MINIMA DISTRIBUTION FOR GLOBAL OPTIMIZATION

XIAOPENG LUO†

Abstract. Minima distribution (MD) establishes a strict mathematical relationship between an arbitrary continuous function on a compact set and its global minima, like the well-known connection $\nabla f(x^*) = 0$ between a differentiable convex function $f$ and its minimizer $x^*$. MD theory provides us with a global monotonic convergence for the minimization of continuous functions on compact sets without any other assumptions; and the asymptotic convergence rate can be further determined for twice continuously differentiable functions. Moreover, a derivative-free algorithm based on MD theory is proposed for finding a stable global minimizer of a possibly highly nonlinear and non-convex function. By means of an extensive testbed it is demonstrated that the MD method converges faster and with more stability than many other acclaimed global methods, and a MATLAB code of the algorithm is provided in appendix for the potential readers' convenience and the reproducibility of the computational results.

Key words. global optimization, structural property, minima distribution, derivative-free

AMS subject classifications. 90C26, 90C30, 90C56, 90C59

1. Introduction. Given a possibly highly nonlinear and non-convex continuous function $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ with the global minima $f^*$ and the set of all global minimizers $X^*$ in $\Omega$, we consider the constrained optimization problem

$$
\min_{x \in \Omega} f(x),
$$

that is, finding $f^*$ and a stable global minimizer in $X^*$; where $\Omega$ is a (not necessarily convex) compact set defined by inequalities $g_i(x) \geq 0$, $i = 1, \cdots, r$.

Generally, finding an arbitrary local minima is relatively straightforward by using classical local methods, for example, a gradient-based approach; however, finding the global minima is far more difficult. As a typical representative of expanding local methods to deal with global problems, random search (RS) is based on the idea of performing parallel local searches starting from multiple initial points. RS appeared in the 1950s [1, 5, 34] and the theoretical investigation was established later [35, 27], moreover, the choice of step size is further considered in [42, 41, 45, 23]. It is usually effective if the number of local minimizers of a target function is not large [3]; however, one cannot see any overall landscape in the RS algorithm since there is no information exchange between those parallel local searches. And such information exchange, in fact, happens to be the focus of the so-called direct searches.

Central to every direct search method is a strategy that generates variations of a set of candidate solutions (often called a population). The best known of these are genetic algorithms (GA) [13, 24], evolution strategies (ES) [36, 43], differential evolution (DE) [47, 32, 9, 8] and so forth [52]. In particular, DE usually performs well for continuous optimization problems [31, 47, 9, 8] although does not guarantee that an optimal solution is ever found. The evolution strategy of DE takes the difference of two randomly chosen candidates to perturb an existing candidate and accepts a new candidate under a greedy or annealing criterion [19]. To some extent, these methods have used partial prior information to guide subsequent searches.

*Funding: The author was supported by NSF grants CHE-1464569 and CHE-1763198.
†Department of Chemistry, Princeton University, Princeton, NJ 08544, USA (luo_works@163.com, xiaopeng@princeton.edu.)
Bayesian optimization (BO) \cite{25, 26} is a typical strategy of making full use of prior information. The BO method is to treat a target function as a random function with a prior distribution and applies Bayesian inference to update the prior according to the previous observations. This updated prior is used to construct an acquisition function to determine the next candidate. The acquisition function, which trade-offs exploration and exploitation so as to minimize the number of function evaluations, can be of different types, such as probability of improvement (PI) \cite{20}, expected improvement (EI) \cite{26, 6} or lower confidence bound (LCB) \cite{7}. BO is a sequential model-based approach and the model is often obtained using a Gaussian process (GP) \cite{18, 12, 38} which provides a normally distributed estimation of the target function \cite{33}. BO has recently emerged as a de-facto method for adjusting hyper-parameters in machine learning \cite{44, 29} and been used for many applications, e.g., \cite{22, 2, 11, 48}.

Although most global methods try to extract as much knowledge as possible from prior information such as previous function evaluations, the connection between the entire landscape of the target function and its global minima is not yet sufficiently clear and precise. More specifically, there is currently the lack of such an essential mathematical relationship; in contrast, there is a well-known relationship between a differentiable convex function and its minima, which is established by the gradient.

Actually, the significance of the gradient is reflected in both structural and constructive aspects for the minimization of a differentiable convex function. Suppose $f$ is differentiable on a convex set $D$, then $f$ is convex if and only if

$$f(y) \geq f(x) + \nabla f(x)(y - x), \quad \forall x, y \in D;$$

see for example Ref. \cite{4}. And this shows that, for the structural aspect, $\nabla f(x^*) = 0$ implies that $f(y) \geq f(x^*)$ holds for all $y \in D$, i.e., $x^*$ is a global minimizer of $f$ on $D$; moreover, for the constructive aspect, one can find the global minimizer by gradually taking steps proportional to the negative of the gradient of $f$ at the current point. The role of this essential correlation is self-evident in convex optimization.

This paper aims to establish a similar mathematical relationship between any continuous function $f$ on a compact set $\Omega \subset \mathbb{R}^n$ and its global minima $f^*$; as a typical case, we will prove that

$$\lim_{k \to \infty} \int_{\Omega} f(x)m^{(k)}(x)dx = f^*, \quad \text{where} \quad m^{(k)}(x) = \frac{e^{-kf(x)}}{\int_{\Omega} e^{-kf(x)}dx},$$

meanwhile, it follows that

$$\lim_{k \to \infty} m^{(k)}(x) = \begin{cases} \frac{1}{\mu(X^*)}, & x \in X^*, \\ 0, & x \in \Omega - X^*, \end{cases}$$

where $\mu(X^*)$ is the $n$-dimensional Lebesgue measure of the set of minimizers $X^* \in \Omega$; and if $x_i^* \in X^*$ is the unique minimizer of $f$ in a certain subdomain $\Omega_i \subset \Omega$, then

$$\lim_{k \to \infty} \int_{\Omega_i} x \cdot m^{(k)}(x)dx = x_i^* \int_{\Omega_i} m^{(k)}(x)dx.$$

Not only that, if $f$ is not a constant on $\Omega$, the monotonic relationship

$$\int_{\Omega} f(x)m^{(k)}(x)dx > \int_{\Omega} f(x)m^{(k+\Delta k)}(x)dx > f^*$$

\cite{27, 26, 6, 12} which provides a normally distributed estimation of the target function \cite{33}. BO has recently emerged as a de-facto method for adjusting hyper-parameters in machine learning \cite{44, 29} and been used for many applications, e.g., \cite{22, 2, 11, 48}.
holds for all $k \in \mathbb{R}$ and $\Delta k > 0$, which implies a series of monotonic containment relationships, and one of them is, for examples,

\[(6) \quad \Omega \supset D_f^{(k)} \supset D_f^{(k+\Delta k)} \supset X^*, \] where $D_f^{(k)} = \left\{ x \in \Omega : f(x) \leq \int_\Omega f(t)m^{(k)}(t)dt \right\}$;

moreover, if $x_i^* \in X^*$ is the unique minimizer of $f$ in a certain subdomain $\Omega_i \subset \Omega$, $f(x - x_i^*) = \psi(\|x - x_i^*\|)$ is radial in $\Omega_i$ and $\psi$ is a non-decreasing function on $\mathbb{R}_+$, then for all $k \in \mathbb{R}$ and $\Delta k > 0$, it holds that

\[(7) \quad \left\| \int_{\Omega_i} (x - x_i^*)m^{(k)}(x)dx \right\|_2 \geq \left\| \int_{\Omega_i} (x - x_i^*)m^{(k+\Delta k)}(x)dx \right\|_2. \]

These results will help us understand the optimization problem itself and distinguish the difference between global approximation and optimization. Based on the proposed theory, we also present a derivative-free algorithm for finding a stable global minimizer of any continuous function on a compact set. Using a monotonic relationship similar to (6), the algorithm gradually simplifies a complex global problem into a simple local problem; meanwhile, the computational requirement is gradually decreasing while the convergence rate is increasing. But for the BO, on the contrary, the computational requirement is increasing while the convergence rate is often decreasing.

The remainder of the paper is organized as follows. The concept of minima distribution (MD) and relevant conclusions (including the monotonic convergence and stability, etc) are fully built in section 2. And then, a derivative-free algorithm based on MD theory is proposed in section 3. Numerical experiments are performed for several illustrative examples in section 4. And we draw some conclusions in section 5.

2. Theory. To establish a mathematical relationship between $f$ and $f^*$, we hope to find an integrable distribution function $m_{f, \Omega} \in \mathcal{L}(\Omega)$ such that

\[f^* = \int_\Omega f(x)m_{f, \Omega}(x)dx \quad \text{and} \quad m_{f, \Omega}(x) > 0 \text{ if } x \in X^*, \quad m_{f, \Omega}(x) = 0 \text{ if } x \in \Omega - X^*.\]

Since the distribution $m_{f, \Omega}$ records all the information for the minimization of function $f$ on $\Omega$, we call it a minima distribution related to $f$ and $\Omega$. In the following, we will first introduce the nascent minima distribution function $m^{(k)}$ related to $f$ and $\Omega$, then define the minima distribution by a weak limit of $m^{(k)}$; meanwhile, we will establish the monotonic convergence, the stability, and some other useful results.

2.1. Nascent minima distribution function.

**Definition 2.1.** Suppose $f$ is a continuous real function defined on a compact set $\Omega \subset \mathbb{R}^n$ and $\rho : \mathbb{R} \to \mathbb{R}$ is monotonically decreasing with $\rho(f(x)) > 0$ for every $x \in \Omega$. For any $k \in \mathbb{N}_0$ or non-negative real number $k$, we define a nascent minima distribution function by

\[(8) \quad m^{(k)}(x) = m_{f, \Omega}^{(k)}(x) = \frac{\tau^k(x)}{\int_\Omega \tau^k(t)dt}, \quad \text{where} \quad \tau(x) = \rho(f(x)).\]

And a typical choice of $\tau$ is the exponential-type, i.e., $\tau(x) = e^{-f(x)}$.

