ADAPTIVE PROBABILISTIC BRANCH AND BOUND FOR LEVEL SET APPROXIMATION

Zelda B. Zabinsky Wei Wang Yanto Prasetio Archis Ghate Joyce W. Yen

Department of Industrial and Systems Engineering University of Washington, Seattle, WA

ABSTRACT

We present a probabilistic branch-and-bound (PBnB) method for locating a subset of the feasible region that contains solutions in a level set achieving a user-specified quantile. PBnB is designed for optimizing noisy (and deterministic) functions over continuous or finite domains, and provides more information than a single incumbent solution. It uses an order statistics based analysis to guide the branching and pruning procedures for a balanced allocation of computational effort. The statistical analysis also prescribes both the number of points to be sampled within a sub-region and the number of replications needed to estimate the true function value at each sample point. When the algorithm terminates, it returns a concentrated sub-region of solutions with a probability bound on their optimality gap and an estimate of the global optimal solution as a by-product. Numerical experiments on benchmark problems are presented.

1 INTRODUCTION

In many optimization problems arising in complex science and engineering systems, the performance functions are not explicitly available, rather, they have to be estimated via black-box simulations or observed from experimental outputs subject to random system noise. Thus stochastic global optimization problems are more difficult than their deterministic counterparts, which themselves are known to be NP-hard.

Many algorithms have been proposed to solve global optimization problems with noisy function evaluations (Alrefaei and Andradóttir 1999, Ghate and Smith 2008, Hong and Nelson 2006, Hu, Fu, and Marcus 2008, Ólafsson 2004, Rubinstein and Kroese 2004, Shi and Ólafsson 2000a, Shi and Ólafsson 2000b, Spall 2003, Zabinsky 2003, Zabinsky 2011). These algorithms typically apply search strategies to discover a solution that is a good estimate of the global optimum. However, mathematical models are only abstract approximations to the real world problems, hence practitioners are interested, not only in the global optimum, but also in how sensitive the optimal solutions are to noisy perturbations present in the underlying systems (Bertsimas, Nohadani, and Teo 2010, Ho, Sreenivas, and Vakili 1992, Ho, Cassandras, Chen, and Dai 2000, Kristinsdottir, Zabinsky, Tuttle, and Csendes 1996, Pintèr 2006). There is a need to derive a quantitative understanding of the quality and sensitivity of prospective solutions. It is desirable to understand the tradeoff between the computational effort spent exploring the solution space for better solutions with the computational effort spent in estimating the true objective function from noisy evaluations.

We present an algorithm developed in two of the co-author's dissertations (Prasetio 2005, Wang 2011) called *probabilistic branch-and-bound* (PBnB) for solving *both* continuous and discrete optimization problems with black-box, noisy function evaluations. The motivation for PBnB is to provide, not only a single near-optimal solution, but also a subset of solutions that are statistically close to the optimal solution. The subset solution can be interpreted as an approximation of the level set at a target threshold. Instead of using a specific value for the threshold, that may be difficult to know apriori, we define a *closeness*

parameter $0 < \delta < 1$ for the target solution threshold in terms of percentile of the objective function's range distribution. For example, the user may be interested in solutions in the top 10% with respect to objective function value (e.g. $\delta = 0.1$) without knowing the actual target value. This is also used in (Ho, Sreenivas, and Vakili 1992, Ho, Cassandras, Chen, and Dai 2000). The quality of approximation is characterized by a second parameter called the *error rate* $0 < \alpha < 1$, which is used in the probability bounds.

As the algorithm proceeds, the feasible region is partitioned and subregions are pruned when it is statistically valid to do so. When the collected sampling information is insufficient for the statistical analysis to make a confident pruning decision, PBnB branches the remaining subregions. At each iteration, PBnB determines both the number of sample points to sample over each of the current contending subregions and the number of replications needed to reliably estimate each true function value. Upon termination, PBnB returns both an incumbent solution and a subset of solutions, along with probability bounds specifying the quality of the solution set (see Theorems 3, 4, and 5).

PBnB resembles other partitioning based stochastic optimization algorithms such as nested partition (NP) (Shi and Ólafsson 2000a, Shi and Ólafsson 2000b) and its variation, two-stage nested partition (TSNP) (Ólafsson 2004). NP and TSNP systematically partition the feasible region into multiple subregions, assess the potential of each subregion, and then focus the sampling effort on the most promising one. However, a major difference between PBnB and NP/TSNP is that PBnB constantly prunes subregions while NP/TSNP keeps all and occasionally backtracks from a nested subregion to the larger one containing it. For NP/TSNP, retention of all subregions with occasional backtracking is essential for its global convergence result. PBnB also converges with an estimate of the global optimum, but in addition, provides a subset of points whose objective function values are statistically indistinguishable according to derived probability bounds. PBnB also shares similarity with TSNP in that both provide a sample allocation rule for determining the number of sample points on each subregion. However, TSNP's prescription is based on the ranking and selection procedure and the concept of indifference zone, whereas PBnB determines the number of sample points based on a percentile estimate of the range of the objective function over the subregion.

