Nonlinear Optimization: Trust Region Algorithms[*]

Ya-xiang Yuan
State Key Laboratory of Scientific and Engineering Computing
Computing Center, Academia Sinica, Beijing 10080, China

Abstract

In this paper, we review the trust region algorithms for nonlinear optimization. The philosophy and the fundamental ideas of trust region algorithms are discussed. Model algorithms for unconstrained optimization, constrained optimization, and nonsmooth optimization are given. Main techniques for global convergence and local superlinear convergence are analyzed.

Key words: constrained optimization, trust region algorithms

1. Introduction

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

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

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

where $f(x)$ and $c_i(x)$ $(i = 1, \ldots, m)$ are real functions defined in $\Re^n$, 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 type of methods for optimization are line search algorithms, which obtain a search direction in each iteration, and search along this direction to obtain a better point. The search direction is normally computed by solving a subproblem that approximates the original problem near the current point, therefore it guarantees that there exist better points along the direction. Most of the known methods for optimization are line search algorithms.

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

[*]this work was partially supported by the state key project "Scientific Computing"

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. Levenberg-Marquardt Method

Levenberg-Marquardt method, first given by Levenberg[15] and re-derived by Marquardt[21], is a method for solving nonlinear equations. This method is often mentioned when the history of trust region algorithms is discussed. The reason is that the technique of trust region is, in some sense, equivalent to that of the Levenberg-Marquardt method.

Consider a system of nonlinear equations

$$f_i(x) = 0, \qquad i = 1, ..., m, \tag{2.1}$$

where $f_i(x)(i = 1, ..., m)$ are continuous differentiable functions in $\Re^n$. We try to compute a least square solution, which means that we need to solve the nonlinear least squares problem

$$\min_{x\in\Re^n} ||F(x)||_2^2 \tag{2.2}$$

where $F(x) = (f_1(x), ...f_m(x))^T$. The Gauss-Newton method for problem (2.2) is iterative, and at the current iterate x_k, the Gauss-Newton step is

$$d_k = -(A(x_k)^T)^+ F(x_k) \tag{2.3}$$

where $A(x) = \nabla F(x)^T$ is the Jacobi matrix, and A^+ is the generalized inverse of A. It is easy to see that the Gauss-Newton step is the minimum norm solution of the subproblem

$$\min_{d\in\Re^n} ||F(x_k) + A(x_k)^T d||_2^2 \tag{2.4}$$

which is an approximation to the original problem (2.2) near the current iterate x_k. One difficulty of using the Gauss-Newton step is that the Jacobi matrix $A(x_k)$ may be ill conditioned, which normally leads to a very big step d_k. And a very long step d_k usually causes the algorithm to break down, because of either numerical overflows or failure in line searches. The Levenberg-Marquardt method chooses the step as follows

$$d_k = -(A(x_k)A(x_k)^T + \lambda_k I)^{-1}A(x_k)F(x_k) \tag{2.5}$$

where $\lambda_k \geq 0$ is a parameter which is updated from iteration to iteration (see, [18]). The original idea of Levenberg-Marquardt method is introducing the parameter λ_k to overcome the ill condition of $A(x_k)$, or in other words, to prevent $||d_k||_2$ being too large.

It is easily seen that d_k given by (2.5) is a stationary point of the convex function:

$$P(d) = ||F(x_k) + A(x_k)^T d||_2^2 + \lambda_k ||d||_2^2 \ . \tag{2.6}$$

Thus, (2.5) is a solution of

$$\min_{d \in \Re^n} ||F(x_k) + A(x_k)^T d||_2^2 + \lambda_k ||d||_2^2 \ . \tag{2.7}$$

Subproblem (2.7) is a modification of (2.4). The additional term $\lambda_k ||d||_2^2$ can be viewed as a penalty term which prevents $||d_k||$ from being too large.

Define

$$\Delta_k = ||(A(x_k)A(x_k)^T + \lambda_k I)^{-1}A(x_k)F(x_k)||_2 \ , \tag{2.8}$$

then for any $||d||_2 \leq \Delta_k$, because d_k is a solution of (2.7), it can be shown that

$$\begin{aligned} ||F(x_k) + A(x_k)^T d||_2^2 &= P(d) - \lambda_k ||d||_2^2 quad \geq P(d_k) - \lambda_k ||d||_2^2 \\ &= ||F(x_k) + A(x_k)^T d_k||_2^2 + \lambda_k(||d_k||_2^2 - ||d||_2^2) \\ &\geq ||F(x_k) + A(x_k)^T d_k||_2^2. \end{aligned} \tag{2.9}$$

This verifies that d_k is also a solution of the following problem

$$\min_{d \in \Re^n} ||F(x_k) + A(x_k)^T d||_2^2 \tag{2.10}$$

$$s.\ t. \qquad ||d||_2 \leq \Delta_k \ . \tag{2.11}$$

Now it is obvious that problem (2.10)-(2.11) is a trust region subproblem, as condition (2.11) is clearly a trust region type constraint. It is in this sense that 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. The following is a trust region algorithm for nonlinear least squares problems:

Algorithm 2.1 *(Trust Region Algorithm for Nonlinear Least Squares)*

```
Step 1  Given $x_1 \in \Re^n$, $\Delta_1 > 0$.

Step 2  Solve (2.10)-(2.11), giving $s_k$;
        If $||F(x_k)||_2 = ||F(x_k + A(x_k)^T s_k||_2$ then stop;
        Compute
```

$$r_k = \frac{||F(x_k)||_2 - ||F(x_k + s_k)||_2}{||F(x_k)||_2 - ||F(x_k) + A(x_k)^T s_k||_2}. \tag{2.12}$$

Step 3 Let

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if } r_k > 0, \\ x_k & \text{otherwise;} \end{cases} \tag{2.13}$$

Set

$$\Delta_{k+1} = \begin{cases} ||s_k||_2 & \text{if } r_k < 0.1, \\ 2\Delta_k & \text{if } r_k > 0.9 \text{ and } ||s_k||_2 > \Delta_k/2, \\ \Delta_k & \text{otherwise;} \end{cases} \tag{2.14}$$

Step 4 $k := k+1$, *go to Step 2.*

In the above algorithm, the trust region radius Δ_k is updated from iteration to iteration directly, while the Levenberg-Marquardt method updates the parameter λ_k, which in turn modifies the value Δ_k from (2.8) implicitly. Modifying Δ_k directly has the advantage of controlling and monitoring the length of d_k easily. Hence, nowadays it is regarded that trust region approach is better than the original Levenberg-Marquardt method. For more details, see [18].

