Research Article

A Derivative-Free Affine Scaling LP Algorithm Based on Probabilistic Models for Linear Inequality Constrained Optimization

Jingbo Zhao,1,2 Peng Wang,1,2 and Detong Zhu3

1Key Laboratory of the Ministry of Education, Hainan Normal University, Haikou, Hainan 570203, China
2Mathematics and Statistics College, Hainan Normal University, Haikou, Hainan 570203, China
3Mathematics and Science College, Shanghai Normal University, Shanghai 200234, China

Correspondence should be addressed to Peng Wang; pengwang621@163.com

Received 6 December 2021; Accepted 12 February 2022; Published 27 March 2022

Academic Editor: Erik Cuevas

Copyright © 2022 Jingbo Zhao 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.

In this paper, a derivative-free affine scaling linear programming algorithm based on probabilistic models is considered for solving linear inequality constrained optimization problems. The proposed algorithm is designed to build probabilistic linear polynomial interpolation models using only \( n+1 \) interpolation points for the objective function and reduce the computation cost for building interpolation function. We build the affine scaling linear programming methods which use probabilistic or random models and affine matrix within a classical linear programming framework. The backtracking line search technique can guarantee monotone descent of the objective function, and by using this technique, the new iterative points are located within the feasible region. Under some reasonable conditions, the global and local fast convergence of the algorithm is shown, and the results of numerical experiments are reported to show the effectiveness of the proposed algorithm.

1. Introduction

1.1. Problem Description Motivation. In this paper, we consider the following nonlinear optimization problem with linear inequality constraints:

\[
\begin{align*}
(P) & \min f(x), \\
& \text{s.t. } Ax \geq b,
\end{align*}
\]

where \( f(x) \) is the general nonlinear twice continuously differentiable functions, but none of their first-order or second-order derivatives is explicitly available. \( \Omega = \{x|Ax \geq b\} \) is the feasible region where \( A = [a_1^T, a_2^T, \ldots, a_l^T]^T \in \mathbb{R}^{\times n} \). Moreover, it is assumed that the interior of the feasible region \( \text{int}\Omega = \{x|Ax > b\} \) is not empty.

The focus of this paper is the analysis of a numerical scheme that utilizes affine scaling randomized models to minimize nonlinear functions with linear inequality constraints. The motivation comes from the algorithm for minimization of so-called black-box functions where values are computed. For such problems, function evaluations are costly and derivatives are typically unavailable and cannot be approximated. Hence, the derivative-free optimization is developed. Moreover, the list of applications including molecular geometry optimization, circuit design, groundwater community problems, medical image registration, dynamic pricing, and aircraft design [1] is diverse and growing. In recent years, some people developed derivative-free algorithms for solving unconstrained and constrained minimization problems. Kolda et al. [2] presented a new generating set search approach for minimizing functions subject to linear constraints. One of their main contributions is a new condition to define the set of conforming search directions that admits several computational advantages. Zhang and Conn [3] developed a framework for a class of derivative-free trust region algorithms for the least-squares minimization. These algorithms are designed to take
advantage of the problem structure by building polynomial interpolation models for each function in the least-squares minimization. Gratton et al. [4] considered an implementation of a recursive model-based active-set trust-region method for solving bound-constrained nonlinear non-convex optimization problems without derivatives using the technique of self-correcting geometry proposed in [5]. Kolda et al. [2] analyzed the oracle complexity of first-order and derivative-free algorithms for smooth non-convex minimization. Billups et al. [6] proposed a derivative-free algorithm for optimizing computationally expensive functions with computational error. The algorithm used the trust region regression method which used weighted regression to obtain more accurate model functions at each trust region iteration. Bueno et al. [7] introduced a new method for solving a constrained optimization problem in which the derivatives of the constraints were available but the derivatives of the objective function were not. The method is based on the inexact restoration framework, and the iteration was divided in two phases. The first phase considered improving the feasibility of the constraints, and the second phase minimized a suitable objective function subjective to a linear approximation of the constraints which must be solved using derivative-free methods. Wild and Shoemaker [8] introduced the global convergence of radial basis function trust region algorithms for derivative-free optimization. Their results extend the recent work of Conn et al. [9] to fully linear models that have a nonlinear term. Xue and Sun [10] proposed a derivative-free trust region algorithm for constrained minimization problems with separable structure, where derivatives of the objective function are not available and cannot be directly approximated. They constructed a quadratic interpolation model of the objective function around the current iterate and used the filter technique to ensure the feasibility and optimality of the iterative sequence.

There is a variety of evidence supporting the claim that randomized models can yield both practical and theoretical benefits for deterministic optimization. As an example, the randomized coordinate descent method for large-scale convex deterministic optimization proposed in [11] yields better complexity results than cyclical coordinate descent. Most contemporary randomized methods generate random directions along which all that may be required is some minor level of descent in the objective function. Within direct search, the use of random positive spanning sets has also been recently investigated [12, 13] with gains in performance and convergence theory for nonsmooth problems. Bandeira et al. [14] considered the use of a probabilistic or random model within a classical trust region framework for the optimization of deterministic smooth general nonlinear functions.

1.2. Contribution and Structure of the Paper. In order to solve the linear inequality constrained optimization problem, the derivative-free affine scaling LP algorithm based on probabilistic models is introduced in this paper. Compared with the traditional derivative-free algorithm, the new derivative-free algorithm proposed in this paper has the following contributions:

1. The randomized numerical algorithm is designed in this paper providing accurate enough approximations such that in each iteration, a sufficiently improving step is produced with higher probability than traditional random schemes.
2. The polynomial interpolation models are built only using \( n + 1 \) interpolation points, which are far less than \( (n + 1)(n + 2)/2 \) interpolation points of building the quadratic interpolation function. The computational cost of building the interpolation function using \( n + 1 \) interpolation point is less than using \( (n + 1)(n + 2)/2 \) interpolation point.
3. The backtracking line search technique is used such that the linear programming subproblem is solved only once for obtaining the search direction, and the total computational effort for completing one iteration is less than the traditional derivative-free algorithm. Meanwhile, this technique also ensures that each iteration point is a feasible interior point.

This paper is organized as follows. In Section 2, we introduce the first-order affine scaling linear programming method based on probabilistic models. The algorithm for solving linear inequality constrained optimization problems is given in Section 3. In Section 4, we give the global convergence of the algorithm. The fast local convergence of the algorithm is given in Section 5. Some numerical results are given in Section 6. Finally, we give some conclusions in Section 7.

