

## **ADAPTIVE PROBABILISTIC BRANCH AND BOUND WITH CONFIDENCE INTERVALS FOR LEVEL SET APPROXIMATION**

Hao Huang

Industrial and Systems Engineering  
University of Washington  
Seattle, WA 98195, USA

Zelda B. Zabinsky

Industrial and Systems Engineering  
University of Washington  
Seattle, WA 98195, USA

### **ABSTRACT**

We present a simulation optimization algorithm called probabilistic branch and bound with confidence intervals (PBnB with CI), which is designed to approximate a level set of solutions for a user-defined quantile. PBnB with CI is developed for both deterministic and noisy problems with mixed continuous and discrete variables. The quality of the results is statistically analyzed with order statistic techniques and confidence intervals are derived. Also, the number of samples and replications are designed to achieve a certain quality of solutions. When the algorithm terminates, it provides an estimation of the desired quantile with confidence intervals, and an approximation level set, including a statistically guaranteed set in the true desirable level set, a statistically pruned set, and a set which is not statistically specified. We also present numerical experiments with benchmark functions to visualize the algorithm and its capability.

### **1 INTRODUCTION**

There are three common problems or characteristics encountered in optimizing extremely complex problems: lack of structure; presence of large uncertainties; and enormously large search spaces (Ho, Cassandras, Chen, and Dai 2000). These characteristics also result in difficult or costly performance evaluation. Stochastic global optimization becomes a powerful approach for these complex problems with estimations of the performance with realizations from black-box simulations or direct observations. Fu (2001) discussed the role of stochastic global optimization in solving these complex systems and problems with system noise. Many global optimization search algorithms are introduced such as gradient-based search algorithms, response surface methodology and random search algorithms. Four key issues are pointed out: stochastic comparisons, family of solutions versus a single point, continuous versus discrete, and convergence and statistical validity (Fu 2001). In ordinal optimization, a family of solutions is considered important, which is also referred to as goal softening (Ho, Cassandras, Chen, and Dai 2000; Ho, Zhao, and Jia 2007). In our algorithm, we also aim to provide a family of solutions that achieve a desirable level set. We also provide a statistically validated quality measure of the results.

Hu, Fu, and Marcus (2007) classified simulation optimization methodologies into two categories. Instance-based algorithms sample new solutions directly from previous solutions, whereas, model-based algorithms sample new solutions based on a probabilistic model, while its parameter is updated iteratively. In the instance-based category, instead of moving from one solution to its neighborhood, algorithms with a partitioning approach systematically partition the solution space in order to narrow down the candidate solutions with a comprehensive understanding of the solution space. The nested partition (NP) family (Shi and Ólafsson 2000, Ólafsson 2004) iteratively partitions the solution space and identifies the most promising subregion. All non-promising subregions are collected in a super region so that the promising subregion may backtrack to its super region if a better solution appears out of the promising subregion. Also, NP has been adapted to mixed-integer/continuous problems (Shi and Ólafsson 2009). With a slightly

different partitioning framework, the widely used algorithm, branch and bound, is introduced into stochastic optimization problems, called stochastic branch and bound (SB&B) (Norkin, Pflug, and Ruszczyński 1998). Xu and Nelson (2013) proposed an algorithm, called empirical stochastic branch-and bound (ESB&B), for discrete decision variable optimization through simulation, rooted with NP and SB&B’s concepts. ESB&B focuses on finding a better solution within restricted computational resources. ESB&B uses the best observed solution to find the promising subregion, which is further partitioned in next iteration. Also, ESB&B calculates an upper confidence bound for each non-promising subregion to allocate further samples in these subregions. Xu and Nelson (2013) provide a statistical convergence of finding the best solution and numerically demonstrate the computational savings on test problems. These partition-based algorithms are mostly focused on finding the best solution, but our proposed algorithm aims to find a family of solutions.

The presented algorithm is an extension of the algorithm introduced in the Winter Simulation Conference 2011, probabilistic branch and bound (PBnB) (Zabinsky, Wang, Prasetio, Ghate, and Yen 2011). PBnB uses order statistics to determine the quality of each subregion for pruning and branching decisions. The algorithm dynamically determines the required number of samples and replications in each current subregions. Unlike other partition-based algorithms, PBnB stops not only further partitioning but also further sampling to increase the efficiency while maintaining a probability bound on the chance of missing desirable region. Furthermore, PBnB concentrates on goal softening, so that the remaining subregions can approximate the target level set of the desirable solutions. The primary difference between Zabinsky, Wang, Prasetio, Ghate, and Yen (2011) and the extension of PBnB presented here is the interval estimation of quantile. So we refer to the present algorithm as PBnB with *confidence intervals* (CI). The CIs are used to identify the target level set directly instead of only assuring the fact of containing desired solution in the current subregions. PBnB with CI includes two types of subregions, pruned and maintained regions, which contain the identified undesirable and desirable solutions, respectively. Each pruning and maintaining decision is processed only while it is statistically valid, which is shown in the performance analysis section. Furthermore, the overall quality of the approximated level set is also established with each pruning and maintaining move. Similar to the previous PBnB, the closeness parameter  $0 < \delta < 1$  and error rate  $0 < \alpha < 1$  are used for determining the target solution threshold in terms of quantile of the objective function’s range distribution and making probability bounds for the quality of approximation (Zabinsky, Wang, Prasetio, Ghate, and Yen 2011). In addition, a new parameter  $\varepsilon$  is introduced as the “tolerance” of the volume of desirable solution within pruned regions and undesirable solution within maintained regions. In this paper, we implement a two-stage procedure for the replications of each sampled point in the current region for the probabilistic correctness of ordering and comparisons for pruning and maintaining.

In Section 2, we describe PBnB with CI in detail. In Section 3, we derive bounds on the algorithm’s performance. Several numerical results are presented in Section 4, and conclusions in Section 5.

## 2 Probabilistic Branch and Bound with Confidence Intervals

The primary goal of the algorithm is to identify the exact set of best  $\delta$  percent solutions, where  $0 < \delta < 1$ . PBnB involves a statistical quantile estimation to decide a “maintaining” set as a good approximation to the true target level set. This algorithm is designed to solve optimization problems in the following form:

$$(\mathcal{P}) \min_{x \in S} f(x) \tag{1}$$

where  $f(x) = E_{\Xi}[g(x, \xi_x)]$  is the expected value of a noisy function  $g(x, \xi_x)$ , and  $\xi_x$  is the noise term. Therefore, the objective function  $f(x)$  can only be estimated. In this case, the distribution of the noise term  $\xi_x$  may be different for each  $x$ , but it is assumed to be normally distributed since the sample mean is a typical way to obtain simulation output, and the Central Limit Theorem could be applied (Chen and He 2005). Let  $y(\delta, S)$  be the  $\delta$  quantile for the objective function, which is defined in terms of probabilities as

$$P(f(X) \leq y(\delta, S)) = \frac{v(\{x : f(x) \leq y(\delta, S)\})}{v(S)} \geq \delta \tag{2}$$

$$P(f(X) < y(\delta, S)) = \frac{v(\{x : f(x) < y(\delta, S)\})}{v(S)} \leq \delta \quad (3)$$

where  $X$  is uniformly distributed in  $S$  and  $v(S)$  denotes the  $n$ -dimensional volume of set  $S$ . We recognize that  $y(\delta, S)$  can also be expressed as

$$y(\delta, S) = \arg \min_{y \in \{f(x) : x \in S\}} \{P(f(X) \leq y | X \in S) \geq \delta\}, \text{ for } 0 < \delta < 1, \quad (4)$$

and we let  $L(\delta, S)$  be our desired set of best  $\delta$  solutions, where

$$L(\delta, S) = \{x \in S : f(x) \leq y(\delta, S)\}, \text{ for } 0 < \delta < 1. \quad (5)$$