Algorithm 2.1 can be modified for solving L_1 norm minimization ([9]) and general norm minimization problems([10]).

3. Unconstrained Optimization

In this section, we consider trust region algortihms for unconstrained optimization problem:

$$\min_{x \in \Re^n} f(x) \tag{3.1}$$

where $f(x)$ is a nonlinear continuous differentiable function in $\Re^n$. At each iteration, a trial step is calculated by solving the subproblem

$$\min_{d \in \Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{3.2}$$

$$s.\ t. \qquad ||d||_2 \le \Delta_k \tag{3.3}$$

where $g_k = \nabla f(x_k)$ is the gradient at the current approximate solution, 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 (3.2)-(3.3). The predicted reduction is defined by the reduction in the approximate model, that is

$$Pred_k = \phi_k(0) - \phi_k(s_k). \tag{3.4}$$

Unless the current point x_k is a stationary point and B_k is positive semi-definite, the predicted reduction $Pred_k$ is always positive. The actual reduction is the reduction in the objective function:

$$Ared_k = f(x_k) - f(x_k + s_k). \tag{3.5}$$

And we define the ratio between the actual reduction and the predicted reduction by

$$r_k = \frac{Ared_k}{Pred_k} \tag{3.6}$$

which is used to decide whether the trial step is acceptable and to adjust the new trust region radius.

A general trust region algorithm for unconstrained optimization can be given as follows.

