updated from iteration to iteration(see, [16]). The original idea of Levenberg-Marquardt method is to overcome the ill condition of $J(x_k)$ by introducing the parameter λ_k, or in other words, to prevent $||d_k||_2$ being too large. It is easy to see that d_k given by (1.4) is also a solution of the following problem

$$\min_{d \in \Re^n} \quad ||F(x_k) + J(x_k)^T d||_2^2 \tag{1.5}$$

$$s.\ t. \quad ||d||_2 \leq \Delta_k\ . \tag{1.6}$$

Because of the constraint (1.6), we can view the classical Levenberg-Marquardt method as a trust region algorithm. Indeed, a trust region algorithm for nonlinear least squares is similar to the Levenberg-Marquardt method, except that the bound Δ_k is updated from iteration to iteration instead of the parameter λ_k. Modern versions of Levenberg-Marquardt method are in fact trust region algorithms.

Pioneer works on trust region methods were done by Powell[21] and [22]. Most researches on trust region algorithms are done in the last twenty years. Now trust region algorithms have attracted attention from more and more researchers. An early review paper was given by More[17]. Recently, Conn, Gould and Toint have finished an enormous monograph on trust region methods[5].

2 A Model Trust Region Algorithm

In order to demonstrate how a trust region algorithm can be constructed, in the following we given a model algorithm for unconstrained optimization problem (1.1). At the $k-$th iteration, the trial step is computed by solving

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{2.1}$$

$$s.\ t. \quad ||d||_2 \le \Delta_k, \tag{2.2}$$

where $g_k = \nabla f(x_k)$ is the gradient at the current iterate x_k, B_k is an $n \times n$ symmetric matrix which approximates the Hessian of $f(x)$ and $\Delta_k > 0$ is a trust region radius. Let s_k be a solution of (2.1)-(2.2). The predicted reduction $Pred_k$ is defined by the reduction in the approximate model, $\phi_k(0) - \phi_k(s_k)$. Unless x_k is a stationary point and B_k is positive semi-definite, $Pred_k$ is always positive. The actual reduction $Ared_k = f(x_k) - f(x_k + s_k)$ is the reduction in the objective function. The ratio between the actual reduction and the predicted reduction $r_k = Ared_k/Pred_k$ plays an very important role in the algorithm. This ratio is used to decide whether the trial step is acceptable and to adjust the new trust region radius.

Now we can give a model trust region algorithm for unconstrained optimization as follows.

Algorithm 2.1 *(Trust Region Algorithm for Unconstrained Optimization)*

Step 1 *Given* $x_1 \in \Re^n$, $\Delta_1 > 0$, $\epsilon \ge 0$, $B_1 \in \Re^{n\times n}$ *symmetric;*
$0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \le \tau_0 \le \tau_2 < 1$, $\tau_2 > 0$, $k := 1$.

Step 2 *If* $||g_k||_2 \le \epsilon$ *then stop; Solve (2.1)-(2.2) giving* s_k.

Step 3 *Compute* $r_k = Ared_k/Pred_k$;

$$x_{k+1} = \begin{cases} x_k & \text{if } r_k \le \tau_0\ , \\ x_k + s_k & \text{otherwise}\ ; \end{cases} \tag{2.3}$$

Choose Δ_{k+1} *that satisfies*

$$\Delta_{k+1} \in \begin{cases} [\tau_3 ||s_k||_2,\ \tau_4\Delta_k] & \text{if } r_k < \tau_2, \\ [\Delta_k,\ \tau_1\Delta_k] & \text{otherwise;} \end{cases} \tag{2.4}$$

Step 4 Update B_{k+1}; $k := k+1$; go to Step 2.