2. First-Order Affine Scaling LP Method Based on Probabilistic Models

In this section, we first introduce the construction method of a stochastic fully linear model and then give the affine scaling linear programming method based on probabilistic models. In the classical trust-region method, there need \( (n+1)(n+2)/2 \) interpolation points to build a quadratic interpolation model \( m_k \) of function \( f \) with in the ball \( B(x_k, \Delta_k) \) where \( x_k \) is the center point and \( \Delta_k \) is the radius of the ball. It is obvious that the number of interpolation points is much larger than the dimension of problem \( (P) \) when the dimension increases. This brings great difficulty to computer in storage and computation. In this section, we will introduce an affine linear programming model based on probabilistic models which need only \( n + 1 \) interpolation points. It means that we will build random fashion polynomial interpolation models. Firstly, we will give the models and some assumptions of the models.

2.1. Probabilistically Fully Linear Models. We propose the polynomial \( m(x) \in P_d^{m} \) as proposed in [15], which is \( (k_{e_f}, k_{e_0}) \) fully linear, and we consider the following linear models, written in the form:
\[ m(x_k + s) = m(x_k) + g^T_k s, \]  
where \( g_k = \nabla m(x_k) \). The following definition shows the accuracy of the model \( m(x_k) \).

**Definition 1** (see [14]). We say that a function \( m \) is a \((\kappa_{eg}, \kappa_{ef})\) fully linear model of \( f \) on \( B(x_k, \Delta_k) \) if, for every \( s \in B(0, \Delta_k) \),

\[
\begin{align*}
\| \nabla f(x_k + s) - m(x_k + s) \| & \leq \kappa_{eg} \Delta_k, \\
\| f(x_k + s) - m(x_k + s) \| & \leq \kappa_{ef} \Delta_k^2.
\end{align*}
\]

**Definition 2** (see [14]). We say that a sequence of random models \( \{ M_k \} \) is probabilistically \((\kappa_{eg}, \kappa_{ef})\) fully linear for a corresponding sequence \( \{ B(x_k, \delta_k) \} \) if the events

\[ P(\mathcal{S}_k | F_k^M) \geq p, \]

where \( F_k^M = \sigma(M_0, M_1, \ldots, M_{k-1}) \) is \( \sigma \)-algebra generated by \( M_0, M_1, \ldots, M_{k-1} \). Furthermore, if \( p \geq (1/2) \), then we say that the random models are probabilistically \((\kappa_{eg}, \kappa_{ef})\) fully linear.

In Definition 2, \( X_k \) and the interpolation radii \( \delta_k \) are the random variables defined over the \( \sigma \)-algebra generated by \( M_0, M_1, \ldots, M_{k-1} \). \( M_k \) depends on \( X_k \) and \( \delta_k \), and hence, \( M_k \) depends on \( M_0, M_1, \ldots, M_{k-1} \).

**2.2. The Affine Scaling Linear Programming Method Based on Probabilistic Models.** Using the scaling matrix \( D_k \) in [15], the form is as follows:

\[ D_k = \text{diag}[A x_k - b]. \]  

(5)

According to the same convergence theory, for any \( x_k \in \text{int}(\Omega) \), the augmented affine scaling linear programming subproblem at the \( k \)-th iteration is defined as

\[
\begin{align*}
(L_k) \min \phi_k(d) & = g^T_k d, \\
\text{s.t.} - \Delta_k & \leq \left( D_k^{(1/2)} A \right) d \leq \Delta_k, \quad i = 1, 2, \ldots, m, \\
- \Delta_k & \leq (d)_i \leq \Delta_k, \quad i = 1, 2, \ldots, n,
\end{align*}
\]

where \( g_k = \nabla m(x_k) \).

Same as reference [15], we have that a “good” decrease of the quadratic objective function in \( (L_k) \) along the gradient \( g_k \) can lead to dual feasibility:

\[
\lim_{k \to \infty} g_k - A^T \lambda_k = 0, \quad \text{and} \quad \lim_{k \to \infty} \| D_k^{(1/2)} \lambda_k \| = 0.
\]

(7)

**3. The Algorithm**

In Section 2, we introduced the affine scaling LP method based on probabilistic models. In this section, we define the specific steps of the derivative-free affine scaling linear programming algorithm based on probabilistic models for solving problem (P).

**3.1. Algorithm**

**3.1.1. Initialization Step.** Fix the positive parameter \( \eta_1, \eta_2, \gamma, \Delta_{\text{max}} \) with \( \gamma > 1 > \eta_2 > \eta_1 > 1/2 \), and \( \omega \in (0, 1) \). Select initial \( k = 0, \Delta_0 = \Delta_{\text{max}} \), and \( x_0 \in \text{int}(\Omega) \).

**3.1.2. Main Step**

(1) Approximate the function \( f \) in \( B(x_k, \Delta_k) \) by the linear function \( m \).

(2) Compute the affine scaling linear programming subproblem \( (L_k) \) and obtain the step \( d_k \).

(3) Choose \( \alpha_k = 1, \omega, \omega^2, \ldots \), until the following inequality is satisfied:

\[
f(x_k + \alpha_k d_k) \leq f(x_k) + \alpha_k g^T_k d_k \text{ with } x_k + \alpha_k d_k \in \Omega.
\]

(8)

(4) Let

\[ h_k = \begin{cases} \alpha_k d_k, & \text{if } x_k + \alpha_k d_k \in \text{int}(\Omega), \\
\theta_k \alpha_k d_k, & \text{otherwise,}
\end{cases} \]

(9)

where \( \theta_k \in (\theta_1, 1) \), for some \( 0 < \theta_1 < 1 \) and \( \theta_k - 1 = O(\|d_k\|) \).

(5) If \( \alpha_k \geq \eta_1 \), then set \( x_{k+1} = x_k + h_k \) and

\[
\Delta_{k+1} = \begin{cases} \gamma^{-1} \Delta_k, & \text{if } \| g_k^{T \Gamma_k} \|^{(1/2)} < \eta_2 \Delta_k, \\
\min\{\gamma \Delta_k, \Delta_{\text{max}}\}, & \text{if } \| g_k^{T \Gamma_k} \|^{(1/2)} \geq \eta_2 \Delta_k.
\end{cases}
\]

(10)

(6) If \( \alpha_k < \eta_1 \), then set \( x_{k+1} = x_k \) and \( \Delta_{k+1} = \gamma^{-1} \Delta_k \). Increase \( k \) by one and go to step 1.

**Remark 1.** Each realization of the algorithm defines a sequence of realizations for the corresponding random variables, in particular, \( m_k = M_k(\omega_k) \), \( x_k = X_k(\omega_k) \), and \( \Delta_k = \delta_k(\omega_k) \).