Also of relevance to PBnB is a branch-and-bound deterministic algorithm by Norkin et al. (Norkin, Pflug, and Ruszczyński 1998). Norkin et al. use stochastic upper and lower estimates of the optimal objective function value in each subregion, whereas PBnB employs order statistics on the information collected from random samples. This leads to different assumptions and results in the theoretical analysis.

The statistical analyses on PBnB provide probability bounds on the performance of the algorithm. Theorem 3 provides an upper bound on the probability that PBnB prunes all desired solutions. Theorem 4 states that each pruning improves the "concentration" of desired solutions in the list of contending subregions. This implies that the volume of the set of undesirable solutions within the remaining subregions, called the "margin," is decreasing over iterations. In this sense, the remaining subregions approximate a portion of the target level set of desirable solutions. This provides valuable sensitivity information as discussed in Section 4.2.1, and to the best of our knowledge, no existing stochastic search algorithm for noisy function optimization possesses this capability. Theorem 5 gives a probabilistic bound on the optimality gap between the best solution encountered and the true optimal objective function value.

In Section 2 we describe the PBnB algorithm in detail, and analyze its performance in Section 3. Several numerical results are presented in Section 4, and the paper is concluded in Section 5.

2 PROBABILISTIC BRANCH-AND-BOUND

We are concerned with solving optimization problems of the form:

$$(\mathscr{P}) \qquad \min_{x \in S} f(x), \tag{1}$$

where $f(x) = E_{\Xi}[g(x, \xi_x)]$ is a real valued objective function over the feasible set *S*, which could be a bounded subset of \mathbb{R}^n or \mathbb{Z}^m . Note $g(x, \xi_x)$ is a real valued function representing the system performance at setting *x* perturbed by random noise ξ_x . We assume that f(x) is not available in closed form, but must be estimated by computer simulations or through experiments.

An important feature of PBnB is its capability of returning a subregion that contains a relatively high "concentration" of high quality solutions. PBnB also returns an estimation of the global optima, which can be considered a by-product of the algorithm. We believe that a quantitative statement on the remaining subregion is of even more value than an estimate of a single solution, especially in the presence of noise.

Given the closeness parameter $0 < \delta < 1$, define the *target threshold* $y(\delta, S)$ as

$$y(\delta, S) = \operatorname{argmin}_{y \in \{f(x): x \in S\}} \{ P(f(X) \le y | X \in S) \ge \delta \},$$
(2)

where X is a random point *uniformly* sampled from S. Thus for a point uniformly sampled from S, the probability that its function value falls below this threshold is δ . It is in this probability/percentile sense that δ provides a measure of how close a function value is to the global minimum.

Associated with the target threshold $y(\delta, S)$, define the *target set* $L(\delta, S) \subseteq S$ as the set of desired solutions, equivalently, the *level set* of feasible points in S whose function values are no larger than $y(\delta, \sigma)$,

$$L(\delta, S) = \{ x \in S : f(x) \le y(\delta, S) \}, \quad 0 < \delta < 1.$$
(3)

For any subset σ of *S*, one can similarly define $y(\delta, \sigma)$ and $L(\delta, \sigma)$.

One should further notice that if $f(\cdot)$ is continuous, by applying (2) to an arbitrary region σ , then

$$P(f(X) \le y(\delta, \sigma) | X \in \sigma) = \delta.$$
(4)

If discontinuities in $f(\cdot)$ are allowed, we have

$$P(f(X) \le y(\delta, \sigma) | X \in \sigma) \ge \delta,$$
(5)

and

$$P(f(X) < y(\delta, \sigma) | X \in \sigma) \le \delta.$$
(6)

Next we present the PBnB algorithm.

Probabilistic Branch-and-Bound (PBnB)

Step 0: Initialize. Set user-defined parameters $0 < \alpha$, $\delta < 1$, and $M \ge 2$. Partition *S* into *M* disjoint subregions $\sigma_1, \ldots, \sigma_M$ according to a partitioning scheme. Set $k = 1, \alpha_1 = \alpha/2, \Sigma_1 = \{\sigma_1, \sigma_2, \ldots, \sigma_M\}$. S

$$N_k = \left\lceil \frac{\ln(lpha_k)}{\ln(1-\delta)}
ight
ceil$$
 and $R_k = \left\lceil \frac{\ln\left(rac{lpha_k}{2(|\Sigma_k|-1)}
ight)}{\ln(0.5)}
ight
ceil$,

where [x] denotes the smallest integer $\geq x$. Generate additional points so there are N_k uniformly distributed sample points $x_{m,j} \in \sigma_m$ for $j = 1, ..., N_k$ in each $\sigma_m \in \Sigma_k$ for $m = 1, ..., |\Sigma_k|$. Evaluate $g(x_{m,j}, \xi_{x_{m,j}}^r)$ with noise $\xi_{x_{m,j}}^r$ for $r = 1, ..., R_k$ and calculate

$$\hat{g}(x_{m,j}) = \sum_{r=1}^{R_k} \frac{g(x_{m,j}, \xi_{x_{m,j}}^r)}{R_k}$$
 and $\hat{h}(\sigma_m) = \min_{j=1,\dots,N_k} \hat{g}(x_{m,j})$

