

Research Article

A Derivative-Free Trust Region Algorithm with Nonmonotone Filter Technique for Bound Constrained Optimization

Jing Gao ¹, Jian Cao ², and Yueting Yang¹

¹The School of Mathematics and Statistics, Beihua University, Jilin 132001, China

²The School of Information Technology and Media, Beihua University, Jilin 132001, China

Correspondence should be addressed to Jian Cao; evercall@163.com

Received 9 January 2018; Revised 2 July 2018; Accepted 19 August 2018; Published 23 October 2018

Academic Editor: Gen Q. Xu

Copyright © 2018 Jing Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a derivative-free trust region algorithm with a nonmonotone filter technique for bound constrained optimization. The derivative-free strategy is applied for special minimization functions in which derivatives are not all available. A nonmonotone filter technique ensures not only the trust region feature but also the global convergence under reasonable assumptions. Numerical experiments demonstrate that the new algorithm is effective for bound constrained optimization. Locally, optimal parameters with respect to overall computational time on a set of test problems are identified. The performance of the best choice of parameter values obtained by the algorithm we presented which differs from traditionally used values indicates that the algorithm proposed in this paper has a certain advantage for the nondifferentiable optimization problems.

1. Introduction

Many of the objective functions in mathematical optimization that occur in engineering are obtained from a mass of numerical experiments and have special characteristics such as being nonconvex for which their first-order or second-order derivatives are unavailable. In this paper, we analyze the solution of the nonlinear problem with bound constraints:

$$\min f(x) \quad (1a)$$

$$\text{subject to } x \in \Omega, \quad (1b)$$

where $\Omega = \{x \in \mathbb{R}^n \mid l \leq x \leq u\}$ and $f(x) : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is a twice continuously differentiable function, but its first-order or second-order derivatives are not explicitly available. It mainly emerges in the field of operations research, management science, industrial engineering, applied mathematics, and network transmission [1] and in engineering disciplines that deal with analytical optimization techniques such as banking business and weather analysis. The unavailable first- or second-order derivatives may result in traditional derivative-based methods like quasi-Newton methods and conjugate gradient methods are not work. Therefore,

researches focus on derivative-free methods which could effectively avoid the use of derivative information.

1.1. Derivative-Free Trust Region Method. Derivative-free technique has been explored to tune parameters of nonlinear optimization methods [2], to automatic error analysis [3, 4] and to design helicopter rotor blade [5, 6] and hydrodynamic [7]. These methods are all special algorithms designed for particular optimizations, but have their usage limitations. In [8–12] another type of derivative-free methods is proposed based on the traditional derivative-based algorithm framework [8, 13–15]. They construct a function with all available derivatives to approximate the original objective function. Conn and Scheinberg and Vicente [8, 13] have already given a derivative-free method under trust region method framework. They construct the trust region subproblem

$$\begin{aligned} m_k(x_k + p) &= m_k(x_k) + q_{m_k}(p) \\ &= m_k(x_k) + g_k^T p + \frac{1}{2} p^T H_{m_k} p, \end{aligned} \quad (2)$$

where $m_k(x_k)$ is the function value at k th iteration point, g_k is the gradient of $m_k(x_k + p)$ at k th iteration point, and H_{m_k} is the

Hessian. Although the function model and the true objective function are meant to coincide the model gradients and Hessian may be (and typically are) different from the original objective function gradient ∇f_k and Hessian H_{f_k} , function (2) defined in [8] is called *fully linear* or *fully quadratic model*, dependent upon the related chosen truncated Taylor series conditions; it must be occasionally updated in order to guarantee that the residual between the approximated and real functions (and more critically, their gradients) is within the related error bounds. In fact, by definition, the function values are essentially the same. We will show the definition of *fully linear model* after a reasonable assumption.

Assumption (A1). Suppose that a level set $\mathcal{L}(x_0)$ and a maximal radius Δ_{\max} are given. Suppose furthermore that f is twice continuously differentiable with Lipschitz continuous Hessian in an appropriate open domain containing the Δ_{\max} neighborhood $\bigcup_{x \in \mathcal{L}(x_0)} \mathcal{B}(x, \Delta_{\max})$ of the set $\mathcal{L}(x_0)$.

Definition 1. Let a function f , which satisfies assumption (A1), be given. A set of model functions $\mathcal{M} = \{m : \mathfrak{R}^n \rightarrow \mathfrak{R}, m \in C^2\}$ is called a fully linear class of models if the following hold.

There exist positive constants κ_{ef} , κ_{eg} , and κ_{blg} , such that, for any $x \in \mathcal{L}(x_0)$, $\Delta \in (0, \Delta_{\max}]$. There is a model function $m(x + p)$ in \mathcal{M} , with Lipschitz continuous gradient and corresponding Lipschitz constant bounded by κ_{blg} , such that

(1) the error between the gradient of the model and the original objective function satisfies

$$\|\nabla f(x + p) - \nabla m(x + p)\| \leq \kappa_{eg} \Delta, \quad \forall p \in \mathcal{B}(0, \Delta), \quad (3)$$

(2) the error between the model and the original objective function satisfies

$$\|f(x + p) - m(x + p)\| \leq \kappa_{ef} \Delta^2, \quad \forall p \in \mathcal{B}(0, \Delta). \quad (4)$$

Such a model m is called fully linear on $\mathcal{B}(x, \Delta)$.

Remark 2. For this class \mathcal{M} there exists an algorithm which we will call a model-improvement algorithm that, in a finite, uniformly bounded (with respect to x and Δ) number of steps can

- (1) either establish that a given model $m \in \mathcal{M}$ is fully linear on $\mathcal{B}(x, \Delta)$
(we will say that a certificate has been provided and the model is certifiably fully linear)
- (2) or find a model $\bar{m} \in \mathcal{M}$ that is fully linear on $\mathcal{B}(x, \Delta)$.

1.2. Affine-Scale Trust Region Method for Bound Constrained Optimization. Since we analyze the bound constrained optimization (1a) and (1b), the trust region subproblem is

$$\begin{aligned} \min \quad & q_{m_k}(p) \\ \text{subject to} \quad & l \leq x_k + p \leq u, \\ & \|p\|_{\infty} \leq \Delta_k. \end{aligned} \quad (5)$$

As the solution of subproblem (5), p_k induces following decrease:

$$\text{Pred}_k(p_k) = -q_{m_k}(p_k) \geq \kappa_1 \chi_{m_k} \min\{1, \Delta_k, \chi_{m_k}\} \quad (6)$$

with constant $\kappa_1 > 0$ independent of k . Hereby, the norm of the projected gradient

$$\chi_{m_k} = \|x_k - P_X(x_k - g_k)\| = \|\hat{g}(x_k)\| \quad (7)$$

is a suitably chosen criticality measurement. In order to obtain a relatively short and elegant convergence result, we describe a concrete implementation by means of a Cauchy step that is defined by an affine-scaled gradient used here have stronger smoothness properties. Similar approaches can be found in [16–18]. Define the diagonal affine-scaling matrix $D_{m_k}(x) \in \mathfrak{R}^{n \times n}$ as

$$D_{m_k}(x)_{ii} = \begin{cases} 1, & l_i = -\infty \text{ and } u_i = +\infty, \\ \min\{x_i - l_i + \gamma \max\{\kappa_D, -(g_k)_i\}, u_i - x_i + \gamma \max\{\kappa_D, (g_k)_i\}\}, & \text{otherwise,} \end{cases} \quad (8)$$

where $\kappa_D > 0$ and $\gamma > 0$ are given constants and $(g_k)_i$ is the i th component of the gradient of $m_k(x_k + p)$. Solve the subproblem

$$\begin{aligned} \min \quad & q_{m_k}(p) \\ \text{subject to} \quad & p = -t D_{m_k}(x_k)^2 g_k \\ & t \geq 0, \\ & p \in X_k = [l - x_k, u - x_k] \cap [-\Delta_k, \Delta_k]^n. \end{aligned} \quad (9)$$

Following the idea of [18], we are able to prove that the solution of this quadratic model (9) also satisfies the decrease (6).

1.3. Nonmonotone Filter Technique. The filter method was first introduced for constrained nonlinear optimization by Fletcher and Leyffer [19] and then it has a wide range of applications in various optimization problems; see [18, 20–23]. In 2005, the filter method has been extended into a filter trust region method by Gould et al [24] for unconstrained optimization and by Sainvrtu [25] for general box constrained optimization. They indicate that the filter method is a reliable and efficient way for nonlinear optimizations. In this paper, we will make a further study on nonmonotone filter method and propose a new algorithm for (1a) and (1b). The main features of this paper are as follows:

- (i) We present a further extension of that filter trust region method by introducing both a suitable nonmonotonicity criterion and a derivative-free strategy for bound constrained optimization.
- (ii) The global convergence of the presented derivative-free trust region method with the nonmonotone filter technique for bound constrained optimization is established.
- (iii) Numerical results indicate that the new algorithm is effective for problems for which the derivative functions are unavailable.