**Remark 2.** Similarly, the definition of \( \alpha_k \) in [15], for the search direction \( d_k \), we have

\[
\alpha_k \leq \min \left\{ \frac{a_i^T x_k - b_i}{a_i^T d_k}, \frac{a_i^T x_k - b_i}{a_i^T d_k} > 0, \quad i = 1, 2, \ldots, l \right\},
\]

(11)

with \( \alpha_k = +\infty \) if \( - (a_i^T x_k - b_i/a_i^T d_k) \leq 0 \) for all \( i \). A key property of the scalar \( \alpha_k \) is that an arbitrary step \( \alpha_k d_k \) to the point \( x_k + \alpha_k d_k \) does not violate any linear inequality constraints. To see this, we first observe that if \( - (a_i^T x_k - b_i/a_i^T d_k) \leq 0 \) for some \( i = 1, 2, \ldots, m \), and hence \( a_i^T x_k - b_i > 0 \) implies \( a_i^T d_k > 0 \). Therefore, for all \( \alpha_k \in (0, +\infty) \),
\[ a_i^T (x_k + \alpha_k d_k) - b_i = a_i^T x_k - b_i + \alpha_k a_i^T d_k > 0, \]

which means that the \( i \)-th linear strict inequality constraint holds.

If \( \alpha_k = (a_i^T x_k - b_i) / a_i^T d_k \) for some \( i \), hence \( a_i^T x_k - b_i > 0 \) implies \( a_i^T d_k < 0 \). We have that from \( \alpha_k = (a_i^T x_k - b_i) a_i^T d_k \),

\[ a_i^T x_k - b_i \leq -\alpha_k a_i^T d_k. \] (13)

Hence, (12) and (13) mean that no matter what cases the inequality \( a_i^T (x_k + \alpha_k d_k) > b_i \) for all \( i = 1, 2, \ldots, I \) holds.

### 4. Global Convergence

In this section, we will introduce the global convergence of our algorithm, which means that the algorithm proposed in this paper converges to the first-order critical points. For the purpose of proving the global convergence of the algorithm, we assume that the initial iterate point \( x_0 \) is given. Then, all the subsequent iterates belong to the level set

\[ L(x_0) = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \}. \] (14)

However, the failed iterates may lie outside this set. In the setting considered in this paper, all potential iterates are restricted to the region

\[ L_{\text{end}}(x_0) = L(x_0) \cup_{x \in L(x_0)} B(x, \Delta_{\text{max}}), \]

where \( \Delta_{\text{max}} \) is the upper bound on the size of the interpolation radius, as imposed by the algorithm, and we define \( \text{conv}(L_{\text{end}}(x_0)) \) to be the convex hull of \( L_{\text{end}}(x_0) \).

Next, we also give the following assumption for the global convergence of algorithm.

**Assumption 1.** Suppose \( x_0 \) and \( \Delta_{\text{max}} \) are given. We assume that \( f \) is continuously differentiable in an open set containing the set \( L_{\text{end}}(x_0) \) and that \( \nabla f \) is Lipschitz continuous on \( \text{conv}(L_{\text{end}}(x_0)) \) with constant \( \kappa_{L,f} \), i.e., for all \( x, y \in L_{\text{end}}(x_0) \), we have that

\[ \| \nabla f(x) - \nabla f(y) \| \leq \kappa_{L,f} \| x - y \|. \] (16)

We also assume that \( f \) is bounded from below on \( L(x_0) \).

**Assumption 2.** Suppose the level set \( L(x_0) \) is bounded.

Based on solving the augmented linear programming subproblem \( (L_k) \), the following lemma shows that \( g_k^T d_k < -\| g_k^T \Gamma_k \|^{1/2} \Delta_k \) holds.

**Lemma 1.** For every realization of the algorithm, let \( d_k \) be the solution of subproblem \( (L_k) \); then,

\[ g_k^T d_k < -\| g_k^T \Gamma_k \|^{1/2} \Delta_k. \] (17)

**Proof.** Let

\[ s_k = \frac{\Gamma_k}{\| g_k^T \Gamma_k \|^{1/2}} \Delta_k. \] (18)

Since

\[ \frac{\left\| D_k^{(1/2)} \Gamma_k \right\|}{\| g_k^T \Gamma_k \|^{1/2}} \Delta_k \leq \frac{\| D_k^{(1/2)} \Gamma_k \|}{\| g_k^T \Gamma_k \|^{1/2}} \Delta_k. \] (19)

Because \( \| g_k^T \Gamma_k \|^{1/2} \leq \| \Gamma_k \|^{1/2} + \| D_k^{(1/2)} \Gamma_k \|^{1/2} \), we have that

\[ \frac{\left\| D_k^{(1/2)} \Gamma_k \right\|}{\| g_k^T \Gamma_k \|^{1/2}} \Delta_k \leq \Delta_k. \] (20)

Hence, we have that \( -\Delta_k \leq (D_k^{(1/2)} \Gamma_k) \Delta_k \leq \Delta_k \). Because \( (1/ \| \Gamma_k \|) \Delta_k = (1/ \| \Gamma_k \|) \sqrt{\| \Gamma_k \|^2 + \| D_k^{(1/2)} \Gamma_k \|^2} \), we have that \( s_k \) is a feasible solution of subproblem \( (L_k) \).

Because \( d_k \) is the solution of subproblem \( (L_k) \), we have that

\[ \bar{\phi}(d_k) \leq \bar{\phi}(s_k). \] (21)

The value of \( \bar{\phi}(s_k) \) is

\[ \bar{\phi}(s_k) = g_k^T s_k \]

\[ = \frac{g_k^T \Gamma_k}{\| g_k^T \Gamma_k \|^{1/2}} \Delta_k. \] (22)

According to the definition of \( \Gamma_k \), we have that

\[ g_k^T \Gamma_k = -g_k^T (g_k - A^T \mu_k) \]

\[ = -g_k^T (g_k + A^T (AA^T + D_k)^{-1} - A^T \mu_k) \]

\[ = -g_k^T \Gamma_k, \]

where \( \mu_k = - (AA^T + D_k)^{-1} A g_k \).

By (21)–(23), we have that

\[ \bar{\phi}(d_k) = g_k^T d_k \leq \bar{\phi}(s_k) \leq -g_k^T \Gamma_k \Delta_k. \] (24)

The following lemma shows that every iterative point of the algorithm is a feasible solution of problem \( (P) \), i.e., \( x_k + \alpha_k d_k \in \Omega \).

**Lemma 2.** Let \( \{ x_k \} \) be a sequence generated by the algorithm; then, we have that there exists \( \alpha_k > 0 \) such that \( x_k + \alpha_k d_k \in \Omega \).