Rank all contending subregions $\sigma_m \in \Sigma_k$ according to $\hat{h}(\sigma_m)$ with $\sigma_{(i)}$ denoting the *i*th best subregion, so that

$$\hat{h}(\boldsymbol{\sigma}_{(1)}) \leq \hat{h}(\boldsymbol{\sigma}_{(2)}) \leq \cdots \leq \hat{h}(\boldsymbol{\sigma}_{(|\boldsymbol{\Sigma}_k|)}).$$

Rank all the sample points $x_{(i),j} \in \sigma_{(i)}$ according to $\hat{g}(x_{(i),j})$ with $x_{(i),(j)}$ denoting the j^{th} best point in the i^{th} best subregion, so that

$$\hat{g}(x_{(i),(1)}) \leq \hat{g}(x_{(i),(2)}) \leq \cdots \leq \hat{g}(x_{(i),(N_k)}).$$

With the above ranking carried out *between* and *within* subregions, $x_{(1),(1)}$ is the incumbent solution with the corresponding objective function value estimate $\hat{g}(x_{(1),(1)})$.

Step 2: Prune. Let

$$g^*(x_{(1),(1)}) = \max_{r=1,\dots,R_k} g\left(x_{(1),(1)},\xi_{x_{(1),(1)}}^r\right)$$

and

$$g_*(x_{(d),(1)}) = \min_{r=1,\dots,R_k} g\left(x_{(d),(1)}, \xi^r_{x_{(d),(1)}}\right) \text{ for } d = 2,\dots,|\Sigma_k|.$$

Define indicator functions \mathscr{I}_d for $d = 2, ..., |\Sigma_k|$ such that

$$\mathscr{I}_{d} = \begin{cases} 1, \text{ if } g^{*}(x_{(1),(1)}) < g_{*}(x_{(d),(1)}); \\ 0, \text{ otherwise.} \end{cases}$$

where $\mathscr{I}_d = 1$ indicates deleting $\sigma_{(d)}$.

Step 3: Branch. Define indicator functions \mathscr{J}_b for each $b \in \{1, ..., |\Sigma_k|\} \setminus \{d : \mathscr{I}_d = 1\}$ such that

$$\mathscr{J}_b = \begin{cases} 1, \text{ if } \sigma_{(b)} \text{ is branchable;} \\ 0, \text{ otherwise.} \end{cases}$$

If $\mathscr{J}_b = 1$, branch $\sigma_{(b)}$ into *M* disjoint subregions $\bar{\sigma}^1_{(b)}, \dots, \bar{\sigma}^M_{(b)}$ according to a partitioning scheme. **Step 4: Update.** Update the list of contending subregions Σ_k by removing all subregions to be deleted and replacing all remaining branchable subregions with new subregions, i.e.,

$$\Sigma_{k+1} = \Sigma_k \setminus \left(\bigcup_{\{d:\mathscr{I}_d=1\}} \sigma_{(d)} \right) \setminus \left(\bigcup_{\{b:\mathscr{I}_b=1\}} \sigma_{(b)} \right) \bigcup \left(\bigcup_{\{b:\mathscr{I}_b=1\}} \bigcup_{j=1}^M \bar{\sigma}_{(b)}^j \right).$$

Step 5: Stop. Terminate PBnB if all $\sigma_m \in \Sigma_{k+1}$ are unbranchable. Output the best-found solution $(x_{(1),(1)}, \hat{g}(x_{(1),(1)}))$ and the remaining subregions in Σ_{k+1} . Otherwise, let $\alpha_{k+1} = \alpha_k/2$, increment k and return to Step 1.

The pruning criterion in Step 2 asserts that the subregion $\sigma_{(d)}$ is to be pruned if the best realization of its best point $x_{(d),(1)}$ is worse than the worst realization of the best region's best point $x_{(1),(1)}$. This criterion gives us quantifiable confidence in pruning a subregion in the presence of noise. If the objective function is non-noisy, PBnB always prunes, leaving a single subregion at each iteration for further consideration. This pruning criterion also helps provide the general lower bound $1 - 2\alpha$ in our main results in Section 3.

Step 5 halves α_k at each iteration, which leads both N_k and R_k to increase at a linear rate. Alternatively, we consider a variation in Section 4.2 where $\alpha_k = \alpha/K_0$ and K_0 is a pre-specified upper bound on the maximum number of iterations during one implementation. Since α_k is fixed for this scheme, both N_k and R_k increase sublinearly.

PBnB iterates until the resulting subregions become unbranchable. A subregion is said to be unbranchable in the discrete case when it contains a singleton, and to be unbranchable in the continuous case when the longest Euclidian distance within the subregion is less than a pre-determined positive value. PBnB terminates when there is no branchable subregion left, thus it will terminate in a finite number of iterations.

3 PERFORMANCE ANALYSIS FOR PBNB

In this section, we provide detailed analyses on the performance of PBnB by deriving probability bounds on the quality of *both* the remaining subregions and the incumbent solution during the course of the algorithm. The development of results starts by assuming the true objective function is available in Proposition 2, and then accounts for noisy evaluations in the three main results in Theorems 3, 4 and 5.

