International Journal of Pure and Applied Mathematics
Volume 44 No. 3 2008, 311-325

SOLVING NONSMOOTH CONVEX OPTIMIZATION WITH A NONMONOTONE TRUST REGION ALGORITHM

Yan Zhao[1 §], Nengzhu Gu[2]
[1,2]School of Science
University of Shanghai for Science and Technology
Shanghai, 200093, P.R. CHINA
[1]e-mail: zhaoyanem@hotmail.com
[2]e-mail: gnzemail@hotmail.com

Abstract: This paper concerns a nonmonotone trust region algorithm for nonsmooth convex optimization problems. The original nonsmooth function was converted into a continuously differentiable function via the Moreau-Yosida regularization, then used approximate values of the converted function and its gradient, the corresponding subproblem can be solved by a nonmonotone scheme. Under suitable assumptions, the algorithm is proved to be global and superlinear convergence.

AMS Subject Classification: 90C30
Key Words: convex optimization, Moreau-Yosida regulariztion, trust region method, convergence

1. Introduction

In this paper we consider the following unconstrained optimization problem

$$\min_{x\in\Re^n} f(x), \tag{1}$$

where $f : \Re^n \to \Re$ is in general nondifferentiable. Throughout this paper, we assume that $f(x)$ is strongly convex, say, there exists a constant $b > 0$ such that

$$f(ax + (1-a)x') \le af(x) + (1-a)f(x') - \frac{1}{2}ba(1-a)\|x - x'\|^2$$

for all $x,\ x' \in \Re^n$ and $a \in (0,\ 1)$. The constant b is called the modulus of

Received: September 30, 2007

§Correspondence author

strong convexity. Associated with problem (1), the continuously differentiable function

$$\min_{x\in\Re^n} F(x), \tag{2}$$

where $F : \Re^n \to \Re$, is so-called the Moreau-Yosida regularization of f. More precisely, $F(x)$ is given as

$$F(x) = \min_{z\in\Re^n}\{f(z) + \frac{1}{2\lambda}\|z - x\|^2\}, \tag{3}$$

where λ is a positive parameter and $\|\cdot\|$ denotes the Euclidean norm. In this study, we fix the value of parameter λ. For convenience of notation, we use both g and ∇F to denote the derivative of F alternatively. We first cite two propositions from [11] to illuminate some basic properties of the gradient of F.

Proposition 1. *Function F is finite-valued, convex and everywhere differentiable with the gradient*

$$g(x) \equiv \nabla F(x) = \frac{1}{\lambda}(x - p(x)),$$

where $p(x)$ is the unique minimizer in (3), i.e.,

$$p(x) = arg\ \min_{z\in\Re^n}\{f(z) + \frac{1}{2\lambda}\|z - x\|^2\}.$$

Moreover,

$$\|g(x) - g(x')\|^2 \le \frac{1}{\lambda}(g(x) - g(x'))^T(x - x')$$

and

$$\|g(x) - g(x')\| \le \frac{1}{\lambda}\|x - x'\|$$

for all $x,\ x' \in \Re^n$.

This proposition shows, in particular, that the mapping $g : \Re^n \to \Re^n$ is globally Lipschitz continuous. Then by Rademacher's Theorem, g is differentiable almost everywhere. Thus here we can define

$$\partial g(x) = co\{V \in \Re^{n\times n} \mid V = \lim_{x_i\to x} \nabla g(x_i),\ x_i \in \Omega_g\},$$

where $\Omega_g = \{x \in \Re^n \mid g \text{ is differentiable at } x\}$. According to the definition of [3], $\partial g(x)$ is called the generalized Jacobian matrix of g at x.

The next proposition formally states the equivalence between problems (1) and (2).

Proposition 2. *The following statements are equivalent:*

1. x minimizes f on $\Re^n$;

2. $x = p(x)$;
3. $g(x) = 0$;
4. *x minimizes F on $\Re^n$.*

It has been shown in [11] that if f is strongly convex with modulus $b > 0$, then F is also strongly convex with modulus $\frac{b}{b\lambda+1}$. The strong convexity of F implies that the optimal solution is unique.

Because the Moreau-Yosida regularization is defined to minimize a problem involving f, therefore, it is practically impossible in general that evaluate the exact values of objective function F and its gradient g at an arbitrary point x. To overcome this difficulty, Fukushima and Qi [7] illustrate the possibility of utilizing approximations of these values instead of their exact values. Motivated by their approach, throughout the remainder of the paper, we will use the approximate values of objective function F and its gradient g. We assume that, for each $x \in \Re^n$ and $\epsilon > 0$, an approximation of $p(x)$ in (3), say, $p^\alpha(x,\ \epsilon)$, is found such that

$$f(p^\alpha(x,\ \epsilon)) + \frac{1}{2\lambda}\|p^\alpha(x,\ \epsilon) - x\|^2 \leq F(x) + \epsilon.$$

Several implementable procedures have been introduced for finding such an approximate minimizer (see, e.g., [6, 1, 4]). Associated with approximate minimizer $p^\alpha(x,\ \epsilon)$, we now introduce two notations, $F^\alpha(x,\ \epsilon)$ and $g^\alpha(x,\ \epsilon)$ to denote the approximations of objective function $F(x)$ and its $g(x)$, respectively, where

$$F^\alpha(x,\ \epsilon) = f(p^\alpha(x,\ \epsilon)) + \frac{1}{2\lambda}\|p^\alpha(x,\ \epsilon) - x\|^2 \tag{4}$$

and

$$g^\alpha(x,\ \epsilon) = \frac{1}{\lambda}(x - p^\alpha(x,\ \epsilon)).$$

The next proposition concerns some properties of $F^\alpha(x,\ \epsilon)$ and $g^\alpha(x,\ \epsilon)$, these properties also can be seen in [7] and [18].

