informs
http://pubsonline.informs.org/journal/ijoc/

INFORMS JOURNAL ON COMPUTING
Vol. 00, No. 0, Xxxxx 2018, pp. 1–17
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

# Surrogate-Based Promising Area Search for Lipschitz Continuous Simulation Optimization

Qi Fan,[a] Jiaqiao Hu[a]

[a] Department of Applied Mathematics and Statistics, State University of New York at Stony Brook, Stony Brook, New York 11794
**Contact:** fancy0829@hotmail.com (QF); jqhu@ams.sunysb.edu (JH)

**Received:** April 19, 2017
**Revised:** October 31, 2017; November 8, 2017
**Accepted:** November 9, 2017
**Published Online:**

https://doi.org/10.1287/ijoc.2017.0801

**Abstract.** We propose an adaptive search algorithm for solving simulation optimization problems with Lipschitz continuous objective functions. The method combines the strength of several popular strategies in simulation optimization. It employs the shrinking ball method to estimate the performance of sampled solutions and uses the performance estimates to fit a surrogate model that iteratively approximates the response surface of the objective function. The search for improved solutions at each iteration is then based on sampling from a promising region (a subset of the decision space) adaptively constructed to contain the point that optimizes the surrogate model. Under appropriate conditions, we show that the algorithm converges to the set of local optimal solutions with probability one. A computational study is also carried out to illustrate the algorithm and to compare its performance with some of the existing procedures.

**History:** Accepted by Bruno Tuffin, Area Editor for Simulation.
**Funding:** This work was supported by the National Science Foundation [Grant CMMI-1634627].

**Keywords:** promising area search • shrinking ball methods • surrogate model approximation

## 1. Introduction

Many complex systems arising in applications from the industrial and science sectors require the use of simulation optimization techniques to improve their performance. In simulation optimization problems, the performance of a system is rarely known with complete certainty and needs to be estimated in a pathwise manner through computer simulation. Thus, in contrast to deterministic optimization, such problems are often characterized by random uncertainties in their performance estimates and therefore require additional simulation effort and special techniques to be expended and employed to deal with the noisy measurement of the objective function.

A well-established class of methods for solving differentiable simulation optimization problems is stochastic approximation (Robbins and Monro 1951, Kiefer and Wolfowitz 1952, Spall 1992). These methods estimate the gradient of the objective function through stochastic simulation and have the same structure as the classical gradient descent algorithms in deterministic optimization. They usually do not rely on precise performance estimates but resort to some forms of averaging to eliminate the estimation error during the course of the iteration. Sample average approximation (Kleywegt et al. 2001, Robinson 1996, Kim et al. 2015) is another approach for approximating the solutions to problems with structural information. The idea is to convert a stochastic problem into a deterministic one by expending a large amount of simulation effort on each visited solution. The resulting problem can then be solved by using conventional mathematical programming solvers, where structural properties such as differentiability and convexity are often exploited.

For nondifferentiable simulation optimization problems, one popular and effective method is to use random search. This encompasses a broad class of algorithms that use a sequence of randomly generated iterates, for example, candidate solutions, probability models, and promising subsets, to approximate the optimal solution. Examples of random search techniques include the stochastic ruler method (Yan and Mukai 1992), simulated annealing (Alrefaei and Andradóttir 1999), the nested partitions method (Shi and Ólafsson 2000), shrinking ball methods (Baumert and Smith 2002, Andradóttir and Prudius 2010), COMPASS (Hong and Nelson 2006, Xu et al. 2010), and model-based approaches (Rubinstein and Kroese 2004, Hu et al. 2008). These algorithms primarily differ in the type of iterates an algorithm produces and the random strategy used to generate the iterates. For recent reviews of random search techniques, see, for example, Andradóttir (2014), Hu (2015), Zabinsky (2015) and references therein.

We briefly describe two of the aforementioned approaches that are most relevant to our work: the shrinking ball method and COMPASS. The shrinking ball method was first introduced by Baumert and Smith (2002) in a pure random search context. The method estimates the objective function value at a sampled point by averaging the performance of all points that fall into a ball centered at it. The estimation bias is then eliminated by gradually sending the radius of the ball to zero, whereas the variance is controlled by adjusting the decreasing speed of the radius to ensure that a sufficient number of points are contained in the ball. This way of estimating the objective function only requires a single function evaluation (simulation) to be performed at every point sampled, and thus appears especially well suited for problems with large or uncountable decision spaces. Andradóttir and Prudius (2010) provide a general analysis for this type of method and propose a stochastic shrinking ball variant in which the choice of the ball radius is based on the number of sampled points within the ball. Recently, Kiatsupaibul et al. (2018) also developed a shrinking ball-based framework for the efficient implementation and analysis of adaptive search algorithms that perform only a single simulation at each visited solution.

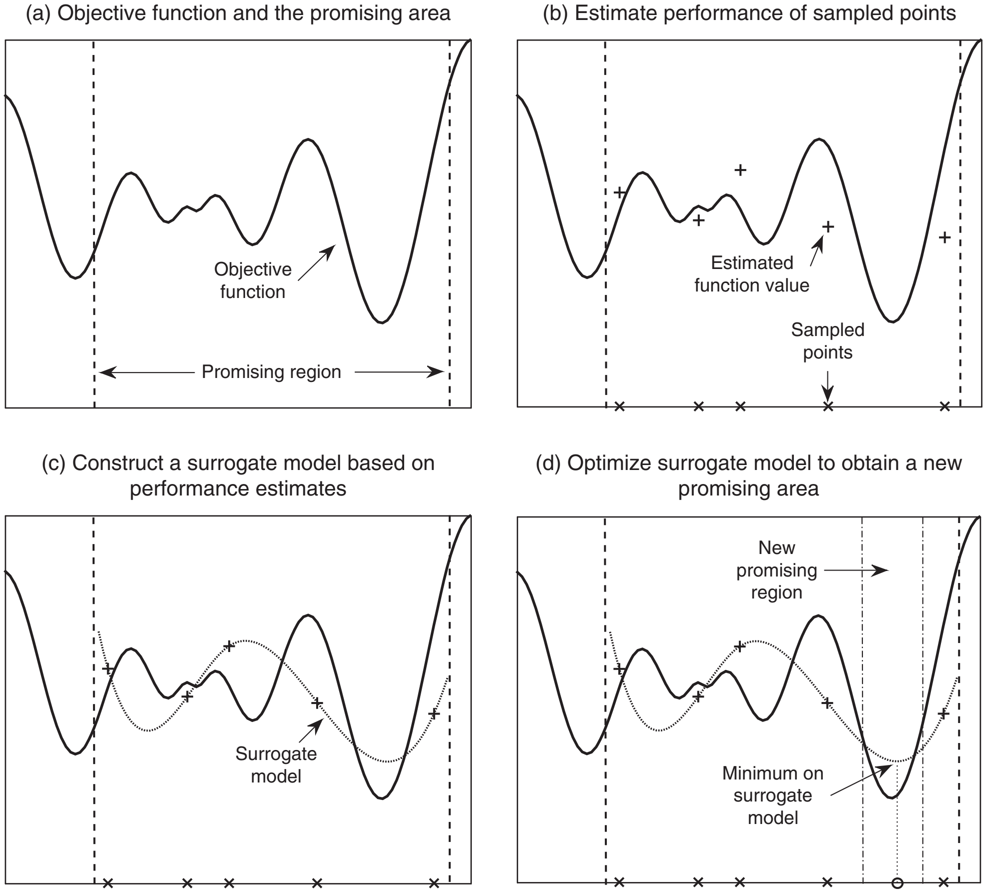
COMPASS (Hong and Nelson 2006) is a discrete simulation optimization algorithm that uses an adaptive neighborhood structure, called the most promising area, in searching for local optimal solutions. Like many other local search techniques, the promising area is designed to explore the structure that good solutions tend to be clustered together, but is constructed in a unique way as the set of points that are closer in distance to the current best sampled solution than to other visited solutions. Later, the idea was extended in Hong and Nelson (2007) to arrive at a general local search framework, and an industrial strength COMPASS algorithm (Xu et al. 2010) was also developed and shown to have very encouraging finite-time performance.

When simulation experiments are computationally expensive, it is often desirable to use surrogate models (metamodels) to represent simulation input-output relations. This has motivated the use of surrogate-based or response surface methods (RSMs) for simulation or "black-box" optimization. The construction of surrogate models can be carried out either in a one-shot space-filling way or through a sequential sampling strategy. In one-shot RSMs, a surrogate/response model is fit to the response values obtained from a set of predetermined design points, and an estimate of the optimal solution is then directly inferred from the model. On the other hand, a sequential RSM often selects design points one at a time by iteratively optimizing a certain criterion function leading to an improved surrogate model. A variety of RSMs have been proposed in the literature (e.g., Jones et al. 1998, Gutmann 2001, Nakayama et al. 2002, Sóbester et al. 2005, Regis and Shoemaker 2007), and their applications to (stochastic) simulation optimization can be found in, for example, Huang et al. (2006), Chang et al. (2013), and Kleijnen (2015). More recently, a surrogate-based optimization algorithm has also been developed in Müller (2017) for solving deterministic problems with multiple conflicting objectives.

In this paper, we draw upon ideas from these highly successful techniques and propose an algorithm called surrogate-based promising area search (SPAS) for solving Lipschitz continuous simulation optimization problems. For a given problem, SPAS proceeds iteratively by constructing and optimizing a sequence of surrogate models, which are approximations of the objective function on promising subsets of the solution space. Each iteration of the algorithm consists of the following three steps (see Figure 1 for a schematic description): (1) Generate a set of candidate solutions by randomly sampling from the promising region constructed in the previous iteration, and use the shrinking ball technique to estimate the performance of the sampled solutions. (2) Use all candidate solutions generated thus far to build a surrogate model of the objective function. (3) Optimize the surrogate model and construct a new promising region that contains the optimal solution to the model. Intuitively, the shrinking ball method reduces the simulation noise at a sampled solution by averaging observations at solutions that are close to it, avoiding the need to allocate multiple simulation replications to the same point. The use of a promising area helps to concentrate the computational effort on subsets of the solution space. Additionally, the surrogate model is able to successively predict the response surface of the objective function by using past sampling information. Note that since the sampling of new solutions is performed within the promising region (as opposed to the entire solution domain), the use of the surrogate model in our approach is not intended to provide a global fit of the underlying response surface, but rather aims to accurately predict the objective function values at unsampled points within the current search area. This facilitates the discovery of better solutions by intensifying the search in the new promising area surrounding the best point predicted by the model. Under some appropriate conditions, we show that the sequence of surrogate model optimizers converges with probability one to the set of local optimal solutions to the original problem.

Our algorithm shares some similarities with a class of algorithms developed under the so-called trust region framework (e.g., Deng and Ferris 2009, Chang et al. 2013, Larson and Billups 2016, Chen et al. 2017), where the common idea is to use a low-order (linear or quadratic) surrogate model to approximate the true response surface over a predefined trust region and then adaptively adjust the size of the region based on the approximation quality of the model. However, unlike our approach, which does not use gradient information, the analysis of trust-region-based methods typically relies on the twice differentiability of the objective function, and some of these algorithms (e.g., Chang et al. 2013) also require the use of gradient and Hessian estimates in constructing local models and determining solution quality. In addition, for highly

**Figure 1.** A6 A Schematic Description of SPAS


(a) Objective function and the promising area
Objective function
Promising region
(b) Estimate performance of sampled points
Estimated function value
Sampled points
(c) Construct a surrogate model based on performance estimates
Surrogate model
(d) Optimize surrogate model to obtain a new promising area
New promising region
Minimum on surrogate model

nonlinear problems, because of the limited approximation capability of low-order models, the trust region radius in these algorithms may become very small, which limits the size of the region to be explored. In contrast, SPAS adopts an interpolation-based fitting strategy and allows for the use of more sophisticated yet practical surrogate models. Such models have been shown efficient in approximating high-dimensional nonlinear functions (e.g., Jones et al. 1998, Gutmann 2001, Regis and Shoemaker 2007), and when used in conjunction with promising region search, may quickly identify areas of the search space with high-quality solutions at no extra simulation effort.

A preliminary version of SPAS has been presented in Fan and Hu (2016). Their algorithm uses a different technique for estimating the performance of sampled solutions and only relies on points generated in the current iteration in fitting surrogate models. Although the algorithm is shown to be convergent, its theoretical analysis requires restrictive assumptions, for example, Hessian information of the surrogate model, which can be difficult to verify in practice. In this work, we generalize the algorithm of Fan and Hu (2016) to allow the reuse of previously generated solutions in algorithm construction. This, in effect, reduces the computational cost and thus may have a significant impact on improving the algorithm's practical efficiency. We provide a complete convergence proof for SPAS under a different, but much weaker, set of conditions than those used in Fan and Hu (2016) and conduct more comprehensive numerical experiments, including additional comparisons with three other alternative approaches, to illustrate its empirical performance.