Algorithm 3.1 *(Trust Region Algorithm for Unconstrained Optimization)*

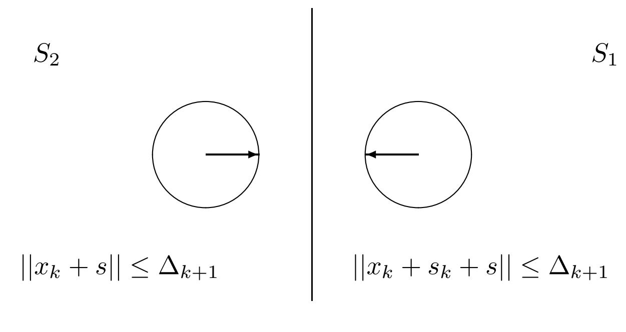
```
Step 1  Given $x_1 \in \Re^n$, $\Delta_1 > 0$, $\epsilon \geq 0$, $B_1 \in \Re^{n\times n}$ symmetric;
        $0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \leq \tau_0 \leq \tau_2 < 1$, $\tau_2 > 0$, $k := 1$.

Step 2  If $||g_k||_2 \leq \epsilon$ then stop;
        Solve (3.2)-(3.3) giving $s_k$.

Step 3  Compute $r_k$;
            $x_{k+1} = \begin{cases} x_k & \text{if } r_k \leq \tau_0 , \\ x_k + s_k & \text{otherwise} ; \end{cases}$     (3.7)
        Choose $\Delta_{k+1}$ that satisfies
            $\Delta_{k+1} \in \begin{cases} [\tau_3||s_k||_2,\ \tau_4\Delta_k] & \text{if } r_k < \tau_2, \\ [\Delta_k,\ \tau_1\Delta_k] & \text{otherwise;} \end{cases}$     (3.8)

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], [11], [19], [27], etc.. The parameter τ_0 is usually zero (e.g. [13], [26]) or a small positive constant (e.g. [9] and [31]). The advantage of using zero τ_0 is that a trial step is accepted whenever the objective function is reduced. Hence it would not throw away a "good point", which is a desirable property especially when the function evaluations are very expensive. Another intuitive argument for preferring $\tau_0 = 0$ is as follows. Consider the case that $r_k > 0$. No matter how small the ratio r_k is, the objective function $f(x)$ has a smaller function value at $x_k + s_k$ than at x_k. Hence intuitively one would expect that the minimum of the objective function should be closer to $x_k + s_k$ than to x_k. In other words, it is more likely that the solution of the original problem is in the half space $S_1 = \{s \mid ||x_k + s_k + s|| \leq ||x_k + s||\}$ instead of $S_2 = \{s \mid ||x_k + s|| \leq ||x_k + s_k + s||\}$ (see Picture 3.1). Normally trust region algorithms reduce the new trust region bound to at most a half of $||s_k||$ whenever s_k is rejected ($x_{k+1} = x_k$), Hence for those algorithms that reject s_k, the trust region for the next iteration will be $\{s \mid ||x_k + s|| \leq \Delta_{k+1} \leq ||s_k||/2\}$ which is a subset of S_2. That contradicts to our above rough analyses that indicate the solution is more likely in S_1. Hence we believe it is better to set $x_{k+1} = x_k + s_k$ in this case, which will enable the next trust region in S_1. That is to say, intuitively it is better to set $x_{k+1} = x_k + s_k$ whenever $r_k > 0$.


S_2
S_1
$||x_k + s|| \leq \Delta_{k+1}$
$||x_k + s_k + s|| \leq \Delta_{k+1}$

Picture 3.1

But, the price we pay for letting $\tau_0 = 0$ is that the global convergence result is only

$$\liminf_{k\to\infty} \quad ||g_k||_2 = 0 \tag{3.9}$$

instead of

$$\lim_{k\to\infty} ||g_k||_2 = 0 \tag{3.10}$$

which can be achieved if $\tau_0 > 0$. However, given a positive tolerance ϵ, (3.9) is also sufficient to guarantee a finite termination of Algorithm 3.1, namely (3.9) ensures that the convergence test $||g_k||_2 \le \epsilon$ in Step 2 of Algorithm 3.1 will be satisfied for some k.

The subproblem (3.2)-(3.3) has been studied by many authors. And the following lemma is well known (for example, see [14] and [20]):

Lemma 3.2 *A vector $d^* \in \Re^n$ is a solution of the problem*

$$\min_{d\in\Re^n} g^T d + \frac{1}{2} d^T B d = \phi(d) \tag{3.11}$$

$$\textit{subject} \quad \textit{to} \qquad ||d||_2 \le \Delta \tag{3.12}$$

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

$$(B + \lambda^* I)d^* = -g \tag{3.13}$$

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

$$\lambda^*(\Delta - ||d^*||_2) = 0. \tag{3.14}$$

Let d^* be a solution of problem (3.11)-(3.12) and λ^* be the multiplier satisfying conditions in the above lemma. If $B + \lambda^* I$ is positive definite, then d^* is uniquely defined by

$$d^* = -(B + \lambda^* I)^{-1} g. \tag{3.15}$$

The case where $B + \lambda^* I$ has zero eigenvalues is called "hard case". In this case, relation (3.13) implies that g is in the range space of $B + \lambda^* I$ and d^* can be written in the form:

$$d^* = -(B + \lambda^* I)^+ g + v, \tag{3.16}$$

where v is a vector in the null space of $B + \lambda^* I$. On other hand, if g is in the range space of $B + \lambda^* I$ then any vector d^* given by (3.16) is also a solution of (3.11)-(3.12) provided that $||d^*||_2 \le \Delta$ and that $\lambda^*(\Delta - ||d^*||_2) = 0$.

Unless in the hard case, λ^* is also the unique solution of the following equation

$$\psi(\lambda) = \frac{1}{||(B + \lambda I)^{-1} g||_2} - \frac{1}{\Delta} = 0. \tag{3.17}$$

Function $\psi(\lambda)$ is well defined for $\lambda \in (-\sigma_n(B), +\infty)$, where $\sigma_n(B)$ is the least eigenvalue of B. $\psi(\lambda)$ is concave and strictly monotonically increasing in $(-\sigma_n(B), +\infty)$ (For example, see [8]). In fact, the first order and second order derivatives of $\psi(\lambda)$ can be easily computed, thus Newton's method can be used to calculate λ^*. The Newton's iteration is

$$\begin{aligned} \lambda_+ &= \lambda - \frac{\psi(\lambda)}{\psi'(\lambda)} \\ &= \lambda - \frac{g^T (B + \lambda I)^{-3} g}{||(B + \lambda I) g||_2^3} \left[\frac{1}{||(B + \lambda I)^{-1} g||_2} - \frac{1}{\Delta} \right]. \end{aligned} \tag{3.18}$$

Based on Newton's iteration (3.18), numerical algorithms for problem (3.11)-(3.12) have been given by [14] and [20].

In the hard case, we have that

$$\lambda^* = -\sigma_n(B), \tag{3.19}$$

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 (3.11)-(3.12). Hence the "real" hard case is that (3.19) is satisfied and $\sigma_n(B) < 0$. For any $\lambda \in (-\sigma_n(B), +\infty)$, Newton's step will normally make the matrix $B + \lambda_+ I$ have negative eigenvalue. Hence Newton's step (3.18) can only be used to adjust the lower bound λ_L. Based on these observations, we suggest to use the Newton's step for an equivalent equation

