Multistart Algorithm for Identifying All Optima of Nonconvex Stochastic Functions

Prateek Jaiswal[*1] and Jeffrey Larson[2]

[1]Department of Statistics, Texas A&M University, College Station, TX 77843, USA
[2]Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439

January 5, 2022

Abstract

We propose a multistart algorithm to identify all local minima of a constrained, nonconvex stochastic optimization problem. The algorithm uniformly samples points in the domain and then starts a local stochastic optimization run from any point that is the "probabilistically best" point in its neighborhood. Under certain conditions, our algorithm is shown to asymptotically identify all local optima with high probability; this holds even though our algorithm is shown to almost surely start only finitely many local stochastic optimization runs. We demonstrate the performance of an implementation of our algorithm on nonconvex stochastic optimization problems, including identifying optimal variational parameters for the quantum approximate optimization algorithm.

1 Introduction

We consider the problem of identifying all local minima of the constrained stochastic optimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}}\, f(x) := \mathbb{E}_{\xi}\left[F(x, \xi)\right], \text{ subject to } x \in \mathcal{D}, \tag{SO}$$

where $F(x, \xi)$ is the observable random function, ξ is a random variable defined on the probability space $(\Omega, \mathbf{F}, P_{\xi})$, and $\mathcal{D} \subset \mathbb{R}^d$ is compact.

Identifying all local minima is relevant in various engineering and scientific applications. For instance, Gheribi et al. [13] describe a multicomponent chemical thermodynamics system where all low-melting compositions—each corresponding to a local minimum of their objective—are desired. While the system in [13] assumes that the evaluation of temperatures is deterministic, in practice, such temperatures measurements would be are accompanied by stochastic noise. Furthermore, in biophysics and biochemistry, an important problem is to understand the transitions in the shape of a protein molecule as it folds itself from a disordered (high-energy) state to a native (least-energy) state. The energy function has multiple local minima, and each local minimum corresponds to an intermediate stable state of a protein molecule. Identifying each of them can be crucial for understanding the protein folding pathways [1, 8]. Li et al. [22] highlight the stochastic nature of the protein folding process. Moreover, as a direct consequence of identifying all local minima, we obtain global minimizers of $f(x)$ as well. Identifying such minima is important in various modern applications, such as parameter tuning of stochastic event simulators [4, 17, 21, 24] and identifying optimal parameters within a quantum approximate optimization algorithm (QAOA) [6, 7].

In this paper, we propose a multistart algorithm for nonconvex stochastic optimization (MANSO) to identify all the local minima of $f(x)$ in $\mathcal{D}$ using state-of-the-art local stochastic optimization methods. (Definition 2 formally defines the meaning of "identifying" a local minimum.) It is designed to extend the popular multistart algorithm Multi-Level Single Linkage (MLSL) [31, 32], which guarantees to find all the

---

*Corresponding author: `jaiswalp@tamu.edu`

local minima of a nonconvex deterministic function. MANSO uniformly samples points in $\mathcal{D}$ and starts a local optimization method, such as stochastic gradient descent, from points that are considered the best points in their neighborhood. By design, MANSO seeks to use as few local optimization runs as possible by carefully shortlisting sampled points based on the noisy observations of $f(x)$. In particular, like MLSL, we construct a specific sequence of radii $\{r_k\}$, where k is the number of iterations of MANSO, which enables us to certify that a uniformly sampled point is indeed probabilistically best to start a local stochastic optimization run.

We also provide theoretical guarantees on the performance of MANSO. Primarily, under assumptions that the stationary points of the true objective function f in $\mathcal{D}$ are sufficiently separated and the local stochastic optimization method is guaranteed to converge to the first-order stationary points with high probability, we prove that MANSO identifies all local minima of f on $\mathcal{D}$ with high probability. Furthermore, we show that MANSO does so while starting only finitely many local stochastic optimization runs.

In addition, we demonstrate the performance of MANSO on two benchmark nonconvex stochastic optimization problems (up to 10 dimensions) using a derivative-free adaptive sampling trust-region stochastic optimization algorithm (ASTRO-DF) [33] as a local stochastic optimization method.

Related Work In general, a nonconvex optimization problem is NP-hard even in the deterministic case, where the true function $f(x)$ can be evaluated at any point $x \in \mathcal{D}$. In the past few decades, extensive research has been conducted to develop stochastic algorithms for solving nonconvex deterministic optimization problems, with the aim of finding global optima. Such stochastic methods can be broadly categorized into multistart [31, 32, 20] and Bayesian optimization(BO) [19, 10] methods. Also, some recent works leverage BO techniques for exploring the domain in the multistart framework [25]. In contrast, the parallel work for stochastic nonconvex optimization still requires much attention.

In the deterministic case, one of the popular approaches is to use a stochastic search technique where randomly sampled points in a compact domain are explored with the help of a local search method and a specific set of rules. These rules are designed to qualify any new randomly sampled point to start a local search, and they depend on the already observed values of the points in the neighborhood of the point being tested. With the help of such rules, these types of algorithms try to avoid finding the same local minima multiple times. Such types of algorithms are known as *multistart* algorithms [31, 32, 23]. Among various multistart algorithms, one of the most popular is MLSL [31, 32], which provides asymptotic guarantees in identifying all the local minima assuming all stationary points of the objective function are separated by a positive distance. Locatelli [23] improves upon this assumption of separated stationary points and introduces N-MLSL (non-monotonic MLSL). Parallel implementation of MLSL is also discussed in [20] with asymptotic guarantees to find all local minima when the true function $f(x)$ can be concurrently evaluated at any $x \in \mathcal{D}$, unlike MLSL. Mathesen et al. [25] propose a novel approach, where the restart points are decided based on ideas developed in BO literature [10, 27, 35]. Instead of randomly selecting the restart points from the search domain, their method chooses points based on a surrogate Gaussian process model of the objective function. Naturally, their surrogate model is updated sequentially using the observed function values at points that are already evaluated. A detailed review of such algorithms, including significant recent developments, can be found in [10]. Similarly, the methods in [18, 29, 30] consider other approaches for utilizing a surrogate model within a multistart framework. Multistart methods that seek improved efficiency have considered early termination of local searches [38] and tunneling and evolutionary strategies [36]. We note that these methods are for deterministic objectives. Extending such approaches to the stochastic setting may yield similar improvements.

Our algorithm extends the MLSL algorithm to the case when we have access only to noisy function values. Like MLSL, we also rely on state-of-the-art local stochastic optimization techniques with first- and second-order convergence guarantees [16, 12, 11, 33]. However, to the best of our knowledge, no work has developed a multistart algorithm for stochastic function evaluations.

Here is a brief roadmap of the paper. In the next section, we define notations and definitions used in the paper, followed in Section 3 by the assumptions required to prove the theoretical properties of MANSO. In Section 4 we describe the details of our proposed method MANSO, and in Section 5 we establish its theoretical properties. In Section 6.5 we describe the parameters of the quantum approximate optimization algorithm, which we will use for testing the MANSO algorithm. In Section 6.1 we present our numerical results. We conclude in Section 7 with a summary and brief description of further work.

2 Notations and Definitions

We use capital calligraphic letters to denote Lebesgue measurable subsets of $\mathbb{R}^d$. The volume (Lebesgue measure) of a set $\mathcal{A}$ is denoted as $m(\mathcal{A})$. We use $|A|$ to denote the cardinality of a discrete set A. Unless otherwise stated, $\|\cdot\|$ denotes the Euclidean norm. We denote the set of natural numbers as $\mathbb{N}$.

We use $\text{Var}_\xi[\cdot]$ and $\text{Cov}_\xi[\cdot, \cdot]$ to represent the variance and covariance, respectively, with respect to the random variable ξ. If $\{A_n(\xi)\}_{n \in \mathbb{N}}$ is a sequence of events and $P_\xi(\{A_n(\xi)\})$ is the probability of event n occurring, then this sequence of events occurs with high probability (w.h.p.) if for any $\delta > 0$ there exists an $n_0 \in \mathbb{N}$ such that $P_\xi(A_n(\xi)) > 1 - \delta$ for any $n \geq n_0$. We index the probability measure of the random variable ξ by ξ itself just to differentiate it from the uniform sampling measure used subsequently in the paper. Moreover, the samples space of the random variable ξ, Ω is arbitrary and the range of ξ is an arbitrary Borel measurable set. When a sequence $\{a_t\}$ is $O(b_t)$, it implies that there exist a $K > 0$ and $t_0 \geq 1$ such that $\forall t \geq t_0$, $a_t \leq Kb_t$. We let $\hat{f}_n(\cdot)$ denote an estimate of $f(x)$ constructed using n i.i.d. measurements of $F(x, \xi)$.

We now establish notation to be used to describe the MANSO algorithm. Let $k \in \mathbb{N}$ denote the number of iterations of MANSO. Let $S_k \subseteq \mathcal{D}$ be the collection of uniformly sampled points up to and including iteration k. Let $A_k \subseteq S_k$ be the collection of sampled points from which a local stochastic optimization method has been started and is still active at iteration k, and let L_k denote the set of points generated from all the local stochastic optimization runs up to and including iteration k. Let $X^* \subset \mathcal{D}$ denote the set of local minima of f in $\mathcal{D}$ and $\hat{X}_k^*$ denote the set of local minima identified before iteration k of MANSO. Let Y^* denote the set of stationary points of f on $\mathcal{D}$ that are not local minima. We let $\{\mathbf{X}_i^a, i \in \mathbb{N}\}$ denote the random sequence of iterates generated by a given local stochastic optimization method started from $a \in \mathcal{D}$. $\{\mathbf{X}_i^a, i \in \mathbb{N}\}$ is a stochastic process defined on the probability space $(\Omega, \mathbf{F}^a, P_\xi)$. Let $\{x_i, i \in \mathbb{N}\}$ be its realization. We represent the filtration (information) available at iteration i of the local stochastic optimization method started from $a \in \mathcal{D}$ as $\{\mathbf{F}_i^a\}$, which is an increasing family of σ-algebras of $\{\mathbf{F}^a\}$ on which the stochastic process $\{\mathbf{X}_i^a\}$ is defined. We also call $\{\mathbf{F}_i^a\}$ a *filtration* at iteration i of the local stochastic optimization (LSO) run started from $a \in \mathcal{D}$.

