AN ADAPTIVE TRUST REGION ALGORITHM FOR LARGE-RESIDUAL NONSMOOTH LEAST SQUARES PROBLEMS

Zhou Sheng, Gonglin Yuan*, Zengru Cui, Xiabin Duan and Xiaoliang Wang

College of Mathematics and Information Science
Guangxi University
Nanning, Guangxi 530004, China

(Communicated by Duan Li)

Abstract. In this paper, an adaptive trust region algorithm in which the trust region radius converges to zero is presented for solving large-residual nonsmooth least squares problems. This algorithm uses the smoothing technique of the approximation function, and it combines an adaptive trust region radius. Moreover, this algorithm differs from the existing methods for solving nonsmooth equations through use of the approximation function of second-order information, which improves the convergence rate for large-residual nonsmooth least squares problems. Under some suitable conditions, the global and local superlinear convergences of the proposed method are proven. The preliminary numerical results indicate that the proposed algorithm is effective and suitable for solving large-residual nonsmooth least squares problems.

1. Introduction. Consider the following nonsmooth least squares problem:

\[
\min_{x \in \mathbb{R}^n} F(x) = \frac{1}{2} f(x)^T f(x),
\]

where \( f : \mathbb{R}^n \to \mathbb{R}^m \) is a nonsmooth function. This problem (1.1) and other type nonsmooth problems \([10, 23, 24, 27]\) have recently appeared in many applications in the medical, image restoration, optimal control, functional approximation and curve fitting fields.

Initially, we consider \( n = m \), which is equivalent to the following nonsmooth nonlinear equation problem

\[
f(x) = 0.
\]

For solving the above nonsmooth equation problem, Qi \([15]\) considers using the smooth plus nonsmooth (SPN) decomposition, namely, \( f \) is decomposed into \( f = p + q \), where \( p : \mathbb{R}^n \to \mathbb{R}^n \) is smooth and \( q : \mathbb{R}^n \to \mathbb{R}^m \) is locally Lipschitzian and relatively small, and two examples of SPN decomposition are given; further examples are given in \([2, 4, 3]\). Qi \([15]\) presents a trust region method for solving
the nonsmooth equation (1.2), and the trail step $d_k$ is obtained by solving the following subproblem:

$$\min \|f(x_k) + \nabla p(x_k)d\|_2$$

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

This method is a very nice numerical method for solving the nonsmooth equation (1.2). However, Sun, Sampaio and Yuan [20] report that (i) this method is used to solve the nonsmooth least squares problem (1.1), in which there is only a slow convergence rate because we only used the first-order information of $p(x)$, and that (ii) we cannot neglect the second-order information of $F(x)$ for solving the larger residual nonsmooth least squares problem (1.1). Hence, they consider using the SPN decomposition and solving the subproblem

$$\min \psi_k(d) = \frac{1}{2}\|f(x_k) + \nabla p(x_k)d\|^2 + \frac{1}{2}d^T V_k d$$

s.t. $\|d\| \leq \tilde{\Delta}_k,$

where $V_k$ is symmetric and carries the second-order information of $F(x)$ and $\tilde{\Delta}_k > 0$.

The ratio between the actual reduction and predicted reduction is defined as

$$\tilde{r}_k = \frac{F(x_k) - F(x_k + d_k)}{\psi(0) - \psi(d_k)}$$

which plays a important role in determining whether to accept the trail step and how to adjust the trust region radius. The next iterate $x_{k+1}$ is computed using the following formula

$$x_{k+1} = \begin{cases} 
  x_k + d_k & \text{if } \tilde{r}_k \geq \mu, \\
  x_k & \text{otherwise},
\end{cases}$$

where $0 < \mu < 1$ is a positive constant. The trust region radius for the next iteration is chosen as

$$\tilde{\Delta}_{k+1} = \begin{cases} 
  (0, \gamma_1 \tilde{\Delta}_k] & \text{if } \tilde{r}_k \leq \mu, \\
  (\gamma_1 \tilde{\Delta}_k, \gamma_2 \tilde{\Delta}_k] & \text{if } \mu < \tilde{r}_k < \mu_1, \\
  [\tilde{\Delta}_k, \min\{\gamma_3 \tilde{\Delta}_k, \overline{\Delta}\}] & \text{otherwise},
\end{cases}$$

where $\overline{\Delta} > 0$, $0 < \mu < \mu_1 < 1$, and $0 < \gamma_1 < \gamma_2 < \gamma_3$.

As is known, one of the most effective methods for problems (1.1) is the trust region method. The trust region method plays an important role in the area of nonlinear optimization, and it has been proven to be a very efficient method (see [17, 21, 22] etc.). Levenberg [9] and Marquardt [12] first applied this method for nonlinear least squares problems, and Powell [14] established the convergence result of this method for unconstrained problems.