The rest of this paper is organized as follows. In Section 2, we begin by introducing the problem setting and then provide a detailed description of the proposed algorithm. In Section 3, we analyze the algorithm and show its strong convergence to the set of local optimal solutions. Illustrative numerical studies including comparisons with two existing approaches are given in Section 4. Finally, in Section 5, we conclude the paper with discussions on some possible future research topics.

## 2. Surrogate-Based Promising Area Search

We consider the following general simulation optimization problem:

A7
$$\min_{x\in\mathbb{X}}\{H(x)=\mathbb{E}[h(x,\phi)]\}, \tag{1}$$

where the solution space $\mathbb{X}$ is a full-dimensional convex, compact subset of $\mathbb{R}^d$ with nonempty interior, $H$ is a deterministic, real-valued objective function, and $\phi$ is a random variable representing the stochastic uncertainty of the underlying system, which, for example, could be a sample path. Note that for a given solution $x$, the objective function value $H(x)$ itself takes the form of an expectation. We assume that the expectation cannot be computed analytically and instead needs to be estimated using the "noisy" sample performance $h$ obtained through computer simulation. To simplify notation, we will sometimes suppress the dependency of $h$ on the sample path $\phi$ and simply write $h(x)$ instead of $h(x, \phi)$. For ease of implementation, we also assume that $\mathbb{X}$ is characterized by simple (deterministic) constraints, for example, box constraints, so that solutions can be easily generated from it; otherwise a penalty function approach can be used by altering the objective function to include a penalty term.

### 2.1. Algorithm Description

We introduce some mathematical notation that will be needed in the rest of this paper. Let $N_k$ be the number of candidate solutions sampled at the $k$th iteration of the algorithm and $\Lambda_k$ be the set of sampled solutions. Let $V_k$ be the collection of all candidate solutions sampled up to the $k$th iteration, $\{r_k\}_{k\geq 1}$ be a sequence of deterministic positive real numbers, and $B(x, r) = \{y \in \mathbb{X}: d(x, y) < r\}$ be an open ball of center $x$ and radius $r$. For two given points $x$ and $y$ in $\mathbb{X}$, we use $d(x, y)$ to denote the Euclidean distance between them, whereas for a set $A \subseteq \mathbb{X}$, the distance between a point $x$ and the set $A$ is defined and denoted by $d(x, A) = \inf_{y\in A} d(x, y)$. Finally, let $S_k$ and $P_k \subseteq \mathbb{X}$ be the respective surrogate model and promising area constructed at the $k$th iteration of SPAS. The detailed algorithmic steps are given next.

### 2.2. Surrogate-Based Promising Area Search (SPAS)

*Step* 0. Set the iteration counter $k = 0$, $V_0 = \varnothing$, and $P_0 = \mathbb{X}$. Specify a small positive constant $\delta > 0$, a sequence of numbers $\{\alpha_k\}_{k\geq 1}$ satisfying $\alpha_k \in [0, 1)\ \forall\, k$, and a shrinking ball strategy $\{r_k\}_{k\geq 1}$.

*Step* 1. Let $k = k + 1$. Uniformly and independently sample a set of $N_k$ candidate solutions $\Lambda_k = \{x_1^k, x_2^k, \ldots, x_{N_k}^k\}$ from the current promising area $P_{k-1}$. Let $V_k = V_{k-1} \cup \Lambda_k$. Obtain the sample performance at each point in $\Lambda_k$ and use the shrinking ball method to construct performance estimates $\tilde{H}_k(x)$ for all $x \in V_k$ as follows:

$$\tilde{H}_k(x) = \alpha_k \frac{\sum_{y\in B(x, r_k)\cap V_k} h(y)}{|B(x, r_k) \cap V_k|} + (1 - \alpha_k)\frac{\sum_{y\in B(x, r_k)\cap \Lambda_k} h(y)}{|B(x, r_k) \cap \Lambda_k|}, \qquad (2)$$

where $|A|$ represents the cardinality of a set $A$.

*Step* 2. Build a surrogate model $S_k(x)$ that interpolates the objective function estimates $\tilde{H}_k(x)$ at all sampled points $x \in V_k$.

*Step* 3. Optimize the surrogate model $S_k(x)$ on $P_{k-1}$ to obtain a minimizer $x_k^*$, that is, $x_k^* \in \arg\min_{x\in P_{k-1}} S_k(x)$. Construct a new promising area $P_k$ based on $x_k^*$ as follows:

$$P_k = \left\{ y \in \mathbb{X}: d(y, x_k^*) \leq d\left( y, x + 2(x - x_k^*)\frac{\delta}{d(x_k^*, x)} \right), \forall\, x \in V_k \right\}.$$

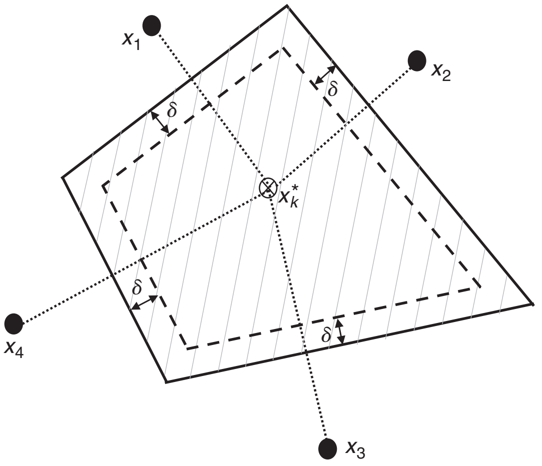
Reiterate from Step 1 until a stopping condition is satisfied.

SPAS starts by taking the initial promising area as the entire solution space and then proceeds to construct and sample from a sequence of promising areas. Each is adaptively formed based on the performance estimates collected at all points sampled up to the current iteration. Note that at Step 1, any sampling measure can be used to generate candidate solutions from the current promising region $P_{k-1}$, provided that such a measure guarantees that any subset in $P_{k-1}$ with a positive Lebesgue measure will also have a positive probability of being sampled. For simplicity, we have used a uniform distribution throughout our discussion.

A subtle issue worth mentioning is that since the construction of promising areas is adaptive, the performance estimates obtained at successive iterations of the algorithm are generally not independent. For example, the sample performance $h(x)$ at a point $x$ generated in the $k$th iteration will affect the shape and size of the promising region $P_k$ obtained at Step 3. This will in turn determine the chance/likelihood of the points to be produced in the next iteration. Thus, for a given sampled solution $x \in V_k$, the observations at points that were generated preceding it are correlated, and a straightforward estimation of its true performance $H(x)$ by averaging past observations (such as the shrinking ball method) will result in an extra bias. We address this issue by taking the estimator $\tilde{H}_k(x)$ as the convex combination of the average of the observations collected at all points in $B(x, r_k) \cap V_k$ and the average of the observations at points in $B(x, r_k) \cap \Lambda_k$ (see Equation (2)). The second average in the combination only depends on points sampled at the current iteration $\Lambda_k$ and does not suffer from the correlation bias, whereas the first term relies on past sampling information and is hence biased. This bias effect is discounted by putting a weight parameter $\alpha_k \in [0, 1)$ that diminishes as more points are generated. Intuitively speaking, since there are only a few points generated in the early iterations, setting the initial values of $\alpha_k$ large (close to 1) helps to effectively use the performance estimates collected at previously sampled points to reduce the variance of the estimator. On the other hand, as sampling gets more focused on the current promising area, the variance of the second term in (2) becomes smaller while the correlation bias accumulated in the first term can be removed by letting $\alpha_k$ decrease to zero.

Step 2 requires the use of an interpolation-based fitting strategy to ensure that $S_k(x) = \tilde{H}_k(x)$ at all $x \in V_k$. The optimization at Step 3 can be carried out using any deterministic optimization algorithm. The underlying assumption is that simulation experiments are much more expensive to run than evaluating the surrogate model, so that $S_k(x)$ can be optimized relatively efficiently without any additional simulation effort. The optimizer $x_k^*$ of $S_k(x)$ is then used at Step 3 to construct a new promising subset $P_k$, which is defined as the set of points in $\mathbb{X}$ whose distances to $x_k^*$ are less than $2\delta$ plus their distances to the set of sampled solutions. The use of the constant $\delta > 0$ ensures $P_k$ to have a nonempty interior and prevents it from degenerating into a single point when the set of sampled points becomes dense in the neighborhood of $x_k^*$. Figure 2 provides a pictorial illustration of how the promising subset is obtained in two dimensions. This is conducted in a way that is very similar to the approach proposed in Hong and Nelson (2006) with the major difference being that now the construction is based on the best point predicted by the surrogate model rather than the one with the current best estimated performance. The intuition is that when the solution space is continuous, the current best sampled solution may be far from being optimal; consequently, using the point to directly construct the promising area may result in the search of new solutions being conducted in a region that is very distant from the set of true (local) optimizers, leading to slow convergence or inferior local solutions. The surrogate model, on the other hand, retains the previous simulation information in predicting the simulation responses at unsampled solutions. Thus, if the model can correctly capture the behavior of the true response surface, then its optimizer would be a more reliable estimate of the true (local) optimal solution than the best sampled solution itself.

**Figure 2.** Graphical Illustration of the Promising Area in Two Dimensions, Where $x_k^*$ Is a Surrogate Model Minimizer and $x_1, \ldots, x_4$ Are Four Sampled Solutions



*Notes.* The region circumscribed by the dashed line represents the set of points that are closer in distance to $x_k^*$ than to the sampled solutions. The promising area (shaded region) is formed by expanding the boundaries of the set by $\delta$ unit(s) in the directions of the vectors $\overrightarrow{x_k^* x_i}$, $i = 1, \ldots, 4$.

It is interesting to observe that in SPAS, if there is no surrogate model and the promising area is taken to be the entire feasible region in all iterations, then the algorithm is identical to the deterministic shrinking ball method discussed in Baumert and Smith (2002) and Andradóttir and Prudius (2010). On the other hand, if the solution space is (discrete) integer ordered, then since each ball $B(x, r_k)$ will only contain $x$ itself (when $r_k$ becomes small enough), the shrinking ball strategy reduces to the usual sample average approximation. Thus, the algorithm (without the surrogate model) reduces to the COMPASS algorithm of Hong and Nelson (2006). In this respect, SPAS can essentially be seen as the extension of COMPASS to continuous simulation optimization.

## 3. Local Convergence of SPAS

In this section, we analyze the asymptotic behavior of SPAS and show its local convergence to the set of optimal solutions of (1). Since the algorithm is randomized, it induces a probability distribution over the set of all sequences of sampled solutions and all possible realizations of performance measures at these solutions. We denote by $P(\cdot)$ and $E[\cdot]$ the probability and expectation taken with respect to this distribution. Probability one convergence of sequences of random events is to be understood with respect to $P$. We also define $\mathcal{F}_k = \sigma\{\Lambda_1, \{h(x), x \in \Lambda_1\}, \ldots, \Lambda_k, \{h(x), x \in \Lambda_k\}\}$, $k = 1, 2, \ldots$ as the sequence of increasing $\sigma$-fields generated by the set of all sampled solutions and their corresponding sample performance measures obtained up to iteration $k$. In the rest of the paper, $F_k$ denotes the uniform sampling measure (conditional on $\mathcal{F}_{k-1}$) used at the $k$th iteration, a sequence $a_k$ is said to be $\Omega(k^n)$ if $\exists\, c > 0$ and $k_0 > 0$, s.t. $\forall\, k \geq k_0$, $a_k \geq ck^n$ and to be $\Theta(k^n)$ if $\exists\, c_1, c_2 > 0$ and $k_0 > 0$, s.t. $\forall\, k > k_0$, $c_1 k^n \leq a_k \leq c_2 k^n$.