The constants τ_i (i=0,..,4) can be chosen by users. Typical values are $\tau_0 = 0, \tau_1 = 2, \tau_2 = \tau_3 = 0.25, \tau_4 = 0.5$. For other choices of those constants, please see [13],[24], [17], etc.. The parameter τ_0 is usually zero (e.g. [13], [23]) or a small positive constant (e.g. [10] and [28]). Intuitively it seems reasonable to choose $\tau_0 = 0$ so that any computed "good" point would not be thrown away([37]). But, the global convergence result for the case $\tau_0 = 0$ is only

$$\liminf_{k\to\infty} \quad ||g_k||_2 = 0. \tag{2.5}$$

A stronger result $\lim_{k\to\infty} ||g_k||_2 = 0$ can be achieved if $\tau_0 > 0$. Recently an example is given by [39] to show that Algorithm 2.1 with $\tau_0 = 0$ may cycle near three points where two of these three points are non-stationary points. However, the weak convergence result (2.5) is sufficient in practical applications. In other words, given a positive tolerance ϵ, (2.5) is also sufficient to guarantee a finite termination of Algorithm 2.1. This ensures that the algorithm will stop at a point x_k satisfying the optimal criterion $||g_k||_2 \le \epsilon$.

3 Trust Region Subproblems

Trust region subproblems are one of the essential parts of trust region algorithms. Since each iteration of a trust region algorithm requires to solve (exactly or inexactly) a trust region subproblem, finding efficient solver for trust region subproblems is very important.

First, we consider subproblem (2.1)-(2.2) which has been studied by many authors. The following lemma is well known (for example, see [14] and [18]):

Lemma 3.1 *A vector $d^* \in \Re^n$ is a solution of the problem*

$$\min_{d\in\Re^n} \qquad g^T d + \frac{1}{2} d^T B d = \phi(d) \tag{3.1}$$

$$s.\ t. \qquad ||d||_2 \le \Delta \tag{3.2}$$

where $g \in \Re^n$, $B \in \Re^{n\times n}$ is a symmetric matrix, and $\Delta > 0$, if and and only if there exists $\lambda^* \ge 0$ such that

$$(B + \lambda^* I)d^* = -g \tag{3.3}$$

and that $B + \lambda^* I$ is positive semi-definite, $||d^*||_2 \le \Delta$ and $\lambda^*(\Delta - ||d^*||_2) = 0$.

To solve (3.1)-(3.2) is equivalent to find the unique λ^* which satisfies the conditions in the above theorem. Unless in the hard case([18]), λ^* is the root of

$$\psi(\lambda) = ||(B + \lambda I)^{-1} g||_2^{-1} - \Delta^{-1} = 0, \tag{3.4}$$

which can be computed by applying Newton's method. In the hard case, we have that $\lambda^* = -\sigma_n(B)$, where $\sigma_n(B)$ is the least eigenvalue of B. If $\sigma_n(B) = 0$, we can easily see that $-B^{+}g$ is a solution of problem (3.1)-(3.2). Hence the "real" hard case is that $\lambda^* = -\sigma_n(B) > 0$. For any $\lambda > -\sigma_n(B)$, the Newton step for (3.4) will normally make the matrix $B + \lambda_+ I$ have negative eigenvalues. Therefore, we need to have some safeguard techniques for updating the lower bound for $-\sigma_n(B)$. Another way is to apply Newton's method to the equation $\tilde{\psi}(\mu) = \psi(\frac{1}{\mu}) = 0$.

If B is positive definite, trust region subproblem (3.1)-(3.2) can be written as a problem that minimizes linear least squares subject to the ball constraint. Thus, it can also be solved by a method by Golub and von Matt[15]. Their method reformulates the problem into a quadrature problem and applies the Gauss quadrature technique.

Recently, semi-definite program is used for subproblem (3.1)-(3.2). Let $t^* = -\lambda^* - g^T s^*$. Then, we have that

$$\begin{pmatrix} t^* & g^T \\ g & B \end{pmatrix} \begin{pmatrix} 1 \\ s^* \end{pmatrix} = \lambda^* \begin{pmatrix} 1 \\ s^* \end{pmatrix}. \tag{3.5}$$

This shows that λ^* is an eigenvalue of the matrix

$$D(t^*) = \begin{pmatrix} t^* & g^T \\ g & B \end{pmatrix}. \tag{3.6}$$

Thus, (3.1)-(3.2) can be transformed into an parametric eigenvalue problem. Indeed, it can be shown that (3.1)-(3.2) is equivalent to

$$\max_{t} \; (\Delta^2 + 1)\sigma_{n+1}(D(t)) - t \tag{3.7}$$

$$s.t. \quad \sigma_{n+1}(D(t)) \leq 0, \tag{3.8}$$

where $\sigma_{n+1}(D(t))$ is the least eigenvalue of $D(t)$. Problem (3.7)-(3.8) can be solved by semi-definite program techniques. For details please see Rendl and Wolkowicz[26].