The paper is therefore in the following way: we present our algorithmic scheme in Section 2. There we first show that the decrease direction described in this paper satisfies the predicted decrease inequality and recall the nonmonotone trust region method from [18, 26] and then make the necessary modifications for a derivative-free version. The global convergence properties of the derivative-free trust region method with nonmonotone filter technique is shown in Section 3. The corresponding numerical results are reported in Section 4 together with some additional tests. Finally, conclusions and further discussions are given.

Notation. Unless otherwise specified, throughout this paper, the norm $\|\cdot\|$ is the 2-norm for a vector and the induced 2-norm for a matrix. Let B denote a closed ball in $\Omega \subset \mathfrak{R}^n$ and $\mathcal{B}(x, \Delta)$ denote the closed ball centered at x , with radius $\Delta > 0$. In addition, $\mathcal{L}(x_0) = \{x \in \Omega \mid f(x) \leq f(x_0)\}$ is the level set about f . We use the subscript f_k and subscript m_k to distinguish the relevant information between the original function and the approximate function; for example, χ_{f_k} is the criticality measurement of $f(x_k)$ and χ_{m_k} is the criticality measurement of $m_k(x_k + p)$. x_k^+ is the trial step in k th iteration. g_k^+ and $H_{m_k}^+$ is the gradient and Hessian of the trial step, respectively.

2. The Derivative-Free Trust Region Algorithm with Nonmonotone Filter Technique

We analyze the behaviors of subproblem (9) with the diagonal matrix defined by (8). Let χ_{m_k} denote the criticality measurement such that

$$\kappa_2 \chi_{m_k} \leq \|D_{m_k}(x_k) g(x_k)\| \quad (10)$$

holds on X_k for some $\kappa_2 > 0$. Fraction of Cauchy decrease condition defined as

$$q_{m_k}(p_k) \leq \alpha q_{m_k}(p_k^c), \quad (11)$$

where $\alpha \in (0, 1)$ is a constant and $q_{m_k}(p_k^c) = q_{m_k}(t^* d_k) = \min\{q_{m_k}(t d_k) : t \leq 0, t d_k \in X_k\}$ with $d_k = D_{m_k}(x_k) g_k$. It is not difficult to prove that both $\|D_{m_k}(x) g(x)\|$ and χ_{m_k} are criticality measurement, i.e., satisfying that $\chi_{m_k} = 0$ if and only if x is the KKT point of problem (1a) and (1b). Furthermore, if $g(x)$ is bounded and uniformly continuous on (l, u) , then $\|D_{m_k}(x) g(x)\|$ is uniformly continuous. The proof is similar to Lemma 6.1 and Lemma 6.2 in [18] except

that we now have to replace $\nabla h(x)$ by the approximated gradient g_k . In order to discuss the global convergence we first provide the following lemma to show that the decrease direction p_k satisfies the predicted decrease inequality (6).

Lemma 3. *Suppose that criticality measurement χ_{m_k} satisfies (10). If $\chi_{m_k} \neq 0$ and if the trial step $x_k + p_k$ satisfies the fraction of Cauchy decrease condition (11), then (6) holds.*

Proof. Considering that $d_k = D_{m_k}(x_k) g_k$ and $q_{m_k}(p_k^c)$ is defined above, firstly we obtain from following inequality that d_k is a decrease direction of $q_{m_k}(x)$ at 0, that is,

$$\begin{aligned} \nabla q_{m_k}(0)^T d_k &= g_k^T d_k = -\|D_{m_k}(x_k) g(x_k)\|^2 \\ &\leq -\kappa_2^2 \chi_{m_k}^2 < 0. \end{aligned} \quad (12)$$

In the case that the maximum stepsize is determined by the trust region constraint $t \|d_k\| \leq \Delta_k$, we obtain

$$\begin{aligned} t_1 &= \min \left\{ \frac{\Delta_k}{|(d_k)_i|}; (d_k)_i \neq 0 \right\} \\ &\geq \frac{\Delta_k}{\|D_{m_k}(x_k)_{ii}\| \|D_{m_k}(x_k)_{ii} g_k\|} \\ &\geq \frac{\Delta_k}{\gamma \kappa_D \|D_{m_k}(x_k) g_k\|}. \end{aligned} \quad (13)$$

In the case that the maximum stepsize is determined by the lower bounds of the set X_k , then

$$\begin{aligned} t_2 &= \min \left\{ \frac{(x_k - l)_i}{|(d_k)_i|}; (d_k)_i < 0 \right\} \\ &= \min \left\{ \frac{(x_k - l)_i}{D_{m_k}(x_k)_{ii} |(D_{m_k}(x_k) g_k)_i|}; (g_k)_i \right. \\ &\quad \left. > 0, (x_k)_i > l_i \right\} \\ &\geq \min \left\{ \frac{(x_k - l)_i}{\min\{\kappa_D, (x_i - l_i)\} \|D_{m_k}(x_k) g_k\|}; (g_k)_i \right. \\ &\quad \left. > 0, (x_k)_i > l_i \right\} \geq \frac{\kappa_D}{\|D_{m_k}(x_k) g_k\|}. \end{aligned} \quad (14)$$

In the same way, the stepsize t_3 admitted by the upper bounds of the set X_k can be estimated:

$$t_3 = \min \left\{ \frac{(u - x_k)_i}{|(d_k)_i|}; (d_k)_i < 0 \right\} \geq \frac{\kappa_D}{\|D_{m_k}(x_k) g_k\|}. \quad (15)$$

In the case $D_{m_k}(x_k)H_{m_k}D_{m_k}(x_k) = 0$, we set $t_4 = +\infty$. Otherwise, in the case that $D_{m_k}(x_k)H_{m_k}D_{m_k}(x_k)$ positive definition as well as $\|H_{m_k}\|$ less than a constant κ_H , the function $q_m(td_k)$, $t \geq 0$, attains its global minimum at $t = t_4$, where

$$\begin{aligned} t_4 &= \frac{-2g_k^T d_k}{d_k^T H_{m_k} d_k} \\ &\geq \frac{2\|D_{m_k}(x_k)g(x_k)\|^2}{\|D_{m_k}(x_k)g(x_k)\|^2 \|D_{m_k}(x_k)H_{m_k}D_{m_k}(x_k)\|} \\ &\geq \frac{2}{\gamma^2 \kappa_D^2 \kappa_H}. \end{aligned} \quad (16)$$

We have $t^* = \min\{t_1, t_2, t_3, t_4\}$. If $t^* < t_4$, then $\|D_{m_k}(x_k)g(x_k)\|^2 > (1/2)\|d_k^T H_{m_k} d_k\|$, and if $d_k^T H_{m_k} d_k > 0$ then

$$\begin{aligned} q_{m_k}(t^* d_k) &= -t^* \|D_{m_k}(x_k)g(x_k)\|^2 \\ &\quad + \frac{(t^*)^2}{2} d_k^T H_{m_k} d_k \\ &< -\frac{t^*}{2} \|D_{m_k}(x_k)g(x_k)\|^2 \\ &= -\frac{\min\{t_1, t_2, t_3\}}{2} \|D_{m_k}(x_k)g(x_k)\|^2. \end{aligned} \quad (17)$$

If $d_k^T H_{m_k} d_k \leq 0$, then

$$Rared_k(p_k) = \max \left\{ f(x_k), \sum_{r=0}^{l_k-1} \lambda_{kr} f(x_{j_{i-r}}) \right\} - f(x_k^+),$$

for the computation of reduction ratio $\rho_k = Rared_k(p_k)/Pred_k(p_k)$ in order to decide whether a step is acceptable or not. The idea behind the update rule (22) is the following: instead of requiring that $f(x_k^+)$ be smaller than $f(x_k)$, it is only required that $f(x_k^+)$ is either less than $f(x_k)$ or less than the weighted mean of the function values at the last $l_k = \min\{i+1, l\}$ successful iterates. Of course, if $l = 1$, then $Rared_k(p) = Ared_k(p)$ and the usual reduction ratio is recovered. Our approach is a slightly stronger requirement than the straightforward idea to replace $Ared_k(p)$ with $\max_{0 \leq r < l_k} f(x_{j_{i-r}}) - f(x_k + p)$. Unfortunately, for this latter choice it does not seem to be possible to establish all the global convergence results that are available for the monotone case. For our approach, however, this is possible without making the theory substantially more difficult.