We now list other important notation:

- Let $r_k = \frac{1}{\sqrt{\pi}} \sqrt[d]{\Gamma(1 + d/2)m(\mathcal{D})\sigma \frac{\log|S_k|}{|S_k|}}$ for some fixed $\sigma > 4$, where Γ is the gamma function. The radius is used in Lemma 2 to define the neighborhood of a candidate "probabilistically best" point.

- Let $\partial\mathcal{D}$ denote the boundary of the set $\mathcal{D}$.

- Let $\mathcal{B}(y; r) := \{x \in \mathcal{D} : \|x - y\| \leq r\}$ represent a ball of radius r centered at any $y \in \mathcal{D}$.

- For $\tau > 0$, let

$$\mathcal{Q}_\tau := \cup_{y \in \partial\mathcal{D}}\{x \in \mathcal{D} : \|x - y\| < \tau, \}. \tag{1}$$

  be the points in $\mathcal{D}$ within τ of the boundary.

- We denote η as the minimum distance between any two distinct stationary points of f on $\mathcal{D}$, that is,

$$\eta = \min_{\{x,y\} \in X^* \bigcup Y^*, x \neq y} \|x - y\|.$$

Next, we define the *domain of attraction for method M* for any local minima.

Definition 1 (Domain of attraction). For any $x^* \in X^*$, the domain of attraction $\mathcal{L}_{x^*}$ is defined as a subset of $\mathcal{D}$ such that if a local stochastic method M is started from any point in it, then it will converge to the local minimum x^* w.h.p. Formally, for any $\epsilon > 0$,

$$\mathcal{L}_{x^*} = \bigcup \left\{ a \in \mathcal{D} : \lim_{k \to \infty} P_\xi(\|\mathbf{X}_k^a - x^*\| > \epsilon) = 0 \right\}. \tag{2}$$

Note that the domain of attraction is defined only for those points that the local stochastic method M will converge to with high probability. Moreover, it is not required that the respective domain of attraction for each $x^* \in X^*$ partition $\mathcal{D}$. We consider Definition 1 to be a reasonable stochastic extension of the domain of attraction for the deterministic case considered in the original MLSL paper and its extensions (e.g., [31, Theorem 4], [23], [20, Assumption 2]).

We also define the event $\omega - identifying\ a\ local\ minimum$, which we use throughout. In our algorithm, ω is a tuning parameter and is given as input by the user.

Definition 2 (ω-identifying local minima). Let $\omega \leq \frac{\eta}{2}$ be fixed, and let $\{\mathbf{X}_i^a\}$ be the sequence of iterates generated by the local method M started from the point $a \in \mathcal{L}_{x^*}$. A local method M has identified the local minimum x^* when the event $\{\|\mathbf{X}_i^a - x^*\| < \omega\}$ occurs for all $i \geq i_0$.

3 Assumptions

We now state the assumptions needed for our theoretical analysis. We group them into assumptions about the objective and domain of the problem (SO), assumptions about the true function f and its estimate computed by using $F(x, \xi)$ at $x \in \mathcal{D}$, and assumptions about the local stochastic optimization method used within the multistart framework.

Assumption 1. We first impose the following conditions on the problem (SO).

1. $\mathcal{D}$ is a compact set, and f is twice continuously differentiable on $\mathcal{D}$.

2. The minimum distance between any two distinct stationary points is positive, that is, $\eta > 0$.

3. There exists $\tau > 0$ such that $(X^* \bigcup Y^*) \bigcap \mathcal{Q}_\tau = \emptyset$, for $\mathcal{Q}_\tau$ defined in (1).

Because the set $\mathcal{D}$ is compact by Assumption 1.1, Assumption 1.2 implies that f has finitely many stationary points in $\mathcal{D}$ and that f is not flat in $\mathcal{D}$. Assumption 1.3 ensures that there are no stationary points near the boundary of $\mathcal{D}$. (The parts of Assumption 1 are the same as those considered in the deterministic case [31].) It is useful to have notation for the set of points in $\mathcal{D}$ within ω of a stationary point of f: For any $\omega \in (0, \eta)$, let

$$\mathcal{T}_\omega := \cup_{x \in \{X^* \cup Y^*\}} \mathcal{B}(x; \omega) \tag{3}$$

Assumption 2. For any $x \in \mathcal{D}$, we assume that the estimate $\hat{f}_n(x)$ of $f(x)$, which is constructed by using n i.i.d. measurements of the measurable function $F(x, \xi)$, satisfies

1. $\mathbb{E}_\xi \left[\hat{f}_n(x) \right] = f(x)$

2. $\text{Var}_\xi \left[\hat{f}_n(x) \right] < \infty$.

Since $\hat{f}_n(\cdot)$ is constructed by using n i.i.d. measurements of the measurable function $F(\cdot, \xi)$, this assumption requires $F(\cdot, \xi)$ to satisfy some regularity conditions. For instance, $\hat{f}_n(\cdot) = \frac{1}{n} \sum_{i=1}^n F(\cdot, \xi_i)$ is an unbiased estimate of f and $\text{Var}_\xi \left[\hat{f}_n(x) \right] < \infty$ if $\text{Var}_\xi [F(x, \xi_i)] < \infty$.

Assumption 3. We make the following assumptions about the local stochastic optimization method M. If $\{\mathbf{X}_i^a\}$ is a random sequence of iterates produced by M when started from $a \in \mathcal{L}_{x^*}$, then it satisfies the following.

1. For any $x^* \in X^*$, $m(\mathcal{L}_{x^*}) > 0$ and $m(\mathcal{L}_{x^*} \setminus \mathcal{Q}_\tau) > 0$

2. For any two local minima $\{x^*, y^*\} \in X^*$, $\mathcal{L}_{x^*} \cap \mathcal{L}_{y^*} = \emptyset$.

3. For any $\nu > 0$ there exist an $i_0 \in \mathbb{N}$, $\omega \in (0, \frac{\eta}{2})$, and a sequence $\{\Lambda_i\}$ satisfying $\Lambda_i \in (0, 1)$ and $\lim_{i \to \infty} \Lambda_i = 0$ such that

Table 1: Conditions defined by tolerances r_k, $\tau > 0$, $\omega > 0$, $n \in \mathbb{N}$, and $\beta \in (0, 1/2)$ to be checked by Algorithm 1 before starting a local optimization run.

> **S1** $\nexists$ a point $z \in \mathcal{B}(a; r_k) \cap (S_k)$, such that $P_\xi \left(\hat{f}_n(z) - \hat{f}_n(a) > \sqrt{\frac{\mathrm{Var}_\xi [\hat{f}_n(z) - \hat{f}_n(a)]}{\beta}} \right) \leq \beta$,
>
> where n is the number of random samples of $f(\cdot)$ at respective points.
>
> **S2** $a \notin \cup_{x \in \hat{X}_{k-1}^*} \mathcal{B}(x; \omega)$, for a given $\omega < \frac{\eta}{2}$, where $\hat{X}_{k-1}^*$ is the collection of approximate local minima up to iteration $(k-1)$.
>
> **S3** $a \notin \mathcal{Q}_\tau$, that is, near the boundary of set $\mathcal{D}$.
>
> **S4** a has not started any LSO.

(i) $P_\xi \{ \|\nabla f(\mathbf{X}_i^a)\|^2 < \nu | a \in \mathcal{L}_{x^*} \} \geq 1 - \Lambda_i$ and

(ii) $P_\xi \left\{ \|\mathbf{X}_i^a - x^*\| < \omega \,\big|\, \|\nabla f(\mathbf{X}_i^a)\|^2 < \nu, a \in \mathcal{L}_{x^*} \right\} = 1$,

for all $i \geq i_0$.

4. For $x, y \in X^*$ and $x \neq y$, we also assume that the sequence of iterates $\{\mathbf{X}_i^a\}$ and $\{\mathbf{Y}_j^b\}$ generated by LSOs started at $a \in \mathcal{L}_x$ and $b \in \mathcal{L}_y$, respectively. Then

$$P_\xi \left\{ \min_{\forall \{i \geq 1\}} \|\mathbf{Y}_j^b - \mathbf{X}_i^a\| > \omega \,\Big|\, \mathbf{F}_{j-1}^b, \mathbf{F}_\infty^a \right\} = 1 \text{ for all } j \in \mathbb{N}.$$

Some of the conditions on the local method M in Assumption 3 are strong conditions that may be difficult to satisfy in practice by a local stochastic optimization method on a nonconvex problem. Yet, we find these assumptions to be a natural stochastic version of their deterministic counterparts: for Assumption 3.3 above is similar to the strictly decent property assumed for the (deterministic) local optimization method [31]. Assumption 3.1 ensures that the domain of attraction for any local minima has positive measure and that $\mathcal{L}_{x^*}$ does not lie entirely in the boundary set $\mathcal{Q}_\tau$. Assumption 3.2 ensures that no two local minima have overlapping domains of attraction. In general, local stochastic methods [12, 11] guarantee convergence to a stationary point only w.h.p.; that is, they satisfy Assumption 3.3(i). In addition, we need Assumption 3.3.1(ii) to ensure that x^* is within an ω-ball of all iterates after a large number of iterations with probability 1, given that the local method is started in that $\mathcal{L}_{x^*}$ and the norm of the gradient at the last iterate (of an LSO run) is small enough (less than ν). Furthermore, in Assumption 3.4, we assume that the iterates (realizations of the sequence of iterates) generated by any two LSO runs started in different domains of attractions are at least ω apart. These conditions are necessary for developing the algorithm and for showing that the algorithm MANSO identifies all the local minima w.h.p. using the LSO method M.