Instead of identifying the exact Lagrange multiplier λ^*, there are algorithms directly computing an approximation to the solution d^* of (3.1)-(3.2). There are mainly three different approaches: the dog-leg method, the 2-dimensional search method and the truncated conjugate gradient method.

The dog-leg method is due to Powell([21], [22]). This method uses the Newton step if it is in the trust region. Otherwise, it finds the Cauchy point which minimizes the objective function along the steepest descent direction. If the Cauchy point is outside the trust region, the truncated Cauchy step is taken as an approximate solution. When the Cauchy point is inside the trust region, the approximate solution is chosen as the intersection of the trust region boundary and the straight line joining the Cauchy point and the Newton step. The piecewise linear curve passing through the origin, the Cauchy point and the Newton step looks like a dog-leg. Therefore the method is called the dog-leg method. This technique is generalized to a double dog-leg method by Dennis and Mei[8].

The 2-dimensional search method is to minimize the objective function in the subspace spanned by the steepest direction and the Newton step within the trust region. The 2-dimensional search method was first suggested by Shultz, Schnabel and Byrd[27] and it can be regarded as an indefinite dog-leg method. On the other hand, the dog-leg step is the truncated conjugate gradient solution of the reduced equation $Bd = -g$ in the 2-dimensional subspace.

The truncated conjugate gradient method is simply to apply the standard conjugate gradient to the minimization of $\phi(d)$. This method is identical as the standard conjugate method as long as the iterates are inside the trust region. If the conjugate gradient method terminates at an point within the trust region, this point is a global minimizer of the objective function. Otherwise, at some iteration, either the new iterate is outside the trust region or a negative curvature direction is computed. In either case, we can get a truncated step which is on the trust region boundary. This method was proposed by Toint[30] and Steihaug[29]. One good property of this method is that the solution computed has a sufficient reduction property which was proved by Yuan[40].

Theorem 3.2 *Assume that ϕ is strictly convex. Let s^* be the exact solution of trust region subproblem, and s^*_{CG} be the truncated CG solution, then*

$$\frac{\phi(0) - \phi(s^*_{CG})}{\phi(0) - \phi(s^*)} \geq \frac{1}{2}. \tag{3.9}$$

It is not possible to prove a similar result for the two-dimensional subspace search method. Examples are given by Byrd, Schnable and Shultz[2] and Yuan[40] showing that the ratio $[\phi(0) - \phi(s^*_{2D})]/[\phi(0) - \phi(s^*)]$ can be arbitrary small, where s^*_{2D} is the 2-dimensional subspace solution.

Now, we consider trust region subproblems for constrained optimization. For simplicity we only consider equality constrained problems. Most trust region subproblems can be viewed as some kinds of modification of the SQP subproblem:

$$\min_{d \in \Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{3.10}$$

$$s.\ t. \quad c_k + A_k^T d = 0, \tag{3.11}$$

where $g_k = \nabla f(x_k)$, $c_k = c(x_k) = (c_1(x_k), ..., c_m(x_k))^T$, $A_k = \nabla c(x_k)^T$ and B_k is an approximate Hessian of the Lagrange function.

Trust region subproblems for constrained optimization can be classified into three groups. The first is the null space method, which decomposes the trial step into a null space step and a range space step, namely $s_k = v_k + h_k$. The range space step v_k and the null space step h_k are also called as the vertical step and the horizontal step respectively. For some constant $\eta \in (0, 1)$, the vertical step v_k minimizes $\|c_k + A_k^T d\|_2^2$ subject to $\|d\|_2 \leq \eta\Delta_k$. The null space step h_k can be computed by solving

$$\min \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{3.12}$$

$$s.t. \quad A_k^T d = 0, \quad \|d\|_2 \le \sqrt{1-\eta^2}\Delta_k. \tag{3.13}$$

The above two subproblems are to minimize quadratic functions subject to ball constraints. This approach is closely related to the SQP-type subproblem:

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{3.14}$$

$$s.\ t. \quad \theta_k c(x_k) + d^T \nabla c(x_k) = 0, \quad \|d\|_2 \le \Delta_k, \tag{3.15}$$

where $\theta_k \in (0,1]$ is a parameter (see Byrd, Schnabel and Shultz [1], Vardi [32], and Omojokun[20]).