**Remark 2.1.** Of course, the exponential-type is not the unique choice of $\tau$; for example, $\tau(x) = \frac{1}{\sqrt{\pi}}e^{-f^2(x)}$ is also a feasible choice in theory, although it is not convenient in practice because we don’t know $f^*$ beforehand.
As illustrated in Figure 1, the most important feature of the nascent minima distribution class is that $m^{(k)}$ retains almost all the information of $f$ in the opposite sense when $k (\neq 0)$ is small; while $m^{(k)}$ gradually highlights the information of $f$ around the set of all global minimizers $X^*$ and compresses other information when $k$ increases. Hence, this process shows a dynamic relationship between $f$ and $X^*$, which will be strictly established later. Clearly, $m^{(k)}$ is a probability density function (PDF) on $\Omega$; especially, $m^{(0)}(x) = 1/\mu(\Omega)$ is the PDF of the continuous uniform distribution on $\Omega$, where $\mu(S) = \int_S dt$ is the $n$-dimensional Lebesgue measure of $S$.

For any $h \in C(\Omega)$ and $\nu \in \mathbb{R}$, define the expectation

$$E^{(k)}(h)^\nu = \int_\Omega h^\nu(t)m^{(k)}(t)dt \quad \text{or} \quad E^{(k)}_\tau(h(x + t))^\nu = \int_\Omega h^\nu(x + t)m^{(k)}(t)dt,$$

then we list some relevant properties below.

**Theorem 2.2.** The nascent minima distribution function satisfies:

(i) The maxima of $m^{(k)}$ is $m^{(k)}(x^*)$ and the set of all maximizers is $X^*$.

(ii) If $f \in C^1(\Omega)$, then $\nabla m^{(k)}(x) = km^{(k)}(x)\nabla\tau(x)/\tau(x)$.

(iii) Suppose $f \in C^2(\Omega)$. If $x^*_1, x^*_2 \in X^*$ and $\nabla^2 f(x^*_1) > \nabla^2 f(x^*_2) > 0$, then

$$\nabla^2 m^{(k)}(x^*_1) < \nabla^2 m^{(k)}(x^*_2) < 0,$$

where $\nabla^2 f$ and $\nabla^2 m^{(k)}$ are the Hessian functions of $f$ and $m^{(k)}$, respectively.

(iv) For every $k \in \mathbb{R}$, it holds that

$$\frac{d}{dk} m^{(k)}(x) = m^{(k)}(x) \left( \log(\tau(x)) - E^{(k)}(\log(\tau)) \right).$$

(v) If $X^*$ has zero $n$-dimensional Lebesgue measure, i.e., $\mu(X^*) = 0$, then

$$\lim_{k \to \infty} m^{(k)}(x) = \begin{cases} \infty, & x \in X^*; \\ 0, & x \notin X^*. \end{cases}$$

(vi) If $X^*$ has nonzero $n$-dimensional Lebesgue measure, i.e., $\mu(X^*) > 0$, then

$$\lim_{k \to \infty} m^{(k)}(x) = \begin{cases} \frac{1}{\mu(X^*)}, & x \in X^*; \\ 0, & x \notin X^*. \end{cases}$$
Remark 2.2. According to the property (iv), it is a very natural thing to choose the exponential-type \( \tau \), although not necessary: and in this case,

\[
\nabla m^{(k)}(x) = -km^{(k)}(x)\nabla f(x) \quad \text{and} \quad \frac{d}{dk} m^{(k)}(x) = m^{(k)}(x) \left( \mathbb{E}^{(k)}(f) - f(x) \right).
\]

Proof. Clearly, (i)-(iv) follow immediately from Definition 2.1. For any \( x' \notin X^* \), let \( p = \tau(x') \), then there must exist an open set \( \Omega_p \) that has nonzero \( n \)-dimensional Lebesgue measure such that \( \tau(t) > p \) if \( t \in \Omega_p \) and \( \tau(t) \leq p \) if \( t \notin \Omega_p \), and further,

\[
m^{(k)}(x') = \frac{p^k}{\int_{\Omega_p} \tau^k(t)dt + \int_{\Omega - \Omega_p} \tau^k(t)dt} \leq \frac{1}{\int_{\Omega_p} p^{-k}\tau^k(t)dt},
\]

since \( p^{-1}\tau(t) > 1 \) for any \( t \in \Omega_p \), the limit of \( \int_{\Omega_p} p^{-k}\tau^k(t)dt \) tends to \( \infty \) as \( k \to \infty \); thus, it holds that

\[
\lim_{k \to \infty} m^{(k)}(x') = 0, \quad \forall x' \notin X^*.
\]

Otherwise, for any \( x'' \in X^* \),

\[
m^{(k)}(x'') = \frac{1}{\int_{X^*} \tau^k(t)dt + \int_{\Omega - X^*} \tau^{-k}(x'') \tau^k(t)dt},
\]

since \( \tau^{-1}(x'')\tau(t) < 1 \) for any \( t \in \Omega - X^* \), the limit of \( \int_{\Omega - X^*} \tau^{-k}(x'') \tau^k(t)dt \) tends to \( 0 \) as \( k \to \infty \); thus, for any \( x'' \in X^* \), it follows that

\[
\lim_{k \to \infty} m^{(k)}(x'') = \begin{cases} 
\infty, & \mu(X^*) = 0; \\
1, & \mu(X^*) > 0;
\end{cases}
\]

this proves (v) and (vi), and the proof is complete. \( \square \)

2.2. Monotonic convergence. The most attractive property of \( m^{(k)} \) is that it provides us with a monotonic convergence of the minimization for all continuous functions on compact sets without any other assumptions. This monotonic convergence strictly confirms the role of \( m^{(k)} \) as a link between \( f \) and \( f^* \).

We first prove the convergence according to the continuity of \( f \).

Theorem 2.3 (convergence). If \( f \in C(\Omega) \), then

\[
\lim_{k \to \infty} \int_{\Omega} f(x)m^{(k)}(x)dx = f^*;
\]

moreover, if \( x^*_i \) is the unique minimizer of \( f \) in any \( \Omega_i \subset \Omega \), then

\[
\lim_{k \to \infty} \int_{\Omega_i} x \cdot m^{(k)}(x)dx = x^*_i \int_{\Omega_i} m^{(k)}(x)dx.
\]

Proof. Given \( \epsilon > 0 \), it follows from the continuity of \( f \) that there exists an open set \( \Omega_e \) such that \( X^* \subset \Omega_e \) and \( f(x) - f^* < \epsilon \) holds for all \( x \in \Omega_e \); further, according to (v) and (vi) of Theorem 2.2, there exists a \( K \in \mathbb{N} \) such that

\[
\int_{\Omega - \Omega_e} m^{(k)}(x)dx < \epsilon
\]
holds for every \( k > K \), hence,

\[
\int_{\Omega} f(x)m^{(k)}(x)dx - f^* = \int_{\Omega} (f(x) - f^*)m^{(k)}(x)dx + \int_{\Omega - \Omega} (f(x) - f^*)m^{(k)}(x)dx < \epsilon(1 - \epsilon) + Rf\epsilon < (1 + Rf)\epsilon,
\]

where \( Rf = \max_{x \in \Omega} f(x) - f^* \) and \( k > K \), which completes the proof of the first identity; and in a similar way one can establish the second one.

To prove the monotonicity, we need the following lemma.

**Lemma 2.4** (Gurland’s inequality [16]). Suppose \( y \) is an arbitrary random variable defined on a subset of \( \mathbb{R} \), if \( g \) and \( h \) are both non-increasing or non-decreasing, then \( \mathbb{E}(g(y) \cdot h(y)) \geq \mathbb{E}(g(y))\mathbb{E}(h(y)) \); if \( g \) is non-decreasing and \( h \) non-increasing, or vice versa, then \( \mathbb{E}(g(y) \cdot h(y)) \leq \mathbb{E}(g(y))\mathbb{E}(h(y)) \).

**Theorem 2.5** (monotonicity). For every \( k \in \mathbb{R} \), the nascent minima distribution function (8) satisfies

\[
\frac{d}{dk} \int_{\Omega} f(x)m^{(k)}(x)dx \leq 0;
\]

especially, if \( \tau \) is the exponential-type \( e^{-f} \) and \( f \) is not a constant on \( \Omega \), then

\[
\frac{d}{dk} \int_{\Omega} f(x)m^{(k)}(x)dx = -\text{Var}^{(k)}(f) < 0,
\]

where \( \text{Var}^{(k)}(f) = \mathbb{E}^{(k)}(f^2) - (\mathbb{E}^{(k)}(f))^2 \).