The following assumptions are used in our analysis:

**Assumption 1 (A1).** *The objective function $H(x)$ is Lipschitz continuous on $\mathbb{X}$ with Lipschitz constant $L_1$.*

**Assumption 2 (A2).** *Conditional on $\mathcal{F}_{k-1}$ and given $\Lambda_k$, the simulation noises $h(x) - H(x)$ at all $x \in \Lambda_k$ are independent with mean zero. In addition, $h(x) - H(x)$ is uniformly bounded on $\mathbb{X}$, that is, there exists $0 < \mathcal{B} < \infty$ such that $|h(x) - H(x)| < \mathcal{B}$ for all $x \in \mathbb{X}$ w.p.1.*

**Assumption 3 (A3).** *The surrogate model $S_k(x)$ satisfies $S_k(x) = \tilde{H}_k(x)$, $\forall\, x \in V_k$. Moreover, all $S_k$'s are Lipschitz continuous on $\mathbb{X}$ with their Lipschitz constants uniformly bounded by $L_2$ for all $k$ w.p.1.*

**Assumption 4 (A4).** *$N_k = \Theta(k^t)$, $r_k = \Omega(k^{-p/d})$ with $\lim_{k\to 0} r_k = 0$, where $t$ and $p$ are two positive constants satisfying $p < t$. The weight parameter $\alpha_k$ satisfies $\alpha_k \in [0,1)\ \forall\, k$ and $\lim_{k\to\infty} \alpha_k = 0$.*

Assumptions A1 and A2 are the respective conditions on the true objective function and the simulation noise. Although the bounded noise in Assumption A2 may appear somewhat restrictive, it is acceptable in many practical situations given that the bounding constant $\mathscr{B}$ is not assumed to be known. For example, the assumption is satisfied when simulation outputs themselves are bounded. In many other cases, for example, when the batch-mean method is used to collect samples in steady-state simulation, the noises are often assumed to follow a normal distribution with finite variance. Thus, by sending the value of $\mathscr{B}$ sufficiently large, we can still obtain a good estimate on the noise distribution by taking the normal distribution truncated at $\mathscr{B}$. Assumption A3 stipulates that the fitting strategy should be interpolation based and requires the surrogate models to be Lipschitz continuous. However, note that the Lipschitz constant $L_2$, as well as $L_1$ in A1, is only used in our theoretical derivations and need not be known in practice. Assumption A4 controls the sample size $N_k$ and the decreasing speed of the shrinking ball radius to ensure that enough observations are collected in balls that shrink with time. It also gives the conditions on the weight parameter when estimating the function value.

Let $\mathscr{M}$ be the set of all local minimizers of (1). Our main result is to show that the sequence of the surrogate model minimizers $\{x_k^*\}_{k\ge 1}$ will converge to $\mathscr{M}$ with probability one. Our analysis proceeds in several steps. First, we prove the following result, which implies that the collection of sampled solutions will eventually become dense in $P_k$.

**Lemma 1.** *For any $\epsilon > 0$ and $x_{k-1} \in P_{k-1}$, define the event $A_k(x_{k-1},\epsilon) = \{\exists\, y \in \Lambda_k,\ d(x_{k-1},y) < \epsilon\}$. If Assumption A4 holds, then $\sum_{k=1}^{\infty} P(\bar{A}_k(x_{k-1},\epsilon)\,|\,\mathscr{F}_{k-1}) < \infty$ w.p.1.*

**Proof.** Note that given $\mathscr{F}_{k-1}$, $x_{k-1}^*$ is completely determined. By construction, it is easy to observe that the $\delta$-neighborhood of $x_{k-1}^*$, $B(x_{k-1}^*,\delta)$, satisfies $B(x_{k-1}^*,\delta) \subseteq P_{k-1}$ for all $k = 1,2,\dots$. In addition, since $P_{k-1}$ is also determined, $x_{k-1}$ becomes a deterministic (arbitrary) point in $P_{k-1}$. To reduce notational burden, we simply denote the point by $x$.

We claim that for any $x \in P_{k-1}$, there exists a constant $p_\epsilon > 0$ that does not depend on $k$ such that $F_k(B(x,\epsilon) \cap P_{k-1}) \ge p_\epsilon$, where recall that $F_k$ is the uniform sampling measure on $P_{k-1}$. For given $u \in \mathbb{X}$ and $r > 0$, let $\tilde{B}(u,r) = \{v \in \mathbb{R}^d\colon d(u,v) < r\}$. Clearly, from the definition of $B(u,r)$, we have $B(u,r) = \tilde{B}(u,r) \cap \mathbb{X}$. Because $\mathbb{X}$ has a nonempty interior, there exist a point $y \in \mathbb{X}$ and a constant $\bar{\epsilon} > 0$ such that $\tilde{B}(y,\bar{\epsilon}) \subseteq \mathbb{X}$. Now consider the set $C(x_{k-1}^*,\bar{\epsilon}) = \{x_{k-1}^* + \varphi e\colon \varphi \ge 0, \|e\| = 1, \cos(\bar{\epsilon}/(2\,\mathrm{dia}(\mathbb{X}))) \le ((y - x_{k-1}^*)/\|y - x_{k-1}^*\|) \cdot e\}$, that is, the cone with its vertex at $x_{k-1}^*$ that has the direction $y - x_{k-1}^*$ and maximum angle $\bar{\epsilon}/(2\,\mathrm{dia}(\mathbb{X}))$, where $\mathrm{dia}(\mathbb{X}) = \max\{d(x,y)\colon x,y \in \mathbb{X}\}$ is the diameter of $\mathbb{X}$. Since $\mathbb{X}$ is convex and compact, by following an argument used in Baumert and Smith (2002, p. 14), it can be directly verified that $C(x_{k-1}^*,\bar{\epsilon}) \cap \tilde{B}(x_{k-1}^*,\bar{\epsilon}/2) \subseteq \mathbb{X}$. This, together with the fact that $B(x_{k-1}^*,\delta) = \tilde{B}(x_{k-1}^*,\delta) \cap \mathbb{X} \subseteq P_{k-1}$, implies $C(x_{k-1}^*,\bar{\epsilon}) \cap B(x_{k-1}^*,\bar{\epsilon}_0) \subseteq P_{k-1}$, where $\bar{\epsilon}_0 = \min\{\bar{\epsilon}/2,\delta\}$. Consequently, there exist an interior point $y_{k-1} \in C(x_{k-1}^*,\bar{\epsilon}) \cap B(x_{k-1}^*,\bar{\epsilon}_0)$ and a constant $\tilde{\epsilon} > 0$, such that $\tilde{B}(y_{k-1},\tilde{\epsilon}) \subseteq C(x_{k-1}^*,\bar{\epsilon}) \cap B(x_{k-1}^*,\bar{\epsilon}_0) \subseteq P_{k-1}$, that is, the promising region $P_{k-1}$ contains a full ball with constant radius $\tilde{\epsilon}$.

Fix the ball $\tilde{B}(y_{k-1},\tilde{\epsilon}) \subseteq P_{k-1}$. For any $x \in P_{k-1}$, now define the cone $C(x,\tilde{\epsilon}) = \{x + \varphi e\colon \varphi \ge 0, \|e\| = 1, \cos(\tilde{\epsilon}/(2\,\mathrm{dia}(\mathbb{X}))) \le ((y_{k-1} - x)/\|y_{k-1} - x\|) \cdot e\}$. Again, by invoking the argument of Baumert and Smith (2002) and noticing that the set $P_{k-1}$ is compact and convex, one can similarly show that $C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\tilde{\epsilon}/2) \subseteq P_{k-1}$. This in turn suggests that $C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\epsilon_0) \subseteq P_{k-1} \cap B(x,\epsilon)$, where $\epsilon_0 = \min\{\epsilon,\tilde{\epsilon}/2\}$. As a result, we obtain

$$
\begin{aligned}
F_k(B(x,\epsilon) \cap P_{k-1}) &\ge F_k(C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\epsilon_0))\\
&= \frac{\mathrm{Vol}(C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\epsilon_0))}{\mathrm{Vol}(P_{k-1})}\\
&\ge \frac{\mathrm{Vol}(C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\epsilon_0))}{\mathrm{Vol}(\mathbb{X})} \triangleq p_\epsilon > 0,
\end{aligned}
$$

where $\mathrm{Vol}(U)$ is the volume of a set $U \subseteq \mathbb{R}^d$ and the definition of $p_\epsilon$ follows because both $\mathrm{Vol}(C(x,\tilde{\epsilon}) \cap \tilde{B}(x,\epsilon_0))$ and $\mathrm{Vol}(\mathbb{X})$ are constants.

Consequently, since the $N_k$ solutions are sampled independently at iteration $k$, we have $P(\bar{A}_k(x_{k-1},\epsilon)|\mathscr{F}_{k-1}) \le (1-p_\epsilon)^{N_k}$ w.p.1 for every $k \ge 1$. Finally, using the condition on $N_k$ in Assumption A4 and applying the Bonferroni inequality, we obtain $\sum_{k=1}^{\infty} P(\bar{A}_k(x_{k-1},\epsilon)|\mathscr{F}_{k-1}) < \infty$ w.p.1. □

The next result shows that for any point $x$ in the promising area $P_{k-1}$, its true objective function value $H(x)$ can be closely approximated by the surrogate model $S_k(x)$ as the number of iterations gets large.

**Lemma 2.** *If Assumptions A1–A4 hold, then for any $\epsilon > 0$ and $x_{k-1} \in P_{k-1}$, we have $\sum_{k=1}^{\infty} P(|S_k(x_{k-1}) - H(x_{k-1})| > \epsilon\,|\mathscr{F}_{k-1}) < \infty$ w.p.1.*

**Proof.** From the conditions on $N_k$ and $r_k$ in Assumption A4, there exists a positive constant $q > 0$ such that $p + q < t$. Define $l_k = \Theta(k^q)$ and let $D_k = \{\forall\, x \in P_{k-1},\ N_k(x,r_k) \ge l_k\}$, where $N_k(x,r_k) = |B(x,r_k) \cap \Lambda_k|$, that is, the number of sampled points in $\Lambda_k$ that lie in the ball $B(x,r_k)$. Since $r_k, \alpha_k \to 0$ as $k \to \infty$ and $N_k = \Theta(k^t)$, there exist $k' \in \mathbb{N}$ and constants $c_1, c_2 > 0$ such that $r_k \le \epsilon/(4L_1)$, $\alpha_k \le \epsilon/(8\mathscr{B})$, $l_k \ge c_1 k^q$, and $N_k \le c_2 k^t$

for all $k \geq k'$. For simplicity, we write the set $B(x, r_k) \cap V_k$ as $B'_k(x)$ and $B(x, r_k) \cap \Lambda_k$ as $B_k(x)$. Let $\epsilon' = \epsilon/(2(L_1 + L_2))$ and denote by $\mathbf{x}_k$ a random variable over $P_{k-1}$ distributed according to $F_k$. Thus, conditional on $\mathscr{F}_{k-1}$, for any given $x_{k-1} = x \in P_{k-1}$ and $k \geq k'$, we have