The other criterion is in the filter step. We prefer a filter mechanism to assess the suitability of x_k^+ . Our strategy is

$$\begin{aligned} q_{m_k}(t^* d_k) &= -t^* \|D_{m_k}(x_k)g(x_k)\|^2 \\ &\quad + \frac{(t^*)^2}{2} d_k^T H_{m_k} d_k \\ &\leq -t^* \|D_{m_k}(x_k)g(x_k)\|^2. \end{aligned} \quad (18)$$

If, on the other hand, $t^* = t_4$, then

$$q_{m_k}(x_k) = -\frac{t_4}{2} \|D_{m_k}(x_k)g(x_k)\|^2. \quad (19)$$

Combining with the assumptions (10) and (11), the conclusion is obtained. \square

There are two criteria in the proposed algorithm, to measure if the trial step $x_k^+ = x_k + p_k$ is acceptable. One is in the trust region method. The new algorithm is based on a nonmonotone decrease criterion. Nonmonotone trust region methods were investigated by Toint [26] and Ulbrich [18]. Let the increasing sequence $(j_i)_{i \geq 0}$ enumerate all indices of accepted steps. Moreover,

$$x_k = x_{j_i}, \quad \forall j_{i-1} < k \leq j_i, \quad \forall i \geq 1. \quad (20)$$

Conversely, if $k \neq j_i$ for all i , then p_k was rejected. In the following we introduce the set of all these ‘‘successful’’ indices j_i by S :

$$S = \{j_i : i \geq 0\} = \{k; \text{trial step } x_k + p_k \text{ is accepted}\}. \quad (21)$$

We follow [18] to choose integer $l, m \geq 0$, fix $\lambda \in (0, 1/m]$, and then compare the predicted decrease promised by the trust region model with a relaxation of the actual decrease

$$\text{where } \lambda_{kr} \geq \lambda, \quad l_k = \min\{i+1, l\}, \quad r = 0, 1, \dots, l_k - 1 \text{ and } \sum_{r=0}^{l_k-1} \lambda_{kr} = 1 \quad (22)$$

inspired by that of [24]: we decide that a trial point x_k^+ is acceptable for the filter \mathcal{F} if and only if

$$\begin{aligned} \forall \widehat{g}_\ell \in \mathcal{F} \\ \exists j \in \{1, \dots, n\} : |\widehat{g}_j(x_k^+)| < |\widehat{g}_{\ell, j}| - \gamma_{\widehat{g}} \|\widehat{g}_\ell\|, \end{aligned} \quad (23)$$

where $\gamma_{\widehat{g}} \in (0, 1/\sqrt{n})$ is a small positive constant and $\widehat{g}_{\ell, j} = \widehat{g}_j(x_\ell)$.

Aiming to solve the nonlinear optimization with unavailable first- (or second-) order derivatives and to guarantee reliable and efficient numerical performance, we now state following derivative-free trust region method.

Algorithm 4 (a derivative-free trust region method with non-monotone filter technique).

Initialization Step. Choose a starting point $x_0 \in \Omega$ and suitable constants $\Delta_{\max}, \Delta_0, \eta_0, \eta_1, \mu_0, \mu_1, \tau_1, \tau_2, \zeta, \varsigma, \varepsilon, \mu, \beta, \omega$, and ι such that $\Delta_{\max} > 0, \Delta_0 \in (0, \Delta_{\max}], 0 \leq \eta_0 \leq \eta_1 < 1, (\eta_1 \neq 0), \mu_0 \in (0, 1/2), \mu_1 \in (0, 1), \tau_1 > 0, \tau_2 > 0, 0 < \zeta < 1 < \varsigma, \varepsilon > 0, \mu > \beta > 0, \omega \in (0, 1)$, and $\iota \in (0, 1)$. Set $\text{NONCONVEX}=0, \text{RESTRICT}=0$, and $k := 0$.

Main Step

Step 1. Construct $m_k(x_k + p)$ as in (2). Get g_k and H_{m_k} .

If $m_k(x_k + p)$ is nonconvex, set $\text{NONCONVEX}=1$.

Step 2. If $\chi_{m_k} \leq \varepsilon$ and $m_k(x_k + p)$ is fully linear and $\Delta_k \leq \iota\chi_{m_k}$, stop; otherwise, set $\tilde{\Delta}_k = \Delta_k$, implement step 9 to obtain $\tilde{\Delta}_k$, and make $m_k(x_k + p)$ fully linear. Update $\Delta_k = \min\{\max\{\tilde{\Delta}_k, \beta\tilde{\chi}_{m_k}\}, \Delta_k\}$.

Step 3. Solve the trust region subproblem (5).

Step 4. Compute the trial point $x_k^+ = x_k + p_k$. Obtain $m_k^+(x_k^+ + p), g_k^+$ and $H_{m_k}^+$.

$$\Delta_{k+1} \in \begin{cases} \min\{\varsigma\Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta_1 \text{ and } \Delta_k < \iota\chi_{m_k}, \\ [\Delta_k, \min\{\varsigma\Delta_k, \Delta_{\max}\}] & \text{if } \rho_k \geq \eta_1 \text{ and } \Delta_k \geq \iota\chi_{m_k}, \\ \zeta\Delta_k & \text{if } \rho_k < \eta_1 \text{ and } m_k(x_k + p) \text{ is fully linear,} \\ \Delta_k & \text{if } \rho_k < \eta_1 \text{ and } m_k(x_k + p) \text{ is not fully linear.} \end{cases} \quad (25)$$

Step 8. If $\rho_k \geq \eta_1$, or $\rho_k \leq \eta_1$ and $\text{RESTRICT}=0$ and $m_k^+(x_k^+ + p)$ is not fully linear on $\mathcal{B}(x_{k+1}, \Delta_{k+1})$, go to Step 1; else set $\text{RESTRICT}=1$, implement Step 9, and then go to Step 2.

Step 9. Set $i = 0$ and set $m_k^{(0)}(x_k + p) = m_k(x_k + p)$. Repeat: increment i by one, $\tilde{\Delta}_k \leftarrow \omega^{i-1}\Delta_k$, and improve $m_k^{(i-1)}(x_k + p)$ by $m_k^{(i)}(x_k + p)$ until it is fully linear (i.e., satisfying the error bounds (3) and (4)).

Set $\tilde{\Delta}_k \leftarrow \Delta_k$ and $\tilde{m}_k(x_k + p) = m_k^{(i)}(x_k + p)$.

Until $\tilde{\Delta}_k \leq \iota(\chi_{m_k})^{(i)}$.

Remark 5. At the beginning of every iteration, Step 5 is encountered with $i = -1$. In this case the sum in (22) is empty and thus we define

$$\text{Rared}_k(p) = f(x_k) - f(x_k + p) = \text{Ared}_k(p). \quad (26)$$

Remark 6. In order to obtain a suitable approximation function, the objective function of the trust region subproblem needs to update if necessary. Step 9 is the model improvement method which has the same principle as the Algorithm 2 proposed in [14, 15].

Step 5. Evaluate ratio

$$\rho_k = \frac{\text{Rared}_k(p_k)}{\text{Pred}_k(p_k)}, \quad (24)$$

by formulas (6) and (22).

If $f(x_k^+) > f_{\text{sup}}$, set $x_{k+1} = x_k$ and $\text{RESTRICT}=1$. Go to Step 7.

Step 6. These are tests to accept the trial step by (20) and (23).

If x_k^+ is acceptable for the filter \mathcal{F} and $\text{NONCONVEX}=0$: set $x_{k+1} = x_k^+$ and $\text{RESTRICT}=0$ and add

g_k^+ to the filter \mathcal{F} if either $\rho_k < \eta_1$.

Elseif $\rho_k \geq \eta_1$, set $x_{k+1} = x_k^+$ and $\text{RESTRICT}=0$;

Elseif $\text{NONCONVEX}=1$, set $f_{\text{sup}} = f(x_{k+1})$ and reinitialize the filter \mathcal{F} to the empty set;

Else set $x_{k+1} = x_k$ and $\text{RESTRICT}=1$.

Step 7. Adjust trust region Δ_k

3. Global Convergence for First-Order Critical Points

The purpose of this section is to provide in-depth introduction to the global convergence properties of Algorithm 4 in first-order case. We recall or state some reasonable assumptions that are assumed to hold for problem (1a) and (1b) in order to get global convergence of the derivative-free trust region method.

Assumptions

(A2) The level set $\mathcal{L}(x_0)$ is bounded.

(A3) There exist positive constants κ_{H_f} and κ_{H_m} such that $\|H_{f_k}\| \leq \kappa_{H_f}$ and $\|H_{m_k}\| \leq \kappa_{H_m}$, respectively, for all $x_k \in \mathcal{L}(x_0)$. And set $\kappa_H = \max\{\kappa_{H_f}, \kappa_{H_m} + 1\}$.

The global convergence property is described by following Theorem 7 which indicates that there exist at less one accumulation point of a sequence generated by the derivative-free trust region method from Algorithm 4 with filter technique which is still a stationary point of the optimization problem (1a) and (1b).

