TRUST REGION ALGORITHMS FOR CONSTRAINED OPTIMIZATION[1]

Ya-xiang Yuan
(*State Key Laboratory of Scientific and Engineering Computing, Computing Center, Academia Sinica, Beijing 100080, China*)

Abstract

We review the main techniques used in trust region algorithms for nonlinear constrained optimization.

1. Trust Region Idea

Constrained optimization is to minimize a function subject to finitely many algebraic equation and inequality conditions. It has the following form

$$\min_{x \in \Re^n} \quad f(x) \tag{1.1}$$

$$\begin{aligned} subject \quad to \qquad c_i(x) &= 0, \qquad i = 1, 2, \ldots, m_e; & (1.2)\\ c_i(x) &\geq 0, \qquad i = m_e + 1, \ldots, m, & (1.3) \end{aligned}$$

where $f(x)$ and $c_i(x)$ $(i = 1, \ldots, m)$ are real functions defined in $\Re^n$, and $m \geq m_e$ are two non-negative integers.

Numerical methods for nonlinear optimization problems can be grouped as two types. One are line search methods and the other are trust region algorithms.

Line search algorithms at each iteration use a direction to carry a line search. The direction is called the search direction, which is normally computed by solving a subproblem that approximates the original problem near the current iterate. A line search means to search for a new point along the search direction. For example, an exact line search is to find a point in the search direction which minimize a certain merit function.

Trust region algorithms are relatively new algorithms. The trust region approach is strongly associated with approximation. Assume we have a current

[1]this work was partially supported by the project "large scale scientific computing" of State Commission of Science and Technology, China

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 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 computations indicate the approximate model fit the original problem quite 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 will be reduced.

The key contents of a trust region algorithm are how to compute the trust region trial step how to decide whether a trial step should be accepted. An iteration of a trust region algorithm has the following form. At the beginning of the iteration, a trust region is available. 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.

Most researches on trust region algorithms are mainly started in the 80s. Hence trust region algorithms are less mature then line search algorithms, and by now the applications of trust region algorithms are not as widely as that of line search algorithms. However, trust region methods have two advantages. One is that they are reliable and robust, another is that they have very strong convergence properties.

2. Trust Region Subproblem

The most important part of a trust region algorithm is how it generates trial steps. The trust region trial step s_k is normally computed by solving a certain subproblem. Such a subproblem is called a trust region subproblem, and it can be viewed as a local approximation to the original nonlinear optimization problem.

Most trust region subproblems for nonlinear optimization can be regarded as some kind of modification of the SQP subproblem of line search algorithm, which has the following form:

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{2.4}$$

$$s.\ t.\quad c_i(x_k) + d^T \nabla c_i(x_k) \;=\; 0 \qquad i = 1, 2, \ldots, m_e; \tag{2.5}$$

$$c_i(x_k) + d^T \nabla c_i(x_k) \quad \geq \quad 0 \qquad i = m_e + 1, \ldots, m \tag{2.6}$$

where $g_k = g(x_k) = \nabla f(x_k)$ and B_k is an approximate Hessian of the Lagrange function.

The first type of trust region subproblems, being a slightly modification of SQP subproblem (2.4)-(2.6), have the following form:

$$\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{2.7}$$

$$\begin{aligned} s.\ t. \quad \theta_k c_i(x_k) + d^T \nabla c_i(x_k) &= 0 \qquad i = 1, 2, \ldots, m_e; & (2.8)\\ \theta_k c_i(x_k) + d^T \nabla c_i(x_k) &\geq 0 \qquad i = m_e + 1, \ldots, m & (2.9)\\ ||d|| &\leq \Delta_k & (2.10)\end{aligned}$$

where $\theta_k \in (0, 1]$ is a parameter (see Byrd, Schnabel and Shultz [1] and Vardi [6]). Parameter θ_k is introduced to overcome the possible nonfeasibility of the linearized constraints (2.5)-(2.6) in the trust region (2.10). Trial steps of the trust region algorithms that apply null space techniques can also be reviewed as solutions of (2.7)-(2.10).