We first state a lemma that prescribes the minimum number of sample points needed on a region so that the best function value can achieve a certain level with a pre-specified probability.

Lemma 1 Let $y(\delta, \sigma_m)$ be the $100\delta^{th}$ percentile point as defined in (2) for some subregion $\sigma_m \in \Sigma$ and $0 < \delta < 1$. Suppose $N^* = \lceil \ln \alpha / \ln(1 - \delta) \rceil$ and $X_{m,1}, \ldots, X_{m,N^*}$ are sampled i.i.d. uniformly from σ_m . Let $f(X_{m,(1)})$ be the minimum of $f(X_{m,1}), \ldots, f(X_{m,N^*})$, then

$$P(f(X_{m,(1)}) \le y(\delta, \sigma_m)) \ge 1 - \alpha.$$
(7)

Proof. By (5), we have

$$P(f(X) > y(\delta, \sigma) | X \in \sigma) \le 1 - \delta,$$

thus,

$$P(f(X_{m,(1)}) > y(\boldsymbol{\delta}, \boldsymbol{\sigma}_m)) \leq (1 - \boldsymbol{\delta})^{N^*},$$

which yields

$$P(f(X_{m,(1)}) \le y(\delta, \sigma_m)) = 1 - P(f(X_{m,(1)}) > y(\delta, \sigma_m)) \ge 1 - (1 - \delta)^{N^*}.$$
(8)

To prove (7), we set the right hand side of (8) greater than or equal to $1 - \alpha$, that is

$$1 - (1 - \delta)^{N^*} \ge 1 - \alpha$$

or

$$N^* \geq \frac{\ln \alpha}{\ln(1-\delta)}.$$

Therefore, if $N^* = \lceil \ln \alpha / \ln(1 - \delta) \rceil$, then (7) is satisfied.

Proposition 2, which assumes all objective function values are precise, provides a probability bound that after *K* iterations, the remaining region contains at least one of the desired solutions (in $L(\delta, S)$). **Proposition 2** Suppose PBnB has progressed to the current K^{th} iteration, with $\alpha_1 \leq \alpha$. At iteration *k* of

Proposition 2 Suppose PBnB has progressed to the current K^m iteration, with $\alpha_1 \le \alpha$. At iteration k of PBnB, let the number of sample points be given as in Step 1, and let σ_p^k denote the pruned region according to the true function value $f(\cdot)$ at iteration k, then

$$P\left(\left(S \setminus \bigcup_{k=1}^{K} \sigma_p^k\right) \bigcap L(\delta, S) \neq \emptyset\right) > 1 - \alpha.$$
(9)

Proof. Initially, PBnB branches *S* into *M* subregions $\sigma_1, \sigma_2, \ldots, \sigma_M$ and samples $N_1 = \left\lceil \frac{\ln \alpha_1}{\ln(1-\delta)} \right\rceil$ on each. Thus the combined number of sample points from all the subregions is $M\left\lceil \frac{\ln \alpha_1}{\ln(1-\delta)} \right\rceil$. Denote the smallest order statistics of function values by $f(X_{S,(1)})$. Note

$$M\left\lceil\frac{\ln\alpha_1}{\ln(1-\delta)}\right\rceil \ge M\left\lceil\frac{\ln\alpha}{\ln(1-\delta)}\right\rceil > \left\lceil\frac{\ln\alpha}{\ln(1-\delta)}\right\rceil$$

thus by applying Lemma 1 to *S*, we get $P(f(X_{S,(1)}) \le y(\delta, S)) \ge 1 - \alpha$, indicating a desirable solution is found with probability at least $1 - \alpha$ during the first iteration. Moreover, as the iterative process of PBnB continues, new incumbents may be found, and at the end of each iteration, the subregion containing the incumbent is always kept, thus the intersection of $L(\delta, S)$ and the remaining subregion is non-empty with probability at least $1 - \alpha$, and the proof is completed.

The proof of Proposition 2 reveals that a desired solution can be highly possibly located within the first iteration. Theorem 3 focuses on analyzing the situation when this is not true and states that by termination the intersection of the remaining region with $L(\delta, S)$ has *positive measure* with probability at least $1 - 2\alpha$.

Theorem 3 For PBnB on Problem \mathscr{P} , assume the distribution of $g(x, \xi_x)$ is symmetric about f(x). Suppose PBnB has progressed to the current K^{th} iteration but the incumbent has not reached the δ^{th} percentile in function values over *S*, i.e., $f(X_{(1),(1)}^k) > y(\delta,S)$ for k = 1, 2, ..., K. At iteration *k* of PBnB, let $\{\sigma_{(1)}^k, \sigma_{(2)}^k, ..., \sigma_{(|\Sigma_k|)}^k\}$ be the set of contending subregions. Let $\sigma_p^k = \bigcup_{\{d:\mathscr{I}_d=1\}} \sigma_{(d)}^k$ denote the pruned region at iteration *k* with \mathscr{I}_d defined in Step 2, then

$$P\left(v\left(\left(S \setminus \bigcup_{k=1}^{K} \sigma_{p}^{k}\right) \bigcap L(\delta, S)\right) > 0\right) > 1 - 2\alpha,$$
(10)

for $0 < \alpha, \delta < 1$, where $L(\delta, S)$ is the set of desired solutions as defined in (3).