$$\tilde{\psi}(\mu) = \psi(\frac{1}{\mu}) = 0. \tag{3.20}$$

Lemma 3.3 *(Powell, 1970) Let S be any subspace in $\Re^n$, and let d_S be any solution of the following problem*

$$\min_{d \in S, ||d||_2 \le \Delta} \phi(d) \ . \tag{3.21}$$

If $g \in S$ then the inequality

$$\phi(0) - \phi(d_S) \ge \frac{1}{2}||g||_2 \min[\Delta, ||g||_2/||B||_2]. \tag{3.22}$$

is satisfied.

Specifically, when $S = \Re^n$ we have that

$$\phi(0) - \phi(d^*) \ge \frac{1}{2}||g||_2 \min[\Delta, ||g||_2/||B||_2]. \tag{3.23}$$

This shows that the reduction in the trust region model will not be very small unless either $||g||_2\Delta$ or $||g||_2^2/||B||_2$ is very small. This property is very important for proving convergence of trust region algorithms.

The global convergence analyses of trust region algorithms depend on the fact that the predicted reduction satisfies (3.23). Hence, instead of solving (3.2)-(3.3) exactly, we can compute a trial step s_k that satisfies

$$\phi_k(o) - \phi_k(s_k) \ge \tau \min\{\Delta_k, ||g_k||_2/||B_k||_2\} \ , \tag{3.24}$$

where τ is some positive constant. A trial step s_k satisfying inequality (3.24) is called a "sufficient reduction" step. To compute a vector s_k satisfying (3.24) is usually much easier than solving (3.2)-(3.3) exactly. The vector s_k can be calculated by dog-leg type techniques or by searching in the two dimensional space spanned by the steepest descent direction and Newton's step. For more details, please see [6], [25], [30] and [34]. The subproblem (3.2)-(3.3) can also be solved approximately by a preconditioned conjugate gradient method which can be regarded as a generalized dog-leg technique (see [32]).

Lemma 3.4 *Assume that $f(x)$ is differentiable and $\nabla f(x)$ is uniformly Lipschitz continuous. Let x_k be generated by Algorithm 3.1 with s_k is so computed that (3.24) is satisfied for all k. If there exists a positive constant δ such that*

$$||g_k||_2 \ge \delta > 0, \quad \forall \ k, \tag{3.25}$$

then there exists a constant $\eta > 0$ *such that*

$$\Delta_k \geq \eta \frac{1}{M_k} \tag{3.26}$$

holds for all k*, where* M_k *is defined by*

$$M_k = 1 + \max_{1 \leq i \leq k} ||B_k||_2. \tag{3.27}$$

Proof If the lemma is not true, then (3.25) holds and there exist a subsequence $\{k_i\}$ such that

$$\lim_{i \to \infty} \Delta_{k_i} M_{k_i} = 0. \tag{3.28}$$

Because $M_k \geq 1$ for all k, (3.28) indicates that $\Delta_{k_i} \to 0$. Due to the monotonicity of M_k, we can assume that $\Delta_{k_i} < \Delta_{k_i - 1}$ for all i. Using the notation $\bar{i} = k_i - 1$, from (3.25), (3.24), (3.28) and the fact that $\Delta_{k+1} \geq \tau_3 ||s_k||_2$ for all k, we can show that

$$\lim_{i \to \infty} ||s_{\bar{i}}||_2 M_{\bar{i}} = 0. \tag{3.29}$$

Inequalities (3.29), (3.25) and (3.24) imply that

$$\phi_{\bar{i}}(0) - \phi_{\bar{i}}(s_{\bar{i}}) \geq \bar{\tau} ||s_{\bar{i}}||_2 \tag{3.30}$$

holds for all large i, where $\bar{\tau}$ is some positive constant. Due to the uniformly Lipschitz continuity of $\nabla f(x)$ and relation (3.29), we have that

$$\begin{aligned} Ared_{\bar{i}} &= f(x_{\bar{i}}) - f(x_{\bar{i}} + s_{\bar{i}}) = -g_{\bar{i}}^T s_{\bar{i}} + O(||s_{\bar{i}}||_2^2) \\ &= \phi_{\bar{i}}(0) - \phi_{\bar{i}}(s_{\bar{i}}) + O(||s_{\bar{i}}||_2^2) + O(||s_{\bar{i}}||_2^2 ||B_{\bar{i}}||_2) \\ &= \phi_{\bar{i}}(0) - \phi_{\bar{i}}(s_{\bar{i}}) + o(||s_{\bar{i}}||_2) \\ &= Pred_{\bar{i}} + o(||s_{\bar{i}}||_2). \end{aligned} \tag{3.31}$$

The above relation and inequality (3.30) show that

$$\lim_{i \to \infty} r_{k_i - 1} = \frac{Ared_{\bar{i}}}{Pred_{\bar{i}}} = 1, \tag{3.32}$$

which shows that

$$\Delta_{k_i} \geq \Delta_{k_i - 1} \tag{3.33}$$

for all large i. This contradicts our assumption that $\Delta_{k_i} < \Delta_{k_i - 1}$ for all i. Hence the lemma is true. $\Box$

The first convergence result for Algorithm 3.1 was given by [24]. Later he showed that global convergence is always guaranteed provided that the matrices B_k satisfy that

$$||B_k||_2 \leq \beta_1 (1 + \sum_{i=1}^{k} ||s_k||_2), \quad \forall\ k \tag{3.34}$$

([26] or

$$||B_k||_2 \leq \beta_1 k, \quad \forall\ k \tag{3.35}$$

([27]), where β_1 is any positive constant. To prove the global convergence of Algortihm 3.1 under condition (3.35), the following lemma is needed.

Lemma 3.5 *(Powell, 1984a) Let $\{\Delta_k\}$ and $\{M_k\}$ be two sequences such that $\Delta_k \geq \nu/M_k \geq 0$ for all k, where ν is a positive constant. Let $\mathcal{I}$ be a subset of $\{1, 2, 3, ...\}$. Assume that*

$$\Delta_{k+1} \leq \tau_1 \Delta_k, \qquad k \in \mathcal{I} \tag{3.36}$$
$$\Delta_{k+1} \leq \tau_4 \Delta_k, \qquad k \notin \mathcal{I} \tag{3.37}$$
$$M_{k+1} \geq M_k, \qquad \forall\ k \tag{3.38}$$
$$\sum_{k \in \mathcal{I}} 1/M_k < \infty \tag{3.39}$$

where $\tau_1 > 1$, $\tau_4 < 1$ are positive constants. Then

$$\sum_{k=1}^{\infty} 1/M_k < \infty. \tag{3.40}$$

Theorem 3.6 *Assume that $f(x)$ is differentiable and $\nabla f(x)$ is uniformly Lipschitz continuous. Let x_k be generated by Algorithm 3.1 with s_k satisfies (3.24). If M_k defined by (3.27) satisfy that*

$$\sum_{k=1}^{\infty} \frac{1}{M_k} = \infty, \tag{3.41}$$

if $\epsilon = 0$ is chosen in Algorithm 3.1, and if $\{f(x_k)\}$ is bounded below, then it follows that

$$\liminf_{k \to \infty} \quad ||g_k||_2 = 0. \tag{3.42}$$

Proof If the theorem is not true, there exists a positive constant δ such that (3.25) holds for all k. Hence, Lemma 3.4 shows that there exists a positive constant ν such that $\Delta_k \geq \nu/M_k$ holds for all k. Define the set

$$\mathcal{I} = \{k \mid \quad r_k \geq \tau_2\}, \tag{3.43}$$

then inequality (3.36) and (3.37) follow from our update formula (3.8). The assumption that $f(x_k)$ is bounded below also implies that