Another trust region subproblem can be derived by exact penalty functions. The first such subproblem is the SL_1QP problem given by Fletcher[11]:

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||c_k + A_k^T d||_1 = \Phi_k(d) \tag{3.16}$$

$$s.t. \quad ||d||_\infty \le \Delta_k. \tag{3.17}$$

This subproblem can be written as a linear program. A similar subproblem based on the L_∞ exact penalty function is given by Yuan[38]. Trust region subproblems based on exact penalty functions are closely related to subproblems of trust region algorithms for nonlinear equations[10]. Algorithms that use (3.16)-(3.17) are also similar to trust region algorithms for nonsmooth optimization[13].

The third kind of trust region subproblems for constrained optimization is the CDT problem given by Celis, Dennis and Tapia[4]. This subproblem replaces the linearized constraints (3.11) by a single quadratic constraint.

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{3.18}$$

$$s.\ t. \quad ||c_k + A_k^T d||_2 \le \xi_k, \quad ||d||_2 \le \Delta_k, \tag{3.19}$$

where $\xi_k \ge 0$ is a parameter. Trust region algorithms that use (3.18)-(3.19) are given by Celis, Dennis and Tapia [4] and Powell and Yuan [25]. The optimal conditions for subproblem (3.18)-(3.19) are given by Yuan[35].

Theorem 3.3 *Assume that* $\xi_k > \min_{||d||_2 \le \Delta_k} ||c_k + A_k^T d||_2$. *Let* d_k^* *is a solution of (3.18)-(3.19), then there exist Lagrange multipliers* $\lambda_k^* \ge 0$ *and* $\mu_k^* \ge 0$ *such that*

$$(B_k + \lambda_k^* I + \mu_k^* A_k A_k^T) d^* = -(g_k + \mu_k^* A_k c_k) \tag{3.20}$$

and the complementarity conditions $\lambda_k^*[\Delta_k - ||d_k^*||_2] = 0$ *and* $\mu_k^*[\xi_k - ||A_k^T d_k^* + c_k||_2] = 0$ *are satisfied. Furthermore, if the multipliers* λ_k^*, μ_k^* *are unique, the matrix* $H(\lambda_k^*, \mu_k^*) = B_k + \lambda_k^* I + \mu_k^* A_k A_k^T$ *has at most one negative eigenvalue.*

If B_k is positive definite, subproblem (3.18)-(3.19) can be solved by dual methods(see [36] and [41]). For general B, the CDT problem is complex and it can also be solved by identifying the two multipliers λ^* and μ^*([3]).

4 Convergence Analysis

To demonstrate how to prove the convergence of a trust region algorithm, we give the outlines of a proof for the convergence of Algorithm 2.1.

The lower bound for the predicted reduction plays an important role in the convergence analysis. The following lemma was proved by Powell[21]:

Lemma 4.1 *Let S be any subspace in $\Re^n$, and let d_S be any solution of the subproblem* $\min_{d\in S,||d||_2\le\Delta}\phi(d)$. *If $g\in S$, then we have that*

$$\phi(0)-\phi(d_S)\ge\frac{1}{2}||g||_2\min[\Delta,||g||_2/||B||_2]. \tag{4.1}$$

Inequality (4.1) shows that the predicted reduction can not be very small unless either $||g||_2\Delta$ or $||g||_2^2/||B||_2$ is very small. This property is crucial for proving convergence of trust region algorithms. Indeed, global convergence can be showed as long as the trial step s_k satisfies

$$\phi_k(o)-\phi_k(s_k)\ge\tau\min\{\Delta_k,||g_k||_2/||B_k||_2\}\ , \tag{4.2}$$

where τ is some positive constant. A trial step s_k satisfying inequality (4.2) is called a "sufficient reduction" step. Trial steps obtained by the dog-leg method, the 2-dimensional search method, and the truncated CG method all satisfy (4.2). The first convergence result for Algorithm 2.1 was given by [21] under the assumption that the matrices B_k are bounded. Later it was showed that the boundedness can be relaxed to $||B_k||_2\le\beta_1(1+\sum_{i=1}^k||s_k||_2)$ and

$$||B_k||_2\le\beta_1 k,\quad\forall\ k, \tag{4.3}$$

where β_1 is any positive constant([23], [24]). To prove the global convergence of Algorithm 2.1 under condition (4.3), the following lemmas are needed.