**Proof.** According to the definition of \( D_k \) and \( -\Delta_k \leq (D_k^{(1/2)} \alpha_k) \Delta_k \), we have that

\[ (b - Ax_k) \Delta_k \leq (D_k^{(1/2)} \alpha_k) \Delta_k \leq (b - Ax_k) \Delta_k. \] (25)

Because \( (Ax_k - b) = a_i^T x_k - b_i \) and \( (D_k^{(1/2)} \alpha_k) = a_i^T d_k \), we have that

\[ (b - a_i^T x_k) \Delta_k \leq (b - a_i^T x_k) \Delta_k \leq (b - a_i^T x_k) \Delta_k. \] (26)
If \( a_i^T d_k \leq 0 \), we have that \( a_i^T (x_k + d_k) - b_i \leq 0 \), i.e., \( \alpha_k = 1 > 0 \). Otherwise, if \( a_i^T d_k > 0 \), then from (26), for \( i \in \{1, 2, \ldots, l\} \), we have that
\[
\frac{a_i^T x_k - b_i}{a_i^T d_k} \geq \frac{1}{\Delta_k} \geq \frac{1}{\Delta_{\max}} > 0. \tag{27}
\]

By (11) and (27), we have that there exists \( \alpha_k > 0 \) such that
\[
A(x_k + \alpha_k d_k) \leq b \text{ holds in the } k\text{-th iteration, i.e., there exists } \alpha_k > 0 \text{ such that } x_k + \alpha_k d_k \in \Omega.
\]

The following lemma states that the interpolation radius converges to zero regardless of the realization of the model sequence \([M_k]\) made by the algorithm.

**Lemma 3.** Suppose Assumptions 1 and 2 hold, for every realization of the algorithm, we have that
\[
limit_{k \to \infty} \Delta_k = 0. \tag{28}
\]

**Proof.** First, we can see that if \( \alpha_k < \eta_1 \) holds as \( k \to \infty \), then (28) holds according to the definition of \( \Delta_k \) and Step 6 of the algorithm. If \( \alpha_k \geq \eta_1 \) and \( |g_k^T \Gamma_k^{(1/2)}| < \eta_2 \Delta_k \) hold as \( k \to \infty \), equation (28) holds according to the definition of \( \Delta_k \) and Step 5 of the algorithm. Otherwise, according to Step 3 of the algorithm, we have that
\[
f(x_k) - f(x_k + \alpha_k d_k) \geq -\alpha_k \beta g_k^T d_k. \tag{29}
\]

By (17), we have that
\[
f(x_k) - f(x_k + \alpha_k d_k) \geq \alpha_k \beta \frac{g_k^T \Gamma_k^{(1/2)}}{\Delta_k}. \tag{30}
\]

Assumption 2 shows that \( f(x) \) is a bounded function; hence, \( f(x_k) - f(x_k + \alpha_k d_k) \to 0 \) as \( k \to \infty \). By (30), we know that \( \alpha_k \beta \frac{g_k^T \Gamma_k^{(1/2)}}{\Delta_k} \to 0 \) as \( k \to \infty \). If \( \alpha_k \to 0 \), then there exists a positive index \( \kappa_0 \) such that \( \alpha_k < \eta_1 \) for all \( k > \kappa_0 \), we have that (28) holds. Otherwise, we have that \( \alpha_k > 0 \) for all \( k \), if \( \lim_{k \to \infty} \Delta_k = 0 \), then the conclusion holds.

Otherwise, we assume that \( \Delta_k > 0 \) for all \( k \to \infty \), then (30) means that \( \lim_{k \to \infty} |g_k^T \Gamma_k^{(1/2)}| = 0 \) holds; according to the modified method of \( \Delta_k \) in Step 5, we have that \( \Delta_k \) will decrease and tend to zero, which is a contradiction. Hence, we have that (28) holds.

The following lemma shows that in the presence of sufficient model accuracy, a successful step will be achieved, provided that the interpolation radius is sufficiently small relative to the size of the model gradient.

**Lemma 4.** Suppose Assumptions 1 and 2 hold, if \( m \) is \((\kappa_{eg}, \kappa_{ef})\) fully linear on \( B(x_k, \Delta_k) \) and
\[
\Delta_k \leq \frac{\omega(1 - \beta) g_k^T \Gamma_k^{(1/2)}}{\eta_1 (\kappa_{Lf} + \kappa_{eg})}, \tag{31}
\]

where \( \kappa_{ef} \) is a positive constant such that \( \|\Gamma_k\| \leq \kappa_{ef} \), then at the \( k\)-th iteration \( \alpha_k \geq \eta_1 \).

**Proof.** Since \([A - D_k^{(1/2)}]\) has full row rank in the compact set \( L(x_k), \mu_k \) is bounded, there exists a constant \( \kappa_\mu > 0 \) such that \( \|\mu_k\| \leq \kappa_\mu \). By the definition of \( \Gamma_k \) and Lemma 2, we have that
\[
\|\Gamma_k\| = \| -g_k - A^T \mu_k \| \leq \| g_k \| + \| A \| \| \mu_k \| \\
\leq \kappa_{eg} + \kappa_{ef} \| A \| \leq \kappa_{Lg} + \kappa_{ef} \kappa_{ef}.
\]

Let \( \kappa_{ef} = \kappa_{Lg} + \kappa_{ef} \kappa_{ef} \), we have that \( \|\Gamma_k\| \leq \kappa_{ef} \).

By accepting the rule of the step \( \alpha_k d_k \), we have that
\[
 f(x_k + \alpha_k d_k) - f(x_k) > \beta \frac{\alpha_k}{\omega} \nabla m(x_k)^T d_k. \tag{33}
\]

Using the mean value theorem, we have the following equality:
\[
f(x_k + \alpha_k d_k) - f(x_k) = \frac{\alpha_k}{\omega} \nabla f \left( x_k + \frac{\alpha_k}{\omega} d_k \right)^T d_k \\
= \frac{\alpha_k}{\omega} \nabla f \left( x_k + \frac{\alpha_k}{\omega} d_k \right) - \nabla f(x_k) \right)^T d_k + \frac{\alpha_k}{\omega} \nabla f(x_k)^T d_k \\
= \frac{\alpha_k}{\omega} \nabla f \left( x_k + \frac{\alpha_k}{\omega} d_k \right) - \nabla f(x_k) \right)^T d_k \\
+ \frac{\alpha_k}{\omega} \left( \nabla f(x_k) - \nabla m(x_k)^T d_k + \frac{\alpha_k}{\omega} m(x_k)^T d_k. \right. \tag{34}
\]

with \( \xi_k \in (0, 1) \). Since \( \nabla f(x) \) is Lipschitz continuously differentiable with constant \( \kappa_{Lg, ef} \), we have that
\[
\left( \nabla f \left( x_k + \frac{\alpha_k}{\omega} d_k \right) - \nabla f(x_k) \right)^T d_k \leq \kappa_{Lg, ef} \frac{\alpha_k}{\omega} \| d_k \|_{\omega} \leq \kappa_{Lg, ef} \frac{\alpha_k}{\omega} \| \Delta_k \|. \tag{35}
\]
Because $m(x_k)$ is a $(\kappa_{cg}, \kappa_{cf})$ fully linear model of $f$ on $B(x_k, \Delta_k)$ and $(\alpha_k/\omega) < 1$, $m(x_k)$ is a $(\kappa_{cg}, \kappa_{cf})$ fully linear model of $f$ on $B(x_k, \alpha_k/\omega \Delta_k)$. According to Definition 1, we have that
\[
\left| \nabla f(x_k) - \nabla m(x_k) \right|^T d_k \leq \frac{\alpha_k}{\omega} \kappa_{cg} \Delta_k^2.
\] (36)