Proof. See (Wang 2011, Chapter 4, Theorem 11).

As noted in (Prasetio 2005, pg. 12), the symmetric assumption on $g(x, \xi_x)$ in Theorem 3 can be relaxed by minimizing arbitrary γ percentile of $g(x, \xi_x)$ instead of the expected value $f(x) = E_{\Xi}[g(x, \xi_x)]$.

Despite the constant deletion of subregions during the algorithm's execution, Theorem 3 shows that PBnB manages to maintain a *fixed* probability bound that the remaining region contains desired solutions and thus keeps the pruning error under control.

Next, we introduce two additional metrics *exclusively* for PBnB on the quality of the remaining region: *concentration* and *margin*. The concentration for a given list of contending subregions measures the proportion of desired solutions, and the margin measures the volume of undesirable solutions in the remaining region, see (11). Thus, they provide more detailed information regarding the quality of the remaining region. Theorem 4 shows that before the incumbent reaches the target threshold $y(\delta, S)$, one more pruning could increase the concentration and decrease the margin, with a probability of at least $1 - 2\alpha$.

Theorem 4 Suppose that up to the K^{th} iteration the incumbent has not reached the δ^{th} percentile in function values over *S*, i.e., $f(X_{(1),(1)}^k) > y(\delta, S)$ for k = 1, 2, ..., K. At iteration *k*, let σ_p^k be the subregion pruned according to Step 2 of the algorithm, \mathscr{C}_k and \mathscr{M}_k be the concentration and margin of the remaining region respectively, i.e.,

$$\mathscr{C}_{k} = \frac{\nu\left(\left(S \setminus \bigcup_{i=1}^{k} \sigma_{p}^{i}\right) \cap L(\delta, S)\right)}{\nu\left(S \setminus \bigcup_{i=1}^{k} \sigma_{p}^{i}\right)}, \quad \mathscr{M}_{k} = \nu\left(S \setminus \bigcup_{i=1}^{k} \sigma_{p}^{i}\right) - \nu\left(\left(S \setminus \bigcup_{i=1}^{k} \sigma_{p}^{i}\right) \cap L(\delta, S)\right), \quad (11)$$

then we have the following for k = 1, 2, ..., K - 1:

. .

$$P(\mathscr{C}_{k+1} \ge \mathscr{C}_k) > 1 - 2\alpha \tag{12}$$

$$P(\mathcal{M}_{k+1} < \mathcal{M}_k) > 1 - 2\alpha.$$
⁽¹³⁾

Proof. See (Wang 2011, Chapter 4, Theorem 12).

In Theorem 4, the concentration relationship assumes $f(X_{(1),(1)}^k) > y(\delta, S)$. If this condition does not hold, then the algorithm has already detected a point in the target set. For example, suppose the target threshold is 5%, but the current value for $f(X_{(1),(1)}^k)$ is even better, at 2%. Then a pruning of point in the 4-5% range may actually decrease the concentration of targeted points, however, the quality of the points in the remaining subregions exceed the target threshold. In the numerical results, we can see how the list of contending subregions approximates portions of the level sets of decreasing function value, even beyond the target threshold.

Theorem 5 makes a qualitative statement about each new record value encountered during the algorithm execution. It needs a technical assumption that the δ used in Step 1 (called δ_1 in the theorem) is strictly less than the target δ .

Theorem 5 Suppose the PBnB algorithm has progressed to the current K^{th} iteration and the closeness parameter during the first iteration δ_1 is strictly less than δ . If a new record point $X_{(1),(1)}^{K+1}$ is found at the $(K+1)^{th}$ iteration, i.e.,

$$f\left(X_{(1),(1)}^{K+1}\right) < f\left(X_{(1),(1)}^{K}\right),$$

then

$$P\left(f\left(X_{(1),(1)}^{K+1}\right) \le y(\delta,S)\right) > 1 - \left(2 + \frac{1}{2^{K+1}}\right)\alpha,$$
(14)

where $y(\delta, S)$ is the δ^{th} percentile in function values over S as defined in (2).

Proof. See (Wang 2011, Chapter 4, Theorem 13).

The lower bound on the right hand side of (14) is increasing with K, and it agrees with the intuition that the new records are more and more likely to hit the target region.

4 NUMERICAL EXPERIMENTS

In this section, we illustrate the performance of PBnB on various test problems. Section 4.1 describes the experimental setup, and we present computational results and discussion in Section 4.2.

4.1 Experimental Setup

We solve Problem (\mathcal{P}) as defined in (1) with

$$f(x) = E_{\Xi}[g(x) + \xi_x],$$

where g(x) is a test function and ξ_x is the additive simulated noise dependent on x.

For continuous problems, the feasible region S is a hyper-rectangle defined by a set of box constraints $l_i \le x_i \le u_i$, where x_i is the *i*th component of a feasible solution $x \in S$. In the finite case, the test problems are discretized versions of their continuous counterparts and S is a set of equally spaced grid points.