4 Statement of the Algorithm

Table 1 lists the conditions that our algorithm checks when deciding where to start an LSO. Algorithm 1 states our algorithm for ω-identifying all local minima of $f(x)$.

Algorithm 1: MANSO

1 Input: LSO method M; $\beta \in (0, \frac{1}{2})$; $\tau > 0$; $\omega > 0$; sampling effort $n \in \mathbb{N}$.

2 Initialize $S_0 = A_0 = \hat{X}_0^* = L_0 = \{\}$,

3 for $k = 0, 1, \ldots$ **do**

4   Uniformly sample a point a in $\mathcal{D}$, evaluate $\hat{f}_n(a)$ and add it to S_k.

5   Start LSO method M from all points in S_k satisfying the conditions listed in Table 1 for a given r_k, β, σ, ω, and sampling effort $n, \forall x \in S_k$. Add those points to A_k.

6   Update L_k by adding the next iterate generated by each LSO started from all the points in A_k.

7   Terminate any LSO run if its current iterate is within 2ω distance of any point in L_{k-1} (from other LSOs) and remove that LSO run from A_k.

8   Update $\hat{X}_k^*$ by adding any new local minima identified during the local searches, and remove that LSO run from A_k.

9   Set $S_{k+1} = S_k$, $\hat{X}_{k+1}^* = \hat{X}_k^*$, $L_{k+1} = L_k$, and $A_{k+1} = A_k$.

Notice that in condition **S1** of Table 1, we have used $n \in \mathbb{N}$ as the number of random samples of $f(\cdot)$ at respective points. We show in Theorem 1 later that for any $n \in \mathbb{N}$ the total number of LSO runs started by Algorithm 1 is finite. Since the results on the diffusion approximation of nonconvex SGD [14] show that the use of smaller batch sizes in batch SGD methods help escape sharp local minima and nondegenerate saddle points, we anticipate that the use of a larger n would be "better and faster" in ω—identifying all the local minima.

Note that MANSO can be viewed as a stochastic analogue of MLSL. That is, if we further assume that (1) $\hat{f}_n(x) \to f(x)$ $P_\xi - a.s.$ as $n \to \infty$ and (2) $\text{Var}_\xi\left[\hat{f}_n(x)\right] \to 0$ as $n \to \infty$, and for any two points $\{x, y\} \in \mathcal{D}$, $\text{Cov}_\xi\left[\hat{f}_n(x), \hat{f}_n(y)\right] \to 0$ as $n \to \infty$, then the above notion (see Table 1, **S1**) of not finding a "probabilistically best" point z in its r_k−neighborhood converges to the original MLSL [31, 32] condition of finding a "better" point z, as $n \to \infty$. Also, while implementing MANSO we estimate $\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(y)\right]$ using samples of $f(\cdot)$ for any $\{x, y\} \in \mathcal{D}$. In step 7 of Algorithm 1, since we do not have prior knowledge about the minimum separation between stationary points, η, we must choose a small enough positive value for each ω and τ.

5   MANSO Asymptotic Analysis

MANSO seeks to use as few LSO runs as possible to ω-identify all local minima of f in $\mathcal{D}$. We will show that the total number of LSO runs started by MANSO is finite, even if the algorithm is run forever. We state our main theoretical results in this section, but to ease presentation, the proofs of lemmas are deferred to the appendix.

We first show a limiting result for the measure of balls around any point $a \in \mathcal{D} \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$, where $\mathcal{Q}_\tau$ and $\mathcal{T}_\omega$ are defined in (1) and (3), respectively.

Lemma 1. Under Assumption 1.1 and 2, for any $a \in \mathcal{D} \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$, $\beta \in (0, 1/2)$, and for all $n \geq 1$

$$\lim_{r \to 0} \frac{m(\mathcal{A}(a; r; n; \beta))}{m(\mathcal{B}(a; r))} \geq \frac{1}{2},$$

where

$$\mathcal{A}(a; r; n; \beta) := \left\{ x \in \mathcal{D} : \|x - a\| \leq r \text{ and } P_\xi\left(\hat{f}_n(x) - \hat{f}_n(a) > \epsilon_n(x; a)\right) \leq \beta \right\},$$

and $\epsilon_n(x; a) = \sqrt{\frac{\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(a)\right]}{\beta}}$.

Specifically, Lemma 1 derives a bound on the measure of the set of points that are not probabilistically best in a ball of radius r around a point a drawn uniformly from the set of points in $\mathcal{D}$ not within ω of a stationary point or τ of the boundary of $\mathcal{D}$. (Not being probabilistically best is measured with respect to β, an integer n, and the tolerance $\epsilon_n(x; a)$.) Lemma 1 shows in the limit as r converges to zero that the set of

not probabilistically best points has a measure of at least half of the ball around a. Note that it is true for any $n \in \mathbb{N}$.

Next, we use Lemma 1 and construct a specific sequence of radius $\{r_k\}$ to show that the probability of starting an LSO run from any previously sampled point is bounded by a term that converges to zero as the number of iterations increases. The proof of the following result uses arguments similar to those used in [31, Theorem 8].

Lemma 2. Let t'_k be the number of LSO runs started from any point in $S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$ during iteration k of Algorithm 1. Then under Assumption 1.1 and 2 and for r_k as defined in Section 2 and used in Table 1 with $\sigma > 0$,

$$P[\{t'_k > 0\}] = O(|S_k|^{1-\frac{\sigma}{2}}).$$

Subsequently, we use the summability of the bound obtained in Lemma 2 on the probability of starting any LSO run from the set of sampled points not within ω of any stationary point or τ of the boundary of $\mathcal{D}$ to show in Theorem 1 that the total number of LSO runs started by Algorithm 1 is finite. Furthermore, condition **S3** in Table 1 ensures that LSO does not start from any point in $\mathcal{Q}_\tau$, and step 7 in Algorithm 1 ensures that the total number of LSO runs started from points sampled within the ω-ball of any stationary point is finite as $k \to \infty$. (The total number of stationary points is finite because of Assumption 1.2.)

That is, the number of LSO runs started by MANSO is finite even if MANSO runs forever.

Theorem 1. Let t_k be the number of LSO runs started by MANSO in iteration k. Then under Assumption 1, $\sum_{k=1}^{\infty} t_k < \infty$ with probability 1.

Proof. For any $\sigma > 4$ and $|S_k| = O(k)$, Lemma 2 implies that

$$\sum_{k=1}^{\infty} P[\{t'_k > 0\}] < \infty, \tag{4}$$

where t'_k is the number of LSO runs started from points in $a \in S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$ during iteration k of Algorithm 1. Note that using the first Borel–Cantelli lemma [5, Theorem 2.3.1], we have from equation (4) that

$$P\left(\cap_{i \geq 1} \cup_{k \geq i} \{t'_k > 0\}\right) = 0.$$

This is equivalent to

$$P\left(\exists i \geq 1 : \forall\, k \geq i, \{t'_k \leq 0\}\right) = 1. \tag{5}$$

Since $t'_k \geq 0, \forall k \geq 1$, the result in (5) implies that $\lim_{k \to \infty} t'_k \to 0$ $P-$ almost surely, that is, with P-probability 1. Since $\{t'_k\}$ is a sequence of natural numbers, it implies that

$$\sum_{k=1}^{\infty} t'_k < \infty. \tag{6}$$

Furthermore, if a point in S_k belongs to $\mathcal{Q}_\tau$, then MANSO (see **S3** in Table 1) does not start a run at that point, since we assumed in Assumption 1.3 that no minimum lies in $\mathcal{Q}_\tau$. Therefore, the probability of starting an LSO run from any point $a \in S_k \cap \mathcal{Q}_\tau$ is zero, and the result in (6) holds true for all $a \in (S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)) \cup \mathcal{Q}_\tau$, that is, for all $a \in S_k \setminus \mathcal{T}_\omega$.

Now consider the last case when $a \in S_k \cap \mathcal{T}_\omega$. Let t''_k be the number of LSO runs started by MANSO in the iteration k from any point $a \in S_k \cap \mathcal{T}_\omega$. Since in step 7 of Algorithm 1 we kill the LSO run if its current iterate is within 2ω distance of an already generated iterate from any of the previous LSO, a run can be started from any $a \in S_k \cap \mathcal{T}_\omega$ at most once. Since Assumption 1.2 implies that there are only a finite number of local minima, there exists a $k_0 \geq 1$ such that for all $k \geq k_0$ the number of LSO runs started at points $a \in S_k \cap \mathcal{T}_\omega$ will be zero. Therefore the number of LSO runs $t''_k \to 0$ as the number of samples increases to infinity. Since $\{t''_k\}$ is a sequence of natural numbers, it implies that

$$\sum_{k=1}^{\infty} t''_k < \infty. \tag{7}$$

Since $t_k = t'_k + t''_k$, the result follows immediately by adding (6) and (7).

□ □

Our next goal is to show that under certain assumptions MANSO will ω-identify all the local minima w.h.p. Recall that X^* is the collection of local minima of f in $\mathcal{D}$. The Assumption 3.3 guarantees that if MANSO starts an LSO run from any point in the domain of attraction $\mathcal{L}_{x^*}$ of a local minimum $x^* \in X^*$, then the LSO method M identifies it w.h.p. In particular, we prove that if a point is sampled in the domain of attraction of a local minimum, then the probability of that local minimum not being identified is sufficiently small for large enough iterations of both MANSO and LSO. Combined with the fact that these domains of attraction are of positive measure and the probability of getting a uniformly sampled point in any domain of attraction approaches 1 as the number of iteration increases (since the number of sampled points increase with each iteration), we show that all the local minima are identified w.h.p.