Based on the concept in [7], note that the ratio $\{\tilde{r}_k\}$ converges to 1 when the sequence generated by the traditional trust region algorithm converges to the problem (1.1). By the updating formula (1.6), it is easy to see that $\tilde{\Delta}_k$ will be larger than a positive constant for all larger $k$. Because $\{\|x_k - x^*\|\}$ converges to zero, the trust region will ultimately not work. Many algorithms with an adaptive trust region radius have been proposed for solving nonlinear equations problems. For example, Zhang and Wang [26] presented a new trust region method with the trust region radius $\overline{\Delta}_k = e^p||f(x_k)||^\delta$ for solving nonlinear equations, where $0 < c < 1$, $p$ is a nonnegative integer, and $\frac{1}{2} < \delta < 1$. Based on the concept in [26], Yuan, Wei and
Lu [25] presented a modified BFGS trust region algorithm, and \( \Delta_k = c^p \| f(x_k) \| \), where \( c \in (0, 1) \), \( p \) is also a nonnegative integer, and the quadratic convergence is obtained under suitable conditions. Fan and Pan [6] also presented a new trust region algorithm, and the trust region radius was chosen as \( \Delta_k = \mu_k \| f(x_k) \|^{\delta} \), where \( \mu_k \) is updated according to \( \tilde{r}_k \), \( \delta \in (0, 1) \); under the local error bound condition, we also obtain the quadratic convergence. For more examples, see [8, 18], among others.

The purpose of this paper is to present an efficient trust region algorithm to solve (1.1). By using the approximation function [3] and adaptive trust region radius [7], the proposed method possesses some good properties.

The remainder of this paper is organized as follows. In the next section, we briefly review some basic results in nonsmooth analysis and state an approximation function. In Section 3, we present an adaptive algorithm for solving problem (1.1), and we prove the global convergence of the proposed method. In Section 4, we prove the superlinear convergence of the algorithm under suitable conditions. Preliminary numerical results are reported in Section 5. We conclude our paper in Section 6.

Throughout this paper, unless otherwise specified, \( \| \cdot \| \) denotes the Euclidean norm of vectors or matrices.

2. Preliminary. In this section, we state an approximation function. First, we provide the definition of an approximate function [3] as follows.

**Definition 2.1.** [3] We call a function \( P : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m \) a point-based smooth approximation function of \( f \) if

(i) for any \( x \in \mathbb{R}^n \), \( P_x(x) \equiv f(x) \);

(ii) for each fixed \( y \in \mathbb{R}^n \), there is a number \( \theta_y > 0 \) such that \( P_y(\cdot) \equiv P(\cdot, y) \) is continuously differentiable in \( S(y, \theta_y) \);

(iii) for each fixed \( x \in \mathbb{R}^n \), there is a number \( \theta_x > 0 \) such that \( P(x) \equiv P(x, \cdot) \) is continuous in \( S(x, \theta_x) \);

(iv) for each fixed \( x \in \mathbb{R}^n \), there is a number \( \tilde{\theta}_x > 0 \) such that \( \frac{\partial}{\partial x} P(x, \cdot) \equiv P'(x) \) is continuous in \( S(x, \tilde{\theta}_x) \).

Where \( S(x, \theta) \) denotes an open ball in \( \mathbb{R}^n \) with center \( x \) and radius \( \theta \).

Some important properties of \( P \) are given in the following [3]:

**Proposition 2.1.** Let \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \) be continuously differentiable. Define \( P \) as:

\[
P_y(x) \equiv f(x), \quad \text{forall } x, y \in \mathbb{R}^n.
\]

Then, \( P \) is a point-based smooth approximation function of \( f \).

**Proposition 2.2.** Let \( f = \phi + \psi \), where \( \phi \) is continuously differentiable and \( \psi \) is continuous. Then, \( P \), which is defined by

\[
P_y(x) = \phi(x) + \psi(y)
\]

is a point-based smooth approximation function of \( f \).

**Remark 2.1.** The point-based smooth approximation function is a generalization of the concept of the splitting function.

**Remark 2.2.** In this paper, based on Definition 1, let \( P_k(x) := P_{x_k}(x) \).
Based on the above propositions, we can now decompose \( f \) as follows:
\[
    f(x) = P_x(x). \tag{2.1}
\]
where \( P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m \) is a point-based smooth approximation function. Thus, we can use \( P_x(x) \) to define the approximation of \( F(x) \) as follows
\[
    F_p(x) = \frac{1}{2}P_x(x)^TP_x(x). \tag{2.2}
\]
Based on the above discussion of \( P_x(x) \), it possesses the following remarkable feature.