$$
\begin{aligned}
&P(|S_k(x)-H(x)|>\epsilon \mid \mathscr{F}_{k-1})\\
&\leq P(|S_k(x)-H(x)|>\epsilon, A_k(x,\epsilon') \mid \mathscr{F}_{k-1})+P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})\\
&=P\left(|S_k(x)-H(x)|>\epsilon, \bigcup_{y\in\Lambda_k} d(x,y)<\epsilon' \,\middle|\, \mathscr{F}_{k-1}\right)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})\\
&\leq |\Lambda_k| P(|S_k(x)-H(x)|>\epsilon, d(x,\mathbf{x}_k)<\epsilon' \mid \mathscr{F}_{k-1})\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1}) \quad \text{where } \mathbf{x}_k \sim F_k\\
&\leq N_k P(|S_k(x)-S_k(\mathbf{x}_k)|+|S_k(\mathbf{x}_k)-H(\mathbf{x}_k)|+|H(\mathbf{x}_k)-H(x)|\\
&\quad >\epsilon, d(x,\mathbf{x}_k)<\epsilon' \mid \mathscr{F}_{k-1})+P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})\\
&\leq c_2 k^t P(|S_k(\mathbf{x}_k)-H(\mathbf{x}_k)|>\epsilon/2 \mid \mathscr{F}_{k-1})\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1}) \qquad (3)\\
&\leq c_2 k^t P(|S_k(\mathbf{x}_k)-H(\mathbf{x}_k)|>\epsilon/2, D_k \mid \mathscr{F}_{k-1})\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1})\\
&=c_2 k^t P\Bigg(\Bigg|\alpha_k \frac{\sum_{y\in B'_k(x_k)}(h(y)-H(\mathbf{x}_k))}{|B'_k(\mathbf{x}_k)|}\\
&\qquad +(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(h(y)-H(\mathbf{x}_k))}{|B_k(\mathbf{x}_k)|}\Bigg|>\epsilon/2, D_k \,\Bigg|\, \mathscr{F}_{k-1}\Bigg)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1})\\
&\leq c_2 k^t P\Bigg(\Bigg|\alpha_k \frac{\sum_{y\in B'_k(x_k)}(h(y)-H(y))}{|B'_k(\mathbf{x}_k)|}\\
&\qquad +(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(h(y)-H(y))}{|B_k(\mathbf{x}_k)|}\Bigg|\\
&\qquad +\Bigg|\alpha_k \frac{\sum_{y\in B'_k(x_k)}(H(y)-H(\mathbf{x}_k))}{|B'_k(\mathbf{x}_k)|}\\
&\qquad +(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(H(y)-H(\mathbf{x}_k))}{|B_k(\mathbf{x}_k)|}\Bigg|\\
&\qquad\qquad >\epsilon/2, D_k \mid \mathscr{F}_{k-1}\Bigg)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1})\\
&\leq c_2 k^t P\Bigg(\Bigg|\alpha_k \frac{\sum_{y\in B'_k(x_k)}(h(y)-H(y))}{|B'_k(\mathbf{x}_k)|}\\
&\qquad +(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(h(y)-H(y))}{|B_k(\mathbf{x}_k)|}\Bigg|>\epsilon/4, D_k \mid \mathscr{F}_{k-1}\Bigg)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1}) \qquad (4)\\
&\leq c_2 k^t P\Bigg(\Bigg|\alpha_k \frac{\sum_{y\in B'_k(x_k)}(h(y)-H(y))}{|B'_k(\mathbf{x}_k)|}\Bigg|\\
&\qquad +\Bigg|(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(h(y)-H(y))}{|B_k(\mathbf{x}_k)|}\Bigg|>\epsilon/4, D_k \mid \mathscr{F}_{k-1}\Bigg)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1})\\
&\leq c_2 k^t P\Bigg(\Bigg|(1-\alpha_k)\frac{\sum_{y\in B_k(x_k)}(h(y)-H(y))}{|B_k(\mathbf{x}_k)|}\Bigg|>\epsilon/8, D_k \mid \mathscr{F}_{k-1}\Bigg)\\
&\quad +P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1}) \qquad (5)\\
&=c_2 k^t \int P\Bigg(\Bigg|\frac{\sum_{y\in B_k(z)}(h(y)-H(y))}{|B_k(z)|}\Bigg|\\
&\qquad\qquad >\frac{\epsilon}{8(1-\alpha_k)}, D_k \mid \mathbf{x}_k=z, \mathscr{F}_{k-1}\Bigg)\\
&\quad \cdot F_k(dz)+P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})+c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1}) \qquad (6)\\
&\leq 2c_2 k^t e^{-(c_1 k^q \epsilon^2)/(32\mathscr{B}^2)}+P(\bar{A}_k(x,\epsilon') \mid \mathscr{F}_{k-1})\\
&\quad +c_2 k^t P(\bar{D}_k \mid \mathscr{F}_{k-1}) \quad \text{w.p.1,}
\end{aligned}
$$

where (3) follows from $d(x, \mathbf{x}_k) < \epsilon'$ and the Lipschitz continuity of $H$ and $S_k$, (4) holds because $r_k \leq \epsilon/(4L_1)$ and that every point $y$ in $B'_k(\mathbf{x}_k)$ or $B_k(\mathbf{x}_k)$ is at most $r_k$ distance away from $\mathbf{x}_k$, and (5) is derived using the bounded noise condition in Assumption A2 and the fact $\alpha_k \mathscr{B} \leq \epsilon/8$. Finally, the first term in (6) follows from an application of Hoeffding's inequality (Hoeffding 1963) to the conditional probability in the integrand.

By using a slight modification of the proof of Lemma 2 in Andradóttir and Prudius (2010), with the feasible region there replaced by $P_{k-1}$ and total sample size by $N_k = \Theta(k^t)$, it can be shown that $P(\bar{D}_k \mid \mathscr{F}_{k-1}) \leq c_3 e^{-k^{q+\varepsilon}}$ w.p.1 for some constant $c_3 > 0$, where $\varepsilon > 0$ satisfies $q + \varepsilon < t - p$. Thus, by applying the result of Lemma 1 and noticing that $\sum_{k=1}^{\infty} k^t e^{-c_1 k^q \epsilon^2/(32\mathscr{B}^2)} < \infty$ and $\sum_{k=1}^{\infty} k^t e^{-k^{q+\varepsilon}} < \infty$, we obtain $\sum_{k=1}^{\infty} P(|S_k(x_{k-1}) - H(x_{k-1})| > \epsilon \mid \mathscr{F}_{k-1}) < \infty$ w.p.1. □

The next result is a strengthened version of Lemma 2, which shows that the objective function $H(x)$ can be closely approximated by the surrogate model $S_k(x)$ uniformly for all points $x$ in the promising area $P_{k-1}$.

**Proposition 1.** *If Assumptions* A1–A4 *hold, then for any* $\epsilon > 0$, $P(\max_{x \in P_{k-1}} |S_k(x) - H(x)| > \epsilon \ i.o.) = 0$.

**Proof.** Select a constant $r \in (0, \epsilon/(2(L_1 + L_2))]$. Let $\bigcup_{v \in \mathbb{X}} B(v, r/2)$ be an open cover of $\mathbb{X}$. Because $\mathbb{X}$ is compact, there exists a finite collection of points $\mathscr{G} = \{v_1, \ldots, v_{\mathscr{S}}\}$ such that $\mathbb{X} \subseteq \bigcup_{v \in \mathscr{G}} B(v, r/2)$. Note that if $B(v, r/2) \cap P_{k-1} \neq \varnothing$, then the distance between any two points contained in $B(v, r/2) \cap P_{k-1}$ is at most $r$. Thus, for every $v \in \mathscr{G}$, we can find a point $v' \in P_{k-1}$ such that $B(v, r/2) \cap P_{k-1} \subseteq B(v', r)$. This, together with the fact that $P_{k-1} \subseteq \bigcup_{v \in \mathscr{G}} [B(v, r/2) \cap P_{k-1}]$, implies the existence of a collection of finite points $\mathscr{G}_{k-1} = \{v_{k-1}^1, v_{k-1}^2, \ldots, v_{k-1}^{s_{k-1}}\}$ satisfying $v_{k-1}^i \in P_{k-1}$ for all $i = 1, 2, \ldots, s_{k-1}$ and $s_{k-1} \leq \mathscr{S}$ so that $P_{k-1} \subseteq \bigcup_{v \in \mathscr{G}_{k-1}} B(v, r)$. Hence, for any $x \in P_{k-1}$, there exists a point $v_{k-1}^j \in \mathscr{G}_{k-1}$ such that $d(x, v_{k-1}^j) \leq r$.

Consequently, by the Lipschitz continuity of $H(x)$ and $S_k(x)$, if $|S_k(v_{k-1}^j) - H(v_{k-1}^j)| \leq \epsilon/2$ for all $j = 1, 2, \ldots, s_{k-1}$, then we must have $|S_k(x) - H(x)| \leq |S_k(x) - S_k(v_{k-1}^j)| + |S_k(v_{k-1}^j) - H(v_{k-1}^j)| + |H(v_{k-1}^j) - H(x)| \leq \epsilon/2 +$

$\epsilon/2 \le \epsilon$ for all $x \in P_{k-1}$. This shows that $P(\forall\, v \in \mathcal{G}_{k-1}, |S_k(v) - H(v)| \le \epsilon/2 \,|\, \mathcal{F}_{k-1}) \le P(\forall\, x \in P_{k-1}, |S_k(x) - H(x)| \le \epsilon \,|\, \mathcal{F}_{k-1})$. It follows that

$$\begin{aligned} &P\Big(\max_{x\in P_{k-1}} |S_k(x)-H(x)| > \epsilon \,|\, \mathcal{F}_{k-1}\Big)\\ &\quad= P(\exists\, x \in P_{k-1},\, |S_k(x)-H(x)| > \epsilon \,|\, \mathcal{F}_{k-1})\\ &\quad\le P(\exists\, v \in \mathcal{G}_{k-1},\, |S_k(v)-H(v)| > \epsilon/2 \,|\, \mathcal{F}_{k-1})\\ &\quad\le \mathcal{S} \max_{j=1,\dots,s_{k-1}} P(|S_k(v^j_{k-1}) - H(v^j_{k-1})| > \epsilon/2 \,|\, \mathcal{F}_{k-1}). \end{aligned}$$

The last inequality holds because the number of points in $\mathcal{G}_{k-1}$ is bounded above by $\mathcal{S}$. Then, we sum over $k$ to get

$$\begin{aligned} &\sum_{k=1}^{\infty} P\Big(\max_{x\in P_{k-1}} |S_k(x)-H(x)| > \epsilon \,|\, \mathcal{F}_{k-1}\Big)\\ &\le \mathcal{S} \sum_{k=1}^{\infty} \max_{j=1,\dots,s_{k-1}} P(|S_k(v^j_{k-1}) - H(v^j_{k-1})| > \epsilon/2 \,|\, \mathcal{F}_{k-1}) < \infty\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{w.p.1,} \qquad (7) \end{aligned}$$

where the finiteness of the last sum follows from the proof of Lemma 2 because $s_{k-1} \le \mathcal{S}$ is finite and $v^j_{k-1} \in P_{k-1}$ $\forall k$. Finally, since the event $\{\max_{x\in P_{k-1}} |S_k(x) - H(x)| > \epsilon\}$ is $\mathcal{F}_k$-measurable, we conclude from the Borel-Cantelli-Lévy lemma (Shiryaev 1996) that

$$\begin{aligned} &P\Big(\max_{x\in P_{k-1}} |S_k(x)-H(x)| > \epsilon \text{ infinitely often}\Big)\\ &= P\Bigg(\sum_{k=1}^{\infty} P(\max_{x\in P_{k-1}} |S_k(x)-H(x)| > \epsilon \,|\, \mathcal{F}_{k-1}) = \infty\Bigg) = 0. \quad \square \end{aligned}$$

Since $x^*_k \in \arg\min_{x\in P_{k-1}} S_k(x)$ and $S_k(x)$ is close to $H(x)$ uniformly over $P_{k-1}$, it is reasonable to expect that $H(x^*_k)$ should also be close to the minimum of the function $H(x)$ over the promising area $P_{k-1}$. This intuition is formalized below.

**Lemma 3.** *If Assumptions A1–A4 hold, then for any $\epsilon > 0$, $P(|H(x^*_k) - \min_{x\in P_{k-1}} H(x)| > \epsilon$ i.o.$) = 0$.*

**Proof.** By conditioning on the $\sigma$-field $\mathcal{F}_{k-1}$, we have

$$\begin{aligned} &P\Big(\Big|H(x^*_k) - \min_{x\in P_{k-1}} H(x)\Big| > \epsilon \,\Big|\, \mathcal{F}_{k-1}\Big)\\ &\quad\le P(|H(x^*_k) - S_k(x^*_k)| > \epsilon/2 \,|\, \mathcal{F}_{k-1})\\ &\qquad + P\Big(\Big|S_k(x^*_k) - \min_{x\in P_{k-1}} H(x)\Big| > \epsilon/2 \,\Big|\, \mathcal{F}_{k-1}\Big)\\ &\quad\le P\Big(\max_{x\in P_{k-1}} |H(x) - S_k(x)| > \epsilon/2 \,\Big|\, \mathcal{F}_{k-1}\Big)\\ &\qquad + P\Big(\Big|\min_{x\in P_{k-1}} H(x) - \min_{x\in P_{k-1}} S_k(x)\Big| > \epsilon/2 \,\Big|\, \mathcal{F}_{k-1}\Big)\\ &\quad\le 2P\Big(\max_{x\in P_{k-1}} |S_k(x) - H(x)| > \epsilon/2 \,\Big|\, \mathcal{F}_{k-1}\Big). \end{aligned}$$

Thus, it follows from Equation (7) in the proof of Proposition 1 that

$$\begin{aligned} &\sum_{k=1}^{\infty} P\Big(\Big|H(x^*_k) - \min_{x\in P_{k-1}} H(x)\Big| > \epsilon \,\Big|\, \mathcal{F}_{k-1}\Big)\\ &\le 2\sum_{k=1}^{\infty} P\Big(\max_{x\in P_{k-1}} |S_k(x) - H(x)| > \epsilon/2 \,\Big|\, \mathcal{F}_{k-1}\Big) < \infty \quad \text{w.p.1.} \end{aligned}$$

Consequently, by noticing that $\{|H(x^*_k) - \min_{x\in P_{k-1}} H(x)| > \epsilon\} \in \mathcal{F}_k$ $\forall k$, a direct application of the Borel-Cantelli-Lévy lemma yields $P(|H(x^*_k) - \min_{x\in P_{k-1}} H(x)| > \epsilon$ i.o.$) = 0$. □

In addition, since $S_k(x)$ is uniformly close to the true objective function $H(x)$ on $P_{k-1}$, Proposition 1 also suggests that the distance between $x^*_k$ and the set of minimizers of $H$ on the promising area $P_{k-1}$ will approach zero as $k$ tends to infinity. This leads to Proposition 2.