Theorem 2. *Under* Assumption *3, and given the sequence of radii $\{r_k\}$ as constructed in* Lemma *2 and used in condition **S1** of* Table *1, MANSO identifies all the local minima of* (SO) *w.h.p.*

Proof. Let a be a point sampled in iteration k of Algorithm 1 in the domain of attraction $\mathcal{L}_{x^*}$ of the local minimum x^* (see Definition 1) for the first time and none of the points sampled before belong to $\mathcal{L}_{x^*}$. Since $m(\mathcal{L}_{x^*}) > 0$ because of Assumption 3.1, the probability of obtaining at least a uniformly sampled point in $\mathcal{L}_{x^*}$ approaches 1 as the number of samples increases to infinity [2]. We also assume that the local minimum x^* has not been identified yet.

Recall $\hat{X}_k^*$ is the collection of approximate local minima identified up to iteration k and that set A_k is the collection of sampled points from which the LSO run has started and is still active up to iteration k. Recall the definition (see Definition 2) of the event that the local minimum is identified at any arbitrary iteration l as

$$I_l =: \{\forall \, l' > l \; \exists p \in A_{l'} \; : \|\mathbf{X}_{l'}^p - x^*\| < \omega\},$$

where $\{x_{l'}^p, l \in \mathbb{N}\}$ is the sequence of iterates generated by LSO at p. Now, let us compute the probability that the local minimum has been identified up to the iteration $\bar{k} \geq k$. Observe that

$$
\begin{aligned}
&P_\xi(I_{\bar{k}}|S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) \\
&= 1 - P_\xi(I_{\bar{k}}^C|S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) \\
&= 1 - \Bigg(P_\xi(\text{LSO has started from } a \text{ in iteration } k, I_{\bar{k}}^C|S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) \\
&\quad + P_\xi(\text{ LSO has not started from } a \text{ in iteration } k \text{ }, I_{\bar{k}}^C|S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) \Bigg).
\end{aligned}
\tag{8}
$$

We first analyze the first probabilistic term in (8). Since LSO starts at a, then by definition of A_k, $a \in A_k$. Even if we sample another point in $\mathcal{L}_{x^*}$ at any iteration $\bar{k} > k$, we will have $a \in A_{\bar{k}}$ because we terminate only the latest LSO in step 7 of MANSO. Also, if at some iteration $\bar{k} > k$ any of the iterates generated by some other sampled point, b (at iteration $k' < k$) in $\mathcal{L}_{y^*}$, for $x^* \neq y^*$, jumps into $\mathcal{L}_{x^*}$, then because of Assumption 3.2 and Assumption 3.4 we will still have $a \in A_{\bar{k}}$ on the event $I_{\bar{k}}^C$, since LSO at a will not be terminated at step 7 of MANSO because

$$P_\xi\left\{\|\mathbf{Y}_{\bar{k}-k'}^b - \mathbf{X}_{\bar{k}-k}^a\| > \omega \Big| \mathbf{F}_{\bar{k}-k'-1}^b\right\} \geq P_\xi\left\{\min_{\forall\{i \geq 1\}} \|\mathbf{Y}_{\bar{k}-k'}^b - \mathbf{X}_i^a\| > \omega \Big| \mathbf{F}_{\bar{k}-k'-1}^b, \mathbf{F}_\infty^a\right\} = 1 \text{ for all } \bar{k} - k \in \mathbb{N}.$$

Consequently, since $a \in \mathcal{L}_{x^*}$ and it remains in $A_{\bar{k}}$ for any $\bar{k} > k$, then because of Assumption 3.3 there exists a $k_0 \in \mathbb{N}$ such that for any iteration $\bar{k} - k > k_0$ of LSO at a, we have

$$P_\xi(\text{LSO has started from } a \text{ in iteration } k, I_{\bar{k}}^C|S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) < \frac{1}{2}\Lambda_{\bar{k}-k}.
\tag{9}$$

Note that we use the same iteration counter for the LSO run and MANSO, since we progress one step of each active LSO in $A_{\bar{k}}$ in each iteration of MANSO (see Step 6). Choosing $\bar{k}$ large enough such that $\bar{k} - k > k_0$, we obtain the last inequality in (9) by using Assumption 3.3, since LSO at a never gets terminated given that the local minimum x^* has not been ω-identified.

Next, we analyze the second probabilistic term in (8). In this case LSO may not start from a in iteration k, since it gets rejected because of any of the following conditions from Table 1 not being satisfied by a in the iteration k. Using these conditions in the third term of (8), we have

$$P_\xi(\text{ LSO has not started from } a \text{ in iteration } k, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset)$$
$$\leq P_\xi(\text{S(1)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) + P_\xi(\text{S(2)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset)$$
$$+ P_\xi(\text{S(3)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) + P_\xi(\text{S(4)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset),$$

where $S(1)$ denotes the event that a does not satisfy condition **S1** from Table 1 and similarly for **S2**, **S3**, and **S4**. The last three cases are straightforward to analyze. First consider event **S2**. Since we assumed that the local minimum x^* has not been identified yet, then the event $\{a \in \cup_{x \in \hat{X}_{k-1}^*} \mathcal{B}(x; \omega)\}$ is of probability measure zero, and hence

$$P_\xi(\text{S(2)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) = 0. \tag{10}$$

For event **S3**, $a \in \mathcal{Q}_\tau$, and we assumed at the beginning that $a \in \mathcal{L}_{x^*}$ as well. But because of Assumption 1.3 $\mathcal{L}_{x^*} \cap \mathcal{Q}_\tau = \emptyset$. Therefore it follows that

$$P_\xi(\text{S(3)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) = 0. \tag{11}$$

Since we assumed at the beginning that for any $\bar{k} < k$ none of the uniformly sampled points belong to $\mathcal{L}_{x^*}$, the final event **S4** is an impossible event, and therefore

$$P_\xi(\text{S(4)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) = 0. \tag{12}$$

For the first event **S1** we assumed that there exists $\bar{a} \in \mathcal{B}(a; r_k) \cap (S_k)$ that does not satisfies **S1** for a point a in $\mathcal{L}_{x^*}$. Now notice that

$$P_\xi(\text{S(1)}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset)$$
$$= P_\xi(\bar{a} \notin \mathcal{L}_{x^*}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset)$$
$$+ P_\xi(\bar{a} \in \mathcal{L}_{x^*}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset). \tag{13}$$

First consider the case when $\bar{a} \notin \mathcal{L}_{x^*}$. Since $r_k \to 0$ as $k \to \infty$ and $a \in \mathcal{L}_{x^*}$, there must exist a $k_0 \geq k$ such that $\bar{a} \notin \mathcal{B}(a; r_{\bar{k}}) \cap (S_{\bar{k}})$ for all $\bar{k} \geq k_0$. Therefore, LSO will start from a at iteration k_0. Using arguments similar to those used in (9), there exists a $k'' \geq k$ such that $\forall \bar{k} \geq k''$

$$P_\xi(\bar{a} \notin \mathcal{L}_{x^*}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) \leq \frac{1}{2} \Lambda_{\bar{k} - k_0}. \tag{14}$$

On the other hand, because of our assumption that for any $\bar{k} < k$ none of the uniformly sampled points belong to $\mathcal{L}_{x^*}$, the case $\bar{a} \in \mathcal{L}_{x^*}$ is an impossible event, and thus

$$P_\xi(\bar{a} \in \mathcal{L}_{x^*}, I_{\bar{k}}^C | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) = 0. \tag{15}$$

Since $m(\mathcal{L}_{x^*}) > 0$ due to Assumption 3.1, the probability of obtaining at least a uniformly sampled point in $\mathcal{L}_{x^*}$ approaches 1 as the number of samples increases to infinity [2]. Therefore $\lim_{\bar{k} \to \infty} P_\xi(S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) = 1$. Now, substituting equation (9)-(15) into (8), we obtain, for large enough $\bar{k}$,

$$P_\xi(I_{\bar{k}} | S_{\bar{k}} \cap \mathcal{L}_{x^*} \neq \emptyset) > 1 - \bar{\Lambda}_{\bar{k}}, \tag{16}$$

where $\bar{\Lambda}_{\bar{k}} = \min\{\Lambda_{\bar{k} - k_0}, \Lambda_{\bar{k} - k}\}$. Therefore, we have shown that any local minimum $x^* \in X^*$ will be identified w.h.p. Also, note that because of Assumption 3.3, $\bar{\Lambda}_{\bar{k}}$ decreases to 0 as the number of iterations $\bar{k}$ of LSO increases with the number of iterations of MANSO.