There are several parameters that are input to the algorithm; the first is  $\delta$ ,  $0 < \delta < 1$ , the desired level of quantile. The user-defined parameter  $\alpha$ ,  $0 < \alpha < 1$ , controls the quality of the results through probability bounds involving  $1 - \alpha$ . Another user-defined parameter  $\varepsilon$  identifies a volume measure to quantify the tolerance of the set approximation to  $L(\delta, S)$ . Usually, the  $\varepsilon$  value can be determined by a certain ratio of the volume of the entire solution set,  $S$ , e.g.,  $\varepsilon = 0.1v(S)$ . In addition, the branching scheme in PBnB with CI is specified by parameter  $B$ , and each current subregion is partitioned into  $B$  equally sized subregions. A subregion is considered unbranchable, when the  $n$ -dimensional volume of a subregion is less than a user-specified positive value. For example, in the discrete case, a subregion may be unbranchable when it contains a singleton. For the continuous case, a subregion may be unbranchable when the longest Euclidian distance within the subregion is less than a pre-determined positive value.

For notational purposes,  $\Sigma_k$  keeps a list of all current subregions at the beginning of iteration  $k$ , and  $\tilde{\Sigma}_k$  is the union of the current subregions. Also,  $\tilde{\Sigma}_k^P$  represents the union of all pruned subregions at the beginning of iteration  $k$ . In addition to pruned subregions, we “maintain” a subregion when it is likely the entire subregion lies in the target level set  $L(\delta, S)$ . The set  $\tilde{\Sigma}_k^M$  denotes the union of all maintained subregions at the beginning of iteration  $k$ . For replication size, we define  $R^o$  as a minimal number of replications,  $R^o \geq 1$ , for new sample points.

PBnB with CI provides two types of results. First, it provides an interval estimate,  $[\hat{f}(Z_{(r)}), \hat{f}(Z_{(s)})]$ , of the  $\delta$  quantile,  $y(\delta, S)$ . Second, it provides the pruned region  $\tilde{\Sigma}_{k+1}^P$  and the maintained region  $\tilde{\Sigma}_{k+1}^M$ , that are used to approximate the target level set,  $L(\delta, S)$ , i.e.,  $\tilde{\Sigma}_{k+1}^M \subset L(\delta, S) \subset S \setminus \tilde{\Sigma}_{k+1}^P$  with high probability and some tolerable loss  $\varepsilon$ . In Section 3, we derive performance results to statistically quantify the quality of the results. Next, we present the PBnB with CI algorithm.

### Probabilistic Branch and Bound with Confidence Intervals (PBnB with CI)

**Step 0.** Initialize: Set user-defined parameters  $\alpha, \delta, \varepsilon, R^o$ , and  $B$ . Partition  $S$  into  $B$  subregions,  $\sigma_1, \dots, \sigma_B$ , and set  $\Sigma_1 = \{\sigma_1, \dots, \sigma_B\}$ ,  $\tilde{\Sigma}_1 = S$ ,  $\tilde{\Sigma}_1^P = \phi$ ,  $\tilde{\Sigma}_1^M = \phi$ ,  $\delta_1 = \delta$ ,  $\alpha_1 = \frac{\alpha}{B}$ ,  $\varepsilon_1 = \frac{\varepsilon}{B}$ ,  $R_0 = R^o$ , and  $k = 1$ .

**Step 1.** Sample: For each subregion  $\sigma_i \in \Sigma_k$ ,  $i = 1, \dots, |\Sigma_k|$ , uniformly sample additional points such that the total number of points in  $\sigma_i$  is

$$N_k^i = \left\lceil \frac{\ln \alpha_k}{\ln \left(1 - \frac{\varepsilon_k}{v(\sigma_i)}\right)} \right\rceil. \quad (6)$$

Denote the sample points in  $\sigma_i$  by  $x_{i,j}$ , for  $j = 1, \dots, N_k^i$  and  $i = 1, \dots, |\Sigma_k|$ . Let  $N_k = \sum_{i=1}^{|\Sigma_k|} N_k^i$  be the total number of sample points in the current region. For each  $x_{i,j} \in \sigma_i$ ,  $j = 1, \dots, N_k^i$  and  $i = 1, \dots, |\Sigma_k|$ , perform  $R_{k-1}$  replications of  $g(x, \xi_x^r)$ , and evaluate the sample mean and sample variance,

$$\hat{f}(x_{i,j}) = \frac{\sum_{r=1}^{R_{k-1}} g(x_{i,j}, \xi_x^r)}{R_{k-1}} \text{ and } S_{\hat{f}}^2(x_{i,j}) = \frac{1}{(R_{k-1} - 1)} \sum_{r=1}^{R_{k-1}} (g(x_{i,j}, \xi_x^r) - \hat{f}(x_{i,j}))^2. \quad (7)$$

**Step 2.** Order Sample Points by Function Value: Order all sampled points  $x_{i,j} \in \tilde{\Sigma}_k$ , for  $j = 1, \dots, N_k^i$  and  $i = 1, \dots, |\Sigma_k|$  by their estimated function value so that

$$\hat{f}(z_{(1)}) \leq \hat{f}(z_{(2)}) \leq \dots \leq \hat{f}(z_{(N_k)}), \text{ where } z_{(q)} = \underset{x_{i,j} \in \tilde{\Sigma}_k \setminus \bigcup_{l=1}^{q-1} z_{(l)}}{\operatorname{arg\,min}} \hat{f}(x_{i,j}).$$