We choose the following four test functions, among which the Rosenbrock function, the Hartmann function and the sinusoidal function are well-known benchmark problems frequently used in the global optimization literature (Ali, Khompatraporn, and Zabinsky 2005).

1. Rosenbrock function $(n = 2, -2 \le x_i \le 2, i = 1, ..., n)$

$$g_1(x) = \sum_{i=1}^{n-1} \left[(1-x_i)^2 + 100(x_{i+1} - x_i^2)^2 \right].$$

The global minimum is located at $x_* = (1,1)$ with $g_1(x_*) = 0$.

2. Norm function $(n = 20, -1000 \le x_i \le 1000, i = 1, ..., n)$

$$g_2(x) = ||x||_2.$$

Clearly the global minimum is located at $x_* = (0, ..., 0)^T$ with $g_2(x_*) = 0$. 3. Hartmann function $(n = 6, 0 \le x_i \le 1, i = 1, ..., n)$

$$g_3(x) = -\sum_{i=1}^4 c_i \exp\left[-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right],$$

where values for *c*, *a*, and *p* are given in (Wang 2011). The global minimum is located at $x_* \approx (0.201690, 0.150011, 0.476874, 0.275332, 0.311652, 0.657301)$ with $g_3(x_*) \approx -3.322368$.

4. Sinusoidal function (n = 10)

$$g_4(x) = -2.5 \prod_{i=1}^n \sin\left(\frac{\pi x_i}{180}\right) - \prod_{i=1}^n \sin\left(\frac{\pi x_i}{36}\right).$$

For the continuous version, $0 \le x_i \le 180, i = 1, ..., 10$, and for the discrete version, $x_i \in \{30, 60, ..., 180\}, i = 1, ..., 10$. The global minimum is at $x_* = (90, ..., 90)$ with $g_4(x_*) = -3.5$.

We set the error rate $\alpha = 0.25$, the closeness parameter $\delta = 0.10$ and the number of subregions M = 3. We consider a subregion to be unbranchable when the longest Euclidian distance in the subregion is less than one percent of the longest Euclidean distance of *S*.

4.2 Results and Discussion

In Section 4.2.1, we illustrate how PBnB approximates portions of the set of desired solutions, thus providing important sensitivity information regarding the behavior of the noisy objective function. We use the two-dimensional Rosenbrock function so that key ideas can be demonstrated using pictures. We report the numerical performance of PBnB and a variation on three high-dimensional problems in Section 4.2.2. Finally, we solve a parameter estimation problem from (Pasupathy and Henderson 2006) in Section 4.2.3.

4.2.1 Illustration of the Sensitivity Information Provided by MPBnB

For the 2-dimensional Rosenbrock function $g_1(x)$, we record the pruning and partitioning processes over the contours, allowing us to visualize how the remaining region iteratively approximates a portion of the target level set.

In Figure 1, we print in the title of each plot the number of contending subregions (# rgn) and the number of total points sampled (# pts) after PBnB terminates in the stated number of iterations (iter.). The number in the center of each rectangle indicates at which iteration this subregion gets pruned. In addition, the global minimum is marked with an asterisk "*" and the incumbent solution at each iteration is marked with a circle "o". We also shade the remaining region for ease of recognition.



Figure 1: Numerical results for 2D Rosenbrock function.

Figure 1 (a) and (b) illustrate the progress of PBnB on the Rosenbrock function, where the innermost "banana" shaped level set corresponds to the function value of 2. The shaded region gives a clear view of how the remaining region maps out a portion of the level set at the desired threshold. The "shape" of the remaining subregion provides insight into the flexibility the practitioner has in choosing values of

various design variables for the complex system at hand, as well as the sensitivity to noise in the objective function. Comparing Figure 1(a) with a low level of noise, $\xi \sim 0.1 \cdot N(0,1)$, to Figure 1(b) with a high level of noise, $\xi \sim 1.0 \cdot N(0,1)$, the remaining region increases in size and possesses a larger margin when the noise scale increases.

PBnB converges rapidly to the level set at the desired threshold in the first few iterations, after that it spends most of the computational effort in smaller subregions. In addition, the global minimum is cautiously kept within the remaining region with a sequence of incumbent solutions closely around it.

4.2.2 Volume Reduction and Incumbent Function Value on High Dimensional Problems

One key feature of PBnB is its ability to return a final list of subregions containing a desired solution with a predetermined probability controllable to the user, and naturally, one would hope that the remaining region is significantly smaller than the initial feasible region *S*. In this section, we provide such information through the *volume ratio* of the remaining region to *S*.

Moreover, we propose and test a variation where instead of halving α at each iteration, we fix $\alpha_k = \alpha/K_0$ where K_0 is a pre-specified upper bound on the maximum number of iterations during one run. This new choice of α_k will leave Theorems 3 and 4 completely intact. Under the fixed α scheme, the expression $\prod_{k=1}^{K} (1 - \alpha/2^k)$ within this inequality would then be replaced with $\prod_{k=1}^{K} (1 - \alpha/K_0)$, which is greater than or equal to $\prod_{k=1}^{K} (1 - \alpha/K) = (1 - \alpha/K)^K$ (since $K \le K_0$) and hence greater than $1 - \alpha$. For Theorem 5, the probability bound $1 - (2 + 1/2^{K+1})\alpha$ in (14) will be modified to $1 - (2 + 1/K_0)\alpha$, however, this change is insignificant because both $\alpha/2^{K+1}$ and α/K_0 are very small numbers.