Theorem 7. Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max}

hold. Suppose furthermore that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$. Then

$$\liminf_{k \rightarrow +\infty} \chi_{f_k} = 0. \quad (27)$$

In order to obtain Theorem 7, the remainder of this section will derive Lemmas 8–15 as support.

Lemma 8. Suppose that Assumptions (A1)–(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. $m_k(x_k + p)$ is the fully linear function on $\mathcal{B}(x_k, \Delta_k)$. Then step 3 of Algorithm 4 will terminate in a finite number of improvement step, if $\chi_{f_k} \neq 0$.

Proof. We prove this result by contradiction. Assume that the loop in Step 9 is infinite. We will show that χ_{f_k} must be zero in this case.

If Step 9 is implemented, we notice that we do not have a certifiably fully linear model $m_k(x_k + p)$ and that the radius Δ_k exceeds $\iota_{\chi_{m_k}}$. Then set $m_k^{(0)}(x_k + p) = m_k(x_k + p)$ and improve the model until it is fully linear on $\mathcal{B}(x_k, \omega^{(0)}\Delta_k)$. If $\chi_{m_k}^{(1)}$ of the resulting model $m_k^{(1)}(x_k + p)$ satisfies $\iota_{\chi_{m_k}^{(1)}} \geq \omega^{(0)}\Delta_k$, the procedure stops with $\tilde{\Delta}_k = \omega^{(0)}\Delta_k \leq \iota_{\chi_{m_k}^{(1)}}$.

Otherwise, that is, $\iota_{\chi_{m_k}^{(1)}} < \omega^{(0)}\Delta_k$ improve the model until it is fully linear on $\mathcal{B}(x_k, \omega\Delta_k)$. Then, estimate whether the procedure stops or not. If not, the radius should be multiplied by ω again, and go on.

The only way for this procedure to be infinite is if

$$\iota_{\chi_{m_k}^{(i)}} < \omega^{i-1}\Delta_k \quad \text{for all } i \geq 1. \quad (28)$$

This construction implies

$$\lim_{i \rightarrow +\infty} \chi_{m_k}^{(i)} = 0. \quad (29)$$

Since each model $m_k^{(i)}(x_k + p)$ was fully linear on $\mathcal{B}(x_k, \omega^{i-1}\Delta_k)$, the bound (3) provides

$$\begin{aligned} \chi_{f_k}^{(i)} &= \|x_k - P_X [x_k - \nabla f_k]\| \\ &\leq \|P_X [x_k - \nabla f_k] - P_X [x_k - (g_k)^{(i)}]\| \\ &\quad + \|x_k - P_X [x_k - (g_k)^{(i)}]\| \\ &\leq \|\nabla f_k - (g_k)^{(i)}\| + \chi_{m_k}^{(i)} \leq \left(\kappa_{eg}\Delta_{\max} + \frac{1}{l}\right) \omega^{i-1}\Delta_k \\ &\quad \text{for all } i \geq 1. \end{aligned} \quad (30)$$

The choice of $\omega \in (0, 1)$ in Algorithm 4 implies that

$$\chi_{f_k} = 0. \quad (31)$$

□

Lemma 9. Suppose that Assumptions (A1)–(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. $m_k(x_k + p)$ is the fully linear function on $\mathcal{B}(x_k, \Delta_k)$.

Suppose furthermore that x_k, p_k, Δ_k , etc. are generated by Algorithm 4, then for all computed indices $i \geq 1$ holds,

$$\begin{aligned} f(x_{j_i}) &< f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-2} \text{Pred}_{j_r}(p_{j_r}) \\ &\quad - \eta_1 \text{Pred}_{j_{i-1}}(p_{j_{i-1}}). \end{aligned} \quad (32)$$

Proof. We note from (6) that $\text{Pred}_k(p_k) > 0$. The proof is by induction. For $i = 1$ we have by (20) and using $\rho_{j_0}(p_{j_0}) > \eta_1$

$$\begin{aligned} f(x_{j_1}) &= f(x_{j_0+1}) = f(x_{j_0}) - \text{Ared}_{j_0}(p_{j_0}) \\ &< f(x_{j_0}) - \eta_1 \text{Pred}_{j_0}(p_{j_0}) \\ &= f(x_0) - \eta_1 \text{Pred}_{j_0}(p_{j_0}). \end{aligned} \quad (33)$$

Now assume that (32) holds for $1, \dots, i$. If $\text{Rared}_{j_i}(p_{j_i}) = \text{Ared}_{j_i}(p_{j_i})$ then, using (32) and $\lambda \leq 1$,

$$\begin{aligned} f(x_{j_{i+1}}) &= f(x_{j_i+1}) = f(x_{j_i}) - \text{Ared}_{j_i}(p_{j_i}) \\ &= f(x_{j_i}) - \text{Rared}_{j_i}(p_{j_i}) \\ &< f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-2} \text{Pred}_{j_r}(p_{j_r}) \\ &\quad - \eta_1 \text{Pred}_{j_{i-1}}(p_{j_{i-1}}) - \eta_1 \text{Pred}_{j_i}(p_{j_i}) \\ &\leq f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-1} \text{Pred}_{j_r}(p_{j_r}) \\ &\quad - \eta_1 \text{Pred}_{j_i}(p_{j_i}). \end{aligned} \quad (34)$$

If $\text{Rared}_{j_i}(p_{j_i}) \neq \text{Ared}_{j_i}(p_{j_i})$ then $\text{Rared}_{j_i}(p_{j_i}) > \text{Ared}_{j_i}(p_{j_i})$, and with $q = \min\{i, l-1\}$ we obtain

$$\begin{aligned} f(x_{j_{i+1}}) &= f(x_{j_i+1}) = \sum_{t=0}^q \lambda_{j_i t} f(x_{j_{i-t}}) \\ &\quad - \text{Rared}_{j_i}(p_{j_i}) < \sum_{t=0}^q \lambda_{j_i t} \left(f(x_0) \right. \\ &\quad \left. - \eta_1 \lambda \sum_{r=0}^{i-t-2} \text{Pred}_{j_r}(p_{j_r}) - \eta_1 \text{Pred}_{j_{i-t-1}}(p_{j_{i-t-1}}) \right) \\ &\quad - \eta_1 \text{Pred}_{j_i}(p_{j_i}). \end{aligned} \quad (35)$$

Using $\{0, \dots, q\} \times \{0, \dots, i-t-2\} \subset \{(t, r); 0 \leq p \leq t, 0 \leq r \leq i-t-2\}$, $\lambda_{j_i 0} + \dots + \lambda_{j_i t} = 1$, and $\lambda_{j_i t} \geq \lambda$, we can proceed as follows:

$$\begin{aligned} f(x_{j_{i+1}}) &< f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-t-2} \left(\sum_{t=0}^q \lambda_{j_i t} \right) \text{Pred}_{j_r}(p_{j_r}) \\ &\quad - \eta_1 \lambda \sum_{t=0}^q \text{Pred}_{j_{i-t-1}}(p_{j_{i-t-1}}) \end{aligned}$$

$$\begin{aligned}
 & -\eta_1 \text{Pred}j_i(p_j) \\
 & < f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-t-2} \text{Pred}j_r(p_{j_r}) \\
 & - \eta_1 \lambda \sum_{r=i-t-1}^{i-1} \text{Pred}j_r(p_{j_r}) - \eta_1 \text{Pred}j_i(p_j) \\
 & = f(x_0) - \eta_1 \lambda \sum_{r=0}^{i-1} \text{Pred}j_r(p_{j_r}) \\
 & - \eta_1 \text{Pred}j_i(p_j). \tag{36}
 \end{aligned}$$

□

Lemma 9 provides the sufficiently descent property at the k th iteration of Algorithm 4.

Lemma 10. Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. Suppose furthermore that x_k, p_k, Δ_k , etc. are generated by Algorithm 4, that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$, and that

$$\Delta_k \leq \min \left\{ 1, \chi_{m_k}, \frac{\kappa_1 (1 - \eta_1) \chi_{m_k}}{2\kappa_{ef}} \right\}. \tag{37}$$

Then for arbitrary $x \in \mathcal{L}(x_0)$ with $\chi_{m_k} \neq 0$, the k th iteration is successful.