By (33), (34), (35), and (36), we have that
\[
-(1 - \beta) \frac{\alpha_k}{\omega} \nabla m(x_k)^T d_k < \left( \frac{\alpha_k}{\omega} \right)^2 \left( \kappa_{lf} + \kappa_{cf} \right) \Delta_k^2.
\] (37)

Dividing $(\alpha_k/\omega)$ in the two sides of (37) and noting $(1 - \beta) > 0$, we have that
\[
\alpha_k > -\frac{\omega(1 - \beta) \kappa_{cg} \Delta_k^2}{\left( \kappa_{lf} + \kappa_{cf} \right) \Delta_k^2} \geq \frac{\omega(1 - \beta) \kappa_{cg} \Delta_k^2}{\left( \kappa_{lf} + \kappa_{cf} \right) \Delta_k^2}.
\] (38)

By (33), we have that $\alpha_k \geq \eta_1$.

Next, we assume that the model used in the algorithm is probabilistically fully linear, and we will state an auxiliary result from the martingale literature that will be useful in our analysis.

**Theorem 1** (see [14]). Let $G_k$ be a submartingale, i.e., a sequence of random variables which, for every $k$, are integrable $E(G_k) < \infty$ and $E(G_k | F_{k-1}^G) \geq G_k$, where $F_k^G = \sigma(G_0, \ldots, G_{k-1})$ is the $\sigma$-algebra generated by $G_0, \ldots, G_{k-1}$, and $E(G_k | F_{k-1}^G)$ denotes the conditional expectation of $G_k$ given the past history of events $F_{k-1}^G$. We assume further that $|G_k - G_{k-1}| \leq M < \infty$ for every $k$. We consider the random events $C = \{ \lim_{k \to \infty} G_k \text{ exists and is finite} \}$ and $D = \limsup_{k \to \infty} G_k = \infty$. Then, $P(C \cup D) = 1$.

**Proof.** The theorem is a simple extension of Theorem 5.3.1 in [16].

Now, we will give the global convergence of the algorithm.

**Theorem 2.** Suppose that the model sequence $\{M_k\}$ is probabilistically $(\kappa_{cg}, \kappa_{cf})$ fully linear for some positive constants $\kappa_{cg}$ and $\kappa_{cf}$. Let $\{x_k\}$ be a sequence of random iterates generated by the algorithm. Then, almost surely,
\[
\liminf_{k \to \infty} \| \nabla f(x_k)^T T_k \| = 0.
\] (39)

**Proof.** According to the definition of $S_i$ in Definition 2, we use the following random walk in [14]:
\[
W_k = \sum_{i=1}^{k} \left( 2 \left( 1_{\tilde{S}_i} - 1 \right) \right),
\] (40)

where $1_{\tilde{S}_i} = 1$ if $\tilde{S}_i$ occurs, $1_{\tilde{S}_i} = 0$ otherwise. It is easily seen that $W_k$ is a submartingale. The proof can be found in Theorem 2 [14]. By Theorem 4.6, we can also see that the event $\{ \limsup_{k \to \infty} W_k = \infty \}$ holds almost surely. Suppose there exist positive constant $\epsilon$ and index $K_1$ such that
\[
P \left( \| \nabla f(x_k)^T T_k \| \geq \epsilon \right) > \epsilon.
\] (41)

For all $k \geq K_1$,
\[
\Delta_k < \left( \frac{\omega(1 - \beta) \kappa_{cg} \Delta_k}{\left( \kappa_{lf} + \kappa_{cf} \right) \Delta_k} \right) ^{\gamma}.
\] (42)

From the definition of fully linear models, we have that
\[
\| \nabla f(x_k) - g_k \| \leq \kappa_{cg} \Delta_k \to 0 \quad \text{as } k \to \infty.
\] (43)

According to Assumption 2, we have that $(A A^T + D_k)$ is a reversible matrix, and by $(A A^T + D_k) \mu_k = -A g_k$ and $(A A^T + D_k) \tilde{\mu}_k = -A \nabla f(x_k)$, we have that
\[
\| g_k - \bar{\mu}_k \| = \left( (A A^T + D_k)^{-1} A (g_k - \nabla f(x_k)) \right) \| g_k - \nabla f(x_k) \|
\leq \| (A A^T + D_k)^{-1} A \| \| g_k - \nabla f(x_k) \|
\leq \kappa_1 \kappa_{cg} \Delta_k,
\] (44)

where $\kappa_1$ is an upper bound of $\| (A A^T + D_k)^{-1} A \|$.

From (43) and (44), we know that $|\mu_k - \bar{\mu}_k| \to 0$. Hence, we have that
\[
\left\| g_k^T T_k - \nabla f(x_k)^T T_k \right\| \leq \left\| g_k^T T_k - g_k^T T_k + g_k^T T_k - \nabla f(x_k)^T T_k \right\|
\leq \left\| g_k^T (T_k - T_k) \right\| + \left\| T_k \right\| \| g_k - \nabla f(x_k) \|
\leq \left\| g_k \right\| \left\| T_k \right\| \| g_k - \nabla f(x_k) \|
\leq \| T_k \| \| g_k - \nabla f(x_k) \|.
\] (45)

Since $\left[ A, D_k \right]^{(1/2)}$ has full row rank in the compact set $L(x_0), \mu_k$ is bounded, there exists a constant $\kappa_{cg} > 0$ such that $\| g_k \| \leq \kappa_{cg}$. Similarly, to prove (32), we have that there exists a positive constant $\kappa_{cf}$ such that $\| T_k \| \leq \kappa_{cf}$. Combining (28), (43), (44), and (45), we can obtain that
\[
\left\| g_k^T T_k - \nabla f(x_k)^T T_k \right\| \to 0,
\] (46)

as $k \to \infty$. By (41) and (46), we obtain that
\[
\left\| g_k^T T_k - \nabla f(x_k)^T T_k \right\| \leq \frac{\epsilon}{2}
\] (47)

Hence, we have that
\[
\left\| g_k^T T_k \right\| \to (1/2) \geq \frac{\epsilon}{2} \geq \eta_2 \Delta_k.
\] (48)

Using Lemma 4 and (28), we obtain that $\alpha_k \geq \eta_1$ and
\[
\left\| g_k^T T_k \right\| \to (1/2) \geq \frac{\epsilon}{2} \geq \eta_2 \Delta_k.
\] (49)

By the construction of the algorithm, and the fact that $\Delta_k \leq (\Delta_{\max}/\gamma)$, we have that $\Delta_k \leq \gamma \Delta_k$. Similar to the proof of Theorem 2 in [14], we have that
\[
\liminf_{k \to \infty} \left\| \nabla f(x_k)^T T_k \right\|^{(1/2)} = 0,
\] (50)

almost surely.
Next, we show that
\[
\lim_{k \to \infty} \left| \nabla f(x_k)^T \tilde{T}_k \right|^{(1/2)} = 0, \quad (51)
\]
almost surely. Before stating and proving the main theorem, we will give the following lemmas.