Lemma 4.2 *Assume that $f(x)$ is differentiable and $\nabla f(x)$ is uniformly Lipschitz continuous. Let x_k be generated by Algorithm 2.1 with s_k satisfying (4.2) for all k. If there exists a positive constant δ such that $||g_k||_2\ge\delta>0$, for all k, then there exists a constant $\eta>0$ such that $\Delta_k\ge\eta/M_k$ holds for all k, where M_k is defined by $M_k=1+\max_{1\le i\le k}||B_k||_2$.*

Lemma 4.3 *Let $\{\Delta_k\}$ and $\{M_k\}$ be two sequences such that $\Delta_k\ge\nu/M_k\ge 0$ for all k, where ν is a positive constant. Let J be a subset of $\{1,2,3,...\}$. Assume that $\Delta_{k+1}\le\tau_1\Delta_k(\forall k\in J)$, $\Delta_{k+1}\le\tau_4\Delta_k(\forall k\notin J)$, $M_{k+1}\ge M_k(\forall k)$ and $\sum_{k\in J}1/M_k<\infty$, where $\tau_1>1$, $\tau_4<1$ are positive constants. Then $\sum_{k=1}^{\infty}1/M_k<\infty$.*

The following result can be proved by using the above two lemmas.

Theorem 4.4 *Assume that $f(x)$ is differentiable and $\nabla f(x)$ is uniformly Lipschitz continuous. Let x_k be generated by Algorithm 2.1 with s_k satisfies (4.2). If M_k defined by $M_k=1+\max_{1\le i\le k}||B_k||_2$ satisfy that $\sum_{k=1}^{\infty}\frac{1}{M_k}=\infty$, if $\epsilon=0$ is chosen in Algorithm 2.1, and if $\{f(x_k)\}$ is bounded below, then it follows that* $\liminf_{k\to\infty}||g_k||_2=0$.

The condition $\sum 1/M_k = \infty$ allows the matrices B_k to be updated by some known quasi-Newton formulae such as Powell's symmetric Broyden (BSP) formula (see, [21], [23]) or by the BFGS method. The above theorem is strengthened by [27] with some additional conditions:

Theorem 4.5 *Under the conditions of Theorem 4.4, if* $\tau_0 > 0$ *and* $\{||B_k||_2\}$ *is bounded, then* $\lim_{k\to\infty} ||g_k||_2 = 0$.

Similar to unconstrained optimization, convergence of trust region algorithms for constrained optimization depends on some lower bound condition of the predicted reduction, which normally has the form:

$$pred_k \geq \delta\epsilon_k \min[\Delta_k, \epsilon_k/||B_k||] \tag{4.4}$$

where δ is some positive constant, and ϵ_k is the violation of the KT conditions.

Local convergence of trust region algorithms can be studied by comparing the trust region trial step to the Newton step or the SQP step. In order to have locally superlinear convergence, the trust region subproblem should be a good approximation of the SQP subproblem. Normally techniques of Dennis and Moré[9] are used to show that the trial step is a superlinear step. It also needs to show the trust region constraint is inactive and the trial step is acceptable for all large k.

5 Combining with other techniques

Trust region can be combined with other techniques in constructing algorithms. Nocedal and Yuan[19] observed that the trial step of trust region algorithms for unconstrained optimization is also a descent direction. So when the trial step is unacceptable, it is still possible to carry out a line search along the direction of the trial step. Nocedal and Yuan[19] presents an algorithm which combines backtracking and trust region.