Let  $d_i = \hat{f}(z_{(i+1)}) - \hat{f}(z_{(i)})$ , where  $i = 1, \dots, N_k - 1$ . Determine  $d^* = \min_{i=1, \dots, N_k-1} d_i$  and  $S^{*2} = \max_{i=1, \dots, N_k} S_{\hat{f}}^2(z_{(i)})$ . Calculate the updated replication number  $R_k = \max \left\{ R_{k-1}, \left( \frac{z_{\alpha_k/2} S^*}{d^*/2} \right)^2 \right\}$ , where  $z_{\alpha_k/2}$  is the  $1 - \alpha_k/2$  quantile of the standard normal distribution. Perform  $R_k - R_{k-1}$  more replications for each sample point. Re-estimate the performance of each sample point with  $R_k$  replications by  $\hat{f}(x_{i,j}) = \frac{\sum_{r=1}^{R_k} g(x_{i,j}, \xi_{x_{i,j}}^r)}{R_k}$ . Within each subregion  $\sigma_i \in \Sigma_k$ , rank all the sample points  $x_{i,j}$  as  $x_{i,(j)}$  representing the  $j$ th best point in subregion, according to the estimated function value, and also update the entire order of all current samples with updated replications, so that

$$\hat{f}(x_{i,(1)}) \leq \hat{f}(x_{i,(2)}) \leq \dots \leq \hat{f}(x_{i,(N_k^i)}), \text{ and } \hat{f}(z_{(1)}) \leq \hat{f}(z_{(2)}) \leq \dots \leq \hat{f}(z_{(N_k)}).$$

**Step 3.** Build Interval Estimates: Determine  $\hat{f}(z_{(r)})$  and  $\hat{f}(z_{(s)})$ , where  $r$  and  $s$  are selected by

$$\max r : \sum_{i=0}^{r-1} \binom{N_k}{i} (\delta_k)^i (1 - \delta_k)^{N_k-i} \leq \frac{\alpha_k}{2} \text{ and } \min s : \sum_{i=0}^{s-1} \binom{N_k}{i} (\delta_k)^i (1 - \delta_k)^{N_k-i} \geq 1 - \frac{\alpha_k}{2}.$$

**Step 4.** Update: For each  $i$ ,  $i = 1, \dots, |\Sigma_k|$ , update the pruning indicator functions  $P_i$  and update the maintaining indicator functions  $M_i$ ,

$$P_i = \begin{cases} 1, & \text{if } \hat{f}(x_{i,(1)}) > \hat{f}(z_{(s)}) \\ 0, & \text{otherwise} \end{cases} \text{ and } M_i = \begin{cases} 1, & \text{if } \hat{f}(x_{i,(N_k^i)}) < \hat{f}(z_{(r)}) \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Update the pruned set  $\tilde{\Sigma}_{k+1}^P$  and the maintained set  $\tilde{\Sigma}_{k+1}^M$

$$\tilde{\Sigma}_{k+1}^P = \tilde{\Sigma}_k^P \bigcup_{i:P_i=1} \sigma_i \text{ and } \tilde{\Sigma}_{k+1}^M = \tilde{\Sigma}_k^M \bigcup_{i:M_i=1} \sigma_i,$$

and determine  $\delta_{k+1}$  by

$$\delta_{k+1} = \frac{\delta_k v(\tilde{\Sigma}_k) - \sum_{i:M_i=1} v(\sigma_i)}{v(\tilde{\Sigma}_k) - \sum_{i:P_i=1} v(\sigma_i) - \sum_{i:M_i=1} v(\sigma_i)}. \quad (9)$$

Also, set  $\alpha_{k+1} = \frac{\alpha_k}{B}$ ,  $\epsilon_{k+1} = \frac{\epsilon_k}{B}$ . If  $P_i = 0$  and  $M_i = 0$ , and if  $\sigma_i$  is branchable, then partition  $\sigma_i$  to  $\bar{\sigma}_i^1, \dots, \bar{\sigma}_i^B$  and update the current set of subregions

$$\Sigma_{k+1} = \{ \bar{\sigma}_i^j : P_i = 0 \text{ or } M_i = 0, j = 1, \dots, B \} \text{ and } \tilde{\Sigma}_{k+1} = \bigcup_{i:P_i=0 \vee M_i=0} \left( \bigcup_{j=1}^B \bar{\sigma}_i^j \right).$$

**Step 5.** Stopping Condition: If all subregions  $\sigma_i \in \Sigma_k$  are not branchable, terminate the algorithm. Otherwise,  $k = k + 1$  and return to **Step 1**.

In Step 1 and Step 2, PBnB with CI samples evenly in the current subregions and orders the samples with their estimated function values. The purpose for evenly sampling is to build quantile confidence

intervals with order statistics. In order to rank the samples correctly under noise, a two-stage procedure is applied that is based on a method from Bechhofer, Dunnett, and Sobel (1954), Gibbons, Olkin, and Sobel (1977). The two-stage procedure to determine the replications are separated in Step 1 and Step 2. The first-stage replications  $R_{k-1}$  are evaluated in Step 1. The original ranking procedure assumes that there is a common variance  $\sigma^2$  but we do not. Step 2 uses the first-stage replications to estimate these differences,  $d_i$  for  $i = 1, \dots, N_k - 1$ . We use the maximum sample variance to update the replication number for the second-stage. Consequently, additional replications may be necessary. The interval estimates are built in Step 3. The estimates  $\hat{f}(z_r)$  and  $\hat{f}(z_s)$  are used to estimate the quantile and the target level set.

In Step 4, the pruned and maintained sets are determined comparing best and worst function values in each subregion with the interval estimates. The purpose of the  $\delta$  update is to relocate the quantile in the current region such that  $y(\delta_k, S \setminus \{\tilde{\Sigma}_k^P \cup \tilde{\Sigma}_k^M\}) = y(\delta, S)$ , under certain assumptions stated in Theorem 1. Since the pruned and maintained regions are not further sampled, the consequence is a non-uniform sampling over the entire sample space  $S$ . Therefore, we need to relocate the quantile  $\delta_k$  to account for this.

PBnB with CI follows a straight-forward stopping condition as in (Zabinsky, Wang, Prasetyo, Ghate, and Yen 2011). This stopping condition assures the algorithm will terminate in a finite number of iterations. Users may wish to terminate earlier, such as when the interval size  $\hat{f}(z_{(s)}) - \hat{f}(z_{(r)})$  is small enough.

### 3 Performance Analysis

In this section, we analyze the performance of PBnB and derive confidence intervals and probability bounds to indicate the quality of our quantile estimation  $[\hat{f}(Z_{(r)}), \hat{f}(Z_{(s)})]$ , the pruned set  $\tilde{\Sigma}^P$  and the maintained set  $\tilde{\Sigma}^M$ . First, we assume that the objective function is deterministic, without the influence of noise. We derive the relocation of quantile in Theorem 1 and use it to derive a confidence interval on  $y(\delta, S)$  in Theorem 2. Second, the quality of pruned and maintained region are discussed with no noise. In Theorem 3, we consider a single pruned subregion's quality, and Theorem 4 derives the probability bounds for the entire pruned region at iteration  $k$ . Similarly, a single maintained subregion's quality is derived in Theorem 5, and Theorem 6 derives the probability bounds for entire maintained region at iteration  $k$ . Third, the impact of the noisy function is discussed in Theorem 8. Corollaries 9, 10, and 11 propose noisy versions of Theorems 2, 4, and 6, respectively.