**Proof.** According to (iv) of Theorem 2.2, we have

\[
m^{(k+\Delta k)}(x) = m^{(k)}(x) + \int_{k}^{k+\Delta k} \frac{dm^{(v)}(x)}{dv} dv
\]

\[
= m^{(k)}(x) + \int_{k}^{k+\Delta k} m^{(v)}(x)(\log(\tau(x)) - \mathbb{E}^{(v)}(\log(\tau))) dv,
\]

then there exists a \( \zeta \in (k, k + \Delta k) \) such that

\[
\frac{\mathbb{E}^{(k+\Delta k)}(f) - \mathbb{E}^{(k)}(f)}{\Delta k} = \frac{1}{\Delta k} \int_{\Omega} f(x) \left( m^{(k+\Delta k)}(x) - m^{(k)}(x) \right) dx
\]

\[
= \frac{1}{\Delta k} \int_{\Omega} \int_{k}^{k+\Delta k} f(x)m^{(v)}(x)(\log(\tau(x)) - \mathbb{E}^{(v)}(\log(\tau))) dv dx
\]

\[
= \frac{1}{\Delta k} \int_{k}^{k+\Delta k} \left( \mathbb{E}^{(v)}(f \log(\tau)) - \mathbb{E}^{(v)}(f)\mathbb{E}^{(v)}(\log(\tau)) \right) dv
\]

\[
= \mathbb{E}^{(\zeta)}(f \log(\tau)) - \mathbb{E}^{(\zeta)}(f)\mathbb{E}^{(\zeta)}(\log(\tau)),
\]

hence, we have

\[
\frac{d\mathbb{E}^{(k)}(f)}{dk} = \lim_{\Delta k \to 0} \frac{\mathbb{E}^{(k+\Delta k)}(f) - \mathbb{E}^{(k)}(f)}{\Delta k} = \mathbb{E}^{(k)}(f \log(\tau)) - \mathbb{E}^{(k)}(f)\mathbb{E}^{(k)}(\log(\tau)).
\]

Further, let \( y = f(x) \), then \( \log(\tau(x)) = \log(\rho(y)) \), and then

\[
\mathbb{E}^{(k)}(f \log(\tau)) - \mathbb{E}^{(k)}(f)\mathbb{E}^{(k)}(\log(\tau)) = \mathbb{E}(y \log(\rho(y))) - \mathbb{E}(y)\mathbb{E}(\log(\rho(y)));
\]
Lemma 2.4 that 

$$\mathbb{E}(y \log(\rho(y))) \leq \mathbb{E}(y \mathbb{E}(\log(\rho(y)))),$$

thus, we have \(\frac{\partial g(k)}{\partial k} \leq 0\), as claimed. Specially, if \(\tau(x) = e^{-f(x)}\), then

$$\mathbb{E}^{(k)}(f \log(\tau)) - \mathbb{E}^{(k)}(f)\mathbb{E}^{(k)}(\log(\tau)) = -\mathbb{E}^{(k)}(f^2) + (\mathbb{E}^{(k)}(f))^2,$$

that is, \(\frac{\partial g(k)}{\partial k} = -\nabla \ar^{(k)}(f) < 0\), and the proof is complete. □

**Theorem 2.6.** If \(x^*_i\) is the unique minimizer of \(f\) in any \(\Omega_i \subset \Omega\), \(f(x - x^*_i) = \psi(||x - x^*_i||_2)\) is radial in \(\Omega_i\) and \(\psi\) is a non-decreasing function on \(\mathbb{R}_+\), then

$$\frac{d}{dk} \left( \int_{\Omega} (x - x^*_i) m^{(k)}(x) dx \right) \leq 0.$$

**Proof.** Let \(r = ||x - x^*_i||_2\), then

$$\frac{d}{dk} \left( \int_{\Omega} (x - x^*_i) m^{(k)}(x) dx \right) \leq \frac{d}{dk} \int_{\Omega} r \cdot m^{(k)}(x) dx,$$

then it holds from (10) that

$$\frac{d}{dk} \int_{\Omega} r \cdot m^{(k)}(x) dx = \mathbb{E}^{(k)}[r \log(\rho(r))] - \mathbb{E}^{(k)}(r)\mathbb{E}^{(k)}[\log(\rho(\psi(r)))],$$

since \(\log(\rho(\psi(r)))\) is monotonically decreasing, the desired result follows. □

2.3. Minima distribution. Now we define a minima distribution to be a weak limit \(m_{f,\Omega}\) such that the identity

$$\int_{\Omega} m_{f,\Omega}(x) \varphi(x) dx = \lim_{k \to \infty} \int_{\Omega} m^{(k)}(x) \varphi(x) dx$$

holds for every smooth function \(\varphi\) with compact support in \(\Omega\). Here are two immediate properties of \(m_{f,\Omega}\):

**Theorem 2.7.** The minima distribution satisfies the following properties:

(i) \(m_{f,\Omega}\) satisfies the identity \(\int_{\Omega} m_{f,\Omega}(x) dx = 1\).

(ii) If \(f\) is continuous on \(\Omega\), then \(f^* = \int_{\Omega} f(x) m_{f,\Omega}(x) dx\).

(iii) If \(x^*_i\) is the unique minimizer of \(f\) in any \(\Omega_i \subset \Omega\), then

$$\int_{\Omega_i} x \cdot m_{f,\Omega}(x) dx = x^*_i \int_{\Omega} m_{f,\Omega}(x) dx.$$

A naive view of the minima distribution is that \(m_{f,\Omega}\) is the pointwise limit

$$m_{f,\Omega}(x) = \lim_{k \to \infty} m^{(k)}(x)$$

that also reserves \(\int_{\Omega} m_{f,\Omega}(x) dx = 1\). From this view, we begin by noting that \(m_{f,\Omega}(x) = 0\) if \(x \notin \mathbb{X}^*\) and deduce that \(f(x) m_{f,\Omega}(x) = f^* \cdot m_{f,\Omega}(x)\) for every \(x \in \Omega\), then we also have \(\int_{\Omega} f(x) m_{f,\Omega}(x) dx = f^* \int_{\Omega} m_{f,\Omega}(x) dx = f^*\), and so forth.
2.4. Stability. The significance of stability is self-evident for optimization, we always expect a sequence of approximate solutions to converge to a flatter valley bottom in the presence of multiple minimizers. In the following, one will see that any method built on the MD theory satisfies this requirement in the sense of probability.

If $\mu(X^*) \neq 0$, then $m_{f,\Omega}$ can be viewed as the PDF of the uniform distribution on $X^*$; if $x^*$ is the unique minimizer of $f$ on $\Omega$, then $m_{f,\Omega}$ is exactly the Dirac delta function $\delta(x-x^*)$; and if $X^*$ is a finite set, i.e., $X^* = \{x^*_i\}_{i=1}^s$, then $m_{f,\Omega}$ can be given by a linear combination of Dirac delta functions

$$m_{f,\Omega}(x) = \sum_{i=1}^s w_i \delta(x-x^*_i) \quad \text{with} \quad \sum_{i=1}^s w_i = 1 \quad \text{and} \quad w_i > 0. \quad (11)$$

According to (iii) of Theorem 2.2, if $f \in C^2(\Omega)$ and $\nabla^2 f(x^*_1) > \nabla^2 f(x^*_2) > \cdots > \nabla^2 f(x^*_s) > 0$, then for given $k \in \mathbb{N}$, it follows that

$$\nabla^2 m^{(k)}(x^*_1) < \nabla^2 m^{(k)}(x^*_2) < \cdots < \nabla^2 m^{(k)}(x^*_s) < 0;$$

and further notice that $m^{(k)}_{\Omega,f}$ is approximately equal to

$$m^{(k)}(x^*_1) + \frac{1}{2}(x - x^*_1)^T \nabla^2 m^{(k)}(x^*_1)(x - x^*_1)$$

in the neighborhood $\Omega^0_i$ of $x^*_i$

Hence, if all points $\{x^*_i\}_{i=1}^s$ are not on the boundary of $\Omega$, we obtain

$$\int_{\Omega_1} m^{(k)}(x)dx < \int_{\Omega_2} m^{(k)}(x)dx < \cdots < \int_{\Omega_s} m^{(k)}(x)dx,$$

and further, the weight coefficients of (11) satisfy $w_1 < w_2 < \cdots < w_s$, as illustrated in Figure 2. It is very helpful to find a stable solution when the target function has more than one global minimizer. Note that this derivation can be extended to any situation that the global minimizer is not unique. So we have the following theorem:

**Theorem 2.8.** Suppose $f \in C^2(\Omega)$, $\nabla^2 f(x^*_1) > \nabla^2 f(x^*_2) > 0$ and two points $x^*_1, x^*_2 \in X^*$ are not on the boundary of $\Omega$.

(i) If $\xi$ is a random sample from a distribution with PDF $m^{(k)}$ $(k > 0)$, then the probability that $\xi$ is near $x^*_2$ is greater than the probability that $\xi$ is near $x^*_1$.

(ii) If $\xi$ is a random sample from a distribution with PDF $m_{f,\Omega}$, then the probability that $\xi$ takes $x^*_2$ is greater than the probability that $\xi$ takes $x^*_1$.

2.5. Significant sets of $m^{(k)}$. Here we will introduce several types of significant set of $m^{(k)}$, which are very useful for both understanding $m^{(k)}$ itself and constructing efficient algorithms. Let’s define the first type of significant set of $m^{(k)}$ as

$$D_f^{(k)} = \{x \in \Omega : f(x) \leq E^{(k)}(f)\}, \quad (12)$$

then it is clear that Theorem 2.5 implies the following containment relationship:

**Theorem 2.9** (monotonic shrinkage). Suppose $\Omega$ is a compact set and $f \in C(\Omega)$. For all $k \in \mathbb{R}$ and $\Delta k > 0$, it holds that

$$\Omega \supset D_f^{(k)} \supset D_f^{(k+\Delta k)} \supset X^*,$$

where $X^*$ is the set of all global minimizers of $f$; especially, if $\tau$ is the exponential-type $e^{-1}$ and $f$ is not a constant on $\Omega$, it holds that $D_f^{(k)} \supset D_f^{(k+\Delta k)}$. 
This result can also be stated in another way. For \( a > 0 \), it holds from (10) that
\[
\frac{d\mathbb{E}^{(k)}(\tau^a)}{dk} = \mathbb{E}^{(k)}(\tau^a \log(\tau^a)) - \mathbb{E}^{(k)}(\tau^a)\mathbb{E}^{(k)}(\log(\tau^a)),
\]
Further, let \( y = \tau^a(x) \), then we have \( \frac{d\mathbb{E}^{(k)}(\tau^a)}{dk} = \mathbb{E}(y \log(y)) - \mathbb{E}(y)\mathbb{E}(\log(y)) \); and it holds from Lemma 2.4 that \( \mathbb{E}(y \log(y)) \geq \mathbb{E}(y)\mathbb{E}(\log(y)) \), thus, \( \frac{d\mathbb{E}^{(k)}(\tau^a)}{dk} \geq 0 \). Hence, if we define the second type of significant set of \( m^{(k)} \) as
\[
(13) \quad D^{(k)}_{\tau^a} = \left\{ x \in \Omega : \tau^a(x) \geq \mathbb{E}^{(k)}(\tau^a) \right\},
\]
then the discriminant condition \( \tau^a(x) \geq \mathbb{E}^{(k)}(\tau^a) \) is equivalent to \( m^{(k+a)}(x) \geq m^{(k)}(x) \) and we have the similar containment relationship:

**Theorem 2.10 (Monotonic shrinkage).** Suppose \( \Omega \) is a compact set and \( f \in C(\Omega) \). For all \( a > 0 \), \( k \in \mathbb{R} \) and \( \Delta k > 0 \), it holds that
\[
\Omega \supset D^{(k)}_{\tau^a} \supset D^{(k+\Delta k)}_{\tau^a} \supset X^*,
\]
where \( X^* \) is the set of all global minimizers of \( f \); especially, if \( \tau \) is the exponential-type \( e^{-f} \) and \( f \) is not a constant on \( \Omega \), it holds that \( D^{(k)}_{\tau^a} \supset D^{(k+\Delta k)}_{\tau^a} \).