Proof. Since $\chi_{m_k} \neq 0$ and $\Delta_k \leq \min\{1, \chi_{m_k}\}$, we obtain from the decrease condition (6)

$$\begin{aligned}
 \text{Pred}_k(p_k) & = -q_{m_k}(p_k) \geq \kappa_1 \chi_{m_k} \min \{1, \Delta_k, \chi_{m_k}\} \\
 & \geq \kappa_1 \chi_{m_k} \Delta_k. \tag{38}
 \end{aligned}$$

On the other hand, since the current model is fully linear on $\mathcal{B}(x_k, \Delta_k)$, then from the bound (3) on the error between the function and the model and from (38), if $\text{Rared}_k(p_k) = \text{Ared}_k(p_k)$, then we have

$$\begin{aligned}
 |\rho_k - 1| & = \left| \frac{\text{Rared}_k(p_k)}{\text{Pred}_k(p_k)} - 1 \right| \\
 & = \left| \frac{f(x_k) - f(x_k + p_k)}{\text{Pred}_k(p_k)} - 1 \right| \\
 & \stackrel{f(x_k)=m_k(x_k)}{\leq} \left| \frac{m_k(x_k + p_k) - f(x_k + p_k)}{\text{Pred}_k(p_k)} \right| \\
 & \leq \frac{2\kappa_{ef}\Delta_k^2}{\kappa_1 \chi_{m_k} \Delta_k} \leq 1 - \eta_1 \tag{39}
 \end{aligned}$$

where we have used the assumption $\Delta_k \leq \kappa_1(1-\eta_1)\chi_{m_k}/(2\kappa_{ef})$ to deduce the inequality.

If $\text{Rared}_k(p_k) > \text{Ared}_k(p_k)$, then we have, with appropriate $x'_k \in [x_k, x_k + p_k]$,

$$\begin{aligned}
 \text{Ared}_k(p_k) & = f(x_k) - f(x_k + p_k) = -\nabla f(x'_k)^T p_k \\
 & = -q_{m_k}(p_k) + [g_k - \nabla f(x'_k)]^T p_k \\
 & \quad + \frac{1}{2} p_k^T H_{m_k} p_k. \tag{40}
 \end{aligned}$$

Since ∇f continuous, there exists $\delta' > 0$ such that

$$\begin{aligned}
 & \|\nabla f(x'_k) - \nabla f(x_k)\|_1 \\
 & \leq \left[(1 - \eta_1) \kappa_1 - \kappa_{eg} - \frac{1}{2} \kappa_{H} \Delta_{\max} \right] \chi_{m_k}, \tag{41}
 \end{aligned}$$

for all $x'_k \in X_k$ with $\|x'_k - x_k\| < \delta$. Hence, for these indices k ,

$$\begin{aligned}
 & \left\| g_k - \nabla f(x'_k) + \frac{1}{2} p_k^T H_{m_k} p_k \right\|_1 \\
 & \leq \|g_k - \nabla f(x_k)\| + \|\nabla f(x'_k) - \nabla f(x_k)\|_1 \\
 & \quad + \frac{1}{2} \kappa_{H} \Delta_k^2 \leq (1 - \eta_1) \kappa_1 \chi_{m_k}. \tag{42}
 \end{aligned}$$

At the same time, we get from (5) and (38) that

$$\|p_k\|_{\infty} \leq \Delta_k \leq \frac{1}{\kappa_1 \chi_{m_k}} \text{Pred}_k(p_k). \tag{43}$$

This inequality implies that

$$\begin{aligned}
 & \left| [g_k - \nabla f(x'_k)]^T p_k + \frac{1}{2} p_k^T H_{m_k} p_k \right| \\
 & \leq (1 - \eta_1) \kappa_1 \chi_{m_k} \|p_k\|_{\infty} \leq (1 - \eta_1) \text{Pred}_k(p_k). \tag{44}
 \end{aligned}$$

This means that

$$\begin{aligned}
 \text{Rared}_k(p_k) & \geq \text{Ared}_k(p_k) \\
 & \geq \text{Pred}_k(p_k) \\
 & \quad - \left| [g_k - \nabla f(x'_k)]^T p_k + \frac{1}{2} p_k^T H_{m_k} p_k \right| \\
 & \geq \eta_1 \text{Pred}_k(p_k). \tag{45}
 \end{aligned}$$

For either case, we obtain the conclusion that $\rho_k \geq \eta_1$; that is, the k th iteration is successful. □

Lemma 11. Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. Suppose furthermore that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$ and that there exists a constant $\kappa_{lbg} > 0$ such that $\chi_{m_k} \geq \kappa_{lbg}$ for all k . Then there is a constant $\kappa_{ubd} > 0$ such that $\Delta_k \geq \kappa_{ubd}$ for all k .

Proof. The proof is the same as for Lemma 5.3 in [8] when $\|p_k\|_{\infty} \leq \Delta_k$ except that we now have to replace $\|g_k\|$ by the criticality measurement χ_{m_k} and use (38) instead of the model decrease defined in [8]. □

Lemma 12. *Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. Suppose furthermore that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$ and that there exists a constant $\kappa_{lbg} > 0$ such that $\chi_{m_k} \geq \kappa_{lbg}$ for all k . Then there can be only finitely many successful nonconvex iterations in the course of the algorithm, i.e., $|S \cap N| < +\infty$.*

Proof. Suppose, for the purpose of obtaining a contradiction, that there are infinitely many successful nonconvex iterations, which we index by $S \cap N = \{k_i\}$. It follows from the fact of $S \cap N = D \cap N$ that the algorithm also guarantees that $\rho_k \geq \eta_1$ for all iteration in $S \cap N$, where $D = \{k \mid \rho_k \geq \eta_1\}$ is the set of sufficient descent iterations, which in turn implies with (6) that, for $k \in S \cap N$,

$$\begin{aligned} \text{Rared}_k(p_k) &\geq \eta_1 [m_k(x_k) - m_k(x_k + p_k)] \\ &\geq \eta_1 \kappa_1 \chi_{m_k} \min \{1, \Delta_k, \chi_{m_k}\} \\ &\geq \eta_1 \kappa_1 \kappa_{lbg} \min \{1, \Delta_k, \kappa_{lbg}\} \end{aligned} \quad (46)$$

where we have used Lemma 8, (A3) and our lower bound on the gradient norm to obtain the last inequality. Combining now this bound with (3), we deduce that

$$\begin{aligned} f(x_0) - f(x_{k+1}) &\geq \sum_{j=0, j \in S \cap N}^k \text{Rared}_k(p_k) \\ &\geq \sum_{j=0, j \in S \cap N}^k \left[\max \left\{ f(x_k), \sum_{r=0}^{k-1} \lambda_{kr} f(x_{j_{i-1}}) \right\} \right. \\ &\quad \left. - f(x_{k+1}) \right] \geq \varsigma_k \eta_1 \kappa_1 \kappa_{lbg} \min \{1, \Delta_k, \kappa_{lbg}\} \end{aligned} \quad (47)$$

where $\varsigma_k = |\{1, \dots, k\} \cap S \cap N|$. As we have supposed that there are infinitely many successful nonconvex iterations, we have that $\lim_{k \rightarrow \infty} \varsigma_k = +\infty$, and $|f(x_0) - f(x_{k+1})|$ is unbounded above, which contradicts the fact that the objective function is bound below; as stated in assumption A1, our initial assumption must then be false and the set $S \cap N$ of successful nonconvex iteration must be finite. \square

Lemma 13. *Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. Suppose furthermore that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$ and that there are only finitely many successful iterations, i.e., $|S| < +\infty$. Then $\lim_{k \rightarrow \infty} \chi_{m_k} = 0$.*

Proof. Let k_0 be the index of the last successful iteration. Then $x^* = x_{k_0+1} = x_{k_0+j}$ and

$$\rho_{k_0+j} < \eta_1 \quad \text{for all } j > 1. \quad (48)$$

Now observe that Restrict is set by the algorithm in the course of every unsuccessful iteration. This flag must thus be set at the beginning of every iteration of index $k_0 + j + 1$ for $j > 0$.