Another trust region subproblem is obtained by replacing the linearized constraints (2.5)-(2.6) by a single quadratic constraint. It can be written as:

$$\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{2.11}$$

$$\begin{aligned} s.\ t. \quad ||(c_k + A_k^T d)^-||_2 &\leq \xi_k & (2.12)\\ ||d||_2 &\leq \Delta_k, & (2.13)\end{aligned}$$

where $c_k = c(x_k) = (c_1(x), ..., c_m(x))^T$, $A_k = A(x_k) = \nabla c(x_k)^T$, $\xi_k \geq 0$ is a parameter and the superscript "-" means that $v_i^- = v_i (i = 1, ..., m_e)$, $v_i^- = \min[0, v_i](i = m_e + 1, ..., m)$. Algorithms that use (2.11)-(2.13) are given by Celis, Dennis and Tapia [2] and Powell and Yuan [5].

Trust region subproblems can also derived by using exact penalty functions. The following trust region subproblem is based on the L_∞ exact penalty function:

$$\min_{d \in \Re^n} \ g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c_k + A_k^T d)^-||_\infty = \Phi_k(d) \tag{2.14}$$

$$s.\ t. \quad ||d|| \leq \Delta_k. \tag{2.15}$$

Trust region subproblems based on exact penalty functions are closely related to subproblems of trust region algorithms for nonlinear systems of equations. Trust region algorithms that compute the trial step by solving (2.14)-(2.15) are also similar to trust region algorithms for nonsmooth optimization.

3. Global Convergence

The convergence properties of trust region algorithms are generally analyzed by considering the descent properties of the trial steps. A suitable approximation of the merit function is used. We call this function approximate merit function. The approximate merit function $\bar{\phi}_k(d)$ is strongly associated with the trust region subproblems. The trial step s_k computed by solving a trust region subproblem will reduced the approximate merit function. In fact, normally the predicted reduction $pred_k$ is defined by $\bar{\phi}_k(0) - \bar{\phi}_k(s_k)$, which is the reduction in the approximate merit function. The approximate merit function also has the required approximation property, that is

$$\bar{\phi}_k(d) - \bar{\phi}_k(0) = P_k(x_k + d) - P_k(x_k) + o(||d||) \tag{3.16}$$

when $||d||$ is very small, where $P_k(x)$ is the merit function that is used to decide whether s_k can be accepted.

To prove convergence of a trust region algorithm, we normally show that the predicted reduction satisfies certain lower bound condition such as

$$pred_k \geq \delta \epsilon_k \min[\Delta_k, \epsilon_k/||B_k||] \tag{3.17}$$

where δ is some positive constant, and ϵ_k is the violation of the KT conditions which is defined by

$$\epsilon_k = ||c_k^-|| + ||g_k - A_k \lambda_k|| \tag{3.18}$$

and λ_k being an approximate multiplier at the current point x_k and it satisfies that $(\lambda_k)_i \geq 0, \quad i > m_e$. Then it is shown that the merit function will remain the same for all large k. That is, there exist a integer k_0 and a merit function $P(x)$ such that $P_k(x) = P(x)$ for all $k \geq k_0$.

If ϵ_k is bounded away from zero, it can be shown that

$$pred_k \geq \bar{\delta} \Delta_k \tag{3.19}$$

for all k, where $\bar{\delta}$ is a positive constant. Using the above inequality and certain condition on the merit function $P(x)$, we can prove that

$$\sum_{k=1}^{\infty} \Delta_k < \infty. \tag{3.20}$$

Thus $\Delta_k \to 0$. This and relation (3.16) imply that

$$r_k = \frac{P(x_k) - P(x_k + s_k)}{pred_k} \to 1. \tag{3.21}$$

The above limit shows that $\Delta_{k+1} \geq \Delta_k$ which contradicts (3.20). Hence it is shown that there exist a subsequence such that $\{\epsilon_k\}$ converges to zero.

Global onvergence results of trust region algorithms depend on the sufficiently reduction condition (3.17) instead of requiring that the trial step s_k solves the trus region subproblem exactly. Hence global convergence is also true when s_k is any approximate solution of the trust region subproblem provided it satisfies condition (3.17).