Proposition 3. *Let $\epsilon,\ \epsilon_x,\ \epsilon_y$ be arbitrary positive numbers and $\epsilon_z = \max\{\epsilon_x,\ \epsilon_y\}$. Suppose that f is strongly convex with modulus b. Then the following inequalities hold:*

1. $F(x) \leq F^\alpha(x,\ \epsilon) \leq F(x) + \epsilon$;
2. $\|p^\alpha(x,\ \epsilon) - p(x)\| \leq \sqrt{2\lambda\epsilon}$;
3. $\|g^\alpha(x,\ \epsilon) - g(x)\| \leq \sqrt{\frac{2\epsilon}{\lambda}}$;
4. $\|g^\alpha(x,\ \epsilon_x) - g^\alpha(y,\ \epsilon_y)\| \leq \frac{1}{\lambda}\|x - y\| + \sqrt{\frac{8\epsilon_z}{\lambda}}$;

5. $(g^{\alpha}(x,\ \epsilon_x) - g^{\alpha}(y,\ \epsilon_y))^T(x-y) \geq \frac{b}{b\lambda+1}\|x-y\|^2 - \sqrt{\frac{8\epsilon_z}{\lambda}}\|x-y\|$;

6. $\|g^{\alpha}(x,\ \epsilon_x) - g^{\alpha}(y,\ \epsilon_y)\|^2 \leq \frac{1}{\lambda}(g^{\alpha}(x,\ \epsilon_x) - g^{\alpha}(y,\ \epsilon_y))^T(x-y) + \sqrt{\frac{32\epsilon_z}{\lambda^3}}\|x-y\| + \frac{6\epsilon_z}{\lambda}$.

Most algorithms proposed for optimization problems, especially for nonsmooth optimization problems are monotonic. However, Grippo, Lampariello and Lucidi [9] shown that monotonic schemes can considerably slow the rate of convergence in the intermediate stages of the minimization process, especially in the presence of narrow curved valley. To avoid this inherent shortcoming of monotonic technique, they introduced a nonmonotone line search technique, which is given as

$$f(x_k + \alpha d_k) \leq \max_{0\leq j\leq m(k)} f(x_{k-j}) + \delta\alpha\nabla f(x_k)^T d_k, \tag{5}$$

where $\delta \in (0,1)$ is a constant, N is an integer and

$$m(k) = \begin{cases} k, & k \leq N; \\ \min\{m(k-1)+1, N\}, & k > N. \end{cases} \tag{6}$$

This technique leads to a breakthrough in nonmonotonic algorithms for general nonlinear optimization problems (see, e.g., [5, 12, 20, 8]).

Recently, Zhang and Hager [22] stated that there exist some limitations with nonmonotone technique (5). First, a good function value generated in any iteration might be excluded due to the max in (5). Second, in many cases, the numerical performance is dependent on the choice of N. Moreover, for any given bound N on the memory, even an iterative method is generating R-linearly convergence for a strongly convex function, the iterates may not satisfy the condition (5) for k sufficiently large. To improve these limitations, they proposed a nonmonotone gradient method for unconstrained optimization. Their method requires an average of the successive function values decreasing, that is to say, the steplength α is computed to satisfy the following line search condition

$$f(x_k + \alpha d_k) \leq C_k + \delta\alpha\nabla f(x_k)^T d_k, \tag{7}$$

where

$$C_k = \begin{cases} f(x_k), & k = 1; \\ (\eta_{k-1}Q_{k-1}C_{k-1} + f(x_k))/Q_k, & k \geq 2, \end{cases} \tag{8}$$

$$Q_k = \begin{cases} 1, & k = 1; \\ \eta_{k-1}Q_{k-1} + 1, & k \geq 2, \end{cases} \tag{9}$$

and $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$. $\eta_{\min} \in (0, \eta_{\max})$ and $\eta_{\max} \in (0,1)$ are two chosen

parameters. The numerical results show that nonmonotone technique (7) is particularly efficient for unconstrained problems. To take the good aspects of this nonmonotone technique, Mo et al [13] incorporated it into trust region method and developed a nonmonotone algorithm. The numerical results indicate that the algorithm is robust and encouraging.

However, as we see, that nonmonotone technique (7) includes complicated parameter η_k and Q_k. Inspired by this observation, Gu and Mo [10] introduce another nonmonotone line search

$$f(x_k + \alpha d_k) \leq D_k + \delta\alpha\nabla f(x_k)^T d_k, \tag{10}$$

where D_k is a simple convex combination of the previous D_{k-1} and f_k, say

$$D_k = \begin{cases} f(x_k), & k = 1; \\ \eta D_{k-1} + (1-\eta) f(x_k) & k \geq 2 \end{cases} \tag{11}$$

for some fixed $\eta \in (0,1)$, or a variable $\eta_k = 1/k$. The numerical results in [10] indicated that nonmonotone technique (10) is as efficient as nonmonotone technique (7). Motivated by nonmonotone technique (10), we propose a nonmonotone trust region method for nonsmooth convex optimization in this study. Our algorithm can viewed as a generalization of the Sagara and Fukushima algorithm [19] from monotone to nonmonotone.