Both \( f \) and \( \tau \) might be very steep or highly oscillating in the main distribution area of \( m^{(k)} \). To enhance numerical stability, for a given sample set \( X = \{x_i\}_{i=1}^N \) and \( Y = f(X) \) we usually consider the normalization
\[
(14) \quad \tau^a(x) = \tau^a_c(x) = e^{-a(f(x)-c)},
\]
where \( a = (\text{Var}(Y))^{-\frac{1}{2}} \) and \( c = \mathbb{E}(Y) \). On the one hand, since \(|(e^{-t})'| \leq 1 \) if \( t > 0 \) and \(|(e^{-t})'| > 1 \) if \( t < 0 \), \( \tau^a_c \) compresses the information where \( f \) is larger than \( \mathbb{E}(Y) \) and enlarges the information where \( f \) is smaller than \( \mathbb{E}(Y) \); on the other hand, \( \tau^a_c \) guarantees that its values are always distributed within a suitable range.

Finally, we will introduce the third type of significant set of \( m^{(k)} \) which can be used to estimate the shrink rate of \( m^{(k)} \). Let
\[
(15) \quad D^{(k)}_0 = \left\{ x \in \Omega : m^{(k)}(x) \geq m^{(0)}(x) = 1/\mu(\Omega) \right\}
\]
with its boundary \( \Gamma^{(k)}_0 = \left\{ x \in \Omega : m^{(k)}(x) = 1/\mu(\Omega) \right\} \). Similarly, we have:
Theorem 2.11 (monotonic shrinkage). Suppose \( \Omega \) is a compact set and \( f \in C(\Omega) \) is not a constant. For all \( k, \Delta k \in \mathbb{R}_+ \), it holds that
\[
\Omega = D_0^{(0)} \supset D_0^{(k)} \supset D_0^{(k+\Delta k)} \supset \Omega^*,
\]
where \( \Omega^* \) is the set of all global minimizers of \( f \).

Proof. First, it follows from \( m^{(0)} = 1/\mu(\Omega) \) that \( \Omega = D_0^{(0)} \); and for any \( x^* \in \Omega^* \) and \( k > 0 \), it holds from \( \int_{\Omega - X^*} \tau^{-k}(x^*) \tau^k(t) dt < \mu(\Omega - X^*) \) that
\[
m^{(k)}(x^*) = \frac{1}{\int_{X^*} dt + \int_{\Omega - X^*} \tau^{-k}(x^*) \tau^k(t) dt} > \frac{1}{\mu(X^*) + \mu(\Omega - X^*)} = \frac{1}{\mu(\Omega)},
\]
that is, \( \Omega^* \subset D_0^{(k)} \) for every \( k \geq 0 \).

Then, for any \( \Delta k \in \mathbb{R}_+ \), according to Hölder’s inequality, it follows that
\[
\int_{\Omega} \tau^k(t) dt < \left( \int_{\Omega} dt \right)^{\frac{\Delta k}{k}} \left( \int_{\Omega} \tau^{k+\Delta k}(t) dt \right)^{\frac{k}{\Delta k}}
\]
when \( f \) is not a constant on \( \Omega \). If \( x \in D_0^{(k+\Delta k)} \), then
\[
\tau^{k+\Delta k}(x) \geq \frac{\int_{\Omega} \tau^{k+\Delta k}(t) dt}{\int_{\Omega} dt},
\]
and together with (16), we have
\[
\tau^k(x) \geq \left( \frac{\int_{\Omega} \tau^{k+\Delta k}(t) dt}{\int_{\Omega} dt} \right)^{\frac{k}{\Delta k}} = \left( \frac{\int_{\Omega} dt}{\int_{\Omega} dt} \right)^{\frac{k}{\Delta k}} \left( \frac{\int_{\Omega} \tau^{k+\Delta k}(t) dt}{\int_{\Omega} dt} \right)^{\frac{k}{\Delta k}} > \left( \frac{\int_{\Omega} \tau^k(t) dt}{\int_{\Omega} dt} \right)^{\frac{k}{\Delta k}},
\]
that is, \( x \in D_0^{(k)} - \Gamma_0^{(k)} \), thus, \( D_0^{(k)} \supset D_0^{(k+\Delta k)} \) for any \( \Delta k \in \mathbb{R}_+ \), as claimed. \( \Box \)

2.6. Shrink rate of \( D_0^{(k)} \). The shrinkage from \( D_0^{(k)} \) to \( D_0^{(k+\Delta k)} \) reflects exactly the difference between \( m^{(k)} \) and \( m^{(k+\Delta k)} \); however, the shrink rate would be slow for a large \( k \). Now we will consider the asymptotic shrink rate.

Let \( f \in C^1(\Omega) \), \( x \in \Gamma_0^{(k)} \) and \( x \) moves to \( x' = x + \Delta x \in \Gamma_0^{(k+\Delta k)} \) when \( k \) continuously increases to \( k' = k + \Delta k \). It follows from (ii) of Theorem 2.2 that
\[
\frac{1}{\mu(\Omega)} = m^{(k')}(x') = m^{(k)}(x) + \frac{k' m^{(k)}(x) \nabla \tau(x)}{\tau(x)} : \Delta x + O(\|\Delta x\|^2);
\]
where \( \| \cdot \| \) is the Euclidean norm.

Moreover, according to (iv) of Theorem 2.2, if \( f \in C^2(\Omega) \), then
\[
\frac{d}{dk} m^{(k)}(x) = \frac{m^{(k)}(x) \nabla \tau(x)}{\tau(x)} \cdot \mathbb{E}_t^{(k)}(x - t) + O \left( \mathbb{E}_t^{(k)}(t - x)^2 \right),
\]
and then, it holds that
\[
m^{(k)}(x) = \frac{1}{\mu(\Omega)} + \frac{\Delta k \nabla \tau(x)}{\mu(\Omega) \tau(x)} \cdot \left( x - \mathbb{E}_t^{(k)}(t) \right) + O \left( \Delta k \| \mathbb{E}_t^{(k)}(t - x)^2 \| \right) + O \left( (\Delta k)^2 \right).
\]

Hence, it holds from (17) and (18) that
\[
\frac{\|\Delta x\|}{\Delta k} = \frac{\|\mathbb{E}_t^{(k)}(t) - x\|}{k + \Delta k} + O \left( \frac{\|\mathbb{E}_t^{(k)}(t) - x\|^2}{k + \Delta k} \right) + O \left( \frac{\|\mathbb{E}_t^{(k)}(t - x)\|^2}{k + \Delta k} \right) + O \left( \frac{\Delta k}{k + \Delta k} \right);
\]
and further, we can obtain the shrink rate as follows:

**Theorem 2.12.** Suppose that $f \in C^2(\Omega)$. If $x \in \Gamma_0^{(k)}$ and $x$ moves to $x + \Delta x \in \Gamma_0^{(k+\Delta k)}$ when $k$ continuously increases to $k + \Delta k$, then the shrink rate

$$
\lim_{\Delta k \to 0} \frac{\|\Delta x\|}{\Delta k} = \frac{\|E_4^{(k)}(t) - x\|}{k} + O\left(\frac{\|E_4^{(k)}(t) - x\|^2}{k}\right) + O\left(\frac{\|E_4^{(k)}(t - x)\|^2}{k}\right),
$$

where $E_4^{(k)}(t) = \int_{\Omega} t'' m^{(k)}(t) dt$ and $\| \cdot \|$ is the Euclidean norm.

**Corollary 2.13.** Under the assumption of Theorem 2.12, if $x^{(k)} \in \Gamma_0^{(k)}$ and $x^{(k)}$ moves to $x^{(k+1)} = x + \Delta x \in \Gamma_0^{(k+1)}$ when $k$ continuously increases to $k + 1$, then

$$
f \left( x^{(k)} \right) - f \left( x^{(k+1)} \right) = O\left( \frac{\|E_4^{(k)}(t) - x^{(k)}\|}{k + 1} \right).
$$

This result can be used to measure the upper bound of the asymptotic convergence rate of an MD based method.

This shows that the shrink rate of $m^{(k)}$ is inversely proportional to $k$. Therefore, the relevant shrinkage is almost independent of $k$ when $\Delta k = (e - 1)k$. In an average sense, it follows from the first mean value theorem for integrals that, the shrinkage from $D_0^{(k)}$ to $D_0^{(e k)}$ can be written as

$$
\int_{k}^{e k} \frac{\|E_4^{(s)}(t) - x(s)\|}{s} ds = \|E_4^{(\zeta)}(t) - x(\zeta)\| \int_{k}^{e k} \frac{1}{s} ds = \|E_4^{(\zeta)}(t) - x(\zeta)\|,
$$

where $x(k) \in \Gamma_0^{(k)}$ is the shrinking path as $k$ continuously increases and $\zeta \in (k, ek)$.