$\square$ $\square$

9

6 Numerical Experiments

We compare implementations of MANSO in their ability to solve difficult synthetic benchmark problems and to identify optimal variational parameters within the quantum approximate optimization algorithm (QAOA) [6]. Our MANSO implementation and scripts to perform our numerical experiments are available:

<div align="center">

https://github.com/prat212/MANSO.git

</div>

6.1 Synthetic Benchmark Experiments

We benchmark our implementation of MANSO on nonconvex optimization problems with large variance in their observations. In particular we fix two non-convex benchmark functions, Branin-Hoo ($d = 2$) and Shekel ($d = 4, 6, 8$ and 10), and make each non-convex objective evaluation stochastic by adding a Gaussian noise with variance 1. We add this significant noise only to make the testing of MANSO rigorous and robust. We generate 10 sample paths of Gaussian noise to create a set of 10 problems each for the Branin and Shekel functions. We seek to find all of the local minima for each problem within a fixed budget of function evaluations, B. We use ASTRO-DF [33] as the local method; it is a derivative-free trust-region stochastic optimizer selected because of its theoretical guarantee to converge to first-order critical points of the objective function. Other optimizers with convergence guarantees to first-order critical points (e.g., [11, 12]) could naturally be used as the local method in MANSO. To improve performance in our numerical experiments, we ensure that there are at most 10 active LSO runs at any given iteration. That is at each iteration k of MANSO, we sample a point uniformly in $\mathcal{D}$ if the total number of active runs is no larger than a fixed threshold, heuristically set to 10 in the experiments. Naturally, MANSO ensures that the conditions listed in Table 1 are satisfied before starting an LSO run from all the sampled points. Note that for condition (S1) in Table 1, for any two points $\{x, a\} \in \mathcal{D}$, our implementation estimates $\mathrm{Var}_\xi \left[\hat{f}_n(x) - \hat{f}_n(a)) \right]$ using the n samples of $f(x)$ and $f(a)$.

We compare the performance of MANSO with drawing points uniformly in the search domain. We compare only with such a random search method because we are unaware of other methods that aim to find all local optima of stochastic nonconvex functions. Our empirical results demonstrate that MANSO outperforms random search in identifying points that are within a small ball of all local minima. As we would like random search to perform equally well across problems independent of problem dimension, we consider each ball around a local minimum to always have a volume that is a small fraction (e.g., $1/1000$) of the volume of the domain. This number is arbitrarily chosen but gives a sense that how fast MANSO and random search can evaluate points near the local minima. We measure the performance of methods using data profiles [26].

6.2 Data profiles

Data profiles present the fraction of problems "solved" from a set of problems $\mathcal{P}$ after a certain number of function evaluations by an implementation of method h in a set of methods $\mathcal{H}$. The set of implementations $\mathcal{H}$ is created by adjusting β, ω, and n of MANSO. For a given objective function, we create different problems by changing the initial random seed. For our comparisons, the set $\mathcal{H}$ contains a uniform random sampling method and different versions of MANSO, obtained by varying its tuning parameters ω, τ, β, and n. Data profiles mark a problem instance $p \in \mathcal{P}$ as solved based on a user-defined test criterion. We use the criterion proposed in [20] to classify that a problem $p \in \mathcal{P}$ is solved: a problem is solved when a point is evaluated near each local minima for the problem. Let there be j local minima for a given problem. Mathematically, we define a test that ensures that a local minima $x^* \in \{x_1^*, x_2^*, \ldots, x_j^*\}$ is identified at level $\rho_d(\zeta)$ after e evaluations, as

$$\exists x \in \mathcal{E}_e \text{ with } \|x - x^*\| \le \rho_d(\zeta), \tag{17}$$

where $\rho_d(\zeta) = \frac{1}{\sqrt{\pi}} \sqrt[d]{\Gamma(1 + d/2)m(\mathcal{D})\zeta}$ and the set $\mathcal{E}_e$ is constructed by sequentially adding points the number of times they are being evaluated by a method $h \in \mathcal{H}$. That is, all the points in $S_k \cup L_k$ till e out of B budget is used. Notice that the volume of a ball of radius $\rho_d(\zeta)$ is ζ times the volume of the search domain $\mathcal{D}$. Hereafter, we use (17) for a problem $p \in \mathcal{P}$, method $h \in \mathcal{H}$ and for all $x^* \in \{x_i^*\}_{i=1}^j$ to compute

$$t_{p,h}(x^*) = \min \left\{ e \ge 1 : \exists x \in \mathcal{E}_e \text{ with } \|x - x^*\| \le \rho_d(\zeta) \right\},$$
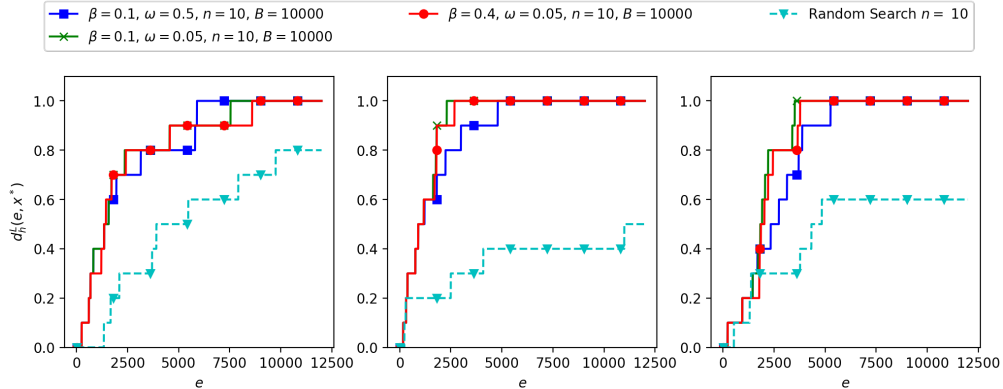
Figure 1: Data profiles for the Branin–Hoo function for finding each local minimum. $\zeta = 10^{-3}$ and $|\mathcal{P}| = 10$.

and define the data profile metric for $e > 0$ function evaluations as

$$d_h(e, x^*) = \frac{|\{p \in \mathcal{P} : t_{p,h}(x^*) \leq e, \}|}{|\mathcal{P}|}. \tag{18}$$

In the next section, we present the details of two benchmark nonconvex functions on which we evaluated the performance of MANSO.

6.3 Benchmark problems

We consider the Branin–Hoo and Shekel (4, 6, 8, and 10 dimensions) functions. Results for $d = 6$ and $d = 8$ Shekel problems appear in Appendix B.

Branin–Hoo function: The Branin-Hoo function [9] is a two-dimensional nonconvex problem with three local minima with the same optimal value. Mathematically, it is defined as

$$f(x^{(1)}, x^{(2)}) = a(x^{(2)} - b(x^{(1)})^2 + cx^{(1)} - r)^2 + s(1 - t)\cos(x^{(1)}) + s, \tag{19}$$

where $a = 1, b = 5.1/(4\pi^2), c = 5/\pi, r = 6, s = 10$ and $t = 1/(8\pi)$ and $\mathcal{D} = [-5, 10] \times [0, 15]$.

Shekel function: The Shekel function is the d-dimensional nonconvex problem with m local minima:

$$f(\mathbf{x}) = -\sum_{i=1}^{m} \left(2^{-d+4} \sum_{j=1}^{d} (x^{(j)} - C_{ij})^2 + c_i \right)^{-1}, \tag{20}$$

where $\mathbf{x} \in [0, 10]^d$, $C_{d \times m} = [x_1^*, x_2^*, x_3^*, \ldots, x_m^*]$ is the set of local minima and $c_i = [w_1, w_2, \ldots, w_m]^T$ is the weights of corresponding local minima; the smallest weight determines the global minima. In our experiments we choose $c_i = \{0.1, 0.2, 0.2, 0.4, 0.4, 0.6, 0.3, 0.7, 0.5, 0.5\}$ for a set of $m = 10$ local minima.

6.4 Experimental analysis

We plot the data profiles for the Branin function ($d = 2$) in Figure 1 and the Shekel $d = 4$ and $d = 10$ functions in Figures 2 and 3, respectively. We observe from the data profiles presented in each plot that MANSO outperforms the uniform random search method in finding a point in a ball of volume 10^{-4} (10^{-3} for the Branin function) times the volume of the domain $\mathcal{D}$ centered at the respective true local minima. Next, we discuss the effect of MANSO hyperparameters on its performance. *Effect of n:* Recall that as n increases the variance in estimates of the function values decrease. However, this confidence is attained at the expense of shedding more budget. Consequently, the number of points evaluated by MANSO decrease. Hence
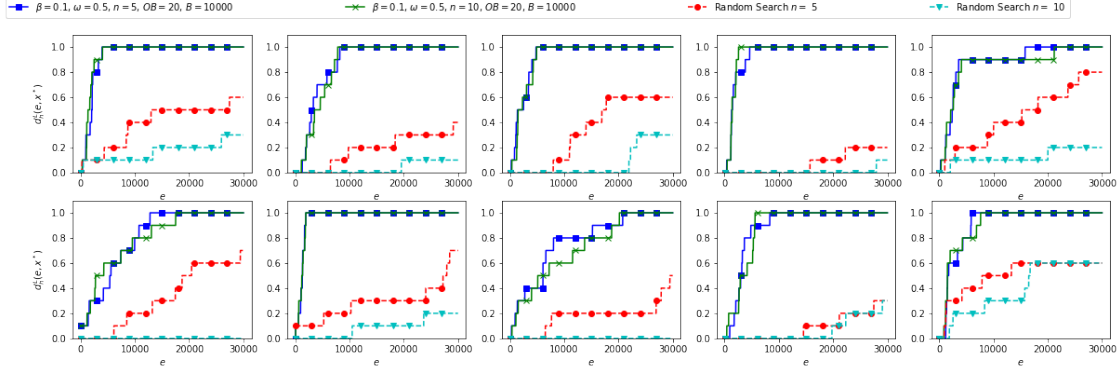
Figure 2: Data profiles for the Shekel-4D function for finding each local minimum. $\zeta = 10^{-4}$ and $|\mathcal{P}| = 10$.

varying n controls the trade-off between exploration and variance in function evaluation. We can observe this effect by comparing the MANSO performance on Shekel-4D function in Figure 2. Nonetheless, it is evident from the plots in Figure 2 that the increasing n does not guarantee that finding the local minima will be faster. *Effect of ω:* Recall that ω is a hyperparameter used in Step 7 of MANSO, to terminate an LSO run if any of its iterates are within 2ω distance of any iterate generated by some other LSO. Intuitively, a larger ω would result in more termination. In Figure 1, the effect of ω can be observed by comparing the green ($\omega = 0.05$) and blue ($\omega = 0.5$) lines. Note that MANSO under the green experiment was able to explore more points and thus identified 2 local minima of the Branin function faster than the blue experiment. In the blue experiments, due to larger ω, the number of points evaluated by MANSO is less than the green as a large portion of the budget is used for evaluating new sampled points.