**Theorem 1** For any iteration  $k \geq 1$ , suppose all previous pruning is correct, that is,  $f(x) > y(\delta, S)$  for all  $x \in \tilde{\Sigma}_k^P$ , and all previous maintaining is correct, that is,  $f(x) \leq y(\delta, S)$  for all  $x \in \tilde{\Sigma}_k^M$ . Then, the  $\delta_k$  updated according to (9) can be used to determine the original quantile  $y(\delta, S)$ , that is

$$y(\delta, S) = y(\delta_k, \tilde{\Sigma}_k). \tag{10}$$

*Proof.* We consider the iterative effect on  $\delta_k$  as subregions are pruned, or maintained, according to (8). We use the superscript  $k$  to denote the subregions pruned  $\{\sigma_i^k : P_i = 1\}$  or maintained  $\{\sigma_i^k : M_i = 1\}$  on iteration  $k$ . By (9),

$$\delta_k = \frac{\delta_{k-1}v(\tilde{\Sigma}_{k-1}) - \sum_{i:M_i=1} v(\sigma_i^{k-1})}{v(\tilde{\Sigma}_{k-1}) - \sum_{i:P_i=1} v(\sigma_i^{k-1}) - \sum_{i:M_i^{k-1}=1} v(\sigma_i^{k-1})} = \frac{\delta_{k-1}v(\tilde{\Sigma}_{k-1}) - \sum_{i:M_i=1} v(\sigma_i^{k-1})}{v(\tilde{\Sigma}_k)}$$

and by applying  $\delta_k = \frac{\delta_{k-1}v(\tilde{\Sigma}_{k-1}) - \sum_{i:M_i=1} v(\sigma_i^{k-1})}{v(\tilde{\Sigma}_k)}$  repeatedly

$$= \frac{\delta_1 v(\tilde{\Sigma}_1) - \sum_{l=1}^{k-1} \sum_{i:M_i=1} v(\sigma_i^l)}{v(\tilde{\Sigma}_k)} = \frac{\delta v(S) - \sum_{l=1}^{k-1} \sum_{i:M_i=1} v(\sigma_i^l)}{v(\tilde{\Sigma}_k)}$$

using the initial setting of  $\delta_1$  and  $\tilde{\Sigma}_1$ , and  $\sum_{l=1}^{k-1} \sum_{i:M_i=1} v(\sigma_i^l) = v(\tilde{\Sigma}_k^M)$  since it denotes the volume of all maintained subregions at the end of the  $k - 1$  iteration, therefore,

$$= \frac{\delta v(S) - v(\tilde{\Sigma}_k^M)}{v(\tilde{\Sigma}_k)}. \tag{11}$$

From (4), we can express  $y(\delta, S)$  as

$$y(\delta, S) = \arg \min_{y \in \{f(x): x \in S\}} \{P(f(X) \leq y | X \in S) \geq \delta\} = \arg \min_{y \in \{f(x): x \in S\}} \left\{ \frac{v(\{x \in S : f(x) \leq y\})}{v(S)} \geq \delta \right\}$$

and subtracting  $\frac{v(\tilde{\Sigma}_k^M)}{v(S)}$  from both sides and then multiplying  $\frac{v(S)}{v(\tilde{\Sigma}_k)}$  on both sides,

$$\begin{aligned} &= \arg \min_{y \in \{f(x): x \in S\}} \left\{ \frac{v(\{x \in S : f(x) \leq y\}) - v(\tilde{\Sigma}_k^M)}{v(\tilde{\Sigma}_k)} \geq \frac{\delta v(S) - v(\tilde{\Sigma}_k^M)}{v(\tilde{\Sigma}_k)} \right\} \\ &= \arg \min_{y \in \{f(x): x \in S\}} \left\{ \frac{v(\{x \in S : f(x) \leq y\}) - v(\tilde{\Sigma}_k^M)}{v(\tilde{\Sigma}_k)} \geq \delta_k \right\} \end{aligned}$$

by (11), and since  $f(x) > y$  for all  $x \in \tilde{\Sigma}_k^P$  and  $f(x) \leq y$  for all  $x \in \tilde{\Sigma}_k^M$

$$\begin{aligned} &= \arg \min_{y \in \{f(x): x \in S\}} \left\{ \frac{v(\{x \in S \setminus \{\tilde{\Sigma}_k^P \cup \tilde{\Sigma}_k^M\} : f(x) \leq y\})}{v(\tilde{\Sigma}_k)} \geq \delta_k \right\} \\ &= \arg \min_{y \in \{f(x): x \in S\}} \left\{ \frac{v(\{x \in \tilde{\Sigma}_k : f(x) \leq y\})}{v(\tilde{\Sigma}_k)} \geq \delta_k \right\} = \arg \min_{y \in \{f(x): x \in S\}} \{P(f(X) \leq y | X \in \tilde{\Sigma}_k) \geq \delta_k\} \\ &= \arg \min_{y \in \{f(x): x \in \tilde{\Sigma}_k\}} \{P(f(X) \leq y | X \in \tilde{\Sigma}_k) \geq \delta_k\} = y(\delta_k, \tilde{\Sigma}_k) \end{aligned}$$

since  $\tilde{\Sigma}_k \subset S$ . □

Since the algorithm stopped sampling in pruned and maintained subregions, the collection of all sampled points are not uniformly distributed over  $S$ . Theorem 1 allows us to use uniformly sampled points in the current region  $\tilde{\Sigma}_k$  to estimate the objective function quantile over the entire set  $S$ . We use Theorem 1 in Theorem 2 to derive a confidence interval for  $y(\delta, S)$  using results from order statistics (Conover 1999).

**Theorem 2** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where there is no noise. A confidence interval on the true objective function quantile  $y(\delta, S)$  is given by

$$P(f(Z_{(r)}) \leq y(\delta, S) \leq f(Z_{(s)})) \geq 1 - \alpha_k \tag{12}$$

where  $Z_{(1)}, \dots, Z_{(N_k)}$  are  $N_k$  samples ordered with function values uniformly sampled from the current region  $\tilde{\Sigma}_k$  at iteration  $k$ ,  $0 < \alpha_k < 1$ , and  $r$  and  $s$  satisfy

$$P(f(Z_{(r)}) > y(\delta_k, \tilde{\Sigma}_k)) \leq \sum_{i=0}^{r-1} \binom{N_k}{i} (\delta_k)^i (1 - \delta_k)^{N_k-i} = \frac{\alpha_k}{2} \tag{13}$$

$$P(f(Z_{(s)}) \geq y(\delta_k, \tilde{\Sigma}_k)) \geq \sum_{i=0}^{s-1} \binom{N_k}{i} (\delta_k)^i (1 - \delta_k)^{N_k-i} = 1 - \frac{\alpha_k}{2}. \tag{14}$$

We say that we have  $1 - \alpha_k$  confidence that  $f(Z_{(r)}) \leq y(\delta, S) \leq f(Z_{(s)})$ .