**2.7. A further remark on the minima distribution.** As mentioned above, the minima distribution is regarded as a weak limit of a sequence of nascent minima distribution functions $\{m^{(k)}\}$. At the end of this section, we want to point out that the way to define $m^{(k)}$ is, of course, not unique.

Similar to the Dirac delta function, $m^{(k)}$ can also be defined by a uniform distribution sequence. For any $k \in \mathbb{N}_0$, we recursively define the sequence of sets

$$D_U^{(0)} = \Omega \quad \text{and} \quad D_U^{(k+1)} = \left\{ x \in D_U^{(k)} : f(x) \leq \frac{1}{\mu(D_U^{(k)})} \int_{D_U^{(k)}} f(t) dt \right\},
$$

where $\mu(D_U^{(k)}) = \int_{D_U^{(k)}} dt$ is the $n$-dimensional Lebesgue measure of $D_U^{(k)}$; then a uniform distribution based nascent minima distribution function can be defined as

$$m_U^{(k)}(x) = \begin{cases} \frac{1}{\mu(D_U^{(k)})}, & x \in D_U^{(k)}; \\ 0, & x \in \Omega - D_U^{(k)}. \end{cases}
$$

And it is clear that

$$
\frac{1}{\mu(D_U^{(k)})} \int_{D_U^{(k)}} f(t) dt = \int_{\Omega} f(t) m_U^{(k)}(t) dt.
$$

Similarly, we have the global convergence

$$
\lim_{k \to \infty} \int_{\Omega} f(t) m_U^{(k)}(t) dt = f^*, \quad \forall f \in C(\Omega),
$$
Algorithm 1 MD Optimization

1: Generate \( X \) of candidates in \( D^{(0)} = \Omega \) by sampling \( \hat{m}^{(0)} \)
2: \textbf{while} termination criteria not satisfied, repeat do
3: \hspace{1em} Update \( k = \max(k, 1/\sqrt{\text{Var}(f(X))}) \) and \( \hat{m}^{(k)} \) according to evaluated \( X \)
4: \hspace{1em} Update \( D^{(k)} \) and \( X = X \cap D^{(k)} \) if outerIter \( > 1 \)
5: \hspace{1em} \textbf{while} termination criteria not satisfied, repeat do
6: \hspace{2em} Update approximation \( \hat{f} \) according to evaluated \( X \)
7: \hspace{2em} Generate \( X' \) of candidates in \( D^{(k)} \) by sampling \( \hat{m}^{(k)} \)
8: \hspace{2em} Find the minimizer \( x' = \arg\min_{x \in D^{(k)}} \hat{f}(x) \)
9: \hspace{2em} Assess the quality of \( \hat{f} \) by \( X' \cup x' \) and update \( X = X \cup X' \cup x' \)
10: \hspace{1em} \textbf{end while}
11: \textbf{end while}

The parameters of the above MD algorithm are \([N_0, N, R, M, \text{tol}]\), where \( N_0 \) is the number of samples at the beginning, \( N \) is the number of samples per update, \( R \) is the recognition accuracy as a termination criterion for the inner loop, \( M \) is the max number of samples for the inner loop, \( \text{tol} \) is the tolerance. One will see the performance of the MD algorithm and its parameters setting in section 4.

We use the function in Figure 1 as an example to illustrate how the MD algorithm is performed. Note that all the inner loops here just perform only one iteration. In the first iteration, as shown in Figure 3(a), the initial points \( X = \{X_i\}_{i=1}^{10} \) were uniformly generated on \( D^{(0)} = [0, 5] \), then \( k \) was determined as 0.79 and an approximation \( \hat{m}^{(0, 0.79)} \) were obtained for generating new candidates \( X' \) in \( D^{(0)} \); and an approximation \( \hat{f} \) was generated to determine the next subset and \( X \) was updated as \( X \cup X' \).

In the second iteration, as shown in Figure 3(b), \( k \) was updated to 1.95 and \( \hat{m}^{(1.95)} \) were created from \( X \) for generating \( X' \) in \( D^{(1.95)} \); then \( D^{(1.95)} \) is determined by \( \hat{f} \).
3.2. Gaussian RBF interpolation of \( f \). To approximate \( f \) on any subset of \( \Omega \), we will briefly review the radial basis function (RBF) methods. In recent decades, RBF approximations have become a powerful tool for scattered data. It leads to algebraic convergence orders for finitely smooth functions and exponential convergence orders for infinitely smooth functions [37, 50, 51]; this result comes from a quantified fact that a differentiable function whose derivatives are bounded cannot attain large values if it is small on a sufficiently dense discrete set [37]. And the key to achieving the best accuracy is to choose the right bandwidth for a given dataset.

Let the dataset \( X = \{X_i\}_{i=1}^N \subset D \) and associated function values \( Y_i = f(X_i) \), then a Gaussian RBF interpolant of \( f \) is required to be of the form

\[
\sum_{i=1}^{N} \alpha_i \phi_\sigma(x - X_i), \quad x \in D \subset \mathbb{R}^n,
\]

where the Gaussian kernel \( \phi_\sigma(x) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\frac{\|x\|^2}{2\sigma^2}} \) and the bandwidth \( \sigma > 0 \); further, the coefficients can be determined by enforcing the interpolation constraints, that is,

\[
\alpha = (\alpha_1, \cdots, \alpha_N)^T = \Phi_\sigma^{-1}Y,
\]

where \( \Phi_\sigma = (\phi_\sigma(X_i - X_j))_{ij} \) and \( Y = (Y_1, \cdots, Y_N)^T \). A well-known way for selecting the bandwidth is the method of cross validation which has long been used in the statistics literature. Here we consider a special case of leave-one-out cross validation (LOOCV) proposed by Rippa [39], i.e., the optimal bandwidth is chosen as

\[
\sigma^* = \arg \min_{\sigma \in \mathbb{R}^+} \left\| \frac{\Phi_\sigma^{-1}Y}{\text{diag}(\Phi_\sigma^{-1})} \right\|_1
\]
with the computational complexity $O(N^3)$. Then, the interpolant can be given as

$$\hat{f}(x) = \sum_{i=1}^{N} \alpha_i^* \phi_\sigma(x - X_i), \text{ where } \alpha^* = \Phi_\sigma^{-1}Y.$$

This interpolation, of course, can also be replaced by a Gaussian process [33]. And moreover, one can use the interpolation $\hat{f}(x)$ to update

$$X = X \cap D_{\tau_n}^{(k)} = \left\{ x \in X : e^{-a(f(x)-c)} \geq \mathbb{E}_t^{(k)}(e^{-a(f(t)-c)}) \right\},$$

where $\tau_n$ is given by (14) and $\mathbb{E}_t^{(k)}(e^{-a(f(t)-c)}) \approx \mathbb{E}(e^{-a(f(X)-c)})$ for a given set $X$.

### 3.3. Mixed Gaussian approximation of $m^{(k)}$

To facilitate the sample generation, we will consider the mixed Gaussian approximation of $m^{(k)}$. In low dimensional space, we could develop the classical Nadaraya-Watson (NW) estimator for this purpose. First recall the NW Gaussian kernel estimator [28, 49] of $\tau^k(x) = e^{-kf(x)}$, i.e.,

$$\hat{\tau}_{NW}^k(x; \sigma) = \frac{\sum_{i=1}^{N} e^{-kY_i} \phi_\sigma(x - X_i)}{\sum_{j=1}^{N} \phi_\sigma(x - X_j)},$$

where $Y_i = f(X_i)$. It is often called the “local constant” estimator. Commonly, the closeness of $\hat{\tau}_{NW}^k$ to its target $\tau^k$ is measured by the size of the mean squared error (MSE) and the asymptotic expression of the MSE of $\hat{\tau}_{NW}^k$ can be given as [21, 30]

$$\text{MSE} (\hat{\tau}_{NW}^k) = O(\sigma^4 + \sigma^{-n}N^{-1}).$$

Consider the following convolution

$$\int_{\mathbb{R}^d} \hat{\tau}_{NW}^k(t; \sigma) \phi_\sigma(x - t)dt = \sum_{i=1}^{N} e^{-kY_i} \int_{\mathbb{R}^d} \frac{\phi_\sigma(t - X_i)}{\sum_{j=1}^{N} \phi_\sigma(t - X_j)} \phi_\sigma(x - t)dt$$

$$= \sum_{i=1}^{N} \frac{e^{-kY_i}}{\sum_{j=1}^{N} \phi_\sigma(X_i - X_j)} \phi_\sigma(x - X_i) + O\left(\frac{1}{N\sigma^n}\right),$$

then the regularized NW Gaussian kernel estimator of $\tau^k$ can be further rewritten as the following form

$$\hat{\tau}^k(x; \sigma) = \sum_{i=1}^{N} \alpha_i \phi_\sigma(x - X_i), \text{ where } \alpha_i = \frac{e^{-kY_i}}{\sum_{j=1}^{N} \phi_\sigma(X_i - X_j)}.$$  

(23)

Here (23) is a mixed Gaussian distribution because the coefficients $\alpha$ are non-negative. Similar to Refs. [21, 30], we know (23) has the same convergence order as (22). And the optimal bandwidth can also determined by the LOOCV method [21]. And further, a regularized Gaussian kernel estimator of $m^{(k)}$ can be given as

$$\hat{m}^{(k)}(x) = \sum_{i=1}^{N} \beta_i \phi_\sigma(x - X_i), \text{ where } \beta_i = \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}.$$  

(24)

So the required samples can be efficiently generated using random sequences.
In high dimensions, the accuracy of such a density estimator may not be good enough. Actually, by Theorem 2.3, if \( f \) has a unique minimizer in \( \Omega \), then the sampling distribution \( m^{(k)} \) can be roughly reduced to only one Gaussian distribution, i.e.,

\[
N \left( X_c, \sigma_c^2 \right),
\]

where \( X_c = \sum_{i=1}^{N} X_i e^{-x_i^2} \) and \( \sigma_c^2 = \text{E}(\text{Var}(X)) \). For the case of multiple global minimizers, practical evidence shows that the approach is often more efficient to find the most stable minimizer in the sense of probability, see section 4. Further, reasonable application of quasi-random sequences \([17, 46]\) can help achieve a stable exploration.