**Proposition 2.** *If Assumptions A1–A4 hold, then*

$$P\Big(\lim_{k\to\infty} d\big(x^*_k, \arg\min_{x\in P_{k-1}} H(x)\big) = 0\Big) = 1.$$

**Proof.** Let $\Omega_1 = \{\lim_{k\to\infty} \max_{x\in P_{k-1}} |S_k(x) - H(x)| = 0\}$. For a given sample path $\omega \in \Omega_1$, let $\varepsilon > 0$ be a small constant and denote the set of minimizers of $H(x)$ on $P_{k-1}(\omega)$ as $\mathcal{M}_{k-1}(\omega) = \arg\min_{x\in P_{k-1}(\omega)} H(x)$. Define $A_{k-1}(\omega) := \{x \in P_{k-1}(\omega): d(x, \mathcal{M}_{k-1}(\omega)) \ge \varepsilon\}$, that is, the set of points in $P_{k-1}(\omega)$ that are at least $\varepsilon$ distance away from $\mathcal{M}_{k-1}(\omega)$. Note that since $H(x)$ is continuous and $P_{k-1}$ is compact, the set of minimizers $\mathcal{M}_{k-1}(\omega)$ is compact. Thus, it can be seen that $A_{k-1}(\omega)$ is also compact, and therefore $H(x)$ attains a minimum value on $A_{k-1}(\omega)$.

Now consider the difference

$$\gamma = \min_{x\in A_{k-1}(\omega)} H(x) - \min_{x\in P_{k-1}(\omega)} H(x).$$

Clearly, since $A_{k-1}(\omega) \cap \mathcal{M}_{k-1}(\omega) = \varnothing$, we must have $\gamma > 0$. Thus, there exists $K_1(\omega)$ sufficiently large such that $\max_{x\in P_{k-1}(\omega)} |S_k(x) - H(x)| < \gamma/2$ for all $k \ge K_1(\omega)$. And because $|\min_{x\in P_{k-1}(\omega)} S_k(x) - \min_{x\in P_{k-1}(\omega)} H(x)| \le \max_{x\in P_{k-1}(\omega)} |S_k(x) - H(x)|$, we get $\min_{x\in P_{k-1}(\omega)} S_k(x) < \min_{x\in P_{k-1}(\omega)} H(x) + \gamma/2$ for all $k \ge K_1(\omega)$.

On the other hand, by the definition of $\gamma$, for any $x \in A_{k-1}(\omega)$, we have $H(x) \ge \min_{x\in A_{k-1}(\omega)} H(x) = \min_{x\in P_{k-1}(\omega)} H(x) + \gamma$. This, when combined with $\max_{x\in P_{k-1}(\omega)} |S_k(x) - H(x)| < \gamma/2$, shows that $S_k(x) > H(x) - \gamma/2 \ge \min_{x\in P_{k-1}(\omega)} H(x) + \gamma/2$ for all $k \ge K_1(\omega)$. Consequently, when $k$ is large enough, we obtain $S_k(x) > \min_{x\in P_{k-1}(\omega)} S_k(x)$ for any $x \in A_{k-1}(\omega)$, that is, no point in $A_{k-1}(\omega)$ can be an optimizer of $S_k(x)$. Thus, since $x^*_k$ is a minimizer of $S_k(x)$, it must be contained in $A^c_{k-1}(\omega)$, that is, $d(x^*_k, \mathcal{M}_{k-1}(\omega)) < \varepsilon$. Hence, the proof is completed by noticing that $\varepsilon$ is arbitrary and $P(\Omega_1) = 1$.

Finally, we arrive at the following convergence result for the SPAS algorithm.

**Theorem 1.** *If Assumptions A1–A4 hold, then*

$$P(\lim_{k\to\infty} d(x_k^*,\mathscr{M})=0)=1.$$

**Proof.** Let $\Omega_2=\{\lim_{k\to\infty} d(x_k^*, \operatorname{argmin}_{x\in P_{k-1}} H(x))=0\}$. From Proposition 2, we know that $P(\Omega_2)=1$. For each path $\omega\in\Omega_2$, since $\lim_{k\to\infty} d(x_k^*(\omega), \operatorname{argmin}_{x\in P_{k-1}(\omega)} H(x))=0$, we can find a $K_2(\omega)>0$ sufficiently large so that for all $k\geq K_2(\omega)$, $\operatorname{argmin}_{x\in P_{k-1}(\omega)} H(x)\cap B(x_k^*(\omega),\delta)\neq\varnothing$, where recall that $\delta>0$ is the parameter used in the algorithm to construct promising areas. In addition, because $B(x_k^*(\omega),\delta)\subseteq P_k(\omega)$ by construction, we have $\operatorname{argmin}_{x\in P_{k-1}(\omega)} H(x)\cap P_k(\omega)\neq\varnothing$. Let $x'\in \operatorname{argmin}_{x\in P_{k-1}(\omega)} H(x)\cap P_k(\omega)$ be arbitrary. It hence follows that

$$\min_{x\in P_k(\omega)} H(x)\leq H(x')=\min_{x\in P_{k-1}(\omega)} H(x).$$

Since $\mathbb{X}$ is compact, $H(x)$ has a lower bound. Thus, the monotone convergence theorem indicates that the sequence of minimum function values $\{\min_{x\in P_k(\omega)} H(x)\}_{k\geq1}$ has a limit, and because this holds for all $\omega\in\Omega_2$, we conclude that the sequence $\{\min_{x\in P_k} H(x)\}_{k\geq1}$ converges w.p.1.

As a result, since $\lim_{k\to\infty}|H(x_k^*)-\min_{x\in P_{k-1}} H(x)|=0$ w.p.1. by Lemma 3, $\{H(x_k^*)\}_{k\geq1}$ must have the same limit as $\{\min_{x\in P_k} H(x)\}_{k\geq1}$. Let $\Omega_3=\{\lim_{k\to\infty} H(x_k^*)=\lim_{k\to\infty}\min_{x\in P_k} H(x)\}$. For each $\omega\in\Omega_3$, because of the compactness of $\mathbb{X}$, the sequence $\{x_k^*(\omega)\}_{k\geq1}$ has a convergent subsequence $\{x_{k_i}^*(\omega)\}_{i\geq1}$ such that $\lim_{i\to\infty} x_{k_i}^*(\omega)=x^*(\omega)\in\mathbb{X}$. Moreover, since $H(x)$ is continuous, it is not difficult to show that $\lim_{i\to\infty}\min_{x\in \mathrm{cl}(B(x_{k_i}^*(\omega)),\delta)} H(x)=\min_{x\in \mathrm{cl}(B(x^*(\omega),\delta))} H(x)$, where $\mathrm{cl}(B(v,r))$ is the closure of the open ball $B(v,r)$. Then the following relation holds:

$$\begin{aligned} H(x^*(\omega)) &= \lim_{i\to\infty} H(x_{k_i}^*(\omega))=\lim_{k\to\infty} H(x_k^*(\omega))\\ &= \lim_{k\to\infty}\min_{x\in P_k(\omega)} H(x)=\lim_{i\to\infty}\min_{x\in P_{k_i}(\omega)} H(x)\\ &\leq \lim_{i\to\infty}\min_{x\in \mathrm{cl}(B(x_{k_i}^*(\omega)),\delta)} H(x)=\min_{x\in \mathrm{cl}(B(x^*(\omega),\delta))} H(x), \end{aligned}$$

which indicates $x^*(\omega)\in\mathscr{M}$. This implies that any limit point of the sequence $\{x_k^*(\omega)\}_{k\geq1}$ is a local minimizer of $H(x)$.

Finally, to prove the desired result, we proceed by contradiction and assume that $\limsup_{k\to\infty} d(x_k^*(\omega),\mathscr{M}):=\bar{d}>0$. This suggests that there is an infinite subsequence of $\{x_k^*(\omega)\}_{k\geq1}$, denoted by $\{x_{k_j}^*(\omega)\}_{j\geq1}$, such that $d(x_{k_j}^*(\omega),\mathscr{M})\geq\bar{d}/2>0$ for all $j$. Then every limit point $x_0^*(\omega)$ of $\{x_{k_j}^*(\omega)\}_{j\geq1}$ satisfies $d(x_0^*(\omega),\mathscr{M})\geq\bar{d}/2>0$, that is, $x_0^*(\omega)\notin\mathscr{M}$. However, according to our previous argument, this contradicts the fact that $x_0^*(\omega)\in\mathscr{M}$, since $x_0^*(\omega)$ itself is also a limit point of the sequence $\{x_k^*(\omega)\}_{k\geq1}$. Therefore, we must have $\limsup_{k\to\infty} d(x_k^*(\omega),\mathscr{M})=0$ for all $\omega\in\Omega_3$, and because $P(\Omega_3)=1$, this result holds w.p.1. □

## 4. Numerical Examples

In this section, we illustrate the performance of SPAS through some computational experiments on a set of optimization benchmark functions and an $(s,S)$-inventory control problem. In the implementation of the algorithm, we have used a Markov chain-based technique discussed in Smith (1984) to sample candidate solutions from the promising region at each iteration. It has been shown that under certain conditions, the points generated by this method are asymptotically uniformly distributed within a given bounded region. In particular, at the $k$th iteration of SPAS, the sampling procedure begins by taking the surrogate model optimizer $x_{k-1}^*\in P_{k-1}$ as a starting point. It then chooses a random direction $\theta\in\mathbb{R}^d$ (i.e., a vector sampled from the uniform distribution on a unit $d$-dimensional hypersphere), and subsequently generates a new point uniformly on the line $\{x\in\mathbb{X}\colon\, x=x_{k-1}^*+\lambda\theta, \lambda\in\mathbb{R}\}\cap P_{k-1}$. These steps are then repeated by using the new point as the initial point and stopped when the required number of points is attained. In our experiments, a warm-up length of 50 epochs is used when collecting the $N_k$ candidate solutions, that is, the first 50 points generated are discarded and the sampled candidate solutions are taken to be the next $N_k$ points in the sequence.

The surrogate model is constructed using the radial basis function (RBF) approximation method (see, e.g., Bishop 1995, Gutmann 2001), which has been successfully used as a curve fitting tool in surrogate-based optimization. The specific approximator considered here is a linear combination of RBFs of the following form: $S_k(x)=\sum_{i=1}^{|V_k|} w_i\psi(\|x-x_i\|)$, where $\psi(r)=r^3$, $x_i$'s are the sampled solutions, and $w_i$'s are the weights of the basis functions, which can be computed by solving a system of linear equations. Note that since $\psi$ is chosen to be a polynomial of degree 3, the derivative of $S_k(x)$ admits an explicit expression and its minimization over the promising area at Step 3 of the algorithm can be conveniently carried out using a straightforward gradient descent method with random restart.

### 4.1. Deterministic Functions with Added Noise

Tests were performed on 10 deterministic functions with added noise. These functions are well known and have been widely used in the literature to investigate the performance of various optimization algorithms. In particular, problems $h_1$, $h_2$, and $h_3$ are unimodal, each with a unique local (global) minimizer. $h_4$ and $h_5$ are low-dimensional problems with a few local minima, while the last five functions $h_6$–$h_{10}$ are highly multimodal with the number of local minima grows exponentially with the problem dimension. In each case, the added noise is assumed to follow a zero-mean truncated normal distribution $\mathcal{TN}(0,\sigma^2)$, which is the normal distribution $\mathcal{N}(0,\sigma^2)$ truncated over the region $[-3\sigma,3\sigma]$.

(1) Beale function with added noise

$$h_1(x,\phi_1)=[1.5-2x_1(1-2x_2)]^2+[2.25-x_1(1-x_2^2)]^2 +[2.625-x_1(1-x_2^3)]^2+1+\phi_1,$$

where $-10\le x_i\le 10$, $i=1,2$ and $\phi_1\sim\mathcal{TN}(0,1)$. The function $H_1(x)=E[h_1(x,\phi_1)]$ has only one minimizer $x_1^*=(3,0.5)$ with function value $H_1(x_1^*)=1$.