4. Local Convergence

Local convergence of trust region algorithms are shown by establishing the equivalence of the trust region trial step and the SQP step. To analyze local convergence, it is always assumed that the sequence $\{x_k\}$ generated by the algorithm converges to x^*. Global convergence results imply that x^* is a KT point.

Let d_k^* be the SQP step that is computed by solving the QP subproblem (2.4)-(2.6). It is well known that under certain conditions the SQP step d_k^* is superlinearly convergent in the sense that

$$\lim_{k\to\infty} ||x_k + d_k^* - x^*||/||x_k - x^*|| = 0. \tag{4.22}$$

Therefore to prove local superlinear convergence

$$\lim_{k\to\infty} ||x_{k+1} - x^*||/||x_k - x^*|| = 0, \tag{4.23}$$

we need to show that

$$||s_k - d_k^*|| = o(||d_k^*||) \tag{4.24}$$

$$x_{k+1} = x_k + s_k \tag{4.25}$$

holds for all large k. In order to have the property (4.24), the trust region subproblem should be a good approximation of the SQP subproblem. The validity of (4.25) depends on suitable choice of the merit function.

For most algorithms, it can be shown that

$$s_k = d_k^* \tag{4.26}$$

if k is sufficiently large and if $||s_k|| < \Delta_k$. Thus it is sufficient to show that the trial step s_k is acceptable and inactive with the trust region bound for all large k. These are not true for some algorithms. For example, the SQP step will not be acceptable if the merit function is nonsmooth. This is the so called Maratos effect. To overcome the Maratos effect, we can either relax the condition for accepting trial steps or compute a second order correction step. Relaxing conditions for accepting trial step can be traced back to the watch-dog technique [3], and second order correction step was first suggested by Fletcher [4].

A second order correction step $\hat{s}_k$ is computed by solving another subproblem that is called second order correction subproblem. The second oder correction subproblem is a slightly modification of the trust region subproblem that used to compute the trial step. Assume that a trial step s_k is calculated. Normaly a second order correction subproblem can be constructed by replacing $c(x_k)$ by $c(x_k+s_k)-A_k^T s_k$ in the trust region subproblem. For example, if the trial step s_k is computed by trust region subproblem (2.14)-(2.15), the second order correction subproblem can be as follows

$$\min \; g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c(x_k+s_k) + A_k^T(d-s_k))^-||_\infty \tag{4.27}$$

$$s.\ t. \quad ||d|| \leq \Delta_k. \tag{4.28}$$

A second order correction step satisfies that $||\hat{s}_k|| = O(||s_k||^2)$. One nice property of second order correction step is that inequality

$$P(x_k + s_k + \hat{s}_k) < P(x_k) \tag{4.29}$$

holds for all large k. Hence if condition (4.24) is satisfied, it follows from (4.22) that that

$$\lim_{k\to\infty} ||x_k + s_k + \hat{s}_k - x^*||/||x_k - x^*|| = 0. \tag{4.30}$$

Relation (4.29) imply that $x_{k+1} = x_k + s_k + \hat{s}_k$ if k is large and if the second order correction step is computed. Trust region algorithms with second order correction techniques compute the second order correction step whenever the trial step s_k is unacceptable. Therefore it can be shown that, if k is large, either $x_{k+1} = x_k + s_k$ or $x_{k+1} = x_k + s_k + \hat{s}_k$. Consequently the superlinear convergence (4.23) follows from (4.30), (4.24) and (4.22).

References

[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] 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.

[3] R.M. Chamberlain, C. Lemarechal, H.C. Pedersen and M.J.D. Powell, "The watchdog techniques for forcing convergence in algorithms for constrained optimization", *Math. Prog. Study* 16(1982) 1-17.

[4] R. Fletcher, "Second order correction for nondifferentiable optimization", in: G.A. Watson, ed., *Numerical Analysis* (Springer, Berlin, 1982) pp. 85-115.

[5] M.J.D. Powell and Y. Yuan, " A trust region algorithm for equality constrained optimization", *Math. Prog.* 49(1991) 189-211.

[6] A. Vardi, "A trust region algorithm for equality constrained minimization: convergence properties and implementation", *SIAM J. Numer. Anal.* 22(1985) 575-591.