4. Numerical examples. We first list some popular test functions in Table 1.

| No. | D | Formula | \( \Omega \) | \( f^* \) | \( X^* \) |
|-----|---|---------|-------------|--------|--------|
| 1   | 2 | \( \cos(x_1^2) + \cos(x_2^2) + 2 \) | \([0, 3.5]^2\) | 0      | \( X_1^* \) |
| 2   | n | \( 20 \exp\left(\frac{-0.02}{\sqrt{n}} \|x\|_2\right) \) | \([-35, 35]^n\) | 0      | \( X_1^* \) |
| 3   | n | \( \sum_{i=1}^{n} |x_i \sin(x_i)| + 0.1|x_i| \) | \([-10, 10]^n\) | 0      | \( X_1^* \) |
| 4   | n | \( \frac{\|x\|^2}{\text{ar}^n} - \prod_{i=1}^{n} \cos\left( \frac{x_i}{\sqrt{7k+1}} \right) + 1 \) | \([-100, 100]^n\) | 0      | \( X_1^* \) |
| 5   | n | \( \|x\|^2 - 10 \sum_{i=1}^{n} \cos(2\pi x_i) + 10n \) | \([-10, 10]^n\) | 0      | \( X_1^* \) |
| 6   | 2 | \( 100(x_2 - x_1^2)^2 + (x_1 - 1)^2 \) | \([-5, 5]^2\) | 0      | \( X_1^* \) |
| 7   | 2 | \( \sin x_1 \exp(1 - \cos x_2)^2 + \cos x_2 \exp(1 - \sin x_1)^2 + (x_1 - x_2)^2 \) | \([-9, 9]^2\) | \( f_7^* \) | \( X_1^* \) |
| 8   | 2 | \( x_1^2(1 - 2.1x_1^2 + \frac{x_1^4}{4}) + x_1 x_2 + 4x_2^2(x_2^2 - 1) \) | \([-3, 3]^2\) | \( f_8^* \) | \( X_1^* \) |
| 9   | 2 | \( [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3(1 - x_1)^2)] \) | \([-2, 2]^2\) | 3      | \( 0, -1 \) |

1. \( X_1^* = \{(\sqrt{\pi}, \sqrt{\pi}), (\sqrt{3\pi}, \sqrt{\pi}), (\sqrt{\pi}, \sqrt{3\pi}), (\sqrt{3\pi}, \sqrt{3\pi})\} \).
2. \( f_7^* = -106.764537 \) and \( X_7^* = \{(4.70104, 3.15294), (-1.58214, -3.13024)\} \).
3. \( f_8^* = -1.03162845 \) and \( X_8^* = \{(-0.0898, 0.7126), (0.0898, -0.7126)\} \).

Here, \( f_1 \) is a very simple 2-D function that has four global minima. \( f_2 - f_5 \) are the Ackley, Alpine, Griewank \([15]\) and Rastrigin functions, respectively. They are some of the most widely used test functions \([52]\) which pose a similar risk for optimization algorithms to be trapped in one of its many local minima. And, \( f_6 \) is the Rosenbrock function \([40]\) and the global minima lies in a long, narrow and parabolic shaped flat valley; \( f_7 \) is the Bird function which has some sharp local minima; \( f_8 \) is the six-hump Camelback function \([10]\) which has six local minima, and two of which are global; and finally, \( f_9 \) is the Goldstein-Price function \([14]\) which has several local minima, and the global minima lies in the flat bottom.
4.1. Stability. We use the test function $f_1$ to demonstrate the stability of the MD algorithm. As mentioned in subsection 2.4, the solution of the MD algorithm converges to a flatter valley with a larger probability. Now it is very clear to see this phenomenon from Figure 4, and we can find that (i) the magnitude of this probability is proportional to the opening size of the corresponding valley, (ii) the sampling form (25) is much more helpful to find the most stable minimizer, and (iii) the the sampling from (25) based MD is more stable than other acclaimed methods. So we will mainly use the sampling from (25) in the remaining comparisons. And one can see that DE is more stable than BO for the case (but BO converges much faster than DE).

4.2. Comparison with BO. Although they are all model-based strategies, the difference between BO and MD is very obvious in Figure 5. For the complicated Ackley function $f_2$, the global approximation of BO quickly locks the valley bottom, but its approximation accuracy at the valley bottom has not improved for a long time, so its exploration is often trapped in one of the local minima of the valley bottom, see Figure 5(c); in contrast, the strategy of MD can more accurately focus exploration near the global minima, see Figure 5(b), so MD gets a better convergence in this case.

For another example, the smooth Rosenbrock function $f_6$, although BO and MD show similar convergence, see Figure 5(d), the difference between BO and MD is also obvious. It seems that the exploration of BO is mainly concentrated in the whole valley bottom, while the exploration of MD is gradually concentrated around the most stable global minima, combined with the stability; as shown in Figures 5(e) and 5(f)). Although it is still too early to say which one is better, in terms of computational costs, MD’s shrinking strategy is indeed much faster than BO’s global approximation.

An extensive comparison between BO and MD is shown in Figure 6. All examples...
Fig. 5. The MAE is the average of the absolute deviations between the current best and $f^*$, similarly hereinafter. The MAEs of MD and BO are generated by repeating 20 and 3 independent runs, respectively. MD’s parameters setting is $[15n, 15n, 0.67n, 150n, 10^{-6}]$ and the acquisition function of BO is selected as the one that best fits the current target function.

support the conclusion that MD converges faster than BO; but it is also worth noting that each example also indicates that BO is better than MD in the early stage – this is the greatest advantage of global approximation. Objectively speaking, the global approximation, as well as the local one, cannot be suitable for all functions. In addition to the approximation method, the main difference between BO and MD is the use of existing function evaluations, one is probabilistic while the other is deterministic. We do not focus on multidimensional problems here because they usually require more function evaluations while BO’s computational cost is very expensive at this time.

4.3. Comparison with RS and DE. As we mentioned, RS is the most typical strategy of expanding local methods to deal with global problems; and DE was proved to be the fastest evolutionary algorithm in the First International IEEE Competition on Evolutionary Optimization [31, 47]. First, the full comparison in 2-D is shown in Figure 7. All examples support the conclusion that MD converges much faster than RS and DE. This is expected because both RS and DE do not make full use of prior information such as previous function evaluations, as we mentioned above.

The comparison in higher dimensions ($n = 5, 10$ and $20$) is shown in Figure 8. All examples also support the same conclusion that MD converges much faster than RS and DE. Although RS is simple and even effective for some limited applications, one can see that it performs poorly for complex multidimensional problems, after all, it is not a true global method, as we mentioned above.

4.4. Potential improvements. Despite the excellent performance of the MD algorithm, we found that there still exist some problems from the practical tests above. Now let’s summarize them below, which are needed to continue to be considered in future research:
Fig. 6. Comparison on some test functions from Table 1. The MAEs of MD and BO are generated by repeating 20 and 3 independent runs, respectively. And MD is running with the same parameters setting $[10n, 10n, 0.67, 150n, 10^{-6}]$.

1. The MD algorithm is a model-based approach which depends on the accuracy and efficiency of the approximate model. The computational complexities of both the LOOCV based RBF interpolation and GP is $O(N^3)$ for $N$ data points. This computation cost largely hampers the wider application. So the multidimensional approximation is of course a very important foundation, and its advance will greatly improve the revolution of global optimization.

2. At this moment, the performance of the MD algorithm still depends on those several predesigned parameters. With a little experience one could often find a set of feasible parameters for an unknown problem with known dimensions; however, this is inconvenient and can sometimes lead to convergence failure. Hence, it will be very helpful to determine or adjust this set of parameters based on the current prior information.

3. Although there are some advantages to using a single Gaussian distribution to roughly approximate $m(k)$, it is clearly not the best choice. For example, the sampling form (24) is much better than (25) for the Rosenbrock function. Therefore, how to effectively improve this sampling distribution is a very important issue for all potential MD-based methods.

5. Conclusions. In this work, we built an MD theory for global minimization of continuous functions on compact sets and proposed a derivative-free algorithm for finding a stable global minimizer of a possibly highly nonlinear and non-convex function. In some sense, the proposed theory breaks through the existing gradient-based theoretical framework and allows us to reconsider the non-convex optimization. On the one hand, it can be seen as a way to understand existing methods; on the other hand, it may also become a new starting point. Thus, we are convinced that the proposed theory and corresponding algorithm will have a thriving future.