*Proof.* At the beginning of any iteration  $k$ , the current set of subregions is  $\Sigma_k$ . According to PBnB, each subregion in the current set  $\tilde{\Sigma}_k$  is uniformly sampled for  $N_k^i = \left\lceil \frac{\ln \alpha_k}{\ln(1 - \frac{\varepsilon_k}{v(\sigma_i)})} \right\rceil$  samples,  $i = 1, \dots, |\Sigma_k|$ . Since each  $v(\sigma_i)$  is the same, the total  $N_k = \sum_{i=1, \dots, |\Sigma_k|} N_k^i$  samples are uniformly sampled from the current region  $\tilde{\Sigma}_k$ . With uniform sampling, we can build the quantile confidence interval, Conover (1999), that is

$$P(f(Z_{(r)}) \leq y(\delta_k, \tilde{\Sigma}_k) \leq f(Z_{(s)})) \geq 1 - \alpha_k,$$

where  $r$  and  $s$  satisfy (13) and (14). From Theorem 1,  $y(\delta, S) = y(\delta_k, \tilde{\Sigma}_k)$ . Then our  $1 - \alpha_k$  confidence interval of  $y(\delta, S)$  is

$$f(Z_{(r)}) \leq y(\delta_k, \tilde{\Sigma}_k) = y(\delta, S) \leq f(Z_{(s)}).$$

□

The confidence interval derived in Theorem 2 provides a statistical measure of quality of the quantile estimation. We now consider the quality of the pruned and maintained regions. Theorem 3 provides a bound on the quality of a *single* pruned subregion  $\sigma_p$ . Theorem 4 extends the bound to consider *all* pruned subregions from iteration 1 to iteration  $k$  in  $\tilde{\Sigma}_{k+1}^P$ .

**Theorem 3** Consider any iteration  $k$  of PBnB on Problem ( $\mathcal{P}$ ) where there is no noise, and suppose  $\sigma_p$  has been pruned on the  $k^{\text{th}}$  iteration. Also, suppose  $y(\delta, S) \leq f(Z_{(s)})$ . Then, the volume of the incorrectly pruned region, i.e.,  $v(L(\delta, S) \cap \sigma_p)$ , is less than or equal to  $\varepsilon_k$  with probability at least  $1 - \alpha_k$ , that is

$$P(v(L(\delta, S) \cap \sigma_p) \leq \varepsilon_k | y(\delta, S) \leq f(Z_{(s)})) \geq 1 - \alpha_k. \quad (15)$$

*Proof.* The probability that the volume of the incorrectly pruned region is less than or equal to  $\varepsilon_k$  is

$$P(v(L(\delta, S) \cap \sigma_p) \leq \varepsilon_k | y(\delta, S) \leq f(Z_{(s)})) = P(v(\{x : f(x) \leq y(\delta, S), x \in \sigma_p\}) \leq \varepsilon_k | y(\delta, S) \leq f(Z_{(s)})). \quad (16)$$

Using (3), and letting  $\delta_p = \frac{\varepsilon_k}{v(\sigma_p)}$ , and since  $X$  is uniformly sampled in  $\sigma_p$ , we have

$$P(f(X) < y(\delta_p, \sigma_p)) = \frac{v(\{x : f(x) < y(\delta_p, \sigma_p), x \in \sigma_p\})}{v(\sigma_p)} \leq \delta_p = \frac{\varepsilon_k}{v(\sigma_p)},$$

then multiplying  $v(\sigma_p)$  on both sides, we have

$$v(\{x : f(x) < y(\delta_p, \sigma_p), x \in \sigma_p\}) \leq \varepsilon_k. \quad (17)$$

Applying (17) to (16), and manipulating the probabilities, we get

$$\begin{aligned} & P(v(\{x : f(x) \leq y(\delta, S), x \in \sigma_p\}) \leq \varepsilon_k | y(\delta, S) \leq f(Z_{(s)})) \\ & \geq P(v(\{x : f(x) \leq y(\delta, S), x \in \sigma_p\}) \leq v(\{x : f(x) \leq y(\delta_p, \sigma_p), x \in \sigma_p\}) | y(\delta, S) \leq f(Z_{(s)})) \\ & \quad - P(v(\{x : f(x) \leq y(\delta, S), x \in \sigma_p\}) > v(\{x : f(x) \leq y(\delta_p, \sigma_p), x \in \sigma_p\}) | y(\delta, S) \leq f(Z_{(s)})). \end{aligned}$$

Associating the volume of a level set with its quantile value, we know that  $v(\{x : f(x) \leq y(\delta, S)\})$  is associated with  $y(\delta, S)$ , and  $v(\{x : f(x) \leq y(\delta_p, \sigma_p), x \in \sigma_p\})$  is associated with  $y(\delta_p, \sigma_p)$ . Hence,

$$\begin{aligned} & = P(y(\delta, S) \leq y(\delta_p, \sigma_p) | y(\delta, S) \leq f(Z_{(s)})) - P(y(\delta, S) = y(\delta_p, \sigma_p) | y(\delta, S) \leq f(Z_{(s)})) \\ & = P(y(\delta, S) < y(\delta_p, \sigma_p) | y(\delta, S) \leq f(Z_{(s)})), \end{aligned}$$

and by the condition of  $f(Z_{(s)})$  and the pruned assumption, we have  $y(\delta, S) \leq f(Z_{(s)}) < f(X_{(p),(1)})$ , therefore,

$$\geq P(f(X_{(p),(1)}) \leq y(\delta_p, \sigma_p) | y(\delta, S) \leq f(Z_{(s)})) = 1 - P(f(X_{(p),(1)}) > y(\delta_p, \sigma_p) | y(\delta, S) \leq f(Z_{(s)})),$$

and since we uniformly sample  $N_k^p$  independent samples in  $\sigma_p$ ,

$$\geq 1 - (1 - \delta_p)^{N_k^p}.$$

Since we desire the probability to be greater than  $1 - \alpha_k$ , we let

$$1 - (1 - \delta_p)^{N_k^p} \geq 1 - \alpha_k \Rightarrow N_k^p \geq \frac{\ln \alpha_k}{\ln(1 - \delta_p)}.$$

Since we let  $\delta_p = \frac{\varepsilon_k}{v(\sigma_p)}$ , and we choose  $N_k^p = \left\lceil \frac{\ln \alpha_k}{\ln(1 - \delta_p)} \right\rceil = \left\lceil \frac{\ln \alpha_k}{\ln(1 - \frac{\varepsilon_k}{v(\sigma_p)})} \right\rceil$ , we have that (15) holds.  $\square$

**Theorem 4** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where there is no noise. The volume of the incorrectly pruned region is less than or equal to  $\varepsilon$  with probability at least  $(1 - \alpha)^2$

$$P(v(L(\delta, S) \cap \tilde{\Sigma}_{k+1}^p \leq \varepsilon) \geq (1 - \alpha)^2. \tag{18}$$

The proof is fairly straightforward, using Theorem 3 and Bernouli's inequality.