$$\begin{aligned}
+\infty &> \sum_{k=1}^{\infty} (f(x_k) - f(x_{k+1})) \\
&\geq \sum_{k \in \mathcal{I}} \tau_2 [\phi_k(0) - \phi_k(s_k)] \\
&\geq \sum_{k \in \mathcal{I}} \tau_2 \tau \delta \min[\Delta_k, \delta/||B_k||_2] \\
&\geq \sum_{k \in \mathcal{I}} \tau_2 \tau \delta \min[\nu, \delta]/M_k
\end{aligned} \tag{3.44}$$

which shows that inequality (3.39) is also true. Now inequality (3.38) follows from the definition of M_k. Therefore, from Lemma 3.5, inequality (3.40) holds, which contradicts (3.41). The contradiction shows that the theorem is true. □

Powell's result is strengthen by Shultz, Schnabel and Byrd (1985) with some additional conditions:

Theorem 3.7 *Under the conditions of Theorem 3.6, if $\tau_0 > 0$ and $\{||B_k||_2\}$ is bounded, then the sequence $\{x_k\}$ generated by Algorithm 3.1 satisfies*

$$\lim_{k \to \infty} ||g_k||_2 = 0. \tag{3.45}$$

The condition (3.41) is weaker than the uniformly boundedness of B_k, and it allows the matrices B_k to be updated by some known quasi-Newton formulae such as Powell's symmetric Broyden (BSP) formula (see, [24], [26]) or by the BFGS method.

Powell(1970a) shows that the superlinear convergence of his trust region algorithm where B_k is updated by the PSB formula under the assumption that the trial steps $s_k(k=1,2,...)$ satisfy a "strict linear independence condition". Similar to that of [7], Powell(1975) establishs the superlinear convergence property of Algorithm 3.1. The following theorem is a slightly generalized form of Powell's superlinear convergence result.

Theorem 3.8 *Assume the trial step s_k computed in Step 2 of Algorithm 3.1 is a solution of subproblem (3.2)-(3.3). If $\epsilon = 0$ and the sequence $\{x_k\}$ generated by Algorithm (3.1) converges to x^*, if $\nabla^2 f(x)$ is continuous in a neighbourhood of x^* and $\nabla^2 f(x^*)$ is positive definite, and if the condition*

$$\lim_{k\to\infty} ||(\nabla^2 f(x^*) - B_k)s_k||_2/||s_k||_2 = 0 \tag{3.46}$$

is satisfied, then the sequence x_k converges to x^* Q-superlinearly in the sense that

$$\lim_{k\to\infty} ||x_{k+1} - x^*||_2/||x_k - x^*||_2 = 0. \tag{3.47}$$

Proof Due to (3.13), for each k, there exists a $\lambda_k \geq 0$ such that

$$(B_k + \lambda_k I)s_k = -g_k. \tag{3.48}$$

Hence we have that

$$s_k^T B_k s_k + s_k^T g_k = -\lambda_k ||s_k||_2^2 \leq 0. \tag{3.49}$$

The positive definiteness of $\nabla^2 f(x^*)$ implies that there exists a positive constant η such that

$$s^T \nabla^2 f(x^*) s \geq \eta ||s||_2^2, \quad \forall\ s \in \Re^n. \tag{3.50}$$

The inequalities (3.49), (3.50) and relation (3.46) show that

$$\begin{aligned} ||s_k||_2 ||g_k||_2 &\geq -s_k^T g_k \geq s_k^T B_k s_k \\ &= s_k^T \nabla^2 f(x^*) s_k + o(||s_k||_2^2) \\ &\geq \eta ||s_k||_2^2 + o(||s_k||_2^2) \end{aligned} \tag{3.51}$$

Thus, it follows that

$$||s_k||_2 \leq \frac{2}{\eta} ||g_k||_2 \tag{3.52}$$

holds for all large k. Similarly it can be shown that

$$\begin{aligned} \phi_k(0) - \phi_k(s_k) &= -g_k^T s_k - \frac{1}{2} s_k^T B_k s_k \\ &= -g_k^T s_k - s_k^T B_k s_k + \frac{1}{2} s_k^T B_k s_k \\ &\geq \frac{1}{2} s_k^T B_k s_k \geq \frac{1}{2} \eta ||s_k||_2^2 + o(||s_k||_2^2). \end{aligned} \tag{3.53}$$

and that

$$\begin{aligned} Pred_k - Ared_k &= f(x_k + s_k) - f(x_k) - g_k^T s_k - \frac{1}{2} s_k^T B_k s_k \\ &= \frac{1}{2} s_k^T (\nabla^2 f(x^*) - B_k) s_k + o(||s_k||_2^2) \\ &= o(||s_k||_2^2). \end{aligned} \tag{3.54}$$

Now relations (3.53) and (3.54) imply that

$$\lim_{k\to\infty} r_k = 1, \tag{3.55}$$

which shows that $\Delta_{k+1} \geq \Delta_k$ and $x_{k+1} = x_k + s_k$ for all large k. Therefore $||s_k||_2 < \Delta_k$, consequently

$$B_k s_k = -g_k \tag{3.56}$$

holds for all sufficiently large k. Now (3.56), (3.46), (3.52) and the fact that $x_{k+1} = x_k + s_k$ for large k shows that

$$\lim_{k\to\infty} \frac{||g_{k+1}||_2}{||g_k||_2} = 0. \tag{3.57}$$

Due to the positive definiteness of $\nabla^2 f(x^*)$, it can be shown that (3.57) is equivalent to (3.47). □

The above theorem requires weaker conditions than the original superlinear convergence result of [26]. For example, we do not assume any boundedness conditions for B_k.

It is also shown by [26] that B_k updated by the PSB formula gives the limit (3.46), consequently superlinear convergence follows. However, Powell's superlinear convergence result requires that B_k is updated at every iteration, even at a failed iteration. As updating B_k requires the evaluation of $g(x_k + s_k)$, therefore even at an unacceptable point $x_k + s_k$ which satisfies $f(x_k + s_k) > f(x_k)$ we still have to compute $g(x_k + s_k)$. To avoid the evaluation of $g(x_k + s_k)$ when s_k is unacceptable, Khalfan (1989) suggests to update B_k by the following formula

$$B_{k+1} = B_k + 2[f(x_k + s_k) - f(x_k) - s_k^T g_k - \frac{1}{2} s_k^T B_k s_k] \frac{s_k s_k^T}{||s_k||_2^4} \tag{3.58}$$

whenever $f(x_k + s_k) \geq f(x_k)$, thus there is no needs to compute $g(x_k + s_k)$ at such iterations. Superlinear convergence remains true after this modification. More details can be seen in [16].

If $B_k = \nabla^2 f(x_k)$ for all k, Algorithm 3.1 is called Newton's method with trust regions. In this special case, a stronger global convergence result can be established.

Theorem 3.9 *Assume that $B_k = \nabla^2 f(x_k)$ for all k, and assume that s_k is an approximate solution of subproblem (3.2)-(3.3) such that the predicted reduction is at least a fraction of the maximum reduction of the model, If $\epsilon = 0$, then the sequence $\{x_k\}$ generated by Algorithm 3.1 satisfies that*

$$\liminf_{k\to\infty} \{||g_k||_2 + max[-\sigma_n(\nabla^2 f(x_k)), 0]\} = 0. \tag{3.59}$$

Moreover, if $\tau_0 > 0$, then we have that

$$\lim_{k\to\infty} \{||g_k||_2 + max[-\sigma_n(\nabla^2 f(x_k)), 0]\} = 0. \tag{3.60}$$