**Proposition 2.3.** Let a function \( P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m \) be a point-based smooth approximation function of \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \). Then, \( F_p(x) = \frac{1}{2}P_x(x)^TP_x(x) \) is a point-based smooth approximation function of \( F : \mathbb{R}^n \rightarrow \mathbb{R}^m \).

The gradient is \( J(x_k)^TP_k(x_k) = \nabla P_k(x_k)^TP_k(x_k) \), where \( J(x_k) = \nabla P_k(x_k) \) is the Jacobian of \( P_k(x_k) \), the Hessian matrix \( \nabla^2 F_p(x_k) = J(x_k)^TJ(x_k) + \sum_{i=1}^{m} P_{ki}(x_k) \nabla^2 P_{ki}(x_k) \).

3. **New algorithm and global convergence.** In this section, based on the concept in [7], we present an adaptive trust region model for solving (1.1), which is motivated by the approximation function, the general trust region method and the secant equation. Now, we describe the trust region method. In each iteration, a trial step \( d_k \) is generated by solving an adaptive trust region subproblem, in which the value of the Jacobian of \( J(x) \) at \( x_k \) is used:
\[
    \begin{align*}
        & \min \quad q_k(d) = \frac{1}{2}\|P_k(x_k) + J(x_k)d\|^2 + \frac{1}{2}d^TB_kd \\
        & \text{s.t.} \quad \|d\| \leq \Delta_k := \lambda_k\|J(x_k)^TP_k(x_k)\|.
    \end{align*} \tag{3.1}
\]
where \( \Delta_k \) is the trust region radius and \( B_k \) is generated by the BFGS update formula
\[
    B_{k+1} = B_k - \frac{B_kd_kd_k^T}{d_k^TB_kd_k} + \frac{y_ky_k^T}{y_k^Td_k}, \tag{3.2}
\]
and \( B_{k+1} \) satisfies the secant equation \( B_{k+1}d_k = y_k \), namely, we have
\[
    (J(x_{k+1})^TJ(x_{k+1}) + B_{k+1})d_k = J(x_{k+1})^TP_{k+1}(x_{k+1}) - J(x_k)^TP_k(x_k), \tag{3.3}
\]
where \( d_k = x_{k+1} - x_k \) and \( y_k = J(x_{k+1})^TP_{k+1}(x_{k+1}) - J(x_k)^TP_k(x_k) - J(x_{k+1})^TP_{k+1}(x_{k+1}) \).

Let \( d_k \) be the optimal solution of (3.1). The actual reduction is defined by
\[
    Ared_k := F_p(x_k) - F_p(x_k + d_k), \tag{3.4}
\]
and we define the predicted reduction as
\[
    Pred_k := q_k(0) - q_k(d_k), \tag{3.5}
\]
Then, we define \( r_k \) as the ratio between \( Ared_k \) and \( Pred_k \)
\[
    r_k := \frac{Ared_k}{Pred_k} \tag{3.6}
\]
Now, we list the steps of the modified trust region algorithm as follows.

**Algorithm 1. Adaptive Trust Region Algorithm**

**Step 0.** Given \( x_0 \in \mathbb{R}^n \) and \( B_0 \in \mathbb{R}^{n \times n} \), let \( 0 \leq \alpha_0 < \alpha_1 < 1, \alpha_2 < 1 \leq \alpha_3, \lambda_0 > 0, \epsilon > 0, \) and \( \Delta_0 := \lambda_0\|J(x_0)^TP_0(x_0)\| \). Set \( k := 0 \).
Step 1. If \( \| P_k(x_k) \| \leq c \), then stop. Otherwise, go to Step 2.

Step 2. Solve the trust region subproblem (3.1) to obtain \( d_k \).

Step 3. Calculate \( r_k = \frac{\Delta_k}{\| P(x_k) \|} \). If \( r_k \leq \alpha_0 \), then we let \( x_{k+1} := x_k \); otherwise, \( x_{k+1} := x_k + d_k \).

Step 4. Set
\[
\lambda_{k+1} = \begin{cases} 
\alpha_2 \lambda_k & \text{if } r_k < \alpha_1, \\
\alpha_3 \lambda_k & \text{if } r_k \geq \alpha_1 \text{ and } \| d_k \| > \frac{1}{2} \Delta_k,
\end{cases}
\]
(3.7)

Step 5. Compute \( \Delta_{k+1} := \lambda_{k+1} \| x_{k+1} \| \), update \( B_k \) by the BFGS formula, \( k := k + 1 \), and go back to Step 1.

Remark 3.1. Our modified algorithm can be considered as an extension of paper [7] from smooth optimization to nonsmooth larger residuals least squares optimization.

To establish the global convergence of Algorithm 1, we will make the following common assumption.