(2) Powell Singular function with added noise

$$h_2(x,\phi_2)=(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4 +10(x_1-x_4)^4+1+\phi_2,$$

where $-10\le x_i\le 10$, $i=1,2,3,4$ and $\phi_2\sim\mathcal{TN}(0,1)$. The function $H_2(x)=E[h_2(x,\phi_2)]$ has only one minimizer $x_2^*=(0,0,0,0)$ with function value $H_2(x_2^*)=1$.

(3) Asymmetric function with added noise ($n=10$)

$$h_3(x,\phi_3)=\sum_{i=1}^{n}[2^{x_i-4}+(6-x_i)]+\phi_3,$$

where $-10\le x_i\le 10$, $i=1,2,\ldots,n$ and $\phi_3\sim\mathcal{TN}(0,25)$. The function $H_3(x)=E[h_3(x,\phi_3)]$ has only one minimizer $x_3^*=(4.529,4.529,\ldots,4.529)$ with function value $H_3(x_3^*)=2.9139n$.

(4) Goldstein-Price function with added noise

$$h_4(x,\phi_4)=(1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2 +6x_1x_2+3x_2^2))(30+(2x_1-3x_2)^2(18-32x_1 +12x_1^2+48x_2-36x_1x_2+27x_2^2))+\phi_4,$$

where $-3\le x_i\le 3$, $i=1,2$ and $\phi_4\sim\mathcal{TN}(0,1)$. The function $H_4(x)=E[h_4(x,\phi_4)]$ has three local minima $(-0.6,-0.4),(1.8,0.2),(1.2,0.8)$ and a global minimizer $x_4^*=(0,-1)$ with function value $H_4(x_4^*)=3$.

(5) Griewank function with added noise ($n=2$)

$$h_5(x,\phi_5)=\frac{1}{4{,}000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+\phi_5,$$

where $-10\le x_i\le 10$, $i=1,2,\ldots,n$ and $\phi_5\sim\mathcal{TN}(0,1)$. The function $H_5(x)=E[h_5(x,\phi_5)]$ has a global minimizer $x_5^*=(0,0,\ldots,0)$ with function value $H_5(x_5^*)=0$.

(6) Styblinski-Tang function with added noise ($n=10$)

$$h_6(x,\phi_6)=\sum_{i=1}^{n}[x_i^4-16x_i^2+5x_i]/(2n)+40.166+\phi_6,$$

where $-10\le x_i\le 10$, $i=1,2,\ldots,n$ and $\phi_6\sim\mathcal{TN}(0,100)$. The function $H_6(x)=E[h_6(x,\phi_6)]$ has a global minimizer $x_6^*=(-2.9035,-2.9035,\ldots,-2.9035)$ with function value $H_6(x_6^*)=1$.

(7) Rastrigin function with added noise ($n=10$)

$$h_7(x,\phi_7)=10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]+\phi_7,$$

where $-5.12\le x_i\le 5.12$, $i=1,2,\ldots,n$ and $\phi_7\sim\mathcal{TN}(0,25)$. The function $H_7(x)=E[h_7(x,\phi_7)]$ has a global minimizer $x_7^*=(0,0,\ldots,0)$ with function value $H_7(x_7^*)=0$.

(8) Schwefel function with added noise ($n=10$)

$$h_8(x,\phi_8)=201.8432n-\sum_{i=1}^{n}x_i\sin(\sqrt{|x_i|})+\phi_8,$$

where $-200\le x_i\le 250$, $i=1,2,\ldots,n$ and $\phi_8\sim\mathcal{TN}(0,100)$. The function $H_8(x)=E[h_8(x,\phi_8)]$ has a global minimizer $x_8^*=(203.814,203.814,\ldots,203.814)$ with function value $H_8(x_8^*)=0$.

(9) Rosenbrock function with added noise ($n=10$)

$$h_9(x,\phi_9)=\sum_{i=1}^{n}100(x_{i-1}-x_i^2)^2+(x_i-1)^2+1+\phi_9,$$

where $-10\le x_i\le 10$, $i=1,2,\ldots,n$ and $\phi_9\sim\mathcal{TN}(0,100)$. The function $H_9(x)=E[h_9(x,\phi_9)]$ has a global minimizer $x_9^*=(1,1,\ldots,1)$ with function value $H_9(x_9^*)=1$.

(10) Trigonometric function with added noise ($n=10$)

$$h_{10}(x,\phi_{10})=\sum_{i=1}^{n}[8\sin^2(7(x_i-0.9)^2)+6\sin^2(14(x_i-0.9)^2) +(x_i-0.9)^2]+\phi_{10},$$

where $-2\le x_i\le 3$, $i=1,2,\ldots,n$ and $\phi_{10}\sim\mathcal{TN}(0,25)$. The function $H_{10}(x)=E[h_{10}(x,\phi_{10})]$ has a global minimizer $x_{10}^*=(0.90009,0.90009,\ldots,0.90009)$ with function value $H_{10}(x_{10}^*)=0$.

The parameters of SPAS are set as follows: $\delta=1$, per iteration sample size $N_k=\max(\sqrt{k},4)$, and the shrinking ball radius is taken to be of the form $r_k=a/k^{p/d}$. Intuitively, since $p$ controls the decreasing speed of $r_k$, its specification should be based on the choice of the sample size $N_k$. For example, if a large number of candidate solutions are allowed at each iteration, then the decreasing rate of $r_k$ can generally be made faster. Therefore, we recommend to choose the value of $p$ close to but smaller than $t$. Because $N_k$ is taken to be on the order of $\sqrt{k}$ (i.e., $t=0.5$), we simply set $p=0.49$ in our experiments. We have experimented with different values of $a$ and found that a good choice of $a$ should depend on the size of region $\mathbb{X}$. Therefore, we set its value to be around 5% of the maximum length of the domain, which yields reasonable performance in all test cases considered. The weight parameter is chosen to have a slow decay rate $\alpha_k=\ln(100)/\ln(100+k)$. This helps to make more efficient use of the past sampling information and thus reduce the variance of the performance estimator when the number of sampled points is small. Note that the above parameter setting satisfies the relevant conditions in Assumption A4 for convergence.

For comparison purposes, we have also applied the simultaneous perturbation stochastic approximation (SPSA) algorithm (Spall 1992) and the STRONG method of Chang et al. (2013) on the 10 testing problems. The former is a stochastic approximation type of algorithm that requires only two function evaluations in estimating the gradient, whereas the latter, as discussed in Section 1, is based on constructing and optimizing surrogate models defined on trust regions. To further illustrate the benefit of using surrogate models in the proposed algorithm, we have also included a simplified version of SPAS, called PAS, in our comparison. PAS has the same structure as SPAS but without the surrogate model approximation step, and the promising region is constructed at each iteration based on the current best sampled solution. Thus, following our discussion at the end of Section 2, PAS is essentially a version of COMPASS applied to continuous simulation optimization. The values of the parameters used in SPAS, PAS, and SPSA are listed in Tables 1 and 2, where for SPSA, $\alpha_k$ and $c_k$ are the respective gain and simultaneous perturbation sizes. These parameters are selected based on trial and error in each case to achieve good performance of SPSA. All parameters used in STRONG are taken to be the same as those recommended in Chang et al. (2013).

Figures 3 and 4 show the performance of the four comparison algorithms, averaged over 50 independent replication runs, on each of the respective test cases. It is easy to observe that SPAS yields reasonably good performance in all cases. In particular, on all unimodal functions $h_1$–$h_3$, the algorithm converges to the unique optimal solution in all runs, whereas in the rest of the cases, SPAS outperforms or at least shows comparable performance to both SPSA and STRONG. In addition to its superior performance in the long run when the number of function evaluations gets large, the algorithm (and PAS) has a significantly faster initial improvement than the other two methods. We conjecture that this is because both SPAS and PAS are population based and initially explore the entire solution space, and thus may quickly identify a good promising area after only a few algorithm iterations. In contrast, the magnitude of improvement in SPSA is governed by the size of the gain parameter, which is often chosen small initially to prevent unstable oscillating behavior of the algorithm (Spall 2003). On the other hand, although the low-order surrogate models in STRONG are easy to construct, they can only accurately approximate the response surface of a nonlinear function over very small regions. This may limit the size of the trust region to be explored, resulting in slow or incremental improvement over time. However, we remark that the performance of STRONG may be improved through finding better algorithm parameter values tailored to the test problems.

The figures also indicate that SPAS performs at least as well as PAS, with superior performance especially manifested on functions with multiple local minima. We see that both algorithms have similar performance on all three unimodal test functions $h_1$–$h_3$; however, SPAS finds better solutions than PAS does on multimodal functions, except for the $h_{10}$ case. This is as expected, since the surrogate model in SPAS implicitly uses previously sampled points to predict objective function values at unsampled locations, whereas such information is not exploited in PAS. Thus, if the surrogate model can adequately capture the general trend of the underlying response curve, then the true performance of the point predicted by the model could be significantly better than that of the current best sampled point. Consequently, by focusing the search around the

**Table 1.** Choices of $r_k$ and $\delta$ in SPAS and PAS

| | $r_k$ | $\delta$ |
|---|---|---|
| $h_1$ | $\frac{1}{(k+1)^{0.49/2}}$ | 1 |
| $h_2$ | $\frac{0.5}{(k+1)^{0.49/4}}$ | 1 |
| $h_3$ | $\frac{1}{(k+1)^{0.49/10}}$ | 1 |
| $h_4$ | $\frac{0.2}{(k+1)^{0.49/2}}$ | 1 |
| $h_5$ | $\frac{1}{(k+1)^{0.49/2}}$ | 1 |
| $h_6$ | $\frac{1}{(k+1)^{0.49/10}}$ | 1 |
| $h_7$ | $\frac{0.5}{(k+1)^{0.49/10}}$ | 1 |
| $h_8$ | $\frac{20}{(k+1)^{0.49/10}}$ | 1 |
| $h_9$ | $\frac{1}{(k+1)^{0.49/10}}$ | 1 |
| $h_{10}$ | $\frac{0.5}{(k+1)^{0.49/10}}$ | 1 |

**Table 2.** Choices of $\alpha_k$ and $c_k$ in SPSA

| | $\alpha_k$ | $c_k$ |
|---|---|---|
| $h_1$ | $\frac{1}{(k+600{,}000)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_2$ | $\frac{1}{(k+1{,}500)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_3$ | $\frac{1}{(k+1)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_4$ | $\frac{1}{(k+5{,}000)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_5$ | $\frac{1}{(k+1)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_6$ | $\frac{1}{(k+100)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_7$ | $\frac{1}{(k+1{,}000)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_8$ | $\frac{10}{(k+1{,}000)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_9$ | $\frac{1}{(k+150{,}000)}$ | $\frac{1}{(k+1)^{0.25}}$ |
| $h_{10}$ | $\frac{1}{(k+1)}$ | $\frac{1}{(k+1)^{0.25}}$ |