Acknowledgments. The author wishes to thank Dr. X. Xu and the editors for their very valuable suggestions of this work.
Appendix. Here is an example to use our MATLAB code of the MD algorithm.

```matlab
clear; clc;
No = 2;
[TF,infB,Fmin] = TFun(No);
[d,lb,ub] = deal(infB{1},infB{2},infB{3});
opt = [10*d; 10^d; 0.67; 150*d; 1e-4];
if size(lb,1)<d, lb = lb*ones(d,1); end
if size(ub,1)<d, ub = ub*ones(d,1); end
d = {lb; ub};
[X,Y,Hdata] = mdopt(TF,d,opt);
```

And the MATLAB functions that implements the MD algorithm and the set of the test functions in Table 1 are given as follows:

```matlab
function [X,Y,Ydata] = mdopt(TF,d,opt)

%MDOPT finds a global minima of a continuous f on a compact set.
%MDOPT attempts to solve problems of the form:
%min TF(X) subject to: iD{1} <= X <= iD{2} (bounds)
%X iD{3}(X) > 0 (nonlinear constraints)
%where opt = [NO; N; R; M; tol] - NO is the number of samples at
%the beginning, N is the number of samples per update, R is the
%recognition accuracy (0.66<R<1) as a termination criterion (TC)
%for the inner loop, M is the max number of samples for the inner
%loop, tol is the tolerance as a TC for the outer loop.
%Default opt = [20*d; 10*d; 0.8; 100*d; 1e-3]

% References
% X Luo, "Minima distribution for global optimization", arXiv
%:1812.03457
% Xiaopeng Luo, 2/13/2019

disp('mdopt start:')
% parameters setting
k = length(d{1});
if nargin<3, opt = [20*d; 20*d; 0.8; 100*d; 1e-3]; end
[NO,N,R,M,tol] = deal(opt(1),opt(2),opt(3),opt(4),opt(5));

% initialization
k = 0;
iter = 1;
iMk = genmk(1,[],[],NO);  % m^-K
iDk = [];  % D^-K
[nss,X] = splD(1,0,0,iMk,d);
Y = TF(X);  % objective
[xb,yb,xc,yc] = updbest(X,Y,zeros(1,d),inf,inf,inf);  % update best
[rec,HX,HY] = mdrecord([1],0,iter,0,X,Y,yl);

while yc>tol
    % update k
    k = max(k,1/std(Y));
    % update tao^-k for both m^-k and D^-k
    EY = mean(Y); T = exp(-k*(Y-EY)); MT = mean(T); IF = isnan(T);
    if max(IF)
        T(IF) = 1; T(~IF) = 0; MT = .5;
    end
    % update the current best and generate m^-k for sampling
    [xb,yb,xc,yc] = updbest(X,Y,xb,yb,xc,yc);
iMk = genmk(0,X,T,N);
    % update D^-k and samples if iter>1
    if iter>1
        [iDk,1] = deal([MT;k;EY;rbf],T>=MT);
        [X,Y] = deal(X1,:),Y(1)
    end
end
```

Minima Distribution

19
% inner loop
[RA, inneriter] = deal(0, 0);
while RA < R && size(X, 1) < M && yc > tol
    inneriter = inneriter + 1;
    % update the approximation
    rbf = adrbf(X, Y);
    % sampling in D^k
    [nss, Xn] = splD(0, inneriter == 1, nss + 1, iMk, iD, iDk);
    eYn = rbf(Xn, rbf);
    % taking minima of the approximation
    iMk0 = {100 * iMk(1); iMk(2); iMk(3)/2};
    [-, Xn2] = splD(0, 1, iMk0, iD, iDk);
    eYn2 = rbf(Xn2, rbf);
    [-, I2min] = min(eYn2, [], 'omitnan');
    [-, Imax] = max(eYn, [], 'omitnan');
    Xn(Imax, :) = Xn2(I2min, :);
    eYn(Imax) = eYn2(I2min);
    % call target function
    Yn = TF(Xn);
    [xb, yb, xc, yc] = updbest(Xn, Yn, xb, yb, xc, yc);
    % termination criteria for inner loop
    eTn = exp(-k * (eYn - EY));
    Tn = exp(-k * (Yn - EY));
    T = [T; Tn];
    [MT, ET] = deal(mean(T), mean(abs(eTn - Tn)));
    % recognition accuracy (RA)
    if k < 10
        RA = sum((eTn > MT) == (Tn > MT)) / length(Tn);
    else
        It = Tn > MT;
        V1 = (eTn(It) - ET) > MT;
        V2 = (eTn(~ It) + ET) <= MT;
        RA = (sum(V1) + sum(V2)) / length(Tn);
    end
    % update, record, and plot
    [X, Y] = deal([X; Xn], [Y; Yn]);
    [rec, HX, HY] = mdrecord(rec, iter, inneriter, k, Xn, Yn, yb, HX, HY);
    mdplot(rec);
end
% update iter
fprintf(' - iteration %d with the best obj = %d; \n', iter, yb);
iter = iter + 1;
end
% end main function

% subfunctions:
function [nss, Xn] = splD(st, fp, nss, iMk, iD, iDk)
%SPLD generates samples Xn by sampling Mk in Dk
if st == 1 % for the beginning
    if nss == 0, nss = randi(100); end
    [LB, UB, d] = deal(id(1, 1), iD(2, 1), length(id(1, 1)));
    ax = net(haltonset(d, ' Skip', nss + 1), iMk(1));
    ax = bsxfun(@plus, bsxfun(@times, ax, (UB - LB)'), LB);
    if length(iD) > 2, ax = ax(id(3, 1)(ax, :)); end
    Xn = ax; nss = nss + size(ax, 1);
else
    [beta, xi, sig] = deal(iMk(1), iMk(2), iMk(3));
    [d, Xn, i] = deal(size(xi, 2), [1], 1);
    while i < length(beta)
        if fp == 1 && i == 1
            nx = beta(i) - 1; X0 = xi(i, :);
        end
    end
end

else
    nx = beta(i); X0 = [];
end

counter = 0; iDk0 = iDk;
while nx>0
    axi = net(haltonset(d,'Skip',nss+1),nx);
    nss = nss + nx;
    axi = norminv(axi,Xi(i,:),sig(i));
    axi = inD(axi,iD,iDk0);
    if isempty(axi), counter = counter + 1; end
    if counter>10 && isempty(iDk)
        counter = 0; iDk0{1} = 0.9*iDk0{1}; sig(i) = 0.9*sig(i);
        if iDk0{1}<1e-6 || sig(i)<1e-6
            axi = net(haltonset(d,'Skip',nss+1),nx);
            nss = nss+nx;
            axi = norminv(axi,Xi(i,:),sig(i));
        end
    end
    if isempty(axi)
        nx = nx-size(axi,1); X0 = [X0; axi];
    end
end  
X0 = X0(1:beta(i,:)); Xn = [Xn; X0]; i=i+1;
end

function X = inD(X,iD,iDk)
if isempty(X), return; end
L = iD{1}; U = iD{2}; d = size(X,2);
if length(iD)>2, iDn = iD{3}; end
if isempty(iDk)
    [TM,a,EY,rbf] = deal(iDk{1},iDk{2},iDk{3},iDk{4});
end
X = X(sum(bsxfun(@ge,X,L') ,2)==d,:);
X = X(sum(bsxfun(@le,X,U') ,2)==d,:);
if length(iD)>2, X = X(iDn(X,:)); end
if isempty(iDk)
    Y = rbfp(X,rbf); T = exp(-a*(Y-EY)); X = X(T>=TM,:);
end
end
function [xb,yb,xchg,ychg] = updbest(X,Y,xb,yb,xchg,ychg)
    [~,I] = min(Y);
    if Y(I)<yb
        [xchg,ychg] = deal(norm(X(I,:)-xb),yb-Y(I));
        [xb,yb] = deal(X(I,:),Y(I));
    end
end
function iMk = genmk(st,X,T,N)
if st==1 % for the beginning
    iMk = {N};
else
    Xc = (T/sum(T))’ * X; Dc = mean(std(X));
    [~,I] = max(T); Xc2 = X(I,:); N2 = 1;
    iMk = {{N-N2, N2}, [Xc; Xc2]; [Dc; Dc/2]};
end
end
function rbf = adrbf(X,Y)
    rbf = fitrgp(X,Y,’KernelFunction’,’squaredexponential’,...
‘FitMethod’,’exact’,’Standardize’,1);
end
function y = rbfp(x,rbf)
y = predict(rbf,x);
end
function mdpplot(rec)
figure(1)
plot(rec(:,4),rec(:,6),'b- .');
set(gca,'yaxislocation','right');
xlabel('Function evaluations');
ylabel('Min observed objective');
title('Minima distribution optimization');
end
function [rec,HX,HY] = mdrecord(rec,SN,iter,k,X,Y,ybest,HX,HY)
if ~isempty(rec)
[smk,emk] = deal(rec(end,4)+1,rec(end,4)+length(Y));
[rec,HX,HY] = deal([rec; SN iter smk emk k ybest],[HX; X],[HY; Y]);
else
[rec,HX,HY] = deal([SN 0 1 length(Y) k ybest],X,Y);
end
end
function [fun,infB,Fmin] = TFun(No,X)
switch No
  case 1
    d = 2; lb = 0; ub = 3.5;
    TXmin = [sqrt(pi) sqrt(pi); sqrt(3*pi) sqrt(3*pi)];
    TYmin = [0; 0; 0; 0];
    TF = @(x) cos(x(:,1).^2) + cos(x(:,2).^2) + 2;
  case {2, 'Ackley', 'ackley', 'ACKLEY'}
    d = 2; lb = -35; ub = 35;
    TXmin = zeros(1,d); TYmin = 0;
    TF = @(x) -20*exp(-0.02*sqrt(sum(x.^2)/d)) ... -exp(sum(cos(2*pi*x)/2)/d)+20+exp(1);
  case {3, 'Alpine', 'alpine', 'ALPINE'}
    d = 2; lb = -50; ub = 50;
    TXmin = zeros(1,d); TYmin = 0;
    TF = @(x) sum(abs(x.*sin(x)) + 0.1*abs(x),2);
  case {4, 'Griewank', 'griewank', 'GRIEWANK'}
    d = 2; lb = -100; ub = 100;
    TXmin = zeros(1,d); TYmin = 0; nn = 1:d;
    TF = @(x)sum(x.^2)/4000 - prod(cos(bsxfun(@times,...
      x,1./sqrt(nn))))/2+1;
  case {5, 'Rastrigin', 'rastrigin', 'RASTRIGIN'}
    d = 2; lb = -10; ub = 10;
    TXmin = zeros(1,d); TYmin = 0;
    TF = @(x)sum(x.^2-10*cos(2*pi*x),2) + 10*d;
  case {6, 'Rosenbrock', 'rosenbrock', 'ROSENBROCK'}
    d = 2; lb = -5; ub = 5;
    TXmin = ones(1,d); TYmin = 0;
    TF = @(x) 100*(x(:,2)-x(:,1).^2).^2+exp(-100*(x(:,1) - x(:,1).^2)).^2+exp(-100*(x(:,2) - x(:,2).^2)).^2;
  case {7, 'Bird', 'bird', 'BIRD'}
    d = 2; lb = -9; ub = 9;
    TXmin = [4.70104, 3.15294; -1.58214, -3.13024];
    TYmin = [-106.764537; -106.764537];
    TF = @(x) sin(x(:,1)).*exp((1-cos(x(:,1))).^2)+... cos(x(:,2)).*exp((1-sin(x(:,1))).^2)+(x(:,1)-x(:,2)).^2;
  case {8, 'Camel6', 'camel6', 'CAMEL6'}
    d = 2; lb = -3; ub = 3;
    TXmin = [-0.0898, 0.7126; 0.0898, -0.7126];
    TYmin = [-1.0316284635; -1.0316284635];
    TF = @(x) (4-2.1*x(:,1).^2+x(:,1).^4/3).*x(:,1).^2 +... x(:,1).*x(:,2)+(4*x(:,2).^2-2-4).*x(:,2).^2;
  case {9, 'Goldstein Price', 'Goldstein', 'goldstein', 'GOLDSTEIN'}
    d = 2; lb = -2; ub = 2;
    TXmin = [0 -1]; TYmin = 3;
    TF = @(x) (1+x(:,1)+x(:,1)).^2+... 3*x(:,1).^2-14*x(:,1)+... 3*(1-x(:,1)).^2-14*(x(:,1)^2+6*x(:,1).*x(:,2)+3*x(:,2)^2)-30*(3+2*x(:,1)+3*x(:,2)^2));
end
12*x(:,1).^2+48*x(:,2)-36*x(:,1).*x(:,2)+27*x(:,2).^2));
otherwise
disp('Does not exist this function!')
end
if nargin>1, fun = TF(X); else, fun = TF; end
if nargin>1, infB = {d; lb; ub}; 
if nargin>2, Fmin = {TXmin; TYmin}; end
end