We test PBnB on the 6-dimensional Hartmann function $g_3(x)$ the 10-dimensional sinusoidal function for both continuous $g_4^c(x)$ and discrete $g_4^d(x)$, and the 20 dimensional norm function $g_2(x)$. To compare the performance of PBnB under the halved α and the fixed α scheme, we choose $K_0 = 20$ for the Hartmann function and $K_0 = 100$ for all others. We use $\xi_x \sim \mu_i \cdot N(0, g_i(x)^2)$ to simulate the additive noise for test function $g_i(x)$ with associated noise scale $0 < \mu_i < 1$. For the Hartmann function and the discrete sinusoidal function μ_i is 0.1, while for the continuous sinusoidal function and the norm function, μ_i is set to 0.05.

We make 100 independent replications on all test functions. In all the plots of Figure 2, the optimal value is represented by the dashed line. We use two vertical axes, with the left one for the estimated function value and the right one for the volume ratio of the remaining region to the initial feasible region *S*. The horizontal axis records the number of points sampled, which is smaller than the typically used number of function evaluations since the objective function is evaluated multiple times (R_k) at each sample point. In addition to Figure 2, we also report detailed performance data in Table 1.

Test	Dim	# points	Mean # fun. eval.		Opt.	Mean best value		Mean vol. ratio	
fun.	п	sampled	halved α	fixed α	value	halved α	fixed α	halved α	fixed α
<i>g</i> 3	6	1e4	1.4e5	1.3e5	-3.32	-3.21	-3.26	0.0033	0.0037
g_4^d	10	2e4	4.8e5	2.5e5	-3.5	-2.86	-3.14	1.04e-5	9.83e-6
g_4^c	10	3e5	9.4e6	5.4e6	-3.5	-2.94	-3.18	1.08e-11	3.13e-13
<i>g</i> ₂	20	3e5	1.1e7	5.5e6	0	560.41	329.51	9.46e-15	4.85e-22

Table 1: Numerical results.

For the Hartmann function, the difference between the two α schemes is insignificant in both the volume ratio and the function value. For the discrete sinusoidal function, although the fixed α scheme does not lead to further reduction in volume of the remaining region, it does improve the estimated function value. For the continuous sinusoidal function and the norm function, as illustrated in Figure 2 (c) and (d), the fixed α scheme prevails against the halved α scheme in both volume reduction and function value during almost the entire course.



Figure 2: Volume reduction and incumbent function value for four test problems.

Aiming at more accurately reflecting the stochastic performance of algorithms, Pasupathy and Henderson (Pasupathy and Henderson 2006) proposed a way of reporting that depicts the distribution of objective function value as a function of time. See Figure 4.5 in (Wang 2011) where we use iteration (k) as a measure of time, and report $P(Y_k \le r)$ for various problem dependent threshold r on the four test problems with the halving α scheme.

4.2.3 Parameter Estimation

Estimating parameters is a common problem and often formulated as a maximum likelihood optimization problem. We consider a problem from Pasupathy and Henderson (Pasupathy and Henderson 2006), where a set of *m* i.i.d data points Y_j , (j = 1, ..., m) are generated from the two-dimensional pdf

$$f(y_1, y_2; x^*) = \frac{e^{-y_1} y_1^{x_1^* y_2 - 1}}{\Gamma(x_1^* y_2)} \frac{e^{-y_2} y_2^{x_2^* - 1}}{\Gamma(x_2^*)}, \quad y_1, y_2 > 0$$

with some *undisclosed* parameter $x^* = (x_1^*, x_2^*)$. The task is to recover x^* given data points Y_j , j = 1, ..., m. Note x^* can be recovered as the maximum likelihood estimator (MLE), we thus have the following optimization problem,

$$\max G_m(x) = \frac{\sum_{j=1}^m \log(f(Y_j; x))}{m}$$

Following Pasupathy and Henderson (Pasupathy and Henderson 2006), we choose $x^* = (2,5), m = 10,000$, and $S = (0,10) \times (0,10)$ to set up the experiment. For this problem, once the data $Y_j, j = 1, ..., m$ is given, the objective function (the log likelihood) is non-noisy, thus at each iteration only the most promising subregion is retained while the others all get pruned. PBnB finds the global optimum with a total amount of 285 sample points in 7 iterations. A graph of the numerical results, in the style of Section 4.2.1, can be found in (Wang 2011, Figure 4.11).

5 CONCLUSIONS

We developed a PBnB framework to solve stochastic global optimization problems with continuous and/or discrete variables. PBnB uses a statistical analysis to dynamically allocate computational effort during the course of its implementation. In particular, at each iteration, it prescribes both the number of points sampled from each subregion and the number of independent replications needed to reliably estimating function values. While PBnB constantly eliminates subregions, it manages to maintain some quality assurance on the remaining region. Upon termination, PBnB returns a small but concentrated region of solutions along with a probability bound on the optimality gap. This feature is not typically available in traditional optimization algorithms.