A direct corollary of the above theorem is that if Newton's method with trust regions converges to x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive semi-definite. Similar to Theorem 3.8, it can be shown that Newton's method with trust regions converges quadratically. More details about Newton's method with trust region techniques can be found in [13], [30], [31] and [32].

A solution s_k of the trust region subproblem (3.2)-(3.3) is also a sufficiently descently direction. Hence simply throwing away s_k whenever $f(x_k + s_k) \geq f(x_k)$ may not be the best choice, as condition $s_k^T g_k < 0$ implies that a line search can be carried out along s_k. [22] gives a trust region algorithm that carries a backtracking line search whenever the computed trial step s_k is unacceptable.

4. Constrained Optimization

For constrained problems, most trust region subproblems can be regarded as some kind of modification of the SQP subproblem of line search algorithm, which has the following form:

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{4.1}$$

$$\begin{aligned} s.\ t. \quad c_i(x_k) + d^T\nabla c_i(x_k) &= 0 \qquad i = 1, 2, \ldots, m_e; & (4.2)\\ c_i(x_k) + d^T\nabla c_i(x_k) &\geq 0 \qquad i = m_e + 1, \ldots, m & (4.3)\end{aligned}$$

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 (4.1)-(4.3), have the following form:

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{4.4}$$

$$\begin{aligned} s.\ t. \quad \theta_k c_i(x_k) + d^T\nabla c_i(x_k) &= 0 \qquad i = 1, 2, \ldots, m_e; & (4.5)\\ \theta_k c_i(x_k) + d^T\nabla c_i(x_k) &\geq 0 \qquad i = m_e + 1, \ldots, m & (4.6)\\ ||d|| &\leq \Delta_k & (4.7)\end{aligned}$$

where $\theta_k \in (0, 1]$ is a parameter (see Byrd, Schnabel and Shultz [2] and Vardi [35]). Parameter θ_k is introduced to overcome the possible nonfeasibility of the linearized constraints (4.2)-(4.3) in the trust region (4.7). Trial steps of the trust region algorithms that apply null space techniques can also be reviewed as solutions of (4.4)-(4.7) (for example, see [23]).

Another trust region subproblem is obtained by replacing the linearized constraints (4.2)-(4.3) by a single quadratic constraint. It can be written as:

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \tag{4.8}$$

$$\begin{aligned} s.\ t. \quad ||(c_k + A_k^T d)^-||_2 &\leq \xi_k & (4.9)\\ ||d||_2 &\leq \Delta_k, & (4.10)\end{aligned}$$

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 (4.8)-(4.10) are given by Celis, Dennis and Tapia [3] and Powell and Yuan [29].

Trust region subproblems can also derived by using exact penalty functions. The following trust region subproblem ([39]) is based on the L_∞ exact penalty function:

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c_k + A_k^T d)^-||_\infty = \Phi_k(d) \tag{4.11}$$

$$s.\ t. \quad ||d|| \leq \Delta_k. \tag{4.12}$$

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 (4.11)-(4.12) are also similar to trust region algorithms for nonsmooth optimization.

Once a trial step s_k is computed by solving the trust region subproblem, the predicted reduction $Pred_k$ is defined by the reduction of some approximate function $\bar{\phi}_k(d)$. It should be noted that in general $\phi_k(d) - \bar{\phi}(d)$. A merit function $P_k(x)$ is used to define the actual reduction $Ared_k$. $P_k(x)$ is normally some penalty function. And the functions $\bar{\phi}_k(d)$ and $P_k(x)$ are so constructed that

$$\bar{\phi}_k(d) - \bar{\phi}_k(0) = P_k(x_k + d) - P_k(x_k) + o(||d||) \tag{4.13}$$

when $||d||$ is very small.

The algorithm can be stated as follows:

Algorithm 4.1 *(Trust Region Algorithm for Constrained Optimization)*

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

Step 2  If $||g_k||_2 \leq \epsilon$ then stop;
        Solve a trust region subproblem, giving $s_k$.

Step 3  Compute $r_k = Pred_k / Ared_k$;
        Set $x_{k+1}$ by (3.7);
        Choose $\Delta_{k+1}$ that satisfies (3.8)

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

Similar to unconstrained optimization, convergence of trust region algorithms for constrained optimization depends on some lower bound condition of the predicted reduction, such as

$$pred_k \geq \delta \epsilon_k \min[\Delta_k, \epsilon_k / ||B_k||] \tag{4.14}$$

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

$$\epsilon_k = ||c_k^-|| + ||g_k - A_k \lambda_k|| \tag{4.15}$$

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

$$pred_k \geq \bar{\delta} \Delta_k \tag{4.16}$$

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

$$\sum_{k=1}^{\infty} \Delta_k < \infty. \tag{4.17}$$

Thus $\Delta_k \to 0$. This and relation (4.13) imply that

$$r_k = \frac{P(x_k) - P(x_k + s_k)}{pred_k} \to 1. \tag{4.18}$$

The above limit shows that $\Delta_{k+1} \geq \Delta_k$ which contradicts (4.17). 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 (4.14) 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 (4.14).

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 (4.1)-(4.3). It is well known that under certain conditions the SQP step d_k^* is superlinearly convergent in the sense that

$$\lim_{k\to\infty} ||x_k + d_k^* - x^*||/||x_k - x^*|| = 0. \tag{4.19}$$

Therefore to prove local superlinear convergence

$$\lim_{k\to\infty} ||x_{k+1} - x^*||/||x_k - x^*|| = 0, \tag{4.20}$$

we need to show that

$$||s_k - d_k^*|| = o(||d_k^*||) \tag{4.21}$$

$$x_{k+1} = x_k + s_k \tag{4.22}$$

holds for all large k. In order to have the property (4.21), the trust region subproblem should be a good approximation of the SQP subproblem. The validity of (4.22) depends on suitable choice of the merit function.

For most algorithms, it can be shown that

$$s_k = d_k^* \tag{4.23}$$

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 [4], and second order correction step was first suggested by Fletcher [12].