This paper is organized as follows. We present the nonmonotone trust region algorithm in Section 2. We discuss the global and superlinear convergence results of the algorithm in Section 3.

2. Algorithm

In this section, we outline the method in the form of algorithm. Associated with problem (4), the subproblem is given by

$$\begin{array}{ll} \min & \frac{1}{2} p^T B_k^\alpha p + g^\alpha(x_k,\ \epsilon_k)^T p = \phi_k(p) \\ s.t. & \|p\| \leq \Delta_k, \end{array} \tag{12}$$

where p is a trial step, $B_k^\alpha \in \Re^{n\times n}$ denotes $B_k^\alpha(x_k, \epsilon_k)$, is an approximate Hessian matrix of F^α at x_k, and Δ_k is a trust region radius computed by

$$\Delta_k = c^i \|g^\alpha(x_k,\ \epsilon_k)\| \|(\hat{B}_k^\alpha)^{-1}\| \tag{13}$$

for some fixed $c \in (0,1)$, where i is a nonnegative integer, $\hat{B}_k^\alpha = B_k^\alpha + \tau_k I$, I is the identify matrix. The safeguarding factor τ_k is to guarantee that $\hat{B}_k^\alpha$ is a positive definite matrix. If B_k is safely positive definite, let $\tau_k = 0$. Otherwise, let τ_k be a positive value such that $\hat{B}_k^\alpha$ positive definite.

Similar to [14], we solve subproblem (12) inaccurately such that $\|p_k\| \leq \Delta_k$ and

$$\phi_k(0) - \phi_k(p_k) \geq \beta \|g^\alpha(x_k,\ \epsilon_k)\| \min\{\Delta_k,\ \frac{\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k^\alpha\|}\} \tag{14}$$

are satisfied, where $\beta \in (0,\ \frac{1}{2}]$. To insure that the algorithm is nonmonotone, we compute the actual reduction of $F^\alpha(x,\ \epsilon)$ by

$$Arep_k = D^\alpha(x_k,\ \epsilon_k) - F^a(x_k + p_k, \epsilon_{k+1}), \tag{15}$$

where $D^\alpha(x_k, \epsilon_k)$ is a simple convex combination of the previous $D^\alpha(x_{k-1}, \epsilon_{k-1})$ and $F^\alpha(x_k, \epsilon_k)$

$$D^\alpha(x_k, \epsilon_k) = \begin{cases} F^\alpha(x_k, \epsilon_k), & k = 1; \\ \mu D^\alpha(x_{k-1}, \epsilon_{k-1}) + (1-\mu) F^\alpha(x_k, \epsilon_k) & k \geq 2 \end{cases} \tag{16}$$

for some fixed $\mu \in (0, 1)$. The predictive reduction of $F^\alpha(x,\ \epsilon)$ is computed by

$$Prep_k = \phi_k(0) - \phi_k(p_k). \tag{17}$$

This mechanism implies that the algorithm does not require objective function values monotonically decreasing.

We now describe the algorithm.

Algorithm 1. (Nonmonotone Trust Region Algorithm)

Step 1. Give $x_1 \in \Re^n$, c, η, μ, β, $r \in (0, 1)$, $\varrho > 0$, $\epsilon_1 > 0$, a symmetric matrix $B_1^\alpha \in \Re^{n \times n}$. Set $k = 1$.

Step 2. Compute $g^\alpha(x_k, \epsilon_k)$. If $\|g^\alpha(x_k, \epsilon_k)\| \leq \delta$, stop.

Step 3. Compute Δ_k, solve subproblem (12) inaccurately, such that $\|p_k\| \leq \Delta_k$ and (14) are satisfied.

Step 4. Compute γ_k, the ratio between actual reduction and predicted reduction by

$$\gamma_k = \frac{Arep_k}{Prep_k}. \tag{18}$$

Step 5. If $\gamma_k < \eta$, set $i = i + 1$, $\epsilon_k = \min\{\epsilon_1 \Delta_k^2, r\epsilon_k\}$, go to Step 3. Otherwise, go to Step 6.

Step 6. Set $x_{k+1} = x_k + p_k$, $\epsilon_{k+1} = r\epsilon_k$. Update symmetric matrix B_k^α.

Step 7. Set $k = k + 1$, $i = 0$, go to Step 2.

Remark 1. B_k^α can be updated by quasi-Newton formula, such as the original BFGS formula or a new BFGS-type formula introduced in [21].

3. Global and Superlinear Convergence

We now turn to analyze the convergence behaviors of Algorithm 1. Before we address some theoretical issues, we would like to make the following assumptions. Under these assumptions we first show that $\lim_{k\to\infty} \|g^\alpha(x_k,\ \epsilon_k)\| = 0$. Then based on this conclusion, we obtain the main results.