Nevertheless, there remain several further issues that warrant additional research. First, PBnB takes uniformly distributed samples on each subregion. It would be interesting to investigate other distributions and incorporate local search heuristics into the analysis. Also, in our numerical implementation, PBnB chooses the dimension corresponding to the longest length of a subregion for partitioning, but we believe more intelligent partitioning schemes would improve the algorithm performance and hence is of much interest.

REFERENCES

- Ali, M. M., C. Khompatraporn, 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.
- Alrefaei, M. H., and S. Andradóttir. 1999. "A Simulated Annealing Algorithm with Constant Temperature for Discrete Stochastic Optimization". *Management Science* 45 (5): 748–764.
- Bertsimas, D., O. Nohadani, and K. M. Teo. 2010. "Robust Optimization for Unconstrained Simulation-Based Problems". *Operations Research* 58 (1): 161–178.
- Ghate, A., and R. L. Smith. 2008. "Adaptive Search with Stochastic Acceptance Probabilities for Global Optimization". *Operations Research* 36 (3): 285–290.
- 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., R. S. Sreenivas, and P. Vakili. 1992. "Ordinal optimization of DEDS". *Discrete Event Dynamic Systems: Theory and Application* 2:61–88.
- Hong, L., and B. L. Nelson. 2006. "Discrete optimization via simulation using COMPASS". *Operations Research* 54 (1): 115–129.
- Hu, J., M. C. Fu, and S. I. Marcus. 2008. "A Model Reference Adaptive Search Method for Stochastic Global Optimization". *Communications in Information and Systems* 8:245–276.
- Kristinsdottir, B. P., Z. B. Zabinsky, M. E. Tuttle, and T. Csendes. 1996. "Incorporating manufacturing tolerances in near-optimal design of composite structures". *Engineering Optimization* 26:1–23.
- 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 Two-Stage Nested Partitions Method for Stochastic Optimization". *Methodology and Computing in Applied Probability* 6:5–27.

- Pasupathy, R., and S. Henderson. 2006, December. "A testbed of simulation-optimization problems". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 255–263. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Pintèr, J. D. 2006. Global optimization: scientific and engineering case studies. New York: Springer.
- Prasetio, Y. 2005. Simulation-based Optimization for Complex Stochastic Systems. Ph. D. thesis, University of Washington.
- Rubinstein, R. Y., and D. P. Kroese. 2004. *The Cross Entropy Method: A Unified Combinatorial Approach* to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning. Springer.
- Shi, L., and S. Ólafsson. 2000a. "Nested partitions method for global optimization". *Operations Research* 48 (3): 390–407.
- Shi, L., and S. Ólafsson. 2000b. "Nested partitions method for stochastic optimization". *Methodology and Computing in Applied Probability* 2 (3): 271–291.
- Spall, J. C. 2003. *Introduction to stochastic search and optimization: estimation, simulation, and control.* Hoboken, N.J.: Wiley-Interscience.
- Wang, W. 2011. Adaptive Random Search for Noisy and Global Optimization. Ph. D. thesis, University of Washington.
- Zabinsky, Z. B. 2003. Stochastic Adaptive Search in Global Optimization. Springer.
- Zabinsky, Z. B. 2011. "Random Search Algorithms". In Wiley Encyclopedia of Operations Research and Management Science, edited by J. J. Cochran, DOI: 10.1002/9780470400531.eorms0704. John Wiley & Sons.

AUTHOR BIOGRAPHIES

ZELDA B. ZABINSKY is a Professor in the Department of Industrial and Systems Engineering at the University of Washington. She is an IIE Fellow, on the editorial board of the Journal of Global Optimization, and a board member of the Pacific Institute of Mathematical Sciences (PIMS). Her research interests are in global optimization under uncertainty for complex systems. Her email address is zelda@u.washington.edu.

WEI WANG received his Ph.D. from the Department of Industrial and Systems Engineering at the University of Washington in 2011. He is currently an Associate Scientist at PROS Revenue Management. His research interests include adaptive random search and simulation optimization. His e-mail is wangwei@uw.edu.

YANTO PRASETIO received his Ph.D. from the Department of Industrial and Systems Engineering at the University of Washington. He returned to his home country of Indonesia to work in financial optimization.

ARCHIS GHATE is an Assistant Professor in the Department of Industrial and Systems Engineering at the University of Washington, Seattle. His research interests focus on theory and algorithms for dynamic optimization under uncertainty with applications to medical treatment planning and service operations management. His e-mail is archis@u.washington.edu.

JOYCE W. YEN is an Affiliate Assistant Professor in the Department of Industrial and Systems Engineering at the University of Washington, and is the Program/Research Manager of the University of Washington's ADVANCE Center for Institutional Change. She is committed to increasing the participation of women in the sciences, engineering, and mathematics. Her e-mail is joyceyen@u.washington.edu.