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 (4.11)-(4.12), the second order correction subproblem can be as follows

$$\min \; g_k^T d + \frac{1}{2} d^T B_k d + \sigma_k ||(c(x_k + s_k) + A_k^T (d - s_k))^- ||_\infty \tag{4.24}$$

$$s.\ t. \quad ||d|| \le \Delta_k. \tag{4.25}$$

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

$$P(x_k + s_k + \hat{s}_k) < P(x_k) \tag{4.26}$$

holds for all large k. Hence if condition (4.21) is satisfied, it follows from (4.19) that that

$$\lim_{k\to\infty} ||x_k + s_k + \hat{s}_k - x^*||/||x_k - x^*|| = 0. \tag{4.27}$$

Relation (4.26) 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 (4.20) follows from (4.27), (4.21) and (4.19).

5. Nonsmooth Optimization

If one of $f(x)$, $c_i(x)(i = 1, ..., m)$ is nonsmooth, or in other words nondifferentiable, problem (1.1)-(1.1) is called a nonsmooth optimization problem, or a nondifferentiable optimization problem. From exact penalty function theory, under certain conditions the constrained nonsmooth problem (1.1)-(1.3) is equivalent to a unconstrained nonsmooth optimization

$$\min_{x\in\Re^n} f(x) + \sigma||c^-(x)||. \tag{5.1}$$

Therefore, it is quite common to study nonsmooth optimization by considering only unconstrained nonsmooth optimization problems.

A special class of nonsmooth optimization problems are "composite nonsmooth optimization" problem

$$\min_{x\in\Re^n} \bar{f}(x) = f(x) + h(F(x)), \tag{5.2}$$

where $F(x) = (f_1(x), f_2(x), ..., f_m(x))^T$, $h(.)$ is a convex functioned defined in $\Re^m$, and $f(x)$, $f_i(x)(i = 1, ..., m)$ are $m+1$ continuous differentiable functions defined in $\Re^n$. It is quite clear that one direct application of (5.2) is to solve constrained smooth optimization problems. This form of the objective function in (5.2) occurs frenquently in discrete approximation and data fitting calculations. Another special subclass of (5.2) is the minimization of some norm of a set of nonlinear equations (see, [9], [10], [17], and [28]). Algorithms for (5.2) can be extended to general nonsmooth optimization (for example, see [5] and [33]).

For the simplification of notation, we denote $g(x) = \nabla f(x)$, $A(x) = \nabla F(x)^T$, and

$$\psi_\rho(x) = h(F(x)) - \min_{||d||\le\rho} [g(x)^T d + h(F(x) + A(x)^T d)], \tag{5.3}$$

$$DF(x; d) = g^T d + \sup_{\lambda\in\partial h(F(x))} \lambda^T A(x)^T d\ , \tag{5.4}$$

where $\rho \ge 0$ and where $\partial h(F(x))$ is the subgradient of $h(.)$, evaluated at $F(x)$. x^* is called a stationary point of $\bar{f}(x)$ if

$$DF(x^*; d) \ge 0, \quad \forall\ d \in \Re^n, \tag{5.5}$$

which is the same as the first order condition of [13].

One can prove that a sequence $\{x_k \mid k = 1, 2, ...\}$ has an accumulation point at which the first order condition holds is equivalent to the limit

$$\liminf_{k\to\infty} \psi_1(x_k) = 0. \tag{5.6}$$

A model trust region algorithm that is first given by Fletcher (1982a). The subproblem in the k-th iteration is

$$\min_{d\in\Re^n} g_k^T d + \frac{1}{2} d^T B_k d + h(F_k + A_k^T d) = \phi_k(d) \tag{5.7}$$

$$s.\ t. \qquad ||d|| \le \Delta_k \tag{5.8}$$

where $||.||$ is a given norm in $\Re^n$ and $\Delta_k > 0$ is the trust region bound in the k-th iteration. It is easy to see that function $\phi_k(d)$ defined in (5.7) is the sum of a quadratic function and a convex function. $\phi_k(d)$ is also convex if B_k is positive semi-definite. Another special case is that $h(F)$ is a polyhedral convex function of the form

$$h(F) = \max_{1\le i\le I} (u_i^T F + \beta_i), \tag{5.9}$$

where $u_i \in \Re^m$, $\beta_i \in \Re$ $(i = 1, ..., I)$ are given vectors and constants respectively, and where I is a positive integer. In this case, $\phi_k(d)$ is a piecewise quadratic function. If the norm $||.||$ in (5.8) is the infinite norm or the 1-norm, subproblem (5.7)-(5.8) can be solved by using techniques such as in [1], and it can can also be rewritten as linearly constrained quadratical programming calculations.

Let s_k be a solution of subproblem (5.7)-(5.8). The prediction reduction and the actual reduction are defined by

$$Pred_k = \phi_k(0) - \phi_k(s_k) \tag{5.10}$$
$$Ared_k = \bar{f}(x_k) - \bar{f}(x_k + s_k). \tag{5.11}$$

The algorithm can be stated as follows.

Algorithm 5.1 *(Trust Region Algorithm for Composite Nonsmooth 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 $\psi_1(x_k) \le \epsilon$ then stop;
        Solve (5.7)-(5.8) giving $s_k$.

Step 3  Compute $r_k = Pred_k/Ared_k$;
        Set $x_{k+1}$ by (3.7);
        Choose $\Delta_{k+1}$ that satisfies (3.8)

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

Similar to (3.23), the following descent condition for the trust region trial step holds.

Lemma 5.2 *Let s_k be a solution of (5.7)-(5.8), then inequality*

$$\phi_k(0) - \phi_k(s_k) \ge \frac{1}{2}\psi_{\Delta_k}(x_k) \min\{1, \psi_{\Delta_k}(x_k)/||B_k||_2\Delta_k^2\} \tag{5.12}$$

holds.

Using the above lemma, we can easily establish the following global convergence result.

Theorem 5.3 *Let $\epsilon = 0$ in Algorithm 5.1, if there exist positive constants τ_5 and τ_6 such that*

$$||B_k||_2 \le \tau_5 + \tau_6 \sum_{i=1}^{k} \Delta_i \tag{5.13}$$

holds for all k, if Δ_k is bounded above and if $\bar{f}(x_k)$ is bounded below, then (5.6) holds, or in other words, $\{x_k\}$ is not bounded away from stationary points of $\bar{f}(x)$.

Similar to Lemma 3.4, we can prove the following lemma (Details can be found in [37]):

Lemma 5.4 *If there exists a constant $\delta > 0$ such that $\psi_1(x_k) \ge \delta$ holds for all k, then there exists a constant $\eta > 0$ such that*

$$\Delta_k \ge \eta \frac{1}{M_k} \tag{5.14}$$

holds for all k.

Now, using the above lemma and Lemma 3.5, one can show the following convergence result.