As a consequence, $\|p_{k_0+j+2}\| \leq \Delta_{k_0+j+2}$ for all $j > 0$. This, (48), and the mechanism of step 7 of the algorithm then imply that

$$\lim_{k \rightarrow \infty} \Delta_k = 0. \quad (49)$$

Assume now, for the purpose of establishing a contradiction, that $\chi_{m_{k_0+1}} \geq \varepsilon$ for some $\varepsilon > 0$. Then Lemma 12 implies that (49) is impossible and we deduce that

$$\chi_{m_{k_0+j}} = 0, \quad \text{for all } j > 0. \quad (50)$$

\square

Lemma 14. *Suppose that Assumptions (A1)-(A3), the error bounds (3) and (4), and the fact that Δ_k is bounded by Δ_{\max} hold. Suppose furthermore that $m_k(x_k + p)$ is fully linear on $\mathcal{B}(x_k, \Delta_k)$ and that $|A| = |S| = +\infty$. Then $\liminf_{k \rightarrow \infty} \chi_{m_k} = 0$.*

Proof. Assume for the purpose of obtaining a contradiction that, for all k large enough,

$$\chi_{m_k} \geq \kappa_{lbg} \quad (51)$$

for some $\kappa_{lbg} > 0$ and define $\{k_i\} = A$. The bound (51) and the Lemma 12 then imply that $|S \cap N|$ is finite and therefore that the filter is no longer reset to the empty set for k sufficiently large. Moreover, since our assumptions imply that $\{\chi_{m_{k_{i+1}}}\}$ is bounded above and away from zero, there must exist a subsequence $k_\ell \subseteq k_{i+1}$ such that

$$\lim_{\ell \rightarrow \infty} \chi_{m_{k_\ell}} = \chi_{m_\infty} \quad \text{with } \chi_{m_\infty} \geq \kappa_{lbg}. \quad (52)$$

By definition of $\{k_\ell\}$, x_{k_ℓ} is acceptable for the filter in each iteration $\ell - 1$. This implies, since the filter is not reset for ℓ large enough, that, for each ℓ sufficiently large, there exists an index $j_\ell \in \{1, \dots, n\}$ such that

$$\left| \widehat{g}_{k_\ell, j_\ell} \right| - \left| \widehat{g}_{k_{\ell-1}, j_\ell} \right| \leq -\gamma_{\widehat{g}} \left\| \widehat{g}_{k_{\ell-1}} \right\|. \quad (53)$$

But (51) implies that $\left\| \widehat{g}_{k_{\ell-1}} \right\| \geq \kappa_{lbg}$ for all ℓ sufficiently large. Hence we deduce from (53) that

$$\left| \widehat{g}_{k_\ell, j_\ell} \right| - \left| \widehat{g}_{k_{\ell-1}, j_\ell} \right| \leq -\gamma_{\widehat{g}} \kappa_{lbg}. \quad (54)$$

for all ℓ sufficiently large. But the left-hand side of this inequality tends to zero when ℓ tends to infinity because of (52), yielding the desired contradiction. Hence the conclusion holds. \square

We now consider the similar conclusion of Lemma 5.7 in [8] that if the model criticality measurement χ_{m_k} converges to 0 on a subsequence, then so does the true criticality measurement χ_{f_k} .

Lemma 15. *For any subsequence k_i such that*

$$\lim_{i \rightarrow +\infty} \chi_{m_{k_i}} = 0 \quad (55)$$

it also holds that

$$\lim_{i \rightarrow +\infty} \chi_{f_{k_i}} = 0. \quad (56)$$

Combining both Lemmas 14 and 15, a global convergent property immediately will be given as Theorem 7, which also illustrates that the criticality step plays an important role in ensuring a subsequence of the iterations being close to the first-order stationarity.

4. Numerical Experiment

In this section, we examine the practical performance of the derivative-free trust region method on two aspects. The comparisons of the numerical results between the proposed derivative-free trust region method and traditional gradient algorithm are firstly given in order to illustrate the effectiveness of Algorithm 4 in solving the general optimization problems. And then the derivative-free trust region algorithm with nonmonotone filter technique to parameter estimation is presented to show the performance of Algorithm 4 to derivative-free optimization problems. All routines are written in Matlab R2009a and run on a PC with 2.66GHz Intel(R) Core(TM)2 Quad CPU and 4G DDR2.

4.1. Numerical Results of the Derivative-Free Trust Region Algorithm with Nonmonotone Filter Technique. The Hock [27] test set is frequently used to test derivative-free algorithms on moderate-size problems. For running the proposed derivative-free trust region algorithm with nonmonotone filter technique, the bound constraints of each problem define the set Ω and linear (not bound) constraints of the original problem were processed by projections. We test 27 simple bound constrained optimization problems (List in Table 1) from Test examples for Nonlinear Programming Codes [27–29]. The values $\Delta_0 = 2, \eta_0 = 0.25, \eta_1 = 0.75, \zeta = 0.4, \varsigma = 2, \iota = 0.5, \beta = 0.3, \varepsilon = 10^{-8}$, and $\omega = 0.5$ are used. At the same time, we introduce two different algorithms traditional quasi-Newton method and conjugate gradient method [30] into this section to measure the objective algorithm efficiency through the tested problems. We denote the two algorithms as Algorithm 1.1 and Algorithm 1.2.

We use the tool of Dolan and Moré [31] to analyze the efficiency of the given three algorithms. Figures 1 and 2 show that the performance of Algorithm 4 is feasible and has the robust property among those three methods. It is not difficult to see from Figure 1 that Algorithm 4 has a huge lead among those three methods in the CPU time for solving each test problems. Simultaneously, Figure 2 illustrates that Algorithm 4 more quickly to optimal performance in terms of the number of function evaluations than other two algorithms.

To measure the efficiency of the proposed algorithm for large-scale optimizations, in this section, we compare this method with Algorithm 2.1 in [32] using three characteristics “NI,” “NF,” and “CPU,” where “NI” presents the number of iterations, “NF” is the calculation frequency of the function, and “CPU” is the time of the process in addressing the tested problems. The numerical results with the corresponding problem are listed in Table 2.

Although we only list the data of dimension of 9000, obviously, the objective algorithm (Algorithm 4) is more effective for large-scale optimizations since the iteration number and CPU time which are two essential aspects to measure the efficiency of an algorithm are less than Algorithm 2.1.

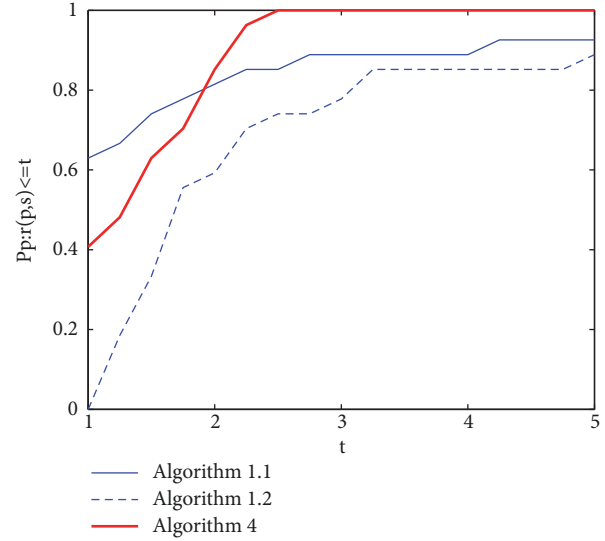


FIGURE 1: Profile comparing the CPU time required performance of different methods.

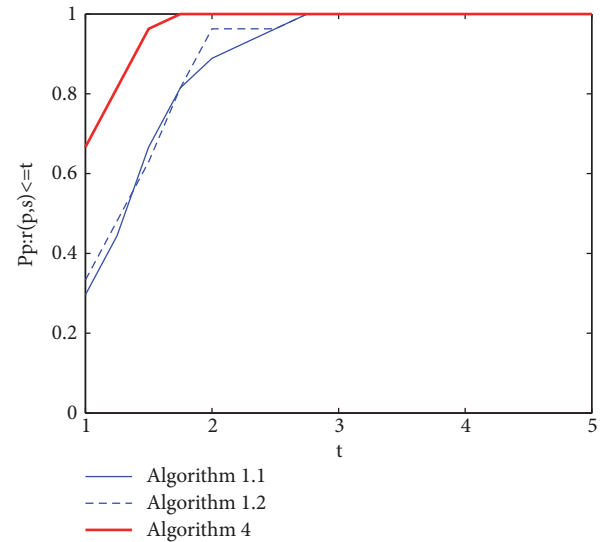


FIGURE 2: Profile comparing the numbers of function evaluations performance of different methods.

4.2. The Derivative-Free Trust Region Algorithm to Parameter Estimation

Algorithm 16 (basic trust region method).

Step 0. An initial point $x_0 \in R^n$ and an initial trust region radius $\Delta_0 > 0$ are given, as well as parameters $0 \leq \eta_0 < \eta_1 < 1$ and $0 < \zeta < 1 < \varsigma$. Compute $f(x_0)$ and set $k = 0$.

Step 1. Solve a trust region subproblem and compute a step p_k which satisfies the sufficient decrease conditions.

TABLE 1: Test problems.

Problem	Dim	x_0	Problem	Dim	x_0
HS1	2	[-2, 1]	HS2	2	[-2, 1]
HS3	2	[10, 1]	HS4	2	[1.125, 0.125]
HS5	2	[0, 0]	HS21	2	[-1, -1]
HS24	2	[1, 0.5]	HS25	3	[100, 12.5, 3]
HS35(Beale's problem)	3	[0.5, 0.5, 0.5]	HS38(Colville No.4)	4	[-3, -1, -3, -1]
HS45	5	[2, 2, 2, 2, 2]	HS110	10	[9, 9, ..., 9]
HS119(Colville No.7)	16	[10, 10, ..., 10]	HS224	2	[0.1, 0.1]
HS232	2	[2, 0.5]	HS247(Helical vally)	3	[-1, 0, 0]
HS250(Rosenbrock's-Post-office-prob.)	3	[10, 10, 10]	HS251	3	[10, 10, 10]
HS253	3	[0, 2, 0]	HS257	4	[-3, -1, -3, -1]
HS262	4	[0, 0, 0, 0]	HS279	8	[0, 0, ..., 0]
HS280	10	[0, 0, ..., 0]	HS299	100	[-1, 2, ..., -1, 2]
HS328(Gear train of minimum inertia)	2	[0.5, 0.5]	HS340(Pascal problem)	3	[1, 1, 1]
HS368(Gear train of minimum inertia)	8	[1, 1, ..., 1]			

TABLE 2: Numerical results.