Non-monotone techniques can also be used within a trust region algorithm. In such an algorithm, it is possible for $f(x_{k+1}) \geq f(x_k)$ at some iterations. The first nonmonotone trust region algorithm was given by Deng, Xiao and Zhou[7], where the condition for accepting a trial step is either $r_k > \tau_0$ or

$$f(x_k + s_k) < \max_{0\leq j\leq m(k)} \{f(x_{k-j}\} - \gamma\Delta_k\|g_k\|. \tag{5.1}$$

$\gamma > 0$ is a constant and $m(k) = \min\{k-1, M\}$, M being a constant integer. The main idea is to require a sufficient reduction after M iterations instead of in every iteration. Another nonmonotone algorithm was given by Toint[31].

Second order correction step is a technique to overcome the so called the Marotos effect, which also exists in line search algorithms. In trust region algorithms for constrained optimization or nonsmooth optimization, it is possible for the trust region constraint to be active at every iteration. Consequently the rate of convergence is only linear even though good second models are used[33].

One way to maintain superlinear convergence is the second order correction technique given by Fletcher[12]. The second order correction step can be computed by solving another subproblem that is a slight modification of the standard trust region subproblem. For example, if the trial step s_k is computed by the SL_1QP subproblem, the second order step $\hat{s}_k$ can be taken as the solution of the following problem

$$\min \ g_k^T(s_k + d) + \frac{1}{2}(s_k + d)^T B_k(s_k + d) + \sigma_k ||c(x_k + s_k) + A_k^T d||_1 \tag{5.2}$$

subject to $\|s_k + d\|_\infty \leq \Delta_k$. When x_k is close to a stationary point where second order sufficient conditions are satisfied, it can be shown that $||\hat{s}_k|| = O(||s_k||^2)$. One nice property of the second order correction step is that it reduces the merit function and maintains the superlinear convergence property[34].

Trust region algorithms can also be combined with interior point techniques. One approach is to use barrier penalty functions to prevent the iterates moving out of the feasible region. Another is use a scaled trust region constraint $\|D_k d\| \leq \Delta_k$ such that the trust region is a subset of the feasible region. For an example of interior trust region algorithms, please see Coleman and Li[6]

Bibliography

1. R. Byrd, R.B. Schnabel and G.A. Shultz, "A trust region algorithm for nonlinearly constrained optimization", *SIAM J. Numer. Anal.* 24(1987) 1152-1170.
2. R. Byrd, R.B. Schnabel and G.A. Shultz, "Approximation solution of the trust region problem by minimization over two-dimensional subspaces", *Math. Prog.* 40(1988) 247-263.
3. X.D. Chen and Y. Yuan, "On local solutions of CDT problem", Report ICM-98-8, ICMSEC, Academia Sinica, 1998. to appear in *SIAM J. Opt.*.
4. M.R. Celis, J.E. Dennis and R.A. Tapia, "A trust region algorithm for nonlinear equality constrained optimization", in: P.T. Boggs, R.H. Byrd and R.B. Schnabel, eds., *Numerical Optimization* (SIAM, Philadelphia, 1985) pp. 71-82.
5. A.R. Conn, N.I.M. Gould and Ph.L. Toint, *Trust Region Methods*, (1999). Monograph, to be published.
6. T.F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds", *SIAM J. Opt.* 6(1996) 418-445.
7. N.Y. Deng, Y. Xiao and F.J. Zhou, "Nonmonotonic trust region algorithm", *JOTA* 26(1993) 259-285.
8. J.E. Dennis and H.H.W. Mei, "Two new unconstrained optimization algorithms which use function and gradient values", *JOTA* 28(1979) 453-482.
9. J.E. Dennis and J.J. Moré, "A characterization of superlinear convergence and its application to quasi-Newton methods", *Math. Comp.* 28(1974) 549-560.