6.5 Variational Parameter Optimization

Quantum approximate optimization algorithm (QAOA) is a hybrid algorithm that uses a parameterized trial quantum state $\psi(x)$ as defined by the parameters x. (The values in $x \in \mathbb{R}^{2p}$ are rotations or angles that parameterize $2p$ unitary operators.) What is desired is parameters x such that when the trial state is measured, the measurement outcome corresponds to the solution of the optimization problem. This is achieved by finding parameters x that give a large expected value for $\psi(x)^T H \psi(x)$, where H is the problem Hamiltonian encoding some classical objective h. Under certain conditions on the Hamiltonian H, $f(x)$ must be evaluated by using a quantum computer. The search for optimal parameters x can therefore be considered as a (classical) numerical optimization problem of the form (SO) with $f(x) = \psi(x)^T H \psi(x)$. The
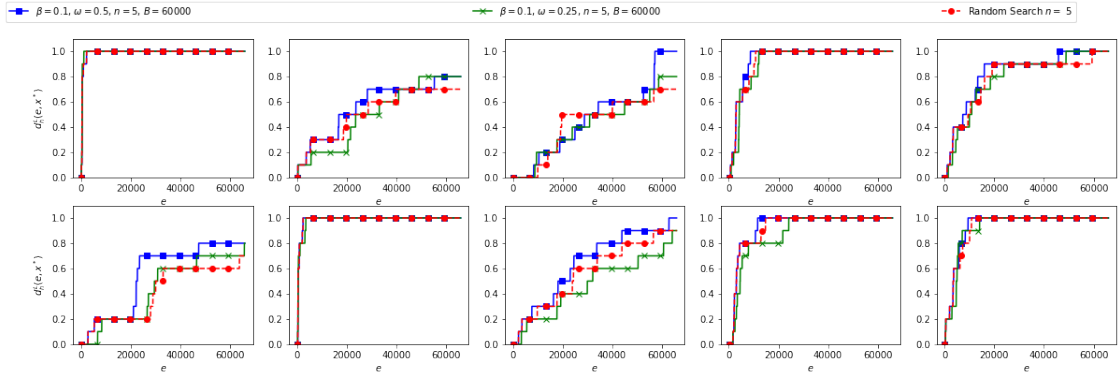

Figure 3: Data profiles for the Shekel-10D function for finding each local minimum. $\zeta = 10^{-4}$ and $|\mathcal{P}| = 10$.
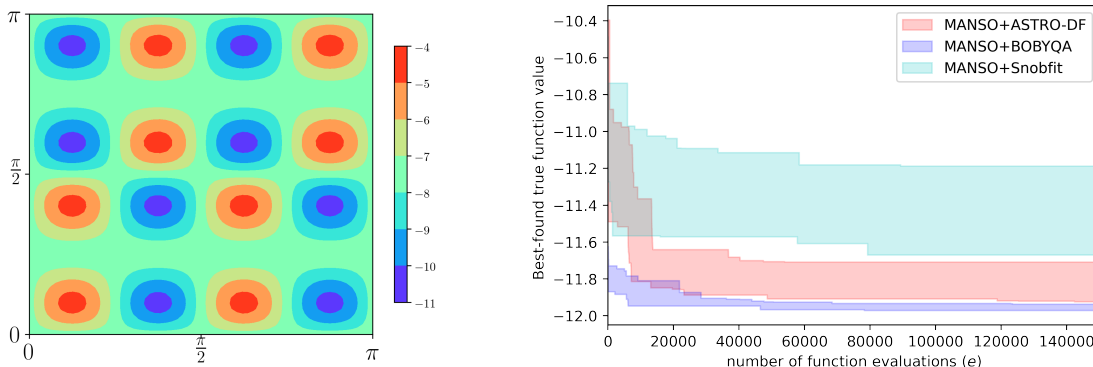
Figure 4: (left) Example landscape for MAXCUT on the Petersen graph with $p = 1$. (right) Performance of MANSO in identifying optimal parameters within QAOA with various local solvers.

stochasticity in the objective arises from not being able to compute the value of observable $\psi(x)^T H \psi(x)$ by using a quantum circuit but rather having to compute the objective from a sample: $f(x) = \sum_{y_i \in \text{sample}} h(y_i)$.

QAOA has nontrivial performance guarantees [6, 7] and requires the execution of only moderately sized quantum circuits, with the depth controlled by the number of steps p. For these reasons, QAOA is an especially promising candidate algorithm for demonstrating quantum advantage on near-term quantum computers. Yet, the quality of the solution produced by QAOA depends critically on the quality of the parameters x used by the algorithm. Identifying such parameters is difficult because the objective landscape is highly nonconvex with many local minima with poor objective values [37, 34]. Figure 4 shows an example contour plot with $p = 1$. While nonglobal optima are not necessarily of interest in the QAOA problem setting, we consider the difficulty of finding a global optimum to be a considerable test of our MANSO implementation.

We consider the problem of using QAOA to find the maximum cut on the Petersen graph with a depth of $p = 5$, that is, $d = 10$. The global optimal value for this problem is -12. The performance of MANSO to identify maximum cut with three local solvers (ASTRO-DF [33], BOBYQA [3], and Snobfit [15]) are summarized in Figure 4. Moreover, we also considered non-MANSO global optimizers such as Bayesian optimization [28] to solve a deterministic version of the QAOA problem. However, the method was significantly slow due to large matrix computation and produced the best candidate global minima with a value -11.74 only after 5000 evaluations. Consequently, we are not comparing MANSO with other approaches such as Bayesian optimization. For each local solver, we run MANSO on the MAXCUT problem with a budget of 150,000 function evaluations and we check for the termination condition in step 7 of MANSO after 500 function evaluations have been performed by the local search method. We also fix $n = 5$, $\omega = 0.01$, $\beta = 0.1$ and $\tau = 0.01$. We repeat each experiment with a given local solver 20 times and plot the range of best function value identified in Figure 4. Although BOBYQA is designed to solve deterministic problems, we use it for this stochastic problem as it has been reported that it empirically performs well on problems with stochastic noise [3]. In particular, it is evident from our experimental result in Figure 4 (right) too that BOBYQA performance is competitive with other stochastic solvers. However, we note that BOBYQA failed on 10 out of 20 experiments as it produced singular Hessian matrices of the noisy QAOA objective.

7   Conclusion

We propose the MANSO algorithm to identify all the local minima of a stochastic nonconvex function. We construct an efficient scheme to judiciously determine when to start a local stochastic optimization run from a sampled point in a compact search domain. We show that under that MANSO starts only finitely many local stochastic optimization runs. We also show that MANSO identifies all the local minima asymptotically with high probability, given that the local stochastic optimization method is guaranteed to converge to a local minimum with high probability. MANSO's theoretical guarantees also require that the sequence of iterates

generated from the local stochastic search started in a domain of attraction and cannot leave that domain with high probability. (Certainly, this is a restrictive assumption for a stochastic optimization method, but it is analogous to the assumptions in the foundational MLSL work [31, 32].) Our experimental results show that MANSO can display strong performance even when coupled with a local optimization that does not satisfy such a restrictive assumption. The assumption that there are no flat regions in the true objective function may be removed by using the techniques developed in [23] for a multistart algorithm MLSL for deterministic nonconvex objectives.

Furthermore, we demonstrate the efficacy of our algorithm on two benchmark problems with dimensions ranging from 2 to 10, and we compare the performance with that of a uniform random search method. We also use MANSO to find the global minima of a highly nonconvex 10-dimensional Peterson graph. We aim to apply MANSO to more complex and higher-dimension benchmark functions and application problems as part of our future work. Similar to [20], an asynchronously parallel version of MANSO can be developed to improve its computational performance.

Acknowledgments

References

[1] Adamcik, J., Mezzenga, R.: Amyloid polymorphism in the protein folding and aggregation energy landscape. Angewandte Chémie International Edition **57**(28), 8370–8382 (2018). doi:10.1002/anie.201713416

[2] Brooks, S.H.: A discussion of random methods for seeking maxima. Operations Research **6**(2), 244–251 (1958). doi:10.1287/opre.6.2.244

[3] Cartis, C., Fiala, J., Marteau, B., Roberts, L.: Improving the flexibility and robustness of model-based derivative-free optimization solvers. ACM Transactions on Mathematical Software **45**(3), 1–41 (2019). doi:10.1145/3338517

[4] di Serafino, D., Gomez, S., Milano, L., Riccio, F., Toraldo, G.: A genetic algorithm for a global optimization problem arising in the detection of gravitational waves. Journal of Global Optimization **48**(1), 41–55 (2010). doi:10.1007/s10898-010-9525-9

[5] Durrett, R.: Probability: Theory and Examples. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press (2010). doi:10.1017/CBO9780511779398

[6] Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv:1411.4028 (2014)

[7] Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. arXiv:1412.6062 (2014)

[8] Floudas, C., Klepeis, J., Pardalos, P.: Global optimization approaches in protein folding and peptide docking. In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 47, pp. 141–171. American Mathematical Society (1999). doi:10.1090/dimacs/047/07

[9] Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling, pp. 195–203. John Wiley & Sons, Ltd (2008). doi:10.1002/9780470770801.app1