Problem	Dim	Algorithm 4			Algorithm 2.1		
		NI	NF	CPU	NI	NF	CPU
Broyden Tridiagonal Function	9000	23	87	14.322854	13	54	9.718862
Extended Powell Function	9000	7	37	0.2094377	9	45	0.218401
Extended Penalty Function	9000	43	193	0.355428	65	252	0.421203
Extended Wood Function	9000	17	73	0.396603	13	50	0.124801
Extended Quadratic Penalty QP1 Function	9000	15	57	0.0953941	18	63	0.124801

Step 2. Compute $f(x_k + p_k)$ and $\rho_k = (f(x_k) - f(x_k + p_k)) / (m_k(x_k) - m_k(x_k + p_k))$. If $\eta_0 \leq \rho_k$, then set $x_{k+1} = x_k + \alpha_k p_k$; otherwise, set $x_{k+1} = x_k$.

Step 3. Set

$$\Delta_{k+1} = \begin{cases} \zeta \|p_k\| & \text{if } \rho_k < \eta_0 \\ \Delta_k & \text{if } \eta_0 < \rho_k < \eta_1 \\ \max[\zeta \|p_k\|, \Delta_k] & \text{if } \eta_1 \leq \rho_k. \end{cases} \quad (57)$$

Increment k by one, and go to Step 1.

Trust region methods generate steps with the help of a quadratic model of the objective function, but they use this model in different ways from line search methods. Trust region methods define a region around the current iterate within which the trust the model to be an adequate representation of the objective function and then choose the step to be the approximate minimizer of the model in this region. The size of the trust region is critical to the effectiveness of each step. There are four parameters found in a trust region update process (57); namely, $\eta_0, \eta_1, \zeta, \varsigma$ are used to adjust the size of the trust region radius. The values are arbitrary and that much better options are available. The classical parameter values

$$s^c = (0.25, 0.75, 0.5, 2) \quad (58)$$

are recommended in the literature [30], but they may not be the best choice if we consider the total CPU time or the

overall number of function evaluations which was necessary to solve a set of optimization problems. This is a derivative-free optimization problem. In this section, an objective is to identify four optimal parameters found in a trust-region update using the proposed Algorithm 4 in this paper. Next we describe the specific representation of this parameter estimation problem.

Let P be a set of optimization problems, $\Omega = \{s \in R^4 \mid 0 \leq \eta_0 < \eta_1 < 1 \text{ and } 0 < \zeta < 1 < \varsigma\}$, $\tau_i(s)$ be the CPU time which is necessary to solve the problem $p_i \in P$ with the parameter values s , and $\phi_i(s)$ be the number of function evaluations. Thus the small-dimensional nondifferentiable optimization problem measuring by the overall CPU time $\psi^{CPU}(s)$ or the overall number of function evaluations $\psi^{feval}(s)$ is written as

$$\begin{aligned} \min_{p \in \Omega} \psi^{CPU}(s) &= \sum_{i \in P} \tau_i(s) \text{ or} \\ \min_{p \in \Omega} \psi^{feval}(s) &= \sum_{i \in P} \phi_i(s) \end{aligned} \quad (59)$$

We consider to minimize the total computing time $\psi^{CPU}(\cdot)$. The test problems in P and their dimension are list in Table 3. The initial point chosen for Algorithm 4 is s^c . Thus, the best set of parameters given by Algorithm 4 is

$$s^* = \{0.238721, 0.923489, 0.352533, 2.304294\}. \quad (60)$$

Table 3 shows the results on the test problems. Timings $\tau_i(\cdot)$ are in seconds. The measure $\phi_i(\cdot)$ is the number of

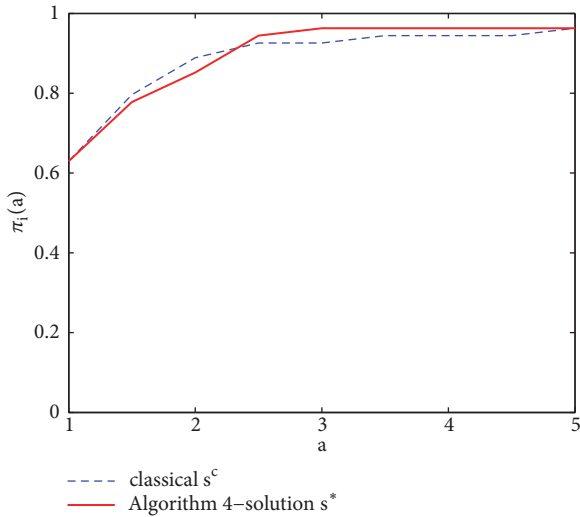


FIGURE 3: Profile comparing the CPU time required performance when using different parameter values.

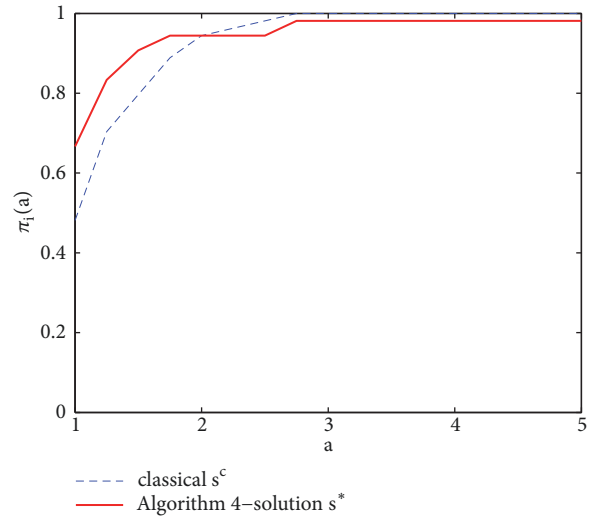


FIGURE 4: Profile comparing the total numbers of function evaluations performance when using different parameter values.

function evaluations in each case. Failures occur when $\phi_i(\cdot) = 1001$. Table 3 demonstrates that this strategy allowed the improvement if the truth initial value from $\psi^{CPU}(s^c) = 71.04$ to $\psi^{CPU}(s^*) = 64.4$ seconds, i.e., an improvement of approximately 10.31 %. The total number of function evaluations is improved from $\psi^{feval}(s^c) = 14221$ to $\psi^{feval}(s^*) = 12752$, i.e., an improvement of approximately 11.52 %.

The function which characterizes the performance profile of parameter values s^c and s^* is the same as [2]. The performance of the computing time $\tau_i(\cdot)$ required to solve each optimization problem in the problem set P , with the parameter values s^c , can be visualized in the profile of Figure 3 which compares the CPU time with the parameter values s^* . The profile of Figure 4 presents a similar comparison, using the number of function evaluations. Both Figures 3 and 4 illustrate that Algorithm 4 has advantage in solving nondifferentiable optimization problems.

5. Conclusion

This paper proposes a derivative-free trust region algorithm with nonmonotone filter technique for bound constrained optimization.

- (i) This algorithm is mainly designed to solve the unavailable derivatives optimization problems in engineering. The proposed algorithm possesses the trust region property and adopts nonmonotone filter technique for bound constrained optimization.
- (ii) The global convergence is provided under the definition of fully linear model. The sufficient descent property is able to make the objective function value descend, and then the iteration sequence $\{x_k\}$ converges to the global limit point if the problems are convex.

- (iii) The preliminary numerical results compared with traditional quasi-Newton method and conjugate gradient method turn out the proposed algorithm is feasible for those of the unavailable derivative functions. Large-scale problems are done by the given problems, which shows that the new algorithms are very effective.

- (iv) Finally, optimal parameters with respect to overall computational time on a set of test problems are identified. We use the proposed algorithm to get a best choice of parameter values which differ from traditionally used values and compare the numerical results affected by two different parameters from the two aspects of the CPU time and the number of function evaluations.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the National Science Foundation of China (Grant nos. 11626037, 11526036, and 11601014); the 13th Five-Year Science and Technology Project of Education Department of Jilin Province (Grant nos. JJKH20170036KJ, JJKH20160046KJ); the Innovation Talent Training Program of Science and Technology of Jilin Province of China (Grant no. 20180519011JH); the Scientific and Technological Planning Project of Jilin Province of China (Grant nos.