**Figure 3.** Averaged Performance of Comparison Algorithms on Test Functions $h_1$–$h_6$

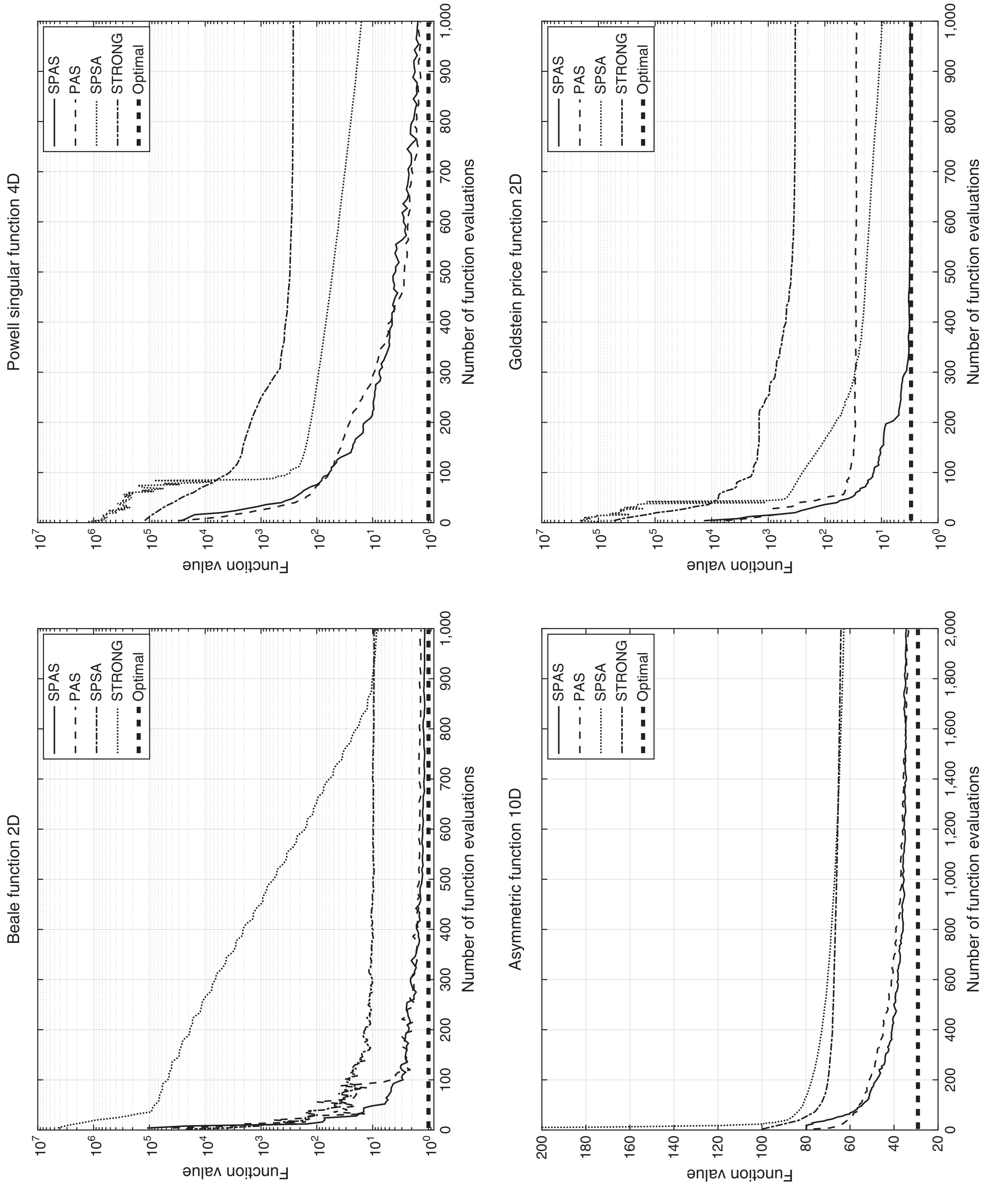


**Figure 3.** (Continued)

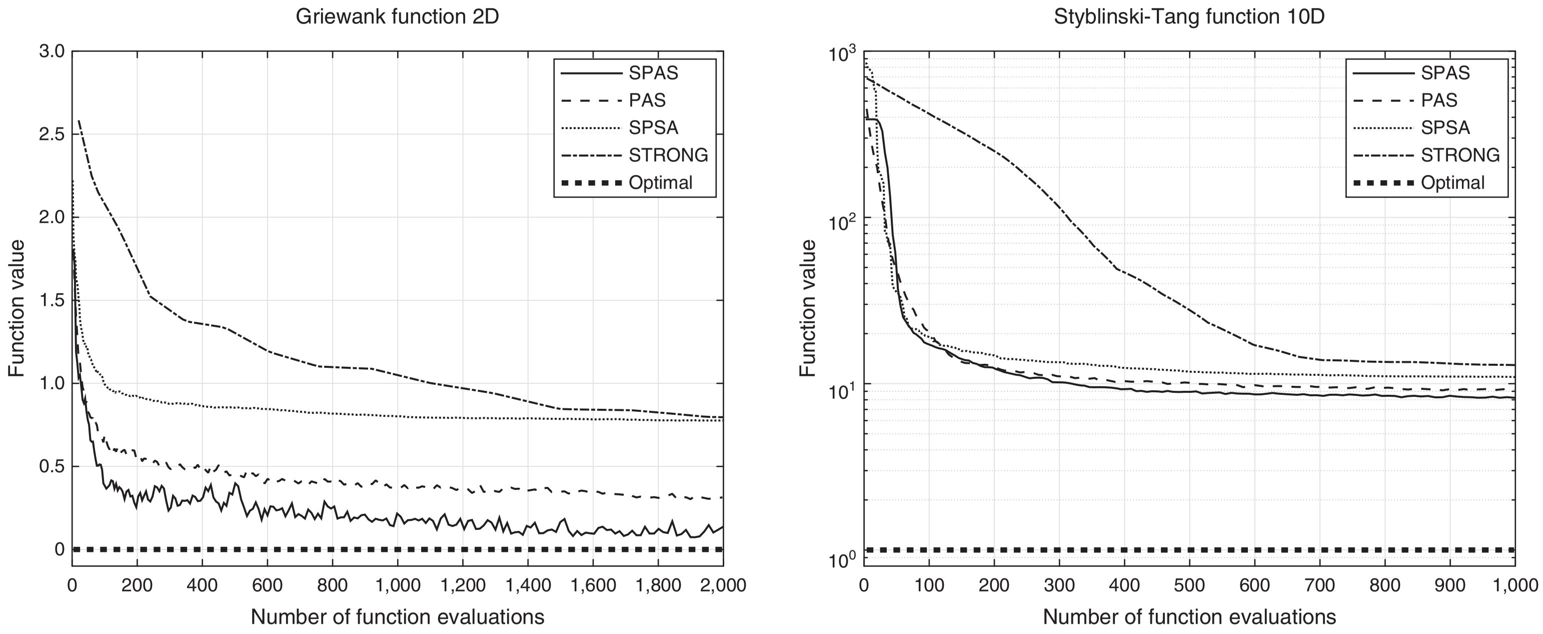
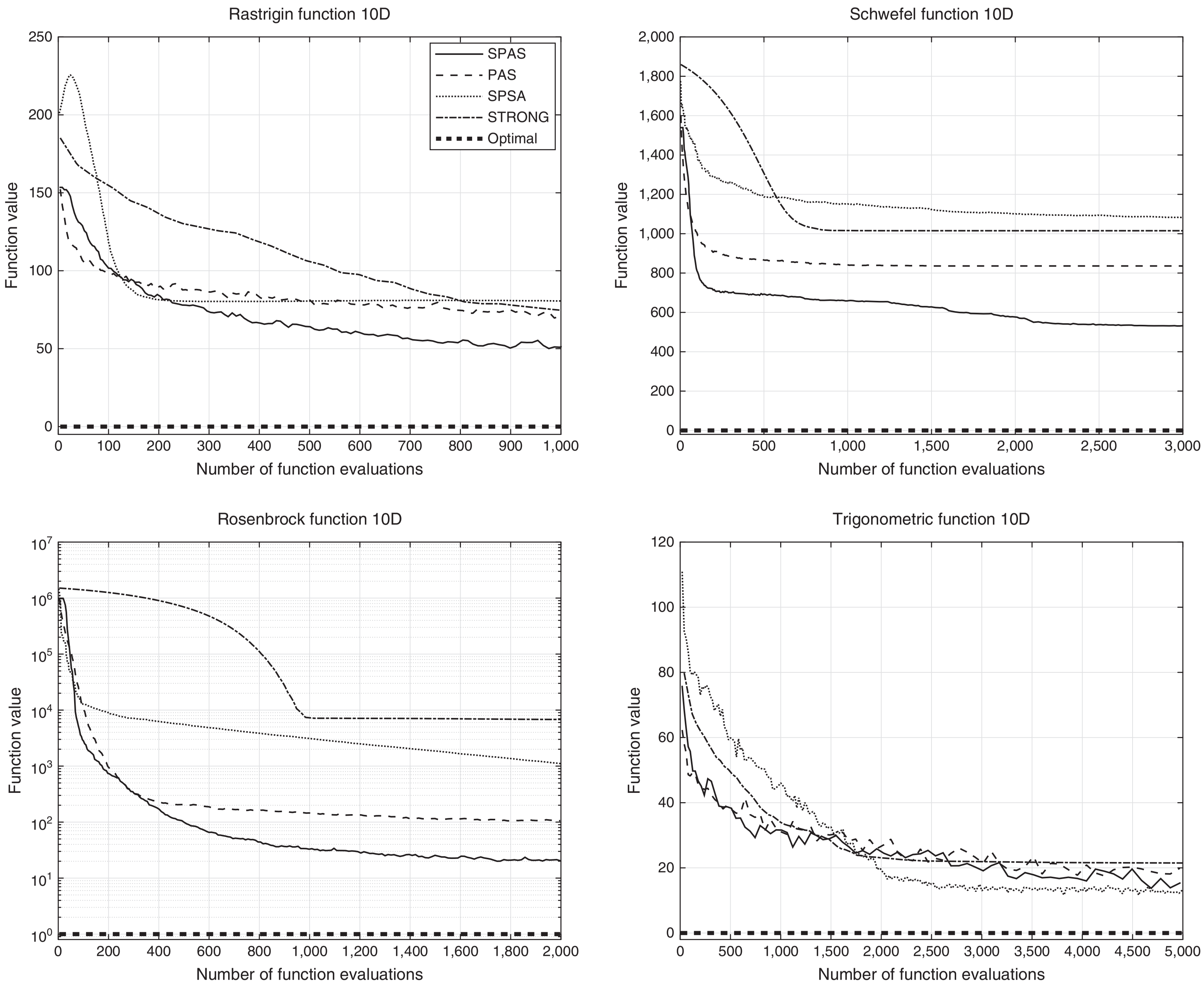


**Figure 4.** Averaged Performance of Comparison Algorithms on Test Functions $h_7$–$h_{10}$



best point predicted by the model, SPAS has the potential to identify better promising areas and prevents the search process from being trapped into inferior local minima in early iterations. Unlike other test functions, $H_{10}$ has a very strong curvature with its shape changing rapidly over very small regions of the solution space. This feature makes the true response curve of the function very hard to predict. An accurate approximation can only be obtained after a significant number of sampled points has been collected, in which case the minimizer of the model will essentially coincide with the best sampled solution, leading to nearly identical performance of both algorithms. As a result, when compared to PAS, we anticipate that the use of surrogate models in SPAS will yield the most computational benefits on multimodal objective functions with relatively smooth response surfaces.

## 4.2. An Inventory Control Example

We consider a discrete-time $(s,S)$ inventory control problem with independent and identically distributed exponentially distributed demands (see e.g., Fu and Healy 1992, Hu et al. 2008). The inventory level is reviewed at the beginning of each time period. When the inventory position (the on-hand inventory plus that on order) falls below the level $s$, an order is placed to increase the inventory position to $S$. We assume that orders are placed and received instantly (i.e., zero order lead times) and unsatisfied demands are fully backlogged.

Let $W_t$ and $\mathcal{D}_t$ be the inventory position and demand in period $t$. Denote by $p$ the per-period per-unit penalty cost for unmet demand, $h$ the per-period per unit inventory holding cost, $c$ the per-unit ordering cost, and $K$ the per-order set-up cost. The dynamic of the inventory position $W_t$ can then be described by the following formula: $W_{t+1}=S-\mathcal{D}_{t+1}$ if $W_t<s$, and $W_{t+1}=W_t-\mathcal{D}_{t+1}$ whenever $W_t\geq s$. The objective is to find the optimal threshold values, $s^*$ and $S^*$, in order to minimize the long-run average cost per period, that is,

$$(s^*,S^*)=\operatorname*{argmin}_{(s,S)\in\mathbb{X}}\left\{J(s,S)=\lim_{t\to\infty}\frac{1}{t}\sum_{i=1}^{t}\Big[I\{W_i<s\}\cdot(K+c(S-W_i))+hW_i^{+}+pW_i^{-}\Big]\right\},$$

where $\mathbb{X}=[0,1{,}000]\times[0,2{,}000]$, $w^+=\max(0,w)$, $w^-=\max(0,-w)$, and $I\{\cdot\}$ is the indicator function.