**Lemma 5** (see [14]). Let \( \{Z_k\} \) be a sequence of nonnegative uniformly bounded random variables and \( \{B_k\} \) be a sequence of Bernoulli random variables such that
\[
P(B_k = 1)\sigma(B_1, B_2, \ldots, B_{k-1} \cdot \sigma(Z_1, Z_2, \ldots, Z_{k-1})) \geq 1/2 \quad (52)
\]
\[
\text{Let } P \text{ be the set of natural numbers } k \text{ such that } B_k = 1 \text{ and } N = N \setminus P, \text{ where } P \text{ and } N \text{ are the random sequences. Then,}
\]
\[
\Prob\left( \left\{ \sum_{i \in P} Z_i \right\} < \infty \cap \left\{ \sum_{i \in N} Z_i \right\} = \infty \right) = 0. \quad (53)
\]
**Proof.** The proof is similar to Lemma 1 in [14].

**Lemma 6** (see [14]). Let \( \{x_k\} \) and \( \{\Delta_k\} \) be sequences of random iterates and random trust region radii generated by the algorithm. We fix \( \varepsilon > 0 \) and define the sequence \( \{K_i\} \) consisting of the natural numbers \( k \) for which \( |\nabla f(x_k)^T \tilde{T}_k| > \varepsilon \), where \( K_i \) is a sequence of random variables. Then,
\[
\sum_{k \in K_i} \Delta_k < \infty, \quad (54)
\]
almost surely.

**Proof.** The proof is similar to Lemma 2 in [14].

Next, we will prove that \( \lim_{k \to \infty} |\nabla f(x_k)^T \tilde{T}_k| = 0 \) holds.

**Theorem 3.** Suppose Assumptions 1 and 2 hold, we suppose that the model sequence \( \{M_k\} \) is probabilistically \( (\kappa_{eg}, \kappa_{ef}) \) fully linear for some positive constants \( \kappa_{eg} \) and \( \kappa_{ef} \). Let \( \{X_k\} \) be a sequence of random iterates generated by the algorithm; then,
\[
\lim_{k \to \infty} |\nabla f(x_k)^T \tilde{T}_k| = 0. \quad (55)
\]
**Proof.** We suppose that \( \lim |\nabla f(x_k)^T \tilde{T}_k| = 0 \) does not hold almost surely. Then, there exists \( \varepsilon > 0 \) such that \( P(|\nabla f(x_k)^T \tilde{T}_k| > 2\varepsilon) > \varepsilon \) holds for infinitely many \( k \). Let \( \{K_i\} \) be a subsequence of the iterations for which \( |\nabla f(x_k)^T \tilde{T}_k| > 2\varepsilon \). According to Theorem 3, we have that there exists a pair of integers \( (k', k'') \) such that \( 0 < k' < k'' \) and \( |\nabla f(x_k)^T \tilde{T}_k| \leq \varepsilon \), \( |\nabla f(x_{k+i})^T \tilde{T}_{k+i}| > \varepsilon \) and \( |\nabla f(x_{k+i})^T \tilde{T}_{k+i}| \leq 2\varepsilon \), hence, for any \( k \in (k', k'') \), we have \( \varepsilon < |\nabla f(x_k)^T \tilde{T}_k| \leq 2\varepsilon \). Now, we consider the sequence of these intervals \( \{(k_i, k_i')\} \). For which \( |\nabla f(x_k)^T \tilde{T}_k| > \varepsilon \) for \( k \in \{k_i\} \), where \( \{x_k\} \) and \( \{\Delta_k\} \) are the realizations of \( \{X_k\} \) and \( \{\delta_k\} \), respectively. Then, for any \( l \), we have that
\[
\varepsilon < |\nabla f(x_k)^T \tilde{T}_k| - |\nabla f(x_{k+i})^T \tilde{T}_{k+i}|
\]
\[
= \left| \sum_{j=k}^{k'-1} \left( |\nabla f(x_j)^T \tilde{T}_j| - |\nabla f(x_{j+i})^T \tilde{T}_{j+i}| \right) \right|
\]
\[
\leq \sum_{j=k}^{k'-1} |\nabla f(x_j)^T \tilde{T}_j - \nabla f(x_{j+i})^T \tilde{T}_{j+i}|
\]
\[
\leq \sum_{j=k}^{k'-1} \left( |\nabla f(x_j)^T \tilde{T}_j - \nabla f(x_{j+i})^T \tilde{T}_{j+i}| - |\nabla f(x_j)^T \tilde{T}_{j+i} - \nabla f(x_{j+i})^T \tilde{T}_{j+i}| \right)
\]
\[
\leq \sum_{j=k}^{k'-1} \left( |\nabla f(x_j) \tilde{T}_j - \nabla f(x_{j+i}) \tilde{T}_{j+i}| + |\tilde{T}_{j+i} - \nabla f(x_{j+i})| \right)
\]
\[
\leq \sum_{j=k}^{k'-1} \left( \kappa_i \tilde{T}_j - \tilde{T}_{j+i} + \kappa_i |\nabla f(x_j) - \nabla f(x_{j+i})| \right).
\]

Since \( \nabla f(x) \) is Lipschitz continuous, we have that
\[
\| \nabla f(x_j) - \nabla f(x_{j+1}) \| \leq \kappa_f \| x_j - x_{j+1} \| \leq \alpha_j \| d_j \| \leq \alpha_j \sqrt{n} \Delta_j.
\]

(57)

\[
\| \Gamma_j - \Gamma_{j+1} \| = \| \nabla f(x_j) - \nabla f(x_{j+1}) \| - \left( AA^T + D_{k_j} \right)^{-1} A \left( \nabla f(x_j) - \nabla f(x_{j+1}) \right) \| \\
\leq \| \nabla f(x_j) - \nabla f(x_{j+1}) \| + \left( AA^T + D_{k_j} \right)^{-1} \| \nabla f(x_j) - \nabla f(x_{j+1}) \| \\
\leq (1 + \kappa_j) \| \nabla f(x_j) - \nabla f(x_{j+1}) \| \\
\leq (1 + \kappa_j) \alpha_j \sqrt{n} \Delta_j.
\]

(58)

Combining (56), (57), and (58), we have that
\[
\varepsilon \leq \alpha_j \sqrt{n} \kappa_2 \left( \Delta_k \sum_{j=0}^{\ell - 1} \Delta_j \right),
\]

(59)

where \( \kappa_2 = \kappa_f (1 + \kappa_1) + \kappa_r^2 \).