Similarly, Theorem 5 derives a bound on the quality of a *single* maintained subregion  $\sigma_m$  using the parameter  $\alpha_k$  and tolerance  $\varepsilon_k$  at iteration  $k$ . Theorem 6 extends the bound to consider *all* maintained subregions from iteration 1 to iteration  $k$  in  $\tilde{\Sigma}_{k+1}^M$ . The proofs are analogous to those of Theorems 3 and 4.

**Theorem 5** Consider any iteration  $k$  of PBnB on Problem  $(\mathcal{P})$  where there is no noise, and suppose  $\sigma_m$  has been maintained. Also, suppose  $y(\delta, S) \geq f(Z_{(r)})$ . Then, the volume of the correctly maintained region, i.e.,  $v(L(\delta, S) \cap \sigma_m)$ , is greater than or equal to  $v(\sigma_m) - \varepsilon_k$  with probability at least  $1 - \alpha_k$ ,

$$P(v(L(\delta, S) \cap \sigma_m) \geq v(\sigma_m) - \varepsilon_k | y(\delta, S) \geq f(Z_{(r)})) \geq 1 - \alpha_k. \tag{19}$$

**Theorem 6** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where there is no noise. The volume of the correctly maintained region is at least  $v(\tilde{\Sigma}_{k+1}^M) - \varepsilon$  with probability at least  $(1 - \alpha)^2$ ,

$$P(v(L(\delta, S) \cap \tilde{\Sigma}_{k+1}^M) \geq v(\tilde{\Sigma}_{k+1}^M) - \varepsilon) \geq (1 - \alpha)^2. \tag{20}$$

The previous analyses (Theorems 1-6) assume the objective function is not noisy, so a precise evaluation  $f(x)$  is sufficient. In Theorem 7, we allow the objective function to have noise and provide rigorous probability bounds for correctly ordering the estimated function values  $\hat{f}(z_{(i)})$  on the  $k$ th iteration. Theorem 8 combines all iterations from 1 to  $k$  and gives a probability bound of  $1 - \alpha$  on the correct ordering.

We use the analysis of a two-stage replication approach, by Bechhofer, Dunnett, and Sobel (1954), in Theorem 7, which has following assumption:

(A1) The noisy function is normally distributed with an unknown common variance  $\sigma^2$ , and at each solution  $z_i \in S$ , the variance can be expressed as  $a_i \sigma^2$  where  $a_i$  is a known constant for each  $i$ .

In PBnB with CI, the constants  $a_i$  are not known, hence, we implement a modified two-stage replication approach (described after Theorem 7).

**Theorem 7** (cf. Bechhofer, Dunnett, and Sobel (1954)) With Assumption (A1), the probability of correct ordering all samples in the current region at iteration  $k$  is

$$P(\hat{f}(Z_{(1)}^k) \leq \hat{f}(Z_{(2)}^k) \leq \dots \leq \hat{f}(Z_{(N_k)}^k) | f(Z_{(1)}^k) \leq f(Z_{(2)}^k) \leq \dots \leq f(Z_{(N_k)}^k)) \geq 1 - \alpha_k, \tag{21}$$

given that we have  $a_i R_0$  as the first stage replication number for each sample point to estimate the common variance by  $S_0^2 = \frac{1}{N_k} \sum_{i=1}^{N_k} \frac{1}{a_i} S_{\hat{f}}^2(z_{(i)})$  and set up the minimum difference desired to be separated as  $d^*$ , and sampling  $R_k = \max\{a_i R_0, 2(\frac{h S_0^*}{d^* \sqrt{2}})^2\}$  in the second stage of the procedure, where  $h$  is a value in the  $H$  c.d.f. of a multivariate student  $t$ 's distribution with  $N_k - 1$  dimensions such that  $H(h) = 1 - \alpha_k$ .



For any iteration  $k$ , we take a conservative approach to achieve the  $1 - \alpha_k$  probability of correct ordering by separating each estimated performance mean  $\hat{f}(z_{(i)})$  with its neighbor by the smallest difference  $d^* = \min_{i=1, \dots, N_k-1} d_i$  of any two neighbors. We also use the largest variance  $S^{*2} = \max_{i=1, \dots, N_k} S_{\hat{f}}^2(z_{(i)})$  so that all ordering is conservative. The implemented two-stage procedure for any iteration  $k$  is

1. Implement  $R_{k-1}$  replications to estimate the function value  $\hat{f}(x)$  and sample variance  $S_{\hat{f}}^2(x)$  and calculate the differences  $d_i$  between ordered samples  $\hat{f}(z_{(i)})$  and  $\hat{f}(z_{(i+1)})$  for  $i = 1, \dots, N_k - 1$ .
2. Implement additional replications  $R_k - R_{k-1}$  where  $R_k = \max\{R_{k-1}, (\frac{z_{\alpha_k/2} S^*}{d^*/2})^2\}$ .

For the implemented two-stage procedure, we drop the assumption of a common variance with  $a_i \sigma^2$  and only assume the noise is normally distributed. In Step 2 of the implemented two-stage procedure we use  $z_{\alpha_k/2}$ , the  $1 - \alpha_k/2$  quantile of the standard normal distribution, together with  $d^*$  and  $S^*$  so that the correct ordering of two function values separated by  $d^*$  with  $S^{*2}$  variance is achieved with probability at least  $1 - \alpha_k$ . Since all other orderings have distance greater than  $d^*$  and variance smaller than  $S^{*2}$ , they also achieve a probability of being ordered correctly of least  $1 - \alpha_k$ . The following Theorem 8 considers the assumptions and rigorous situation in Theorem 7 to derive a rigorous bound for iteration 1 to  $k$ .

**Theorem 8** With Assumption (A1), the probability of correct ordering from iteration 1 to iteration  $k$  is

$$P\left(\bigcap_{l=1}^k \left(\hat{f}(Z_{(1)}^l) \leq \hat{f}(Z_{(2)}^l) \leq \dots \leq \hat{f}(Z_{(N_l)}^l) \mid f(Z_{(1)}^l) \leq f(Z_{(2)}^l) \leq \dots \leq f(Z_{(N_l)}^l)\right)\right) \geq 1 - \alpha, \quad (22)$$

where  $Z_{(j)}^l$  is the  $j^{\text{th}}$  ordered sampled point at iteration  $l$ .