Assumption A. (i) The level set \( \Omega = \{ x \in \mathbb{R}^n | F_P(x) \leq F_P(x_0) \} \) is bounded.

(ii) The \( J(x) \) is bounded, namely, there exists a positive constant \( M_1 > 0 \) such that
\[ \| J(x) \| \leq M_1 \quad \forall x \in \mathbb{R}^n. \]

(iii) There exists a positive constant \( M_2 > 0 \) such that \( \| B_k \| \leq M_2 \) holds for all \( k \).

Similar to the famous result of Powell [14], we provide the following lemma.

Lemma 3.1. If \( d_k \) is the solution of (3.1), then
\[
\text{Pre} \ d_k \geq \frac{1}{2} \| J(x_k) \| P_k(x_k) \|^2 \min \left\{ \lambda_k, \frac{1}{\| J(x_k) \| P_k(x_k) + B_k \|} \right \}. \quad (3.8)
\]

Proof. Because \( d_k \) is the solution of (3.1), for any \( \alpha \in [0, 1] \), we obtain
\[
\text{Pre} \ d_k \geq q_k(0) - q_k \left( -\alpha \lambda_k \| J(x_k) \| P_k(x_k) \| J(x_k) \| P_k(x_k) \| J(x_k) \| P_k(x_k) \| \right) \\
= \alpha \lambda_k [J(x_k) \| P_k(x_k) \| [J(x_k) \| P_k(x_k) \|] \\
- \frac{1}{2} \alpha^2 \lambda_k^2 [J(x_k) \| P_k(x_k) \| [J(x_k) \| J(x_k) \| + B_k \| J(x_k) \| P_k(x_k) \|] \\
\geq \alpha \lambda_k \| J(x_k) \| P_k(x_k) \|^2 - \frac{1}{2} \alpha^2 \lambda_k^2 \| J(x_k) \| P_k(x_k) \|^2 \| J(x_k) \| J(x_k) \| + B_k \|.
\]

Therefore, we have
\[
\text{Pre} \ d_k \\
\geq \max_{0 \leq \alpha \leq 1} \left\{ \alpha \lambda_k \| J(x_k) \| P_k(x_k) \|^2 - \frac{1}{2} \alpha^2 \lambda_k^2 \| J(x_k) \| P_k(x_k) \|^2 \| J(x_k) \| J(x_k) \| + B_k \| \right \} \\
\geq \frac{1}{2} \| J(x_k) \| P_k(x_k) \|^2 \min \left\{ \lambda_k, \frac{1}{\| J(x_k) \| J(x_k) \| + B_k \|} \right \}.
\]

The proof is complete.