Assumption 1. The level set

$$\pounds(x_1,\ \epsilon_1) = \{x \in \Re^n \mid F^\alpha(x,\epsilon) \le F^\alpha(x_1,\ \epsilon_1)\}$$

is bounded for any given $x_1 \in \Re^n$ and $\epsilon_1 > 0$.

Assumption 2. $\|B_k^\alpha\|$ does not grow too rapidly.

We first give an important lemma.

Lemma 4. *Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then*

$$F^\alpha(x_{k+1},\epsilon_{k+1}) \le D^\alpha(x_{k+1},\epsilon_{k+1}) \le D^\alpha(x_k,\ \epsilon_k). \tag{19}$$

Proof. By the definition of $D^\alpha(x,\epsilon)$, we have

$$D^\alpha(x_{k+1},\epsilon_{k+1}) - F^\alpha(x_{k+1},\epsilon_{k+1}) = \mu(D^\alpha(x_k,\ \epsilon_k) - F^\alpha(x_{k+1},\epsilon_{k+1})) \tag{20}$$

and

$$D^\alpha(x_{k+1},\epsilon_{k+1}) - D^\alpha(x_k,\ \epsilon_k) = (1-\mu)(-D^\alpha(x_k,\ \epsilon_k) + F^\alpha(x_{k+1},\epsilon_{k+1})). \tag{21}$$

Because the algorithm does not generate a new iterate until an acceptable trial step is found, therefore we deduce from (18) and (14) that

$$D^\alpha(x_k,\epsilon_k) - F^\alpha(x_{k+1},\epsilon_{k+1}) \ge \eta\beta\|g^\alpha(x_k,\ \epsilon_k)\| \min\{\Delta_k,\ \frac{\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k^\alpha\|}\}. \tag{22}$$

Combining (20) and (22), we obtain that

$$D^\alpha(x_{k+1},\epsilon_{k+1}) - F^\alpha(x_{k+1},\epsilon_{k+1}) \ge \mu\eta\beta\|g^\alpha(x_k,\ \epsilon_k)\| \min\{\Delta_k,\ \frac{\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k^\alpha\|}\} \ge 0. \tag{23}$$

On the other hand, combining (21) and (22), we have that

$$D^\alpha(x_{k+1},\epsilon_{k+1}) - D^\alpha(x_k,\epsilon_k) \le -(1-\mu)\eta\beta\|g^\alpha(x_k,\ \epsilon_k)\| \min\{\Delta_k,\ \frac{\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k^\alpha\|}\} \le 0. \tag{24}$$

(23) and (24) indicate that (19) holds. □

The following lemma cited from [8] concerns the relation between p_k and $g^\alpha(x_k,\ \epsilon_k)$.

Lemma 5. *Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then there exists a positive constant $\bar{c}$ such that*

$$\|p_k\| \le \bar{c}\|g^\alpha(x_k,\ \epsilon_k)\|,\ k = 1,\ 2, \cdots .$$

Proof. The proof is similar to that of Lemma 4.2 in [8]. □

Next, we illustrate that Algorithm 1 is well defined. It is enough to prove that algorithm terminates in finite steps between Step 3 and Step 5.

Lemma 6. *Suppose that Assumptions 1 and 2 hold. Then Algorithm 1 does not cycle infinitely.*

Proof. Suppose first, for the purpose of deriving a contradiction, that algorithm cycles infinitely between Step 3 and Step 5. Consequently, $\lim_{k\to\infty}\Delta_k = 0$, $\lim_{k\to\infty}\epsilon_k = \mathrm{O}(\Delta_k^2) = 0$, $\gamma_k < \eta$ for k sufficiently large. Using the definition of γ_k and noting that $\eta \in (0, 1)$, we have $D^\alpha(x_k, \epsilon_k) - F^\alpha(x_{k+1}, \epsilon_{k+1}) < \eta(-\phi_k(p_k)) < -\phi_k(p_k)$. This relation leads to $D^\alpha(x_k, \epsilon_k) - F^\alpha(x_{k+1}, \epsilon_{k+1}) + \phi_k(p_k) < 0$. Now, we have from Lemma 4 that

$$F^\alpha(x_k, \epsilon_k) - F^\alpha(x_{k+1}, \epsilon_{k+1}) + \phi_k(p_k) \le D^\alpha(x_k, \epsilon_k) - F^\alpha(x_{k+1}, \epsilon_{k+1}) + \phi_k(p_k) < 0. \tag{25}$$

Using Proposition 3, Talyor's expansion and Proposition 1, we have

$$\begin{aligned}
0 &< F^\alpha(x_k + p_k, \epsilon_{k+1}) - F^\alpha(x_k, \epsilon_k) - \phi_k(p_k) \\
&\le F(x_k + p_k) + \epsilon_{k+1} - F(x_k) - \frac{1}{2}p_k^T B_k p_k - g^\alpha(x_k, \epsilon_k)^T p_k \\
&= -\frac{1}{2}p_k^T B_k p_k + (g(x_k + t_k p_k) - g^\alpha(x_k, \epsilon_k))^T p_k + r\epsilon_k \\
&= -\frac{1}{2}p_k^T B_k p_k + (g(x_k + t_k p_k) - g(x_k) + g(x_k) - g^\alpha(x_k, \epsilon_k))^T p_k + r\epsilon_k \\
&\le \frac{1}{2}\|B_k\|\|p_k\|^2 + \frac{t_k}{\lambda}\|p_k\|^2 + \sqrt{\frac{2\epsilon_k}{\lambda}}\|p_k\| + \epsilon_k,
\end{aligned} \tag{26}$$

where $t_k \in (0,\ 1)$.