TABLE 3: Numerical results of the parameter estimation.

Problem name	n	$\tau_i(s^c)$	$\tau_i(s^*)$	$\phi_i(s^c)$	$\phi_i(s^*)$
3PK	30	0.14	0.04	71	55
ARGLINA	200	0.64	0.34	3	3
BIGGS6	6	0.12	0.02	24	23
BOX3	3	0.02	0.02	6	6
BROWNAL	200	0.12	0.17	4	7
BROWNBS	2	0.22	0.32	23	27
BROWNDEN	4	0.02	0.02	10	9
CHNROSNB	50	0.32	0.43	66	70
CLIFF	2	0.20	0.02	27	19
CUBE	2	0.02	0.02	45	40
DECONVU	61	1.34	0.93	90	38
DENSCHND	3	0.55	0.22	31	29
DIXMAANK	15	0.10	0.02	9	9
DJTL	2	0.02	0.02	168	147
ERRINROS	50	0.22	0.11	71	73
GENROSE	500	23.37	19.55	461	413
GROWTHLS	3	0.82	0.47	179	157
GULF	3	0.03	0.02	54	49
HAIRY	2	0.04	0.09	78	51
HEART6LS	6	0.25	0.25	1001	1001
HEART8LS	8	0.25	0.18	259	178
HIELOW	3	3.09	1.29	11	6
HIMMELBF	4	0.44	0.62	201	174
HUMPS	2	0.44	0.41	1001	613
LOGHAIRY	2	0.44	0.44	1001	1001
MANCINO	100	21.33	16.79	20	24
MARATOSB	2	0.03	0.37	1001	1001
MEYER3	3	0.42	0.49	1001	472
OSBORNEA	5	0.06	0.04	69	77
OSBORNEB	11	0.11	0.16	20	53
PALMERIC	8	0.54	0.94	1001	874
PALMERID	7	0.02	0.02	49	69
PALMER2C	8	0.71	0.71	1001	1001
PALMER3C	8	0.31	0.26	646	983
PALMER4C	8	0.03	0.06	80	548
PALMER6C	8	0.07	0.15	229	145
PALMER7C	8	0.36	0.84	1001	1001
PALMER8C	8	0.55	1.09	268	233
PENALTY2	200	0.99	0.77	13	15
PFITILS	3	0.23	0.65	625	368
PFIT2LS	3	0.01	0.37	242	176
PFIT3LS	3	0.27	0.49	275	377
PFIT4LS	3	0.21	0.43	510	269
SENSORS	100	8.77	11.77	22	20
SISSER	2	0.02	0.02	10	12
SNAIL	2	0.06	0.06	75	81
TOINTGOR	50	0.08	0.10	7	9
TOINTPSP	50	0.05	0.05	16	20
TOINTQOR	50	0.02	0.02	16	10

TABLE 3: Continued.

Problem name	n	$\tau_i(s^c)$	$\tau_i(s^*)$	$\phi_i(s^c)$	$\phi_i(s^*)$
VARDIM	200	0.09	0.22	26	71
VAREIGVL	50	0.03	0.03	20	15
VIBRBEAM	8	1.63	0.95	1001	533
WATSON	12	0.77	0.48	11	4
YFITU	3	0.05	0.05	72	93
Total	ψ	71.04	64.40	14221	12752

20160520108JH, 20170101037JC); the PhD Start-Up Fund of Natural Science Foundation of Beihua University and the Youth Training Project Foundation of Beihua University (Grant no. 2017QNJJL10).

References

- [1] S. Nie, T. Gao, and Q. Ji, "An information theoretic feature selection framework based on integer programming," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, pp. 3584–3589, Cancun, Mexico, December 2016.
- [2] C. Audet and D. Orban, "Finding optimal algorithmic parameters using derivative-free optimization," *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 642–664, 2006.
- [3] N. J. Higham, "Optimization by direct search in matrix computations," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 2, pp. 317–333, 1993.
- [4] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2nd edition, 2002.
- [5] A. Booker, P. Frank, Dennis Jr., D. Moore, and D. Serafini, "Managing surrogate objectives to optimize a helicopter rotor design - Further experiments," in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, Mo, USA, 1998.
- [6] A. J. Booker, J. Dennis, P. D. Frank, D. B. Serafini, and V. Torczon, "Optimization using surrogate objectives on a helicopter test example," in *Computational Methods for Optimal Design and Control*, J. T. Borggaard, J. Burns, E. Cliff, and S. Schreck, Eds., vol. 24, pp. 49–58, Birkhäuser, Boston, Mass, USA, 1998.
- [7] R. Duvigneau and M. Visonneau, "Hydrodynamic design using a derivative-free method," *Structural and Multidisciplinary Optimization*, vol. 28, no. 2-3, 2004.
- [8] A. R. Conn, K. Scheinberg, and L. N. Vicente, "Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 387–415, 2009.
- [9] X. Fang and Q. Ni, "A new derivative-free conjugate gradient method for large-scale nonlinear systems of equations," *Bulletin of the Australian Mathematical Society*, vol. 95, no. 3, pp. 500–511, 2017.
- [10] H. Ghanbari and K. Scheinberg, "Black-Box optimization in machine learning with trust region based derivative free algorithm," *Industrial and Systems Engineering*, 2017.
- [11] H. Zhang, A. R. Conn, and K. Scheinberg, "A derivative-free algorithm for least-squares minimization," *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 3555–3576, 2010.
- [12] H. Zhang and A. R. Conn, "On the local convergence of a derivative-free algorithm for least-squares minimization,"

- Computational optimization and applications*, vol. 51, no. 2, pp. 481–507, 2012.
- [13] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, Society for Industrial and Applied Mathematics, Mathematical Programming Society, Philadelphia, Pa, USA, 2009.
- [14] J. Gao and D. Zhu, “An affine scaling derivative-free trust region method with interior backtracking technique for bounded-constrained nonlinear programming,” *Journal of Systems Science & Complexity*, vol. 27, no. 3, pp. 537–564, 2014.
- [15] J. Gao and J. Cao, “A class of derivative-free trust-region methods with interior backtracking technique for nonlinear optimization problems subject to linear inequality constraints,” *Journal of Inequalities and Applications*, vol. 2018, article 108, 2018.
- [16] T. F. Coleman and Y. Li, “An interior trust region approach for nonlinear minimization subject to bounds,” *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 418–445, 1996.
- [17] C. Kanzow and A. Klug, “On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints,” *Computational optimization and applications*, vol. 35, no. 2, pp. 177–197, 2006.
- [18] M. Ulbrich, “Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems,” *SIAM Journal on Optimization*, vol. 11, no. 4, pp. 889–917, 2001.
- [19] R. Fletcher and S. Leyffer, “Nonlinear programming without a penalty function,” *Mathematical Programming*, vol. 91, no. 2, pp. 239–269, 2002.
- [20] R. Fletcher, N. I. Gould, S. Leyffer, P. L. Toint, and A. Wächter, “Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming,” *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 635–659, 2002.
- [21] R. Fletcher, S. Leyffer, and P. L. Toint, “On the global convergence of a filter—SQP algorithm,” *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 44–59, 2002.
- [22] C. C. Gonzaga, E. Karas, and M. Vanti, “A globally convergent filter method for nonlinear programming,” *SIAM Journal on Optimization*, vol. 14, no. 3, pp. 646–669, 2003.
- [23] A. Wächter and L. T. Biegler, “Global and local convergence of line search filter methods for nonlinear programming,” Tech. Rep. CAPD B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Penn, USA, 2001.
- [24] N. I. Gould, C. Sainvitu, and P. . Toint, “A filter-trust-region method for unconstrained optimization,” *SIAM Journal on Optimization*, vol. 16, no. 2, pp. 341–357, 2005.
- [25] C. Sainvitu and P. L. Toint, “A filter-trust-region method for simple-bound constrained optimization,” *Optimization Methods & Software*, vol. 22, no. 5, pp. 835–848, 2007.
- [26] P. L. Toint, “Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints,” *Mathematical Programming*, vol. 77, no. 1, Series A, pp. 69–94, 1997.
- [27] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, Springer, New York, NY, USA, 1981.
- [28] C. A. Floudas et al., *Handbook of Test Problems in Local And Global Optimization*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.
- [29] K. Schittkowski, *More Test Examples for Nonlinear Programming Codes*, vol. 282 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, Germany, 1987.
- [30] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 2006.
- [31] E. D. Dolan and J. J. Moré, “Benchmarking optimization software with performance profiles,” *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.
- [32] G. Yuan and W. Hu, “A conjugate gradient algorithm for large-scale unconstrained optimization problems and nonlinear equations,” *Journal of Inequalities and Applications*, vol. 2018, article 113, 2018.



Hindawi

Submit your manuscripts at
www.hindawi.com