REFERENCES

[1] R. L. Anderson, Recent advances in finding best operating conditions, J. Am. Stat. Assoc., 48 (1953), pp. 789–798.
[2] J. Azimi, A. Fern, and X. Z. Fern, Batch bayesian optimization via simulation matching, IEEE Trans. on Syst., 1 (2010), pp. 109–117.
[3] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization, Journal of Machine Learning Research, 13 (2012), pp. 281–305.
[4] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, New York, 2004.
[5] S. H. Brooks, A discussion of random methods for seeking maxima, Operations Research, 6 (1958), pp. 244–251.
[6] A. D. Bull, Convergence rates of efficient global optimization algorithms, J. Mach. Learn. Res., 12 (2011), pp. 2879–2904.
[7] D. D. Cox and S. John, SDO: A statistical method for global optimization, in Multidisciplinary Design Optimization: State-of-the-Art, 1997, pp. 315–329.
[8] S. Das, S. S. Mullick, and P. N. Suganthan, Recent advances in differential evolution - an updated survey, Swarm and Evolutionary Computation, 27 (2016), pp. 1–30.
[9] S. Das and P. N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. on Evolutionary Computation, 15(1) (2011), pp. 4–31.
[10] L. Dixon and G. P. Szegö, Towards global optimization, in Proceedings of a Workshop at the University of Cagliari, Italy, North Holland, 1975.
[11] P. Feliot, J. Bect, and E. Vazquez, A bayesian approach to constrained single- and multi-objective optimization, Journal of Global Optimization, 67 (2017), pp. 97–133.
[12] C. A. Floudas and C. E. Gounaris, A review of recent advances in global optimization, Journal of Global Optimization, 45 (2009), pp. 3–38.
[13] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[14] A. Goldstein and J. Price, On descent from local minima, Mathematics of Computation, 25 (1971), pp. 569–574.
[15] A. O. Grie wang, Generalized descent for global optimization, JOTA, 34 (1981), pp. 11–39.
[16] J. Gurland, The teacher’s corner: an inequality satisfied by the expectation of the reciprocal of a random variable, The American Statistician, 21(2) (1967), pp. 24–25.
[17] J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numer. Math., 2 (1960), pp. 84–90.
[18] D. R. Jones, M. Schonlau, and W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global Optimization, 13 (1998), pp. 455–492.
[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science, 220 (1983), pp. 671–680.
[20] H. J. Kushner, A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise, J. Basic Engineering, 86 (1964), pp. 97–106.
[21] Q. Li and J. S. Racine, Nonparametric Econometrics: Theory and Practice, Princeton University Press, Princeton, N.J., 2007.
[22] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot, Auton. Robots, 27(2) (2009), pp. 93–103.
[23] S. F. Marsi, G. A. Bekey, and F. B. Safford, A global optimization algorithm using adaptive random search, Applied Mathematics and Computation, 7 (1980), pp. 353–375.
[24] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[25] J. Mockus, On bayesian methods for seeking the extremum, Optimization Techniques, (1974),
[26] J. Mockus, V. Tiesis, and A. Zilinskas, The application of bayesian methods for seeking the extremum, Toward Global Optimization, 2 (1978), pp. 117–129.

[27] V. A. Mutseniyeks and L. A. Rastrigin, Extremal control of continuous multi-parameter systems by the method of random search, Engineering Cybernetics, 1 (1964), pp. 82–90.

[28] E. A. Nadaraya, On estimating regression, Theory of Probability & Its Applications, 9(1) (1964), pp. 141–142.

[29] T. D. Nguyen, S. Gupta, S. Rana, and S. Venkatesh, Stable bayesian optimization, International Journal of Data Science and Analytics, 6 (2018), pp. 327–339.

[30] A. Pagan and A. Ullah, Nonparametric Econometrics, Themes in Modern Econometrics, Cambridge University Press, Cambridge, 1999.

[31] K. Price and R. Storn, Minimizing the real functions of the ICEC’96 contest by differential evolution, in Proceedings of IEEE International Conference on Evolutionary Computation (ICEC’96), May 1996, pp. 842–844.

[32] K. Price, R. Storn, and J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer, Berlin, Heidelberg, 2005.

[33] C. E. Rasmussen and C. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, MA, 2006.

[34] L. A. Rastrigin, Extremal control by the method of random scanning, Automation and Remote Control, 21 (1960), pp. 891–896.

[35] L. A. Rastrigin, The convergence of the random search method in the extremal control of a many-parameter system, Automation and Remote Control, 24 (1963), pp. 1337–1342.

[36] I. Rechenberg, Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973.

[37] C. Rieger and B. Zwicknagl, Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning, Adv Comput Math, 32 (2010), pp. 103–129.

[38] L. M. Rios and N. V. Sahinidis, Derivative-free optimization: a review of algorithms and comparison of software implementations, Journal of Global Optimization, 56 (2013), pp. 1247–1293.

[39] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Advances in Computational Mathematics, 11 (1999), pp. 193–210.

[40] H. H. Rosenbrock, An automatic method for finding the greatest or least value of a function, The Computer Journal, 3(3) (1960), pp. 175–184.

[41] G. Schrack and M. Choi, Optimized relative step size random searches, Mathematical Programming, 10 (1976), pp. 230–244.

[42] M. A. Schumer and K. Steiglitz, Adaptive step size random search, IEEE Transactions on Automatic Control, AC-13 (1968), pp. 270–276.

[43] H. P. Schwefel, Evolution and Optimum Seeking, Wiley-Interscience, New York, 1995.

[44] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE, 104 (2016), pp. 148–175.

[45] B. V. Sheela, An optimized step size random search (OSSRS), Computer Methods in Applied Mechanics and Engineering, 19 (1979), pp. 99–106.

[46] I. M. Sobol, Distribution of points in a cube and approximate evaluation of integrals, Zh. Vych. Mat. Mat. Fiz., 7 (1967), pp. 784–802.

[47] R. Storn and K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization, 11 (1997), pp. 341–359.

[48] M. Thomas and R. Schwartz, A method for efficient bayesian optimization of self-assembly systems from scattering data, BMC Systems Biology, 12 (2018), pp. 65–82.

[49] G. S. Watson, Smooth regression analysis, Sankhyā: The Indian Journal of Statistics, Series A, 26(4) (1964), pp. 359–372.

[50] H. Wendland, Scattered Data Approximation, Cambridge Monogr. Appl. Comput. Math. 17, Cambridge University Press, Cambridge, UK, 2005.

[51] H. Wendland and C. Rieger, Approximate interpolation with applications to selecting smoothing parameters, Numer. Math., 101 (2005), pp. 729–748.

[52] X. S. Yang, Nature-Inspired Optimization Algorithms, Elsevier Insight, London, 2014.
No. 1, n = 2
No. 2, n = 2
No. 3, n = 2
No. 4, n = 2
No. 5, n = 2
No. 6, n = 2
No. 7, n = 2
No. 8, n = 2
No. 9, n = 2

Fig. 7. Comparison on the test functions from Table 1. The MAEs are generated by repeating 20 independent runs. RS is running with 50 parallel local searches which start with 50 randomly selected points from the 2-dimensional Halton sequence, respectively. DE is running with the same parameters setting $NP = 25$, $F = 0.5$ and $CR = 0.7$ with $tol = 10^{-5}$. And MD is running with the same parameters setting $[10n, 10n, 0.67, 150n, 10^{-6}]$. 
Fig. 8. Comparison on some test functions from Table 1. The MAEs are generated by repeating 20 independent runs. RS is running with 50 parallel local searches which start with 50 randomly selected points from the relevant Halton sequence, respectively. DE is running with the same parameters setting $NP = 20$, $F = 0.5$ and $CR = 0.1$ with $tol = 10^{-6}$. And MD is running with the parameters setting $[10m, 10m, 0.67, 150m, 10^{-5}]$ for $f_2$ and $f_3$, $[100n, 10m, 0.67, 200n, 10^{-5}]$ for $f_4$ and $[100n, 10m, 0.67, 150n, 10^{-5}]$ for $f_5$. 