Using $\|p_k\| \le \Delta_k$, (25), (26) and (14) we have

$$\begin{aligned}
|\gamma_k - 1| &= \left|\frac{D^\alpha(x_k, \epsilon_k) - F^\alpha(x_k + p_k,\ \epsilon_{k+1}) + \phi_k(p_k)}{-\phi_k(p_k)}\right| \\
&\le \left|\frac{F^\alpha(x_k, \epsilon_k) - F^\alpha(x_k + p_k,\ \epsilon_{k+1}) + \phi_k(p_k)}{-\phi_k(p_k)}\right|
\end{aligned}$$

$$= \frac{F^\alpha(x_k+p_k,\ \epsilon_{k+1}) - F^\alpha(x_k,\epsilon_k) - \phi_k(p_k)}{-\phi_k(p_k)}$$

$$\leq \frac{(\|B_k\| + \frac{2t_k}{\lambda})\Delta_k^2 + 2\sqrt{\frac{2\epsilon_k}{\lambda}}\Delta_k + 2\epsilon_k}{2\beta\|g^\alpha(x_k,\ \epsilon_k)\| \min\{\Delta_k,\ \frac{\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k\|}\}}. \tag{27}$$

Since x_k is not an optimal solution of problem (1), we have from Step 2 of the algorithm that there exists a constant $\delta > 0$ such that $\|g^\alpha(x_k,\ \epsilon_k)\| \geq \delta$ for any ϵ_k sufficiently small. Therefore, from (27) and Step 5 of algorithm, we get

$$|\gamma_k - 1| \leq \frac{(\|B_k\| + \frac{2t_k}{\lambda})\Delta_k^2 + 2\sqrt{\frac{2\epsilon_k}{\lambda}}\Delta_k + 2\epsilon_k}{2\beta\delta \min\{\Delta_k,\ \frac{\delta}{\|B_k\|}\}}$$

$$\leq \frac{(\|B_k\| + \frac{2t_k}{\lambda})\Delta_k^2 + 2\sqrt{\frac{2\epsilon_1}{\lambda}}\Delta_k^2 + 2\epsilon_1\Delta_k^2}{2\beta\delta \min\{\Delta_k,\ \frac{\delta}{\|B_k\|}\}}. \tag{28}$$

Using Assumption 2, we see that the right-hand side of (28) tends to zero as $\lim_{k\to\infty}\Delta_k = 0$. This implies $\gamma_k \to 1$ for k sufficiently large, which contradicts $\gamma_k < \eta$. □

Based on Assumption 1 and Lemma 4, we show that $\{x_k\}$ generated by algorithm is contained in the level set.

Lemma 7. *Assume that Assumption 1 holds, then sequence* $\{(x_k,\epsilon_k)\} \subset \pounds(x_1,\ \epsilon_1)$ *and* $\{D^\alpha(x_k,\epsilon_k)\}$ *is bounded below.*

Proof. By (16), we have $D^\alpha(x_1,\epsilon_1) = F^\alpha(x_1,\epsilon_1)$. Lemma 4 implies that $F^\alpha(x_{k+1},\epsilon_{k+1}) \leq D^\alpha(x_k,\epsilon_k) \leq F^\alpha(x_1,\epsilon_1)$. Since Assumption 1 implies that $\{F^\alpha(x_k,\epsilon_k)\}$ is bounded below, we know that $\{D^\alpha(x_k,\epsilon_k)\}$ is also bounded below. □

In order to establish the global convergence of the algorithm, we need to use Assumption 2, this assumption has been used in other researches (see, e.g. [14]). Here we define a sequence

$$M_k = \max_{1\leq j\leq k}(\tau_j + \|B_j^\alpha\|) \tag{29}$$

for all k. $\|B_k^\alpha\|$ does not grow too rapidly, which implies sequence $\{M_k\}$ satisfies

$$\sum_{k=1}^{\infty} 1/M_k = \infty. \tag{30}$$

Lemma 8. *Suppose that Assumptions 1 and 2 hold, sequence* $\{x_k\}$ *is*

generated by Algorithm 1. Then

$$\lim_{k\to\infty} \|p_k\| = 0. \tag{31}$$

Proof. Lemma 6 implies that $\gamma_k \geq \eta$ for k sufficiently large, therefore we have from the definition of γ_k that

$$D^\alpha(x_k, \epsilon_k) - F^\alpha(x_k + p_k, \epsilon_{k+1}) \geq -\eta\phi_k(p_k).$$

This inequality leads to

$$F^\alpha(x_k + p_k, \epsilon_{k+1}) \leq D^\alpha(x_k, \epsilon_k) + \eta\phi_k(p_k). \tag{32}$$