Theorem 3.1. Let Assumption A hold, and \( \{ x_k \} \) is a sequence generated from Algorithm 1. Then, Algorithm 1 either stops finitely or satisfies
\[
\lim_{k \to \infty} \| J(x_k) \| P_k(x_k) \| = 0. \quad (3.9)
\]
Proof. By contradiction, suppose that there is a positive constant $\epsilon_0$ such that
\[
\|J(x_k)^T P_k(x_k)\| \geq \epsilon_0. \tag{3.10}
\]
From Algorithm 1, we can define
\[
\Gamma = \{ k | r_k \geq \alpha_1 \}.
\]
Then, from Assumption A, Lemma 3.1 and (3.10), we obtain
\[
+\infty > \sum_{k=1}^{\infty} [F_P(x_k) - F_P(x_k + d_k)] \\
\geq \sum_{k \in \Gamma} \alpha_1 [g_k(0) - q_k(d_k)] \\
\geq \frac{1}{2} \sum_{k \in \Gamma} \alpha_1 \|J(x_k)^T P_k(x_k)\|^2 \min \left\{ \lambda_k, \frac{1}{\|J(x_k)^T J(x_k) + B_k\|} \right\} \\
\geq \frac{1}{2} \sum_{k \in \Gamma} \alpha_1 \epsilon_0 \min \left\{ \lambda_k, \frac{1}{M_1^2 + M_2} \right\}. \tag{3.11}
\]
If $\Gamma$ is finite, from (3.7), it is clear that $\lambda_{k+1} = \alpha_2 \lambda_k$ for all sufficiently large $k$. Because $\alpha_2 < 1$, then
\[
\lim_{k \to \infty} \lambda_k = 0. \tag{3.12}
\]
If $\Gamma$ is infinite, by the definition of $\Delta_k$ and (3.11), we have
\[
\sum_{k \in \Gamma} \min \left\{ \lambda_k, \frac{1}{M_1^2 + M_2} \right\} < +\infty.
\]
Therefore, we must have
\[
\lim_{k \to \infty} \lambda_k = 0. \tag{3.13}
\]
However, from Assumption A(ii), (iii), Lemma 3.1, (3.10) and (3.13), we have
\[
|r_k - 1| = \left| \frac{\text{Are} \, d_k}{\text{Pre} \, d_k} - 1 \right| = \left| \frac{\text{Are} \, d_k - \text{Pre} \, d_k}{\text{Pre} \, d_k} \right| \\
\leq \left| \frac{\|P_k(x_k) + J(x_k)d_k\|O(||d_k||^2) + O(||d_k||^4)}{\|P_k(x_k) + J(x_k)d_k\|O(||d_k||^2) + O(||d_k||^4)} - \frac{1}{2} d_k^T B_k d_k \right| \\
\leq \left| \frac{\|P_k(x_k) + J(x_k)d_k\|O(||d_k||^2) + O(||d_k||^4) + \frac{1}{2} d_k^T B_k d_k}{\|P_k(x_k) + J(x_k)d_k\|O(||d_k||^2) + O(||d_k||^4)} \right| \\
\leq \frac{M_2 ||d_k||^2 + 2||P_k(x_k) + J(x_k)d_k||O(||d_k||^2) + O(||d_k||^4)}{\epsilon_0^2 \min \{ \lambda_k, \frac{1}{M_1^2 + M_2} \}} \\
\leq \frac{O(||d_k||^2)}{\lambda_k} \\
\leq \frac{O(\Delta_k^2)}{\lambda_k} \to 0.
\]
Then, we obtain
\[
r_k \to 1.
\]
Therefore, by above equation, it follows that there exists a constant $\overline{\lambda} > 0$ such that $\lambda_k > \overline{\lambda}$ holds for all sufficiently large $k$, namely, $\lim_{k \to \infty} \lambda_k > \overline{\lambda} > 0$, which contradicting the equation (3.13), thus, equation (3.10) is not true. Hence, we have that (3.9) holds, and the proof is complete. \qed
4. Superlinear convergence. In this section, we shall prove the superlinear convergence under suitable conditions. From [7], we know that our new trust region radius converges to zero. Hence, to prove that the algorithm attains superlinear convergence under suitable conditions, we need to show that the constraint is inactive when \( d_k \) is the solution of (3.1), which means that the trail step takes the quasi-Newton step. Namely, the sequence \( \{\lambda_k\} \) is bounded. Now, let us consider the following lemma.

**Lemma 4.1.** If \( d_k \) is the solution of (3.1), assume that \( \{x_k\} \) is a sequence generated when Algorithm 1 converges to \( x^* \), \( \nabla^2 F_P(x) \) is continuous in a neighborhood of \( x^* \), and \( \nabla^2 F_P(x^*) \) is positive definite, where \( \nabla^2 F_P(x) = J(x)^T J(x) + \sum_{i=1}^{m} P_{ki}(x) \nabla^2 P_{ki}(x) \). Then, the sequence \( \{\lambda_k\} \) is bounded.

**Proof.** By contradiction, suppose that there is a subsequence \( \{\lambda_l\} \) such that \( \lambda_{\lambda_l} \rightarrow +\infty \). Because the inequality

\[
||d_k|| > \frac{1}{2} \lambda_k ||J(x_k)^T P_k(x_k)||. \tag{4.1}
\]

holds, by the positive definiteness of \( \nabla^2 F_P(x^*) \), it follows that for all large \( k \), there exist \( L > l > 0 \) such that

\[
L||x_k - x^*|| \geq ||J(x_k)^T P_k(x_k)|| \geq l||x_k - x^*||, \tag{4.2}
\]

\[
L||x_k - x^*||^2 \geq F_P(x_k) - F_P(x^*) \geq l||x_k - x^*||^2. \tag{4.3}
\]

Therefore, from (4.1), (4.2) and (4.3), we obtain

\[
L||x_k - x^*||^2 \geq F_P(x_k + d_k) - F_P(x^*)
\]

\[
\geq l||x_k + d_k - x^*||^2 
\]

\[
\geq l(||x_k + d_k - x^*||^2 - 2||x_k + d_k - x^*|| \cdot ||d_k|| + ||d_k||^2)
\]

\[
= l(||d_k|| - ||x_k - x^*||^2)^2 
\]

\[
\geq l\left(\frac{1}{2} \lambda_l^2 ||x_k - x^*|| - ||x_k - x^*||^2\right)^2 
\]

\[
= l\left(\frac{1}{2} \lambda_l l - 1\right)^2 ||x_k - x^*||^2,
\]

which is a contradiction to \( \lambda_{\lambda_l} \rightarrow +\infty \). Hence, the sequence \( \{\lambda_k\} \) is bounded, and the proof is complete.