From (28), then, for any \( l \) large enough, \( \Delta_k \leq (e/4 \alpha_j \kappa_r \kappa_f) \). Similar to the proof of (38), we have that \( \alpha_j > 0 \) for all index \( j \in [k] \); hence, \( \sum_{j=k+1}^{\ell - 1} \Delta_j > (e/2) > 0 \), which give us \( \sum_{j=k}^{\ell - 1} \Delta_j = \infty \), which contradicts Lemma 6; hence, the conclusion is true.

\[\square\]

5. Local Convergence

In this section, we will show the local convergence of the algorithm, i.e., we prove that our algorithm superlinearly converges to the first-order stable point of problem (P).

According to the definition of \( \bar{\Gamma}(x) \), we have that

\[ 0 < f(x_k + d_k) - f(x_k) - \beta g_k^T d_k = \nabla f(x_k)^T d_k + o(\| d_k \|) \]

\[ = \left( \nabla f(x_k) - g_k \right)^T d_k + (1 - \beta) g_k^T d_k + o(\| d_k \|) \]

\[ \leq (1 - \beta) g_k^T d_k + o(\| d_k \|) + \| \nabla f(x_k) - g_k \| \| d_k \|. \]

(60)

Because \( \| \nabla f(x_k) - g_k \| \leq \kappa_{o} \Delta_k \) and \( \lim_{k \to \infty} \kappa_{o} \Delta_k = 0 \), then \( \| \nabla f(x_k) - g_k \| \to 0 \) as \( k \to \infty \). Hence, we divide \( \| d_k \| \) on both sides of inequality in (60), and we have that

\[ (1 - \beta) \frac{g_k^T d_k}{\| d_k \|} + o(\| d_k \|) + \frac{\| \nabla f(x_k) - g_k \|}{\| d_k \|} \]

(61)

By the preserving character of the limit and (61), we have that

\[ \lim_{k \to \infty} \frac{g_k^T d_k}{\| d_k \|} > 0, \]

(62)

which is contrary with \( g_k^T d_k \leq 0 \) in Lemma 1. Hence, the assumption is not true, and for sufficiently large \( k \), then the step \( \alpha_k \equiv 1 \).

The following theorem shows that the algorithm is local superlinear convergence to optimal solution \( x^* \).

\[ \rho \| x_{k+1} - x^* \|^2 \leq (x_{k+1} - x^*)^T D_k^{(2)} A(x_{k+1} - x^*). \]

(64)

By (64), we have that

\[ \rho \| x_{k+1} - x^* \|^2 \leq (x_{k+1} - x^*)^T D_k^{(2)} A(x_{k+1} - x^*). \]
We use three different levels of accuracy: 2, 3, and 5. Then the number of function evaluations needed by each solver to reach the best performance by any solver is limited to \( \eta_p \) and \( \eta_P \) for each problem \( p \) and solver \( s \), respectively. For each problem \( p \) and solver \( s \), we define the performance ratio as:

\[
r_{p,s} = \min \{ r_{p,u} : u \in S \}.
\]

In order to compare the performance of solver \( s \) on problem \( p \) with the best performance by any solver \( S \) on \( P \), we define the performance ratio as

\[
r_{p,s} = \frac{\tau_{p,s}}{\min \{ \tau_{p,u} : u \in S \}}.
\]

The values on the left side of the figures represent the efficiency of each solver, and the values on the right side give information about the robustness of the solvers. This means that the best solver is the highest on the figures.

Figure 2 shows the comparison results of using our algorithm, COBYLA, and SDPEN to solve the test problems in Table 1 under termination accuracy level \( 2 (\epsilon = 10^{-2}) \) and \( 3 (\epsilon = 10^{-3}) \), respectively. It is obvious from the figures that the speed of the red curve (representing our algorithm) is faster than the other two curves, and when \( t = 1 \), the values of the red curve are higher than the other two curves. This shows that our algorithm performs better than the other two algorithms when we solve the test problems in Table 1, and our algorithm has high computational efficiency under the condition of the low termination level.
Table 1: Description of the test problems.

| Problem | n | Ineq. | Problem | n | Ineq. | Problem | n | Ineq. |
|---------|---|-------|---------|---|-------|---------|---|-------|
| HS10    | 2 | 1     | HS33    | 3 | 2     | HS90    | 4 | 1     |
| HS11    | 2 | 1     | HS34    | 3 | 2     | HS91    | 5 | 1     |
| HS12    | 2 | 1     | HS35    | 3 | 1     | HS92    | 6 | 1     |
| HS13    | 2 | 1     | HS36    | 3 | 1     | HS93    | 6 | 2     |
| HS15    | 2 | 2     | HS37    | 3 | 2     | HS94    | 6 | 4     |
| HS16    | 2 | 2     | HS43    | 4 | 3     | HS95    | 6 | 4     |
| HS17    | 2 | 2     | HS57    | 2 | 1     | HS96    | 6 | 4     |
| HS18    | 2 | 2     | HS64    | 3 | 1     | HS97    | 6 | 4     |
| HS19    | 2 | 2     | HS65    | 3 | 1     | HS98    | 6 | 4     |
| HS21    | 2 | 1     | HS66    | 3 | 2     | HS100   | 7 | 4     |
| HS22    | 2 | 1     | HS70    | 4 | 1     | HS101   | 7 | 6     |
| HS23    | 2 | 1     | HS72    | 4 | 2     | HS102   | 7 | 6     |
| HS29    | 3 | 1     | HS76    | 4 | 3     | HS103   | 7 | 6     |
| HS30    | 3 | 1     | HS88    | 2 | 1     | HS104   | 8 | 6     |
| HS31    | 3 | 1     | HS89    | 3 | 1     | HS105   | 8 | 1     |
| ------  | --- | ---   | ------  | --- | ---   | ------  | --- | ---   |
| HS117   | 15 | 5     |

Figure 1: The CPU time of interpolating 100 times.

Figure 2: Performance profile for the number of iterates in level 2.
Figure 3 records the results of the number of iterations spent by the three algorithms in solving the test problem in Table 1 when the termination level is 3 ($\varepsilon = 10^{-3}$). It can be clearly seen from Figure 2 that when the termination level is increased to 3, the performance rate of the three algorithms is lower than that of accuracy level 2. However, when $t = 1$, the value of the red curve is still significantly higher than the other two curves, and the speed of red curve tending to 1 is significantly faster than the other two curves. This shows that after the accuracy level is improved, the number of iterations spent by our algorithm in solving the test problem is still less than that of the other two algorithms, and its computational efficiency is still high.

Figure 4 records the calculation results of three algorithms when the termination level is 5 ($\varepsilon = 10^{-5}$). It can be seen from Figure 3 that although the computational efficiency of the three algorithms has decreased significantly at a high accuracy level, the red curve still shows a high level. Whether from the value when $t = 1$ or the speed tending to 1, the red curve shows good characteristics, that is, under the condition of high accuracy level, the number of iterations spent by our algorithm in solving the test problem in Table 1 is still less than that of the other two algorithms.