Now using the definition of $D^\alpha(x_k, \epsilon_k)$ and (32), we have

$$\begin{aligned} D^\alpha(x_{k+1}, \epsilon_{k+1}) &= \mu D^\alpha(x_k, \epsilon_k) + (1-\mu)F^\alpha(x_{k+1}, \epsilon_{k+1}) \\ &\leq \mu D^\alpha(x_k, \epsilon_k) + (1-\mu)D^\alpha(x_k, \epsilon_k) + (1-\mu)\eta\phi_k(p_k) \\ &= D^\alpha(x_k, \epsilon_k) + (1-\mu)\eta\phi_k(p_k), \end{aligned} \tag{33}$$

(33) gives us

$$\sum_{k=1}^{\infty} -(1-\mu)\eta\phi_k(p_k) \leq \sum_{k=1}^{\infty}(D^\alpha(x_k, \epsilon_k) - D^\alpha(x_{k+1}, \epsilon_{k+1})).$$

Lemma 7 together with this inequality give

$$\sum_{k=1}^{\infty} -\phi_k(p_k) < \infty. \tag{34}$$

Since $\Delta_k = c^i\|g^\alpha(x_k,\ \epsilon_k)\|\|(\hat{B}_k^\alpha)^{-1}\|$ and $\hat{B}_k^\alpha = B_k^\alpha + \tau_k I$, Lemma 6 implies that there exists a constant c^{min} such that $c^i \geq c^{min}$, therefore we have from (29) that

$$\begin{aligned} \Delta_k = c^i\|g^\alpha(x_k,\ \epsilon_k)\|\|(\hat{B}_k^\alpha)^{-1}\| &\geq \frac{c^i\|g^\alpha(x_k,\ \epsilon_k)\|}{\|\hat{B}_k^\alpha\|} \\ &= \frac{c^i\|g^\alpha(x_k,\ \epsilon_k)\|}{\|B_k^\alpha + \tau_k I\|} \geq \frac{c^{min}\|g^\alpha(x_k,\ \epsilon_k)\|}{\|M_k\|}. \end{aligned} \tag{35}$$

Due to (14), (35), (29) and Lemma 5, we obtain that

$$\begin{aligned} \sum_{k=1}^{\infty} -\phi_k(p_k) &\geq \sum_{k=1}^{\infty} \beta\|g^\alpha(x_k, \epsilon_k)\| \min\{\Delta_k, \frac{\|g^\alpha(x_k, \epsilon_k)\|}{\|B_k^\alpha\|}\} \\ &\geq \sum_{k=1}^{\infty} \beta\|g^\alpha(x_k, \epsilon_k)\| \min\{\frac{c^{min}\|g^\alpha(x_k,\ \epsilon_k)\|}{\|M_k\|}, \frac{\|g^\alpha(x_k, \epsilon_k)\|}{\|M_k\|}\} \\ &= \sum_{k=1}^{\infty} \beta\|g^\alpha(x_k, \epsilon_k)\| \frac{c^{min}\|g^\alpha(x_k,\ \epsilon_k)\|}{\|M_k\|} \geq \sum_{k=1}^{\infty} \frac{\beta c^{min}\|p_k\|^2}{\bar{c}^2 M_k}. \end{aligned} \tag{36}$$

Now, for the purpose of deriving a contradiction, we suppose that (31) does not hold, that is, there exists a constant ν such that

$$\|p_k\| \geq \nu, \qquad \forall\ k. \tag{37}$$

Using (34), (37) and the monotonicity of $\{M_k\}$, we have from (36) that

$$\sum_{k=1}^{\infty} \frac{\beta c^{min} \nu}{\bar{c}^2 M_k} \leq \infty.$$

The above inequality implies

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

this contradicts (30). Therefore $\lim_{k\to\infty} \|p_k\| = 0$. □

Theorem 9. *Suppose that Assumptions 1 and 2 hold, sequence $\{x_k\}$ is generated by Algorithm 1. Then either some $\{x_k\}$ satisfies the termination criteria and the algorithm terminates or*

$$\lim_{k\to\infty} \|g^{\alpha}(x_k,\ \epsilon_k)\| = 0. \tag{38}$$

Proof. Suppose that (38) does not hold, i.e., there exist a positive constant δ such that $\|g^{\alpha}(x_k, \epsilon_k)\| \geq \delta$ infinitely often. Using (14), (34) and (35), we obtain that

$$\begin{aligned}
\infty &\geq \sum_{k=1}^{\infty} \beta \|g^{\alpha}(x_k, \epsilon_k)\| \min\{\Delta_k, \frac{\|g^{\alpha}(x_k, \epsilon_k)\|}{\|B_k^{\alpha}\|}\} \\
&\geq \sum_{k=1}^{\infty} \beta \|g^{\alpha}(x_k, \epsilon_k)\| \min\{\frac{c^{min}\|g^{\alpha}(x_k,\ \epsilon_k)\|}{\|M_k\|}, \frac{\|g^{\alpha}(x_k, \epsilon_k)\|}{\|M_k\|}\} \\
&= \sum_{k=1}^{\infty} \beta \|g^{\alpha}(x_k, \epsilon_k)\| \frac{c^{min}\|g^{\alpha}(x_k, \epsilon_k)\|}{\|M_k\|} \\
&\geq \sum_{k=1}^{\infty} \frac{\beta \delta^2 c^{min}}{M_k}.
\end{aligned} \tag{39}$$

Inequality (39) leads to

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

This result contradicts (30). Therefore $\lim_{k\to\infty} \|g^{\alpha}(x_k,\ \epsilon_k)\| = 0$. □

Based on the above theorem, we obtain the main convergence result.