**Theorem 4.1.** Under the conditions of Lemma 4.1, and if the condition

\[
\lim_{k \to \infty} \frac{\left[ \sum_{i=1}^{m} P_{ki}(x) \nabla^2 P_{ki}(x) - B_k \right] d_k}{||d_k||} = 0 \tag{4.4}
\]

holds, then the sequence \( \{x_k\} \) converges to \( x^* \) Q-superlinearly.

**Proof.** By the definiteness of \( \nabla^2 F_P(x^*) \) and (4.4), it follows that for all sufficiently large \( k \), there exists a positive constant \( \eta \) such that

\[
d_k^T B_k d_k \geq l||d_k||^2. \tag{4.5}
\]

Note that \( d_k \) is a solution for solving (3.1); therefore, there exists \( \nu_k \geq 0 \) such that \( (J(x_k)^T J(x_k) + B_k + \nu_k I)d_k = -J(x_k)^T P_k(x_k) \). Thus, we have

\[
Pre d_k = \frac{1}{2} d_k^T (J(x_k)^T J(x_k) + B_k) d_k + \nu_k ||d_k||^2. \tag{4.6}
\]
By the continuity of \( \nabla^2 F_P(x) \) and (4.4), we have
\[
A_{red} k = -J(x_k)^T P_k(x_k) d_k - \frac{1}{2} d_k^T \nabla^2 F_P(x^*) d_k + o(\|d_k\|^2)
\]
\[
= Pre d_k + \frac{1}{2} d_k^T [B_k - \sum_{i=1}^m P_{ki}(x) \nabla^2 P_{ki}(x)] d_k + o(\|d_k\|^2)
\]
\[
= Pre d_k + o(\|d_k\|^2).
\] (4.7)

Thus, we obtain
\[
|\nu_k - 1| = \left| \frac{A_{red} k - Pre d_k}{Pre d_k} \right| \\
\leq \frac{o(\|d_k\|^2)}{(\nu_k + \frac{1}{2})\|d_k\|^2} \to 0.
\]
Namely, \( \lim_{k \to \infty} \nu_k = 1 \).

Note that the sequence \( \{\lambda_k\} \) is bounded from Lemma 4.1, and we have \( \|d_k\| \leq \frac{1}{2} \Delta_k \) for all large \( k \). Therefore, the sequence \( \{x_k\} \) generated by Algorithm 1 is a quasi-Newton sequence that is superlinearly convergent to \( x^* \) under the assumed conditions. For details regarding the superlinear convergence, see [13]. This completes the proof. \( \square \)

5. Numerical results. In this section, we report numerical results of our adaptive trust region algorithm (ATR, in short) for solving large-residual nonsmooth least squares problems. We compared our algorithm with the traditional trust region algorithm (TTR, in short) of paper [5]. Here, we obtain the trial step \( d_k \) in the trust region subproblem (3.1) by the cgtrust method [19] for both codes. Both algorithms were implemented in Matlab R2010b, and the numerical experiments were performed on a PC equipped with CPU Intel Pentium(R) Dual-Core CPU at 3.20 GHz, 2.00 GB of RAM, and the Windows 7 operating system. The parameters were chosen as follows: \( \alpha_0 = 0.00001, \alpha_1 = 0.1, \alpha_2 = 0.3, \alpha_3 = 1.5 \) and \( \lambda_0 = 1 \). We stopped the algorithm when the condition \( \|J(x_k)^T P_k(x_k)\| \leq 10^{-5} \) was satisfied. According to Conn et.al [5], we choose \( \Delta_0 = 1 \), and update \( \Delta_k \) by the following formula:
\[
\Delta_{k+1} = \begin{cases} 
0.25\Delta_k & \text{if } r_k < 0.1, \\
\Delta_k & \text{if } 0.1 \leq r_k \leq 0.9, \\
3\Delta_k & \text{if } r_k > 0.9.
\end{cases}
\] (5.1)

To explain the ATR is efficient for finding global minimizers of the problems, we consider following examples.

Example 5.1. (this example is a modified problem in [11]). Let \( f(x) = (f_1(x), f_2(x), f_3(x))^T \) and
\[
\begin{align*}
f_1(x) &= x_1^2 + x_2^2 + x_1x_2; \\
f_2(x) &= \sin x_1 + \max\{0, x_2\}; \\
f_3(x) &= \cos x_2 + \max\{0, x_1\}.
\end{align*}
\] (5.2)
According the Proposition 3 of Chen and Qi [3], we have following equations for the second equation and third equation of (5.2).

\[
P_2(x) = \begin{cases} 
  f_2(x) & \text{if } |x_2| \geq w; \\
  \sin x_1 + \frac{x_2^2}{2} + \frac{x_2^2 + w^2}{4w} & \text{if } |x_2| < w;
\end{cases}
\]  
(5.3)

\[
P_3(x) = \begin{cases} 
  f_3(x) & \text{if } |x_1| \geq w; \\
  \cos x_2 + \frac{x_1^2}{2} + \frac{x_1^2 + w^2}{4w} & \text{if } |x_1| < w,
\end{cases}
\]  
(5.4)

for the above equations (5.3) and (5.4), we choose \( w = 0.625 \) and the initial point \( x_0 = (3, 1)^T \).