[10] Frazier, P.I.: Bayesian optimization. In: Recent Advances in Optimization and Modeling of Contemporary Problems, pp. 255–278. INFORMS TutORials in Operations Research (2018). doi:10.1287/educ.2018.0188

[11] Ghadimi, S., Lan, G.: Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Mathematical Programming **156**(1-2), 59–99 (2015). doi:10.1007/s10107-015-0871-8

[12] Ghadimi, S., Lan, G., Zhang, H.: Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Mathematical Programming **155**(1-2), 267–305 (2014). doi:10.1007/s10107-014-0846-1

[13] Gheribi, A.E., Robelin, C., Digabel, S.L., Audet, C., Pelton, A.D.: Calculating all local minima on liquidus surfaces using the FactSage software and databases and the mesh adaptive direct search algorithm. The Journal of Chemical Thermodynamics **43**(9), 1323–1330 (2011). doi:10.1016/j.jct.2011.03.021

[14] Hu, W., Li, C.J., Li, L., Liu, J.G.: On the diffusion approximation of nonconvex stochastic gradient descent. Annals of Mathematical Sciences and Applications **4**(1), 3–32 (2019). doi:10.4310/amsa.2019.v4.n1.a1

[15] Huyer, W., Neumaier, A.: SNOBFIT – stable noisy optimization by branch and fit. ACM Transactions on Mathematical Software **35**(2), 1–25 (2008). doi:10.1145/1377612.1377613

[16] Jin, C., Liu, L.T., Ge, R., Jordan, M.I.: On the local minima of the empirical risk. In: Advances in Neural Information Processing Systems, pp. 4896–4905 (2018)

[17] Krishnamoorthy, M., Schulz, H., Ju, X., Wang, W., Leyffer, S., Marshall, Z., Mrenna, S., Müller, J., Kowalkowski, J.B.: Apprentice for event generator tuning. EPJ Web of Conferences **251**, 03060 (2021). doi:10.1051/epjconf/202125103060

[18] Krityakierne, T., Shoemaker, C.A.: SOMS: SurrOgate MultiStart algorithm for use with nonlinear programming for global optimization. International Transactions in Operational Research **24**(5), 1139–1172 (2015). doi:10.1111/itor.12190

[19] Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. Journal of Basic Engineering **86**(1), 97 (1964). doi:10.1115/1.3653121

[20] Larson, J., Wild, S.M.: Asynchronously parallel optimization solver for finding multiple minima. Mathematical Programming Computation **10**(3), 303–332 (2018). doi:10.1007/s12532-017-0131-4

[21] Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A.: Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research **18**(1), 6765–6816 (2017)

[22] Li, Z., Scheraga, H.A.: Monte Carlo-minimization approach to the multiple-minima problem in protein folding. Proceedings of the National Academy of Sciences of the United States of America **84**(19), 6611–6615 (1987). doi:10.1073/pnas.84.19.6611

[23] Locatelli, M.: Relaxing the assumptions of the multilevel single linkage algorithm. Journal of Global Optimization **13**(1), 25–42 (1998). doi:10.1023/a:1008246031222

[24] Maclaurin, D., Duvenaud, D., Adams, R.: Gradient-based hyperparameter optimization through reversible learning. In: International Conference on Machine Learning, pp. 2113–2122 (2015)

[25] Mathesen, L., Pedrielli, G., Ng, S.H., Zabinsky, Z.B.: Stochastic optimization with adaptive restart: A framework for integrated local and global learning. Journal of Global Optimization **79**(1), 87–110 (2020). doi:10.1007/s10898-020-00937-5

[26] Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization **20**(1), 172–191 (2009). doi:10.1137/080724083

[27] Nguyen, V., Rana, S., Gupta, S., Li, C., Venkatesh, S.: Budgeted batch Bayesian optimization with unknown batch sizes. arXiv:1703.04842 (2017)

[28] Nogueira, F.: Bayesian Optimization: Open source constrained global optimization tool for Python (2014–). URL https://github.com/fmfn/BayesianOptimization

[29] Peri, D., Tinti, F.: A multistart gradient-based algorithm with surrogate model for global optimization. Communications in Applied and Industrial Mathematics **3**(1) (2012). doi:10.1685/journal.caim.393

[30] Regis, R.G., Shoemaker, C.A.: A quasi-multistart framework for global optimization of expensive functions using response surface models. Journal of Global Optimization **56**(4), 1719–1753 (2012). doi:10.1007/s10898-012-9940-1

[31] Rinnooy Kan, A.H.G., Timmer, G.T.: Stochastic global optimization methods part I: Clustering methods. Mathematical Programming **39**(1), 27–56 (1987). doi:10.1007/bf02592070

[32] Rinnooy Kan, A.H.G., Timmer, G.T.: Stochastic global optimization methods part II: Multi level methods. Mathematical Programming **39**(1), 57–78 (1987). doi:10.1007/bf02592071

[33] Shashaani, S., Hashemi, F.S., Pasupathy, R.: ASTRO-DF: A class of adaptive sampling trust-region algorithms for derivative-free stochastic optimization. SIAM Journal on Optimization **28**(4), 3145–3176 (2018). doi:10.1137/15m1042425

[34] Shaydulin, R., Safro, I., Larson, J.: Multistart methods for quantum approximate optimization. In: Proceedings of the IEEE High Performance Extreme Computing Conference (2019). doi:10.1109/hpec.2019.8916288

[35] Wessing, S., Preuss, M.: The true destination of EGO is multi-local optimization. In: IEEE Latin American Conference on Computational Intelligence (2017). doi:10.1109/la-cci.2017.8285677

[36] Zheng, R., Li, M.: Multistart global optimization with tunnelling and an evolutionary strategy supervised by a martingale. Engineering Optimization pp. 1–19 (2021). doi:10.1080/0305215x.2021.1940989

[37] Zhou, L., Wang, S.T., Choi, S., Pichler, H., Lukin, M.D.: Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Physical Review X **10**(2) (2020). doi:10.1103/physrevx.10.021067

[38] Žilinskas, A., Gillard, J., Scammell, M., Zhigljavsky, A.: Multistart with early termination of descents. Journal of Global Optimization **79**(2), 447–462 (2019). doi:10.1007/s10898-019-00814-w

A   Proofs

Below are the proofs of Lemma 1 and Lemma 2.

Proof of Lemma 1. First, consider the sets

$$\mathcal{G}(a;r;n) := \{x \in \mathcal{D} : \|x - a\| \le r \text{ and } [f(x) - f(a)] < \epsilon_n(x;a)\} \text{ and} \tag{21}$$

$$\mathcal{C}(a;r) := \{x \in \mathcal{D} : \|x - a\| \le r \text{ and } [f(x) - f(a)] < 0\}. \tag{22}$$

Now, observe that using Chebyschev's inequality and Assumption 2, for all $x \in \mathcal{G}(a;r;n)$,

$$
\begin{aligned}
&P_\xi\left(\hat{f}_n(x) - \hat{f}_n(a) > \epsilon_n(x;a)\right) \\
&\quad \le P_\xi\left(\left|[\hat{f}_n(x) - f(x)] - [\hat{f}_n(a) - f(a)]\right| > \epsilon_n(x;a) - [f(x) - f(a)]\right) \\
&\quad \le \frac{1}{(\epsilon_n(x;a) - [f(x) - f(a)])^2} \mathbb{E}_\xi\left[\left([\hat{f}_n(x) - f(x)] - [\hat{f}_n(a) - f(a)]\right)^2\right] \\
&\quad = \frac{1}{(\epsilon_n(x;a) - [f(x) - f(a)])^2}\left(\text{Var}_\xi\left[\hat{f}_n(x)\right] + \text{Var}_\xi\left[\hat{f}_n(a)\right] - 2\,\text{Cov}_\xi\left[\hat{f}_n(x), \hat{f}_n(a)\right]\right) \\
&\quad = \frac{\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(a)\right]}{(\epsilon_n(x;a) - [f(x) - f(a)])^2}. \tag{23}
\end{aligned}
$$

Since $x \in \mathcal{G}(a;r;n)$ implies $\epsilon_n(x;a) > [f(x) - f(a)]$, therefore

$$
\begin{aligned}
&\left\{x \in \mathcal{D} : \frac{\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(a)\right]}{(\epsilon_n(x;a) - [f(x) - f(a)])^2} < \beta\right\} \\
&\quad = \left\{x \in \mathcal{D} : f(x) - f(a) < \epsilon_n(x;a) - \sqrt{\beta^{-1}\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(a)\right]}\right\}. \tag{24}
\end{aligned}
$$

Next, recall the definition of $\mathcal{A}(a;r;n;\beta)$, and observe that equation (23) and (24) together imply that

$$
\begin{aligned}
&\left\{x \in \mathcal{D} : \|x - a\| \le r \text{ and } f(x) - f(a) < \epsilon_n(x;a)\right. \\
&\qquad\qquad\qquad \left. - \sqrt{\beta^{-1}\text{Var}_\xi\left[\hat{f}_n(x) - \hat{f}_n(a)\right]}\right\} \subseteq \mathcal{A}(a;r;n;\beta), \tag{25}
\end{aligned}
$$

for all $x \in \mathcal{G}(a;r;n)$. Since $\epsilon_n(x;a) = \sqrt{\frac{\text{Var}_\xi[\hat{f}_n(x) - \hat{f}_n(a)]}{\beta}}$, equation (25) implies that

$$\mathcal{C}(a;r) \cap \mathcal{G}(a;r;n) \subseteq \mathcal{A}(a;r;n;\beta) \cap \mathcal{G}(a;r;n). \tag{26}$$

Observe that $\mathcal{C}(a;r) \subseteq \mathcal{G}(a;r;n)$ and $\mathcal{A}(a;r;n;\beta) \cap \mathcal{G}(a;r;n) \subseteq \mathcal{A}(a;r;n;\beta)$; Therefore it follows from (26) that

$$\mathcal{C}(a;r) \subseteq \mathcal{A}(a;r;n;\beta). \tag{27}$$

Recall that $\mathcal{T}_\omega$ is the union of ω-radius balls centered at stationary points of f in $\mathcal{D}$. Now consider $a \in \mathcal{D} \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)\}$. Define the set

$$\mathcal{E}(a;r;\rho) := \left\{x \in \mathcal{D} : \|x - a\| \le r \text{ and } \nabla f(a)^T(x - a) + \frac{1}{2}\rho r^2 \le 0\right\},$$

where ρ is the largest eigenvalue of $\nabla^2 f(x)$ for $x \in \mathcal{D}$. Using the Taylor expansion of f around a, we know that for all $x \in \mathcal{D}$, with $\|x - a\| \le r$, there exists a $\theta \in [0, 1]$ such that

$$f(x) - f(a) = \nabla f(a)^T (x - a) + \frac{1}{2}(x - a)^T \nabla^2 f(a + \theta(x - a))(x - a). \tag{28}$$

For ease of reference, let $H = \nabla^2 f(a + \theta(x - a)) = H$ and $v = x - a$.