Theorem 10. *Suppose that Assumptions 1 and 2 hold, sequence $\{x_k\}$ is*

generated by Algorithm 1. Then

$$\lim_{k\to\infty} \|g(x_k)\| = 0. \tag{40}$$

Proof. The algorithm implies that $\epsilon_k \leq \epsilon_1 \Delta_k^2$ for all k, this inequality together with Proposition 3 (3) yield that

$$\begin{aligned} \|g(x_k)\| &= \|g(x_k) - g^\alpha(x_k,\ \epsilon_k) + g^\alpha(x_k,\ \epsilon_k)\| \\ &\leq \|g(x_k) - g^\alpha(x_k,\ \epsilon_k)\| + \|g^\alpha(x_k,\ \epsilon_k)\| \\ &\leq \sqrt{\frac{2\epsilon_k}{\lambda}} + \|g^\alpha(x_k,\ \epsilon_k)\| \leq \sqrt{\frac{2\epsilon_1}{\lambda}}\Delta_k + \|g^\alpha(x_k,\ \epsilon_k)\|. \end{aligned}$$

Note that the updating rule of Δ_k implies that $\Delta_k \to 0$ as $\|g^\alpha(x_k, \epsilon_k)\| \to 0$, therefore (40) holds. □

Let x^* be an arbitrary accumulation point of sequence $\{x_k\}$. By Theorem 10, we know that x^* minimizes F. Notice that objective function F is strong convex, optimal solution is unique. Therefore, $\{x_k\}$ converges to x^* uniquely.

In the rest of this section is to establish the superlinear convergence of the algorithm. To this end, we assume further that $g(x)$ is semismooth. Since $g : \Re^n \to \Re^n$ is globally Lipschiz continuous and semismooth, by the conclusions of [15], we can define the directional derivative of $g(x)$ at x, as

$$g'(x,\ d) = \lim_{t\to 0} \frac{g(x+td) - g(x)}{t}.$$

If $d \to 0$,

$$g(x+d) = g(x) + g'(x,\ d) + o(\|d\|). \tag{41}$$

Using Lemma 5 of [17], there exists a $V_k \in \partial g(x_k)$ for each k such that

$$g'(x_k,\ d) = V_k d,\ \forall d \in \Re^n. \tag{42}$$

Similarly, for $\bar{x}$, there is a $\bar{V} \in \partial g(\bar{x})$ such that $g'(\bar{x},\ d) = \bar{V}d,\ \forall d \in \Re^n$. In other words, there exists a corresponding matrix V_k for any given x_k.

We now prove the following superlinear convergence theorem.

Theorem 11. *Suppose that Assumptions 1 and 2 hold, $g(x)$ is semismooth. Suppose also that $\epsilon_k = o(\|p_k\|^2)$,*

$$\lim_{k\to\infty} \frac{\|(B_k^\alpha - V_k)p_k\|}{\|p_k\|} = 0 \tag{43}$$

and

$$\frac{\|B_k^\alpha p_k + g_k\|}{\|g_k\|} \leq \xi_k \tag{44}$$

are satisfied, where $\lim_{k\to\infty}\xi_k = 0$. *Then* $\{x_k\}$ *converges to* x^* *superlinearly.*

Proof. By (41) and (42), we have

$$\begin{aligned}\frac{\|g_{k+1}\|}{\|p_k\|} &= \frac{\|g_k + B_k^\alpha p_k + (V_k - B_k^\alpha)p_k + o(\|p_k\|)\|}{\|p_k\|} \\ &\le \frac{\|g_k + B_k^\alpha p_k\|}{\|g_k\|}\frac{\|g_k\|}{\|p_k\|} + \frac{\|(V_k - B_k^\alpha)p_k\|}{\|p_k\|} + \frac{o(\|p_k\|)}{\|p_k\|} \\ &\le \xi_k\frac{\|g_k\|}{\|p_k\|} + \frac{\|(V_k - B_k^\alpha)p_k\|}{\|p_k\|} + \frac{o(\|p_k\|)}{\|p_k\|}. \end{aligned} \tag{45}$$

From Proposition 1, we have

$$\|g_k\| \le \frac{1}{\lambda}\|p_k\| + \|g_{k+1}\|.$$

Dividing both sides by $\|p_k\|$, we get

$$\frac{\|g_k\|}{\|p_k\|} \le \frac{1}{\lambda} + \frac{\|g_{k+1}\|}{\|p_k\|}.$$

The above inequality together with (45) yield that

$$\frac{\|g_{k+1}\|}{\|p_k\|} \le \xi_k(\frac{1}{\lambda} + \frac{\|g_{k+1}\|}{\|p_k\|}) + \frac{\|(V_k - B_k^\alpha)p_k\|}{\|p_k\|} + \frac{o(\|p_k\|)}{\|p_k\|},$$

which implies

$$\frac{\|g_{k+1}\|}{\|p_k\|} \le \frac{(\frac{1}{\lambda}\xi_k + \frac{\|(V_k - B_k^\alpha)p_k\|}{\|p_k\|} + \frac{o(\|p_k\|)}{\|p_k\|})}{(1-\xi_k)}. \tag{46}$$