Figure 1 reports the results of the CPU time of constructing interpolation functions in 100 time iteration using $(n + 1)$ and $(n + 1)(n + 2)/2$ interpolation points, respectively. From the result, we can see that the interpolation function can be established faster by using our algorithm. Lemma 1 shows that the linear interpolation model has the same drop as the two interpolation trust region model. Hence, our algorithm is more effective than the algorithm using two interpolation function.

In order to test the effectiveness of our algorithm in solving problems with large dimensions, we selected 6 problems in Table 2, calculated them with our algorithm, COBYLA, and SDPEN under different dimensions, and recorded the calculation results of level 5.

When the accuracy level is 5 (the termination accuracy is $\varepsilon = 10^{-5}$), we use our algorithm to calculate the test problems in Table 2. At the same time, we use COBYLA and SDPEN to calculate the test problems in Table 2, compare the number of iterations spent by the three algorithms in calculating the test problems, and record the comparison results in Figure 5.

From Figure 5, we can see that the red curve representing our algorithm tending to 1 is the fastest, that is, the red curve tends to 1 faster than the curves representing the other two algorithms. Moreover, we can also see that when $t = 1$, the computational efficiency of our algorithm has reached 0.38, while COBYLA rectification has reached 0.25 and SDPEN has only reached 0.11, which shows that our algorithm not only spends less the number of iterations than COBYLA and
SDPEN in solving low-dimensional problems but also shows high computational efficiency. Moreover, the number of iterations is still lower than that of COBYLA and SDPEN. In short, our algorithm has high computational efficiency when calculating the actual test problems.

7. Conclusion

This paper has described the derivative-free linear programming methods based on probabilistic models for linear inequality constrained optimization. The algorithm is based on methods built by linear polynomial interpolations, which use only \( n + 1 \) interpolation point and far less than \( (n + 1)(n + 2)/2 \) interpolation point in quadratic polynomial interpolations and reduce the computation cost of building interpolation function. The affine scaling linear methods which use probabilistic or random models are built to obtain the descent direction. In this paper, we use the backtracking line search technique to ensure that the step length along the direction of descent is in the feasible region, and the objective function is reduced. Under some reasonable conditions, the global and local fast convergence of the algorithm is shown and the results of numerical experiments are reported to show the effectiveness of the proposed algorithm [21–26].

Data Availability

The test problem data used to support the findings of this study have been deposited in the (test examples for nonlinear programming codes) repository (DOI: 10.1007/BF00934594).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors gratefully acknowledge the partial supports of the National Science Foundation of China Grant (11371253) of China. The authors gratefully acknowledge the partial supports of the Natural Science Foundation of Hainan province Grant (120MS029).

References

[1] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-free Optimization*, MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA, 2009.

[2] T. G. Kolda, R. M. Lewis, and V. Torczon, "Stationarity results for generation set search for linearly constrained optimization," *SIAM J. Optim.*, vol. 17, pp. 943–968, 2006.

[3] 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.

[4] S. Gratton, P. L. Toint, and A. Troltzsch, "An active-set trust-region method for derivative-free nonlinear bound-constrained optimization," *Optimization Methods and Software*, vol. 26, no. 4-5, pp. 873–894, 2011.

[5] K. Scheinberg and P. L. Toint, "Self-correcting Geometry in Model-Based Algorithms for Derivativefree Unconstrained Optimization," Technical Report TR09/06, Department of Mathematics, FUNDP - University of Namur, Namur, Belgium, 2009.

Table 2: Description of the large-dimension test problems.

| Problem                | Bounded | Problem     | Bounded |
|------------------------|---------|-------------|---------|
| Rastrigin’s function   | 50 [−5.12, 5.12] | Egg holder  | 50 [−512, 512] |
|                        | 100 [−5.12, 5.12] |            | 100 [−512, 512] |
|                        | 150 [−5.12, 5.12] |            | 150 [−512, 512] |
|                        | 200 [−5.12, 5.12] |            | 200 [−512, 512] |
|                        | 50 [−512, 512]   |            | 50 [−500, 500] |
| Schwefel’s function    | 100 [−512, 512] | Rana’s function | 100 [−500, 500] |
|                        | 150 [−512, 512] |            | 150 [−500, 500] |
|                        | 200 [−512, 512] |            | 200 [−500, 500] |
|                        | 50 [−10, 10]    |            | 50 [0, 10] |
| Stretch V sine wave function | 100 [−10, 10] | Keane’s function | 100 [0, 10] |
|                        | 150 [−10, 10]   |            | 150 [0, 10] |
|                        | 200 [−10, 10]   |            | 200 [0, 10] |

Figure 5: Performance profile for the number of iterates.
[6] S. C. Billups, J. Larson, and P. Graf, "Derivative-free optimization of expensive functions with computational error using weighted regression," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 27–53, 2013.

[7] L. F. Bueno, A. Friedlander, J. M. Martínez, and F. N. C. Sobral, "Inexact restoration method for derivative-free optimization with smooth constraints," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1189–1213, 2013.

[8] S. M. Wild and C. Shoemaker, "Global convergence of radial basis function trust-region algorithms for derivative-free optimization," *SIAM Review*, vol. 55, no. 2, pp. 349–371, 2013.

[9] 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.

[10] D. Xue and W. Sun, "On convergence analysis of a derivative-free trust region algorithm for constrained optimization with separable structure," *Science China Mathematics*, vol. 57, no. 6, pp. 1287–1302, 2014.

[11] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[12] C. Audet and J. E. Dennis, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 188–217, 2006.

[13] L. N. Vicente and A. L. Custódio, "Analysis of direct searches for discontinuous functions," *Mathematical Programming*, vol. 133, no. 1-2, pp. 299–325, 2012.

[14] A. S. Bandeira, K. Scheinberg, and L. N. Vicente, "Convergence of trust-region methods based on probabilistic models," *SIAM Journal on Optimization*, vol. 24, no. 3, pp. 1238–1264, 2014.

[15] P. Wang and D. Zhu, "A derivative-free affine scaling trust region methods based on probabilistic models with new nonmonotone line search technique for linear inequality constrained minimization without strict complementarity," *International Journal of Computer Mathematics*, vol. 96, pp. 1–29, 2019.

[16] R. Durrett, *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, Cambridge, UK, fourth edition, 2010.

[17] M. J. D. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, S. Gomez and J.-P. Hennart, Eds., Kluwer Academic, Dordrecht, Netherlands, pp. 51–67, 1994.

[18] G. Liuzzi, S. Lucidi, and M. Sciandrone, "Sequential penalty derivative free methods for constrained optimization," *SIAM J. Optim.*, vol. 20, pp. 39–58, 2010.

[19] W. Hock and K. Schittkowski, "Test examples for nonlinear programming codes. Lect." *Notes Econ.Math.Syst.*, vol. 187, 1981.

[20] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.