**Example 5.2.** (this example is a modified Trigonometric Problem in [1]). Let

\[
\begin{align*}
  f_1(x) &= -d_1 + (- e_1 + \sum_{j=1}^{2} (a_{1j} \sin x_j + b_{1j} \cos x_j))^2; \\
  f_2(x) &= -d_2 + (- e_2 + \sum_{j=1}^{2} (a_{2j} \sin x_j + b_{2j} \cos x_j))^2 + \min\{0, x_1\}; \\
  f_3(x) &= -d_3 + (- e_3 + \sum_{j=1}^{2} (a_{3j} \sin x_j + b_{3j} \cos x_j))^2 + \min\{0, x_2\},
\end{align*}
\]

where \( d_1 = 1, d_2 = 2, d_3 = 3, a_{ij}, b_{ij} (i = 1, 2, 3; j = 1, 2) \) are random integers in \([-10, 10]\) and \( e_1, e_2, e_3 \) are random numbers in \([0, 1]\).

Similar to Example 5.1, we have

\[
P_2(x) = \begin{cases} 
  f_2(x) & \text{if } |x_2| \geq w; \\
  -d_2 + (- e_2 + \sum_{j=1}^{2} (a_{2j} \sin x_j + b_{2j} \cos x_j))^2 + \frac{x_1^2}{2} - \frac{x_2^2 + w^2}{4w} & \text{if } |x_2| < w;
\end{cases}
\]  
(5.6)

\[
P_3(x) = \begin{cases} 
  f_3(x) & \text{if } |x_1| \geq w; \\
  -d_3 + (- e_3 + \sum_{j=1}^{2} (a_{3j} \sin x_j + b_{3j} \cos x_j))^2 + \frac{x_2^2}{2} + \frac{x_1^2 + w^2}{4w} & \text{if } |x_1| < w,
\end{cases}
\]  
(5.7)

we also choose \( w = 0.625 \) and the initial point \( x_0 = (-1, -1)^T \).

The computational results are shown in Figures 1 and 2 with ATR algorithm and TTR algorithm which indicate that our ATR algorithm is effective and suitable to solve large-residual nonsmooth least squares problems.

6. **Conclusions.** In this paper, we proposed an adaptive trust region algorithm for solving large-residual nonsmooth least squares problems. This algorithm uses the approximation functions of Chen and Qi [3], and it combines a new trust region strategy with the trust region radius converging to zero. In contrast to the nonsmooth equations, the existing methods only use the first-order information of the smooth approximation function. For large-residual nonsmooth least squares
problems, we must consider the second-order information of the smooth approximation function because the second-order information \( \sum_{i=1}^{n} P_{ki}(x)\nabla^2 P_{ki}(x) \) does not converge to zero. Because the trust region radius of the traditional trust region algorithm will be larger than a positive constant, the trust region ultimately does not work. We extended the approach of Fan and Yuan [7], and we applied the algorithm to large-residual nonsmooth least squares problems. Finally, global and local superlinear convergence are established under some mild conditions, and the preliminary numerical results of our algorithm indicate the promising behaviour to solve large-residual nonsmooth least squares problems.

**Figure 1.** The value of \( \| F(x_k) \| \) with iteration \( k \) for Example 5.1

**Figure 2.** The value of \( \| F(x_k) \| \) with iteration \( k \) for Example 5.2
Acknowledgments. We would like to thank the editor and two anonymous referees for their valuable comments and suggestions which greatly improve this paper. This work is supported by the National Natural Science Foundation of China (11261006 and 11661009), the Guangxi Science Fund for Distinguished Young Scholars (No. 2015GXNSFGA139001) and the Basic Ability Promotion Project of Guangxi Young and Middle-aged Teachers (No. 2017KY0019).