10. I.S. Duff, J. Nocedal, and J.K. Reid, "The use of linear programming for the solution of sparse sets of nonlinear equations", *SIAM J. Sci. Stat. Comput.* 8(1987) 99-108.
11. R. Fletcher, "A model algorithm for composite NDO problem", *Math. Prog. Study* 17(1982) 67-76.
12. R. Fletcher, "Second order correction for nondifferentiable optimization", in: G.A. Watson, ed., *Numerical Analysis* (Springer-Verlag, Berlin, 1982) pp. 85-115.
13. R. Fletcher, *Practical Methods of Optimization* (second edition) (John Wiley and Sons, Chichester, 1987)
14. D.M. Gay, "Computing optimal local constrained step", *SIAM J. Sci. Stat. Comp.* 2(1981) 186-197.
15. G.H. Golub and U. von Matt, "Quadratically constrained least squares and quadratic problems", *Numerische Mathematik* 59(1991) 561-580.
16. J.J. Moré, "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.
17. J.J. Moré, "Recent developments in algorithms and software for trust region methods", in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of the Art* (Springer-Verlag, Berlin, 1983) pp. 258-287.
18. J.J. Moré and D.C. Sorensen, "Computing a trust region step", *SIAM J. Sci. Stat. Comp.* 4(1983) 553-572.
19. J. Nocedal and Y. Yuan, "Combining trust region and line search techniques", in: Y. Yuan, ed. *Advances in Nonlinear Programming*, (Kluwer, 1998), pp. 153–175.
20. E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph. D. Thesis, University of Colorado at Boulder, 1989.
21. 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. (1970a)
22. M.J.D. Powell, "A hybrid method for nonlinear equations", in: P. Robinowitz, ed., *Numerical Methods for Nonlinear Algebraic Equations* (Gordon and Breach Science, London, 1970) pp. 87-144. (1970b)
23. 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* (Acadmic Press, New York, 1975) pp. 1-27.
24. M.J.D. Powell, "Nonconvex 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.

25. M.J.D. Powell and Y. Yuan, “ A trust region algorithm for equality constrained optimization”, *Math. Prog.* 49(1991) 189-211.
26. F. Rendl and H. Wolkowicz, “A semidefinite framework for trust region subproblems with applications to large scale minimization”, *Math. Prog.* 77(1997) 273-299.
27. 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 J. Numer. Anal.* 22(1985) 47-67.
28. D.C. Sorensen, “Newton’s method with a model trust region modification”, *SIAM J. Numer. Anal.* 20(1982) 409-426.
29. T. Steihaug, “The conjugate gradient method and trust regions inlarge scale optimization”, *SIAM J. Numer. Anal.* 20(1983) 626-637.
30. Ph. L. Toint, “Towards an efficient sparsity exploiting Newton method for minimization”, in: I.S. Duff, ed., *Sparse Matrices and Their Uses* (Academic Press, London, 1981) pp. 57-88.
31. Ph.L. Toint, “A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints”, *Math. Prog.* 77(1997) 69-94.
32. A. Vardi, “A trust region algorithm for equality constrained minimization: convergence properties and implementation”, *SIAM J. Numer. Anal.* 22(1985) 575-591.
33. Y. Yuan, “An example of only linearly convergence of trust region algorithms for nonsmooth optimization”, *IMA J. Numer. Anal.* 4(1984) 327-335.
34. Y. Yuan, “On the superlinear convergence of a trust region algorithm for nonsmooth optimization”, *Math. Prog.* 31(1985) 269-285.
35. Y. Yuan, “On a subproblem of trust region algorithms for constrained optimization”, *Math. Prog.* 47(1990) 53-63.
36. “A dual algorithm for minimizing a quadratic function with two quadratic constraints”, *J. Comp. Math.* 9(1991) 348-359.
37. “Nonlinear Optimization: Trust region algorithms”, in: S.T. Xiao, F. Wu, eds., *Proceedings of the Third Chinese SIAM Conference* (Tsinghua University Press, Beijing, 1994), pp.84-102.
38. Y. Yuan, “On the convergence of a new trust region algorithm”, *Numerische Mathematik* 70(1995) 515-539.
39. Y. Yuan, “An example of non-convergence of trust region algorithms”. in: Y. Yuan, ed. *Advances in Nonlinear Programming*, (Kluwer, 1998), pp. 205–215
40. Y. Yuan, “On the truncated conjugate gradient method”, Report, ICM99-003, ICMSEC, Chinese Academy of Sciences. Beijing, China, 1999.
41. Y. Zhang, “Computing a Celis-Dennis-Tapia trust region step for equality constrained optimization”, *Math. Prog.* 55(1992) 109-124.