**Table 3.** Four Test Cases of the Inventory Problem

| Case | $E[\mathcal{D}_t]$ | $p$ | $K$ | $J(s^*,S^*)$ |
|---|---|---|---|---|
| 1 | 20 | 1 | 10 | 40.00 |
| 2 | 20 | 10 | 100 | 102.68 |
| 3 | 200 | 10 | 100 | 740.95 |
| 4 | 200 | 100 | 1,000 | 1,470.30 |

In the simulation experiments, we fix $h=c=1$ and consider four test cases by varying the mean demand $E[\mathcal{D}_t]$ and the values of $p$ and $K$; the optimal objective function values in all cases can be computed analytically (e.g., Fu and Healy 1992) and are given in Table 3. SPAS is implemented based on the following parameter setting: $\delta=1$, $r_k=25/(k+1)^{0.49/2}$, $N_k=\max\{\sqrt{k},4\}$, and the exponential demand distribution is truncated over the interval $[0,5E[\mathcal{D}_t]]$, which corresponds to a tail probability of $e^{-5}$. For SPSA, we set $\alpha_k=100/(k+1)$, $c_k=10/(k+1)^{0.25}$. In STRONG, we choose the initial trust region radius $\Delta=20$ and the threshold $\tilde{\Delta}=12$; the values of the rest of the parameters remain unchanged from Chang et al. (2013). The initial solutions in SPSA and STRONG are uniformly selected from the feasible region $\mathbb{X}$, and the value of $J(s,S)$ is estimated in all algorithms by simulating the $(s,S)$-policy for 250 periods with a warm-up length of 50 periods and then averaging the cost accumulated over the remaining 200 periods.

The simulation results, averaged over 30 independent runs for each algorithm, are reported in Table 4, where $N_{\text{rep}}$ indicates the total number of simulation replications (i.e., the number of $(s,S)$ pairs evaluated), and the entries represent the averaged function value $J$ at the final (best) sampled solutions. The results are similar to those obtained in Section 4.1, with SPAS providing either superior or comparable performance to PAS in all cases. Both SPAS and PAS significantly outperform SPSA and STRONG. In addition, note that the estimates obtained by SPAS have significantly smaller standard errors than those of PAS in three out of the four cases, which suggests that the use of the surrogate model could lead to more stable and robust algorithm performance.

In Table 5, we also recorded the computational run times (on a PC with a 2.6 GHz Intel Xeon CPU and 16 GB

**Table 4.** Performance of Different Algorithms on the Four Test Cases (Standard Errors in Parentheses)

| Case | $N_{\text{rep}}$ | SPAS | PAS | SPSA | Strong |
|---|---|---|---|---|---|
| 1 | 200 | 40.47 (0.16) | 98.12 (22.90) | 422.29 (36.09) | 636.50 (61.03) |
| 2 | 200 | 103.94 (0.24) | 147.47 (18.85) | 426.78 (36.14) | 585.47 (66.83) |
| 3 | 1,000 | 750.83 (2.12) | 756.51 (4.32) | 841.29 (10.25) | 941.59 (29.52) |
| 4 | 1,000 | 1,499.01 (5.84) | 1,498.68 (5.74) | 1,664.13 (63.44) | 2,476.70 (322.37) |

**Table 5.** Computational Run Times (in Seconds) of Different Algorithms (Standard Errors in Parentheses)

| Case | $N_{rep}$ | SPAS | PAS | SPSA | Strong |
|---|---|---|---|---|---|
| 1 | 200 | 6.24 (0.11) | 1.98 (0.03) | 0.84 (1.8e−3) | 0.99 (0.01) |
| 2 | 200 | 6.17 (0.09) | 1.89 (0.03) | 0.86 (3.7e−3) | 1.02 (0.03) |
| 3 | 1,000 | 48.47 (0.28) | 23.37 (0.19) | 4.13 (3.6e−3) | 5.16 (0.36) |
| 4 | 1,000 | 47.69 (0.36) | 24.64 (0.16) | 4.49 (1.5e−2) | 6.02 (0.48) |

memory) of different comparison algorithms. Although SPAS finds better solutions than the other algorithms do, it is clear from the table that it is also the slowest among the four algorithms. This is primarily because of the computational time required to construct and optimize the surrogate model, which is significantly longer than the time used in conducting simulation runs. Consequently, we expect that the advantage of SPAS will become evident when the cost of simulation is high and dominates the computational overhead incurred by the algorithm.

## 5. Conclusions and Future Research

In this paper, by integrating ideas from the shrinking ball method, surrogate model approximation, and promising region search, we have proposed a novel approach, called SPAS, for solving Lipschitz continuous simulation optimization problems. Under appropriate conditions, we have shown that the algorithm converges almost surely to the set of local optimal solutions. The performance of SPAS has been illustrated on a set of 10 benchmark testing problems and an inventory control example. Empirical results on these examples indicate that the algorithm is promising and may significantly outperform some existing methods that exploit gradient information.

Although our discussion of SPAS has been based on a specific promising area construction scheme, the algorithm offers the flexibility of using other suitable neighborhood structures or construction procedures. In addition, there are a wide variety of surrogate models that can be employed, ranging from simple polynomial regressions to sophisticated neural network models. For example, instead of the RBF method considered here, another potentially promising approach is Kriging, which is also an interpolation-based approximation technique. From this perspective, SPAS can be viewed as a general framework for Lipschitz simulation optimization. Thus, an important line of research is to investigate the use of other procedures for constructing promising areas and surrogate models within the framework and to compare the computational efficiency in terms of algorithm performance between different strategies.

The convergence of SPAS requires the sample size $N_k$ to increase polynomially with the number of algorithm iterations. This is in contrast to the shrinking ball methods of Andradóttir and Prudius (2010) and the recent development in Kiatsupaibul et al. (2018), where only a single candidate solution is sampled at each iteration. The cause of this difference is that in latter approaches, the sampling is performed from the entire fixed feasible region; whereas in SPAS, the candidate points are generated from random subsets so that a point sampled at an earlier time cannot be guaranteed to lie within the promising area constructed at a later iteration. However, our intuition is that while the underlying promising areas vary with $k$, their intersection may contain an invariant set with positive volume as $k$ becomes large. Thus, although currently we were only able to prove the convergence of SPAS when the sample size increases, we conjecture that the same convergence result could be warranted in the case when $N_k$ is held constant, but the theoretical analysis might require a totally different approach. This is clearly a future research issue that merits investigation.

The shrinking ball method improves the algorithm efficiency by allowing search (exploring better solutions) and evaluation (obtaining better estimates at current solutions) to be conducted at the same time. However, the use of the method in SPAS may not have reached its full potential yet. In particular, because of the correlation issue mentioned in Section 2.1, the contribution of historically sampled points is not fully utilized in performance estimation and needs to be discounted to eliminate the induced bias effect. Recently, this correlation bias issue has also been noted and successfully addressed in Kiatsupaibul et al. (2018) under a different adaptive search framework by using a martingale-based approach. So it will be interesting to study whether the martingale approach proposed there can also be usefully applied to our setting. However, since the sampling regions in SPAS are random, the theoretical analysis is likely to require more technical manipulations.

## Acknowledgments

The authors thank the department editor, associate editor, and two anonymous reviewers for their helpful comments and suggestions, which have led to an improved paper. The authors also thank Dr. Hong Wan at Purdue University for providing the MATLAB code of the STRONG algorithm.

## References

Alrefaei MH, Andradóttir S (1999) A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Sci.* 45:748–764.

Andradóttir S (2014) A review of random search methods. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 277–292.

Andradóttir S, Prudius AA (2010) Adaptive random search for continuous simulation optimization. *Naval Res. Logist.* (*NRL*) 57:583–604.

Baumert S, Smith RL (2002) Pure random search for noisy objective functions. University of Michigan Technical Report, Ann Arbor.

Bishop CM (1995) *Neural Networks for Pattern Recognition* (Oxford University Press, New York).

Chang KH, Hong LJ, Wan H (2013) Stochastic trust-region response-surface method (strong): A new response-surface framework for simulation optimization. *INFORMS J. Comput.* 25(2):230–243.

A15 Chen R, Menickelly M, Scheinberg K (2017) Stochastic optimization using a trust-region method and random models. *Math. Programming*, https://doi.org/10.1007/s10107-017-1141-8.

Deng G, Ferris MC (2009) Variable-number sample-path optimization. *Math. Programming* 117:81–109.

Fan Q, Hu J (2016) Simulation optimization via promising region search and surrogate model approximation. Roeder TMK, Frazier PI, Szechtman R, Zhou E, eds. *Proc. 2016 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 649–658.

Fu MC, Healy KJ (1992) Simulation optimization of $(s, S)$ inventory systems. Swain JJ, Goldsman D, Crain RC, Wilson JR, eds. *Proc. 1992 Winter Simulation Conf.* (IEEE Press, Piscataway, NJ), 506–514.

Gutmann HM (2001) A radial basis function method for global optimization. *J. Global Optim.* 19:201–227.

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 58:13–30.

Hong LJ, Nelson BL (2006) Discrete optimization via simulation using compass. *Oper. Res.* 54:115–129.

A16 Hong LJ, Nelson BL (2007) A framework for locally convergent random-search algorithms for discrete optimization via simulation. *ACM Trans. Model. Comput. Simul.* 17, http://dx.doi.org/10.1145/1276927.1276932.

Hu J (2015) Model-based stochastic search methods. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 319–340.

Hu J, Fu MC, Marcus SI (2008) A model reference adaptive search method for stochastic global optimization. *Commun. Inform. Systems* 8:245–276.

Huang D, Allen T, Notz W, Zeng N (2006) Global optimization of stochastic black-box systems via sequential kriging metamodels. *J. Global Optim.* 34:441–466.

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. *J. Global Optim.* 13:455–492.

A17 Kiatsupaibul S, Smith RL, Zabinsky ZB (2018) Single observation adaptive search for continuous simulation optimization. *Oper. Res.* Submitted.

Kiefer J, Wolfowitz J (1952) Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.* 23:462–466.

Kim S, Pasupathy R, Henderson SG (2015) A guide to sample average approximation. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 207–243.

Kleijnen JPC (2015) Model-based stochastic search methods. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 81–104.

Kleywegt A, Shapiro A, Homem-De-MelloHu T (2001) The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* 12:479–502.

Larson J, Billups SC (2016) Stochastic derivative-free optimization using a trust region framework. *Comput. Optim. Appl.* 64:619–645.

Müller J (2017) SOCEMO: Surrogate optimization of computationally expensive multiobjective problems. *INFORMS J. Comput.* 29(4):581–596.

Nakayama H, Arakawa M, Sasaki R (2002) Simulation-based optimization using computational intelligence. *Optim. Engrg.* 3:201–214.

Regis R, Shoemaker C (2007) Improved strategies for radial basis function methods for global optimization. *J. Global Optim.* 37:113–135.

Robbins H, Monro S (1951) A stochastic approximation method. *Ann. Math. Statist.* 22:400–407.

Robinson SM (1996) Analysis of sample-path optimization. *Math. Oper. Res.* 21(3):513–528.

Rubinstein RY, Kroese DP (2004) *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-Carlo Simulation* (*Information Science and Statistics*) (Springer-Verlag, Secaucus, NJ).

Shi L, Ólafsson S (2000) Nested partitions method for stochastic optimization. *Methodology Comput. Appl. Probab.* 2:271–291.

Shiryaev AN (1996) *Probability*, Second Ed. (Springer-Verlag, New York).

Smith RL (1984) Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Oper. Res.* 32(6):1296–1308.

Sóbester A, Leary S, Keane A (2005) On the design of optimization strategies based on global response surface approximation model. *J. Global Optim.* 33:31–59.

Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Automatic Control* 37:332–341.

Spall JC (2003) *Introduction to Stochastic Search and Optimization* (John Wiley & Sons, Springer, Hoboken, NJ).

Xu J, Nelson BL, Hong JL (2010) Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *ACM Trans. Model. Comput. Simul.* 20:3:1–3:29.

Yan D, Mukai H (1992) Stochastic discrete optimization. *SIAM J. Control Optim.* 30:594–612.

Zabinsky ZB (2015) Stochastic adaptive search methods: Theory and implementation. Fu MC, ed. *Handbook of Simulation Optimization* (Springer, New York), 293–318.

## Author Queries

A1 Au: Please confirm names, affiliations, and email addresses are okay as set. Provide author(s) ORCID number(s) if applicable.

A2 Au: Please provide institutional emails.

A3 Au: Please confirm keywords.

A4 Au: Please confirm heading levels are correct.

A5 Au: Revised to "also developed a shrinking ball-based framework." Please check.

A6 Au: Please check that all figures and tables are set correctly.

A7 Au: Please verify that all equations are set correctly.

A8 Au: Because the definition for SPAS already appears at the first occurrence of the abbreviation (as per style), they do not need to appear together in subsequent occurrences. Please remove either the definition or the abbreviation here.

A9 Au: Revised to section 2.2.

A10 Au: Please check revisions to assumption headings.

A11 Au: Per style, sentences should not begin with variables. Please revise.

A12 Au: Because it only occurs once, i.i.d. written out instead of abbreviated.

A13 Au: Should this be SPAS?

A14 Au: Please include issue numbers with all journal references.

A15 Au: Please update.

A16 Au: Please update.

A17 Au: Please update.