*Proof.* The ordering on each iteration is independent, so we can apply Theorem 7 to yield

$$P\left(\bigcap_{l=1}^k \left(\hat{f}(Z_{(1)}^l) \leq \hat{f}(Z_{(2)}^l) \leq \dots \leq \hat{f}(Z_{(N_l)}^l) \mid f(Z_{(1)}^l) \leq f(Z_{(2)}^l) \leq \dots \leq f(Z_{(N_l)}^l)\right)\right) \geq \prod_{l=1}^k \left(1 - \frac{\alpha}{B^l}\right), \quad (23)$$

and then apply Bernoulli's inequality repeatedly to yield the final result.  $\square$

With the effect of the noisy function, we can now derive noisy versions of Theorem 2, Theorem 4, and Theorem 6 in Corollary 9, Corollary 10, and Corollary 11, to quantify the quality of solution with noise.

**Corollary 9** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where (A1) is assumed. The probability of all confidence intervals correctly capturing the objective function quantile  $y(\delta, S)$  from iteration 1 to  $k$  is

$$P\left(\bigcap_{l=1}^k \left(\hat{f}(Z_{(r)}^l) \leq y(\delta, S) \leq \hat{f}(Z_{(s)}^l)\right)\right) \geq (1 - \alpha)^2. \quad (24)$$

Similarly, Theorems 4 and 6 for pruned and maintained regions' quality holds for the noisy case given the ordering of all samples in the current region is correct. As in Corollary 9, we derive the quality of pruned and maintained regions with noise using conditional and marginal probabilities and invoking Theorem 8.

**Corollary 10** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where (A1) is assumed. The volume of the incorrectly pruned region is bounded by

$$P(v(L(\delta, S) \cap \tilde{\Sigma}_{k+1}^P) \leq \varepsilon) \geq (1 - \alpha)^3. \quad (25)$$

**Corollary 11** Consider any iteration  $k$  of PBnB on  $(\mathcal{P})$  where (A1) is assumed. The volume of the correctly maintained region is bounded by

$$P(v(L(\delta, S) \cap \tilde{\Sigma}_{k+1}^M) \geq v(\tilde{\Sigma}_{k+1}^M) - \varepsilon) \geq (1 - \alpha)^3. \quad (26)$$

Combining Corollaries 9, 10, and 11, we can bound the level set  $L(\delta, S)$  by  $\tilde{\Sigma}_{k+1}^M \subset L(\delta, S) \subset S \setminus \tilde{\Sigma}_{k+1}^P$  with  $(1 - \alpha)^3$  probability that at most  $\varepsilon$  volume is incorrectly pruned and maintained in  $\tilde{\Sigma}_{k+1}^P$  and  $\tilde{\Sigma}_{k+1}^M$ .

#### 4 Numerical Experiments

In this section, we illustrate the capability and performance of PBnB with CI on two test problems. We consider the problem ( $\mathcal{P}$ ) defined in (1) with  $f(x) = E_{\Xi}[g(x) + \xi_x]$  where  $g(x)$  is a test function and  $\xi_x$  is the additive noise depending on  $x$ . The feasible region  $S$  for continuous problems is defined as the hyper-rectangle with box constraints,  $l_i \leq x_i \leq u_i$ , where  $x_i$  is the  $i^{th}$  component in feasible region  $S$ . We choose two test functions, the Rosenbrock function and Sinusoidal function, which are widely used in the global optimization literature (Ali, Khompatporn, and Zabinsky 2005). We present the results in two dimensions to visualize the partitioning and level set approximation easily. We set the parameters as follows:  $\delta = 0.1$ ,  $\alpha = 0.05$ , and  $\varepsilon$  is 2.5 percent of the volume of  $S$ . Also,  $B = 2$  and the unbranchable subregion is defined as the longest Euclidian distance in the subregion less than one percent of the longest Euclidean distance of  $S$ . For both test functions, we tested with no noise,  $\xi_x = 0$ , and with noise,  $\xi_x = N(0, 1)$ .

As in Theorem 2 and Corollary 9, PBnB with CI provides an interval estimation of the quantile,  $y(\delta = 0.1, S)$ , every iteration. Table 1 shows the numerical estimation of the 2-dimensional Rosenbrock and Sinusoidal function with no noise. The sampling number based on PBnB with CI is able to reduce the length of the confidence interval iteratively. The reduction can be observed by the iterative confidence interval from the two test functions in Table 1.

The level set approximation results of PBnB with CI are shown in Figure 1. From left to right, the panels show the non-noisy Rosenbrock function, the noisy Rosenbrock function, the non-noisy Sinusoidal function, and the noisy Sinusoidal function. Each rectangle represents a subregion in the algorithm. The number in a subregion represents the iteration that the subregion is pruned or maintained. The white rectangles are pruned, the light gray (green in color) rectangles are maintained, and the dark gray (blue in color) rectangles are the last current subregions. In Figure 1, we can see the best 10 percent level set,  $L(\delta = 0.1, S)$ , of the Rosenbrock function is approximated almost perfectly for both non-noisy and noisy functions. We can see the best 10 percent level set,  $L(\delta = 0.1, S)$ , of the non-noisy Sinusoidal function is approximated almost perfectly but the noisy version is not. The primary reason is that  $N(0, 1)$  noise is relatively large for the function value of the Sinusoidal function, where the difference between the optimal solution and the 10 percent quantile is about 1.25. In contrast, the difference between the optimal solution and the 10 percent quantile for the Rosenbrock function is about 9.45.

The number of sampled points for the Rosenbrock function are 287,968 and 289,754 for the non-noisy and noisy versions. Also, the number of sampled points for the non-noisy and noisy Sinusoidal function are 252,563 and 855,877. The test functions include real-valued variables, but the algorithm is still not very efficient on the number of points sampled. It is a result of a conservative approach on designing the sample size to achieve volume loss at most  $\varepsilon$  of incorrect pruning and maintaining with  $(B - 1)$  subregions for each iteration. Also, in order to comprehensively represent a subregion by order statistics, the number of samples is determined regardless of the size of the subregion. The benefit of using order statistics is that we need very few assumptions on the objective function, and the order statistics can be efficient for a very large feasible region. A disadvantage of order statistics is that a small feasible region may be practically enumerated. Our future research is to improve the efficiency of the algorithm by implementing an adaptive design of the sample size.

#### 5 Conclusion

We developed the PBnB with CI algorithm to identify a family of solutions to stochastic global optimization problems. The algorithm is designed to solve problems with mixed continuous and discrete variables. The basic framework of PBnB with CI is to iteratively partition the domain to subregions, and prune or maintain the subregion using an interval estimation of quantile. The algorithm dynamically allocates computational resources within each iteration. The algorithm provides an  $1 - \alpha$  confidence interval of the desired quantile  $[\hat{f}(z_{((r))}), \hat{f}(z_{((s))})]$ , and a pruned set  $\tilde{\Sigma}^P$  and a maintained set  $\tilde{\Sigma}^M$  with probability bounds and tolerable loss.

Table 1: Length of confidence interval on  $y(\delta, S)$  for non-noisy Rosenbrock and Sinusoidal functions.