Theorem 5.5 *Theorem 5.3 is still true if condition (5.13) is replaced by*

$$\sum_{k=1}^{\infty} \frac{1}{M_k} = \infty, \tag{5.15}$$

where M_k is defined in the previous section.

Unfortunately, Algorithm 5.1 may not converge superlinearly.

Lemma 5.6 *(Yuan, 1984) For any given $0 \le \tau_0 < \tau_2 < 1 < \tau_1$, there exist $\tau_3, \tau_4 \in (0,1)$ such that by suitable choices of initial point and initial trust region bound, Algorithm 5.1 applied to the problem given in the beginning of this section may converge only linearly, though $B_k = G^*_\infty$, and strict complementarity and second order sufficiency conditions are satisfied.*

To overcome this difficulty, Fletcher (1982b) presents a trust region algorithm with a second order correction. On some iterations, the following "second order correction" subproblem

$$\begin{aligned} \min_{d \in \Re^n} \hat{\phi}_k(d) \equiv \phi_k(s_k + d) & \\ + h(F(x_k + s_k) + A_k^T d) - h(F(x_k) + A_k^T (s_k + d)) & \qquad (5.16) \end{aligned}$$

$$s.\ t. \qquad ||s_k + d|| \le \Delta_k, \tag{5.17}$$

is also solved. It is shown that Fletcher's second order correction method is superlinearly convergent ([38]).

References

[1] R.H. Bartels, A.R. Conn and J.W. Sinclair, "Minimization techniques for piecewise differentiable functions: the L_1 solution to an overdetermined linear system", *SIAM J. Numer. Anal.* 15(1978) 224-241.

[2] 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.

[3] 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.

[4] 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.

[5] J.E. Dennis, Jr., S.-B. Li and R.A. Tapia, "A unified approach to global convergence of trust region methods for nonsmooth optimization", Report, TR89-5, Dept. of Math. Rice University, Houston, USA, 1989.

[6] J.E. Dennis and H.H.W. Mei, "Two new unconstrained optimization algorithms which use function and gradient values", *J. Opt. Theory and Applns.* 28(1979) 453-482.

[7] J.E. Dennis and J.J. Moré, "A characterization of superlinear convergence and its application to quasi-Newton methods", *Mathematics of Computation* 28(1974) 549-560.

[8] J.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983)

[9] 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.

[10] M. El Hallabi and R.A. Tapia, "A global convergence theory for arbitrary norm trust-region methods for nonlinear equations" Report MASC TR 87-25, Rice University, Houston, USA. (revised 1993)

[11] R. Fletcher, "A model algorithm for composite NDO problem", *Math. Prog. Study* 17(1982) 67-76. (1982a)

[12] R. Fletcher, "Second order correction for nondifferentiable optimization",

in: G.A. Watson, ed., *Numerical Analysis* (Springer-Verlag, Berlin, 1982) pp. 85-115. (1982b)

[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] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares", *Qart. Appl. Math.* 2(1944) 164-166.

[16] H.F.H. Khalfan, *Topics in quasi-Newton methods for unconstrained optimization.* PhD thesis, University of Colorado, 1989.

[17] K. Madsen, "An algorithm for the minimax solution of overdetermined systems of nonlinear equations", *J. Inst. Math. Appl.* 16(1975) 1-20.

[18] 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.

[19] 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.

[20] J.J. Moré and D.C. Sorensen, "Computing a trust region step", *SIAM J. Sci. Stat. Comp.* 4(1983) 553-572.

[21] D.W. Marquardt, "An algorithm for least-squares estimation of nonlinear inequalities", *SIAM J. Appl. Math.* 11(1963) 431-441.

[22] J. Nocedal and Y. Yuan, "Combining trust region and line search techniques", Report NAM06, EECS, Northwestern University, USA, 1991.

[23] E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph. D. Thesis, University of Colorado at Boulder, 1989.

[24] 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)

[25] 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)

[26] 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.

[27] 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.

[28] M.J.D. Powell, "On the rate of convergence of variable metric algorithms for unconstrained optimization", in: Z. Ciesielki and C. Olech, eds., *Proceeding of the International Congress of Mathematicians* (Elsevier, New York, 1984) pp. 1525-1539. (1984b)

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

[30] 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.

[31] D.C. Sorensen, "Newton's method with a model trust region modification", *SIAM J. Numer. Anal.* 20(1982) 409-426.

[32] T. Steihaug, "The conjugate gradient method and trust regions inlarge scale optimization", *SIAM J. Numer. Anal.* 20(1983) 626-637.

[33] P. Terpolili, "Trust regin method in non-differentiable optimization: the use of exact and inexact local method", presented at 14th Intern. Symposium on Mathematical Programming, Amsterdam, The Netherlands, 1991.

[34] S.W. Thomas, "Sequential estimation techniques for quasi-Newton algorithms", Technical Report, TR 75-227, Dept of Computer Science, Cornell University, Ithaca, NY, USA, 1975.

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

[36] Y. Yuan, "An example of only linearly convergence of trust region algorithms for nonsmooth optimization", *IMA J. Numerical Analysis* 4(1984) 327-335.

[37] Y. Yuan, "Conditions for convergence of trust region algorithms for nonsmooth optimization", *Mathematical Programming* 31(1985) 220-228. (1985a)

[38] Y. Yuan, "On the superlinear convergence of a trust region algorithm for nonsmooth optimization", *Mathematical Programming* 31(1985) 269-285. (1985b)

[39] Y. Yuan, "A new trust region algorithm for nonlinear optimization", in: D. Bainov and V. Covachev, eds., *Proceedings of the Frist International Colloquium on Numerical Analysis* (VSP, Zeist, 1993) pp. 141-152.