REFERENCES

[1] M. Al-Baali and R. Fletcher, Variational methods for non-linear least-squares, J. Oper. Res. Soc., 36 (1985), 405–421.
[2] X. Chen, On the convergence of Broyden-like methods for nonlinear equations with nondifferentiable terms, Ann. Institut. Statist. Math., 42 (1990), 387–401.
[3] X. Chen and L. Qi, A parameterized Newton method and a quasi-Newton method for non-smooth equations, Comput. Optim. Appl., 5 (1994), 157–179.
[4] X. Chen and T. Yamamoto, On the convergence of some quasi-Newton methods for nonlinear equations with nondifferentiable operators, Computing, 49 (1992), 87–94.
[5] A. R. Conn, N. I. M. Gould and Ph. L. Toint, Trust-Region Methods, Society for Industrial and Applied Mathematics SIAM, Philadelphia, PA, 2000.
[6] J. Fan and J. Pan, An improve trust region algorithm for nonlinear equations, Comput. Optim. Appl., 48 (2011), 59–70.
[7] J. Fan and Y. Yuan, A new trust region algorithm with trust region radius converging to zero, in Proceedings of the 5th International Conference on Optimization: Techniques and Applications (December 2001, Hongkong) (ed. D. Li), 2001, 786–794.
[8] L. Hei, A self-adaptive trust region algorithm, J. Comput. Math., 21 (2003), 229–236.
[9] K. Levenberg, A method for the solution of certain nonlinear problem in least squares, Quarterly Journal of Mechanics and Applied Mathematics, 2 (1944), 164–168.
[10] S. Liu, Z. Wang and C. Liu, On convergence analysis of dual proximal-gradient methods with approximate gradient for a class of nonsmooth convex minimization problems, J. Ind. Manag. Optim., 12 (2016), 389–402.
[11] K. Madsen, An algorithm for minimax solution of overdetermined systems of nonlinear equations, Rep. TP 559, AERE Harwell, England, 1973.
[12] B. Martinet, Régularisation d’inéquations variationelles par approximations successives, Rev. Fr. Inform. Rech. Oper., 4 (1970), 154–158.
[13] J. Moré, Recent developments in algorithms and software for trust region methods, in Mathematical Programming: The State of Art (eds. A. Bachem, M. Grötschel and B. Korte), Springer, Berlin, 1983, 258–287.
[14] M. J. D. Powell, Convergence properties of a class of minimization algorithms, in Nonlinear Programming (Q. L. Mangasarian, R. R. Meyer and S. M. Robinson), Vol. 2, Academic Press, New York, 1974, 1–27.
[15] L. Qi, Trust region algorithms for solving nonsmooth equations, SIAM J. Optimization, 5 (1995), 219–230.
[16] L. Qi, Z. Wei and G. Yuan, An active-set projected trust region algorithm with limited memory BFGS technique for box constrained nonsmooth equations, Optimization, 62 (2013), 857–878.
[17] Z. Sheng, A. Ouyang and L. B. Liu, et al., A novel parameter estimation method for Muskingum model using new Newton-type trust region algorithm, Math. Probl. Eng., (2014), Art. ID 634852, 7 pp.
[18] Z. Shi and J. Guo, A new trust region method for unconstrained optimization, J. Comput. and Appl. Math., 213 (2008), 509–520.
[19] T. Steihaug, The conjugate gradient method and trust regions in large scale optimization, SIAM J. Numer Anal., 20 (1983), 626–637.
[20] W. Sun, R. J. B. Sampaio and J. Yuan, Quasi-Newton trust region algorithm for non-smooth least squares problems, Appl. Math. Comput., 105 (1999), 183–194.
[21] G. Yuan, S. Lu and Z. Wei, A new trust-region method with line search for solving symmetric nonlinear equations, Intern. J. Comput. Math., 88 (2011), 2109–2123.
[22] G. Yuan, X. Lu and Z. Wei, BFGS trust-region method for symmetric nonlinear equations, J. Comput. and Appl. Math., 230 (2009), 44–58.
[23] G. Yuan, Z. Meng and Y. Li, A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations, *J. Optim. Theo. Appl.*, 168 (2016), 129–152.

[24] G. Yuan, Z. Wei and G. Li, A modified Polak-Ribière-Polyak conjugate gradient algorithm for nonsmooth convex programs, *J. Compu. and Appl. Math.*, 255 (2014), 86–96.

[25] G. Yuan, Z. Wei and X. Lu, A BFGS trust-region method for nonlinear equations, *Computing*, 92 (2011), 317–333.

[26] J. Zhang and Y. Wang, A new trust region method for nonlinear equations, *Math. Methods Oper. Res.*, 58 (2003), 283–298.

[27] S. Zhou, Y. Li and L. Kong, et al., A smoothing iterative method for quantile regression with nonconvex $l_p$ penalty, *J. Ind. Manag. Optim.*, 12 (2016), 93–112.

Received July 2015; revised June 2016.

E-mail address: szhou03@live.com
E-mail address: glyuan@gxu.edu.cn
E-mail address: cuizengru@163.com
E-mail address: xiabinduan@126.com
E-mail address: xliangwang@126.com