Note that $\xi_k \to 0$ and $\|p_k\| \to 0$ as $k \to \infty$, thus (43) and (46) gives us

$$\lim_{k\to\infty}\frac{\|g_{k+1}\|}{\|p_k\|} = 0. \tag{47}$$

Due to Proposition 3 and assumption condition $\epsilon_k = o(\|p_k\|^2)$, we obtain

$$\begin{aligned}\frac{\|g^\alpha(x_{k+1},\ \epsilon_{k+1})\|}{\|p_k\|} &\le \frac{\sqrt{\frac{2\epsilon_k}{\lambda}} + \|g_{k+1}\|}{\|p_k\|} \\ &\le \frac{o(\|p_k\|)}{\|p_k\|} + \frac{\|g_{k+1}\|}{\|p_k\|}, \end{aligned} \tag{48}$$

(47) and (48) yield

$$\lim_{k\to\infty}\frac{\|g^\alpha(x_{k+1},\ \epsilon_{k+1})\|}{\|p_k\|} = 0. \tag{49}$$

By Lemma 5 and (49),

$$\lim_{k\to\infty}\frac{\|g^\alpha(x_{k+1},\ \epsilon_{k+1})\|}{\bar{c}\|g^\alpha(x_k,\epsilon_k)\|} \le \lim_{k\to\infty}\frac{\|g^\alpha(x_{k+1},\ \epsilon_{k+1})\|}{\|p_k\|} = 0,$$

where $\bar{c}$ is a constant. Thus we prove the superlinear convergence result. □

References

[1] A. Auslender, Numerical methods for nondifferentiable convex optimization, *Math. Prog. Study*, **30** (1987), 102-126.

[2] X. Chen, M. Fukushima, Proximal quasi-Newton methods for nondifferentiable convex optimization, *Math. Prog.*, **85** (1999), 313-334.

[3] F. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, (1983).

[4] R. Correa, C. Lemaréchal, Convergence of some algorithm for convex minimization, *Math. Prog.*, **62** (1993), 261-275.

[5] N. Deng, Y. Xiao, F. Zhou, Nonmonotone trust region algorithm, *J. Optim. Theory Appl.*, **76** (1993), 259-285.

[6] M. Fukushima, A descent algorithm for nonsmooth convex optimization, *Math. Prog.*, **30** (1984), 163-175.

[7] M. Fukushima, L. Qi, A globally and superlinearly convergent algorithm for nonsmooth convex minimization, *SIAM J. Optim.*, **4** (1996), 1106-1120.

[8] J. Fu, W. Sun, Nonmonotone adaptive trust-region method for unconstrained optimization problems, *Appl. Math. Comput.*, **163** (2005), 489-504.

[9] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM J. Numer. Anal.*, **23** (1986), 707-716.

[10] N. Gu, J. Mo, Incorporating nonmonotone strategies into trust region method for unconstrained optimization, *Computers and Mathematics with Applications*, To Appear.

[11] J. Hiriart-Urruty, C. Lemaréchal, *Convex Analysis and Minimization Algorithm*, Springer-Verlag, Berlin, Germany, (1993).

[12] X. Ke, J. Han, A class of nonmonotone trust region algorithms for unconstrained optimization, *Science in China*, Series A, **41** (1998), 927-932.

[13] J. Mo, C. Liu, S. Yan, A nonmonotone trust region method based on non-increasing technique of weighted average of the successive function values, *J. Comput. Appl. Math.*, Doi:11.1016/j.cam.2006.10.070 (2006).

[14] J. Nocedal, Y. Yuan, Combining trust region and line search techniques, In: *Advances in Nonlinear Programming* (Ed. Y. Yuan), Kluwer Academic Publishers, Dordrecht (1998), 153-175.

[15] L. Qi, Convergence analysis of some algorithms for solving nonsmooth equations, *Math. Oper. Res.*, **18** (1993), 227-244.

[16] L. Qi, X. Chen, A preconditioning proximal Newton for nondifferentiable convex optimization, *Math. Prog.*, **76** (1997), 411-429.

[17] L. Qi, J. Sun, A nonsmooth version of Newton's method, *Math. Prog.*, **58** (1993), 353-368.

[18] A. I. Rauf and M. Fukushima, Globally convergent BFGS method for non-smooth convex optimization, *J. Optim. Theory Appl.*, **104** (2000), 539-558.

[19] N. Sagara, M. Fukushima, A trust region method for nonsmooth convex optimization, *J. Indust. Manage. Optim.*, **1** (2005), 171-180.

[20] W. Sun, Nonmonotone trust region method for solving optimization problems, *Appl. Math. Comput.*, **156** (2004), 159-174.

[21] Z. Wei, G. Li, L. Qi, New quasi-Newton methods for unconstrained optimization problems, *Appl. Math. Comput.*, **175** (2006), 1156-1188.

[22] H. Zhang, W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.*, **14** (2004), 1043-1056.

[23] J. Zhang, X. Zhang, A nonmonotone adaptive trust region method and its convergence, *Comput. Math. Appl.*, **45** (2003), 1469-1477.