Since f is twice continuously differentiable by Assumption 1.1, its Hessian is always real and symmetric, satisfying $v^T H v \le \rho v^T v$ for all $v \in \mathbb{R}^d$. It follows from (28) that

$$f(x) - f(a) \le \nabla f(a)^T (x - a) + \frac{1}{2}\rho(x - a)^T (x - a) \le \nabla f(a)^T (x - a) + \frac{1}{2}\rho r^2. \tag{29}$$

Equation (29) implies that $\mathcal{E}(a; r; \rho) \subseteq \mathcal{C}(a; r) \subseteq \mathcal{A}(a; r; n; \beta)$. For a given $a \in \mathcal{D}$, $m(\mathcal{B}(a; r)) \le r^d \frac{\pi^{d/2}}{\Gamma(1+d/2)}$, where $\Gamma(\cdot)$ is the gamma function. Now using the lower bound on $m(\mathcal{E}(a; r; \rho))$ derived in Lemma 7 of [31], we obtain

$$\lim_{r \to 0} \frac{m(\mathcal{A}(a; r; n; \beta))}{m(\mathcal{B}(a; r))} \ge \lim_{r \to 0} \frac{m(\mathcal{E}(a; r; \rho))}{m(\mathcal{B}(a; r))} \ge \lim_{r \to 0} \frac{1}{2} - \frac{\pi^{-d/2}}{2p\Gamma(1 + \frac{d-1}{2})} \frac{\rho r}{2} = \frac{1}{2}, \tag{30}$$

where $p = \min\{\|\nabla f(a)\|, a \in \mathcal{D} \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)\}$ for any $k \in \mathbb{N}$. (Note that $p > 0$ by construction.) Therefore (30) implies that for all $a \in \mathcal{D} \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$

$$\lim_{r \to 0} \frac{m(\mathcal{A}(a; r; n; \beta))}{m(\mathcal{B}(a; r))} \ge \frac{1}{2}.$$

$\square$ $\qquad$ $\square$

Proof of Lemma 2. For any $a \in S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$ and using the fact that S_k is sampled uniformly, for a vanishing sequence $\{r_k\}$ the Lemma 1 implies that there exists a $k_0 \ge 1$ such that for all $k \ge k_0$,

$$P[\{t'_k > 0\}] \le \bigcup_{a \in S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)} \left(1 - \frac{m(\mathcal{A}(a; r_k; n; \beta))}{m(\mathcal{D})}\right)^{|S_k|-1}$$

$$\le |S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)| \left(1 - \frac{m(\mathcal{A}(a; r_k; n; \beta))}{m(\mathcal{D})}\right)^{|S_k|-1}$$

$$\le |S_k| \left(1 - \frac{1}{2}\frac{m(\mathcal{B}(a; r_k))}{m(\mathcal{D})}\right)^{|S_k|-1}, \tag{31}$$

where the first inequality bounds the probability that for any sampled point in $S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)$ none of the remaining sampled points are in $\mathcal{A}(a; r_k; n; \beta)$ (see (**S1**) of Table 1). The second inequality follows from Boole's inequality. The last inequality in (31) is due to Lemma 1 and the fact that $|S_k \setminus (\mathcal{Q}_\tau \cup \mathcal{T}_\omega)| < |S_k|$.

Recall that for any $a \in \mathcal{D}$ and r_k sufficiently small such that $\mathcal{B}(a; r_k) \subseteq \mathcal{D}$, $m(\mathcal{B}(a; r_k)) = r_k^d \frac{\pi^{d/2}}{\Gamma(1+d/2)}$. Combined with this fact, for any $\sigma > 0$ and choosing a sequence $r_k = \frac{1}{\sqrt{\pi}} \sqrt[d]{\Gamma(1 + d/2)m(\mathcal{D})\sigma \frac{\log|S_k|}{|S_k|}}$, we have

$$P[\{t'_k > 0\}] \le |S_k| \left(1 - \frac{1}{2}\frac{m(\mathcal{B}(a; r_k))}{m(\mathcal{D})}\right)^{|S_k|-1} = |S_k| \left(1 - \frac{\sigma}{2}\frac{\log|S_k|}{|S_k|}\right)^{|S_k|-1}.$$

Since $|S_k| \to \infty$ as $k \to \infty$, the fact that $e^{-\frac{\sigma}{2}\frac{\log|S_k|}{|S_k|}} \ge 1 - \frac{\sigma}{2}\frac{\log|S_k|}{|S_k|}$ implies

$$P[\{t'_k > 0\}] \le |S_k| \left(e^{-\frac{\sigma}{2}\frac{(|S_k|-1)\log|S_k|}{|S_k|}}\right) = |S_k|^{1 - \frac{\sigma}{2}\frac{(|S_k|-1)}{|S_k|}} = O(|S_k|^{1-\frac{\sigma}{2}}). \tag{32}$$

$\square$ $\qquad$ $\square$
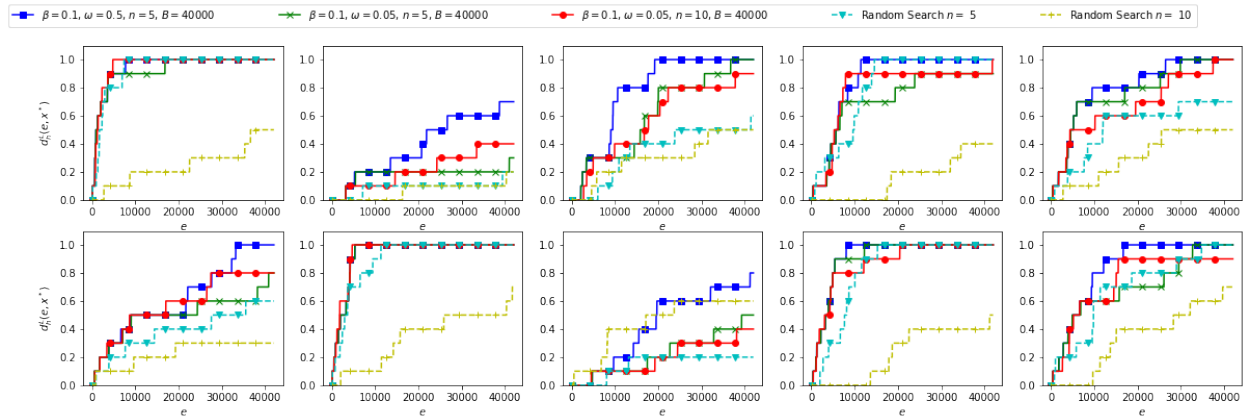
B Shekel with $d = 6$ and $d = 8$

18

Figure 5: Data profiles for the Shekel-6D function for finding each local minimum. $\zeta = 10^{-4}$ and $|P| = 10$.
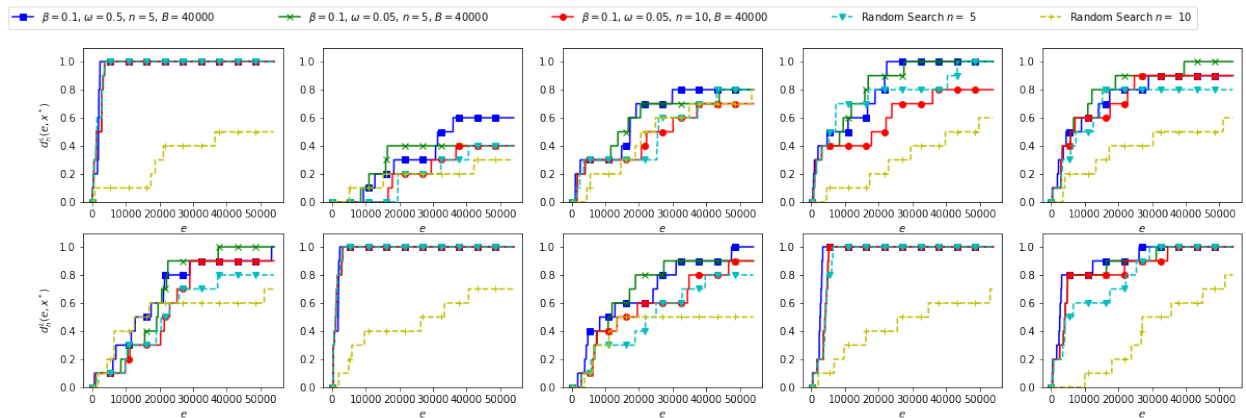

Figure 6: Data profiles for the Shekel-8D function for finding each local minimum. $\zeta = 10^{-4}$ and $|P| = 10$.