Iteration	Rosenbrock			Sinusoidal		
	CI lower bound	CI upper bound	Half-width	CI lower bound	CI upper bound	Half-width
1	8.3832	32.3042	11.9605	-2.6236	-2.0226	0.3005
2	8.3465	25.4378	8.5457	-2.3382	-2.0864	0.1259
3	6.611	17.4311	5.4101	-2.369	-2.1011	0.1340
4	7.9182	15.0424	3.5621	-2.36	-2.1492	0.1054
5	8.4537	12.7701	2.1582	-2.3034	-2.1773	0.0631
6	8.5411	11.7635	1.6112	-2.3183	-2.2125	0.0529
7	9.1013	11.2579	1.0783	-2.3036	-2.2189	0.0423
8	9.2688	10.8394	0.7853	-2.2884	-2.2349	0.0268
9	9.2972	10.2729	0.4879	-2.2814	-2.2454	0.0180
10	9.4958	10.0955	0.2999	-2.2722	-2.2483	0.0120
11	9.6386	10.0187	0.1901	-2.2651	-2.2499	0.0076
12	9.7014	9.9399	0.1193	-2.2636	-2.2527	0.0054
13	9.7544	9.8896	0.0676	-2.2625	-2.255	0.0038
14	9.7806	9.8632	0.0413	-2.2624	-2.2571	0.0027

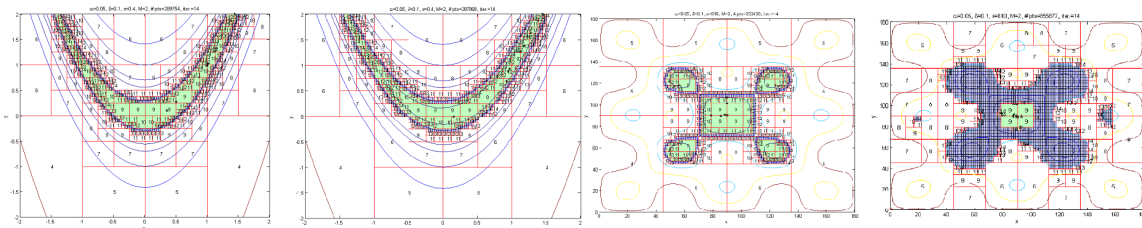


Figure 1: Rosenbrock function ( $\xi_x = 0$ ,  $\xi_x = N(0, 1)$ ) and Sinusoidal function ( $\xi_x = 0$  and  $\xi_x = N(0, 1)$ ).

The analysis in Theorems 2, 4, and 6 assumes  $f(x)$  is no noise, and then the influence of noise on the probability bounds are provided in Corollaries 9, 10, and 11.

There is potential to further improve the algorithm considering some issues highlighted in the discussion of the numerical results. First, the sample size and replication number could be reduced since the current design is conservative. Since we prune and maintain subregions with a tolerance of losing part of the good region and bad region, this could be taken into consideration in the accuracy of quantile relocation. In addition, the partitioning scheme (e.g., parameter  $B$ ) could be dynamically adjusted.

**ACKNOWLEDGMENTS**

This work has been funded in part by the Department of Laboratory Medicine at Seattle Children’s Hospital, and by NSF grant CMMI-1235484.

**REFERENCES**

Ali, M. M., C. Khompatporn, and Z. B. Zabinsky. 2005. “A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems”. *Journal of Global Optimization* 31(4):635–672.

Bechhofer, R. E., C. W. Dunnett, and M. Sobel. 1954. “A Two-Sample Multiple Decision Procedure for Ranking Means of Normal Populations with a Common Unknown Variance”. *Biometrika* 41:170–176.

Chen, C. H., and D. He. 2005. “Intelligent Simulation for Alternatives Comparison and Application to Air Traffic Management”. *Systems Science and Systems Engineering* 14(1):37–51.

- Conover, W. J. 1999. *Practical Nonparametric Statistics, 3rd Edition*. 3rd ed. New York: John Wiley & Sons, Inc.
- Fu, M. C. 2001. "Optimization for Simulation: Theory vs. Practice". *INFORMS Journal on Computing* 14:192–215.
- Gibbons, J. D., I. Olkin, and M. Sobel. 1977. *Selecting and Ordering Populations: A New Statistical Methodology*. New York: Wiley.
- Ho, Y. C., C. G. Cassandras, C. H. Chen, and L. Dai. 2000. "Ordinal optimisation and simulation". *Journal of the Operational Research Society* 51:490–500.
- Ho, Y. C., Q. C. Zhao, and Q. S. Jia. 2007. *Ordinal optimization: Soft optimization for hard problems*. Berlin, Germany: Springer.
- Hu, J., M. C. Fu, and S. I. Marcus. 2007. "A Model Reference Adaptive Search Method for Global Optimization". *Operations Research* 55(3):549–568.
- Norkin, V. I., G. C. Pflug, and A. Ruszczyński. 1998. "A Branch and Bound Method for Stochastic Global Optimization". *Mathematical Programming* 83:425–450.
- Ólafsson, S. 2004. "Two-Stage Nested Partitions Method for Stochastic Optimization". *Methodology and Computing in Applied Probability*.
- Shi, L., and S. Ólafsson. 2000. "Nested Partitions Method for Stochastic Optimization". *Methodology and Computing in Applied Probability*.
- Shi, L., and S. Ólafsson. 2009. *Nested Partitions Method, Theory and Applications*. New York: Springer.
- Xu, W. L., and B. L. Nelson. 2013. "Empirical Stochastic Branch-and-bound for Optimization via Simulation". *IIE Transactions* 45:7:685–698.
- Zabinsky, Z. B., W. Wang, Y. Prasetyo, A. Ghate, and J. Yen. 2011. "Adaptive Probabilistic Branch and Bound for Level Set Approximation". In *Proceedings of the 2011 Winter Simulation Conference*, edited by J. H. K. P. W. S. Jain, R. R. Creasey and M. Fu, 46–57. Phoenix, Arizona: Institute of Electrical and Electronics Engineers, Inc.

## AUTHOR BIOGRAPHIES

**HAO HUANG** is a PhD student in the Department of Industrial and Systems Engineering at the University of Washington. His research interests include simulation optimization and healthcare applications. His e-mail is [haoh7493@uw.edu](mailto:haoh7493@uw.edu).

**ZELDA B. ZABINSKY** is a Professor in the Department of Industrial and Systems Engineering at the University of Washington, with adjunct appointments in the departments of Electrical, Mechanical, and Civil and Environmental Engineering. She is an IIE Fellow, and has published numerous papers in the areas of global optimization, algorithm complexity, and optimal design of composite structures. She received an Erskine Fellowship from the University of Canterbury, Christchurch, New Zealand to collaborate internationally on global optimization methods. Professor Zabinsky's research interests are in global optimization under uncertainty for complex systems. Her email address is [zelda@u.washington.edu](mailto:zelda@u.washington.edu).