

EMPIRICAL STOCHASTIC BRANCH-AND-BOUND FOR OPTIMIZATION VIA SIMULATION

Wendy Lu Xu
Barry L. Nelson

Department of Industrial Engineering and Management Sciences
Northwestern University
2145 Sheridan Road, Evanston, IL 60208, USA

ABSTRACT

We introduce a new method for discrete-decision-variable optimization via simulation that combines the stochastic branch-and-bound method and the nested partitions method in the sense that we take advantage of the partitioning structure of stochastic branch and bound, but estimate the bounds based on the performance of sampled solutions as the nested partitions method does. Our Empirical Stochastic Branch-and-Bound algorithm also uses improvement bounds to guide solution sampling for better performance.

1 INTRODUCTION

For large-scale and complicated stochastic optimization problems, a closed-form objective function may not exist, and therefore it has to be estimated through simulation. Optimization via simulation (OvS) provides solutions to this type of problem. [Andradottir \(1998\)](#), [Fu \(2002\)](#), and [Fu, Glover, and April \(2005\)](#) give extensive overviews of the OvS literature. Most algorithms for discrete-decision-variable optimization via simulation (DOvS) are based on adaptive random search. The two research threads that are the roots of our work are the Nested Partitions (NP) and the Stochastic Branch-and-Bound (SB&B) methods.

NP is a globally convergent framework proposed by [Shi and Olafsson \(2000\)](#). At each iteration, the algorithm identifies a most promising region. When a better solution is found inside the current most promising region, the region is partitioned for further exploration. If a better solution is found outside of the current most promising region, NP backtracks to its super region or to the whole feasible region. The intention of NP is to concentrate computational effort where good solutions appear to be, and the search is guided by the estimated performance of randomly sampled solutions.

Branch-and-Bound algorithms (B&B) are widely used to solve deterministic combinatorial optimization problems. [Norkin, Ermoliev, and Ruszczyński \(1998\)](#) and [Norkin, Pflug, and Ruszczyński \(1998\)](#) adapt the B&B idea to the stochastic setting. Their SB&B algorithm iteratively partitions the feasible region into smaller and smaller subregions; estimates bounds on the objective function for these subregions by solving bounding problems; and selects as the record set the subregion with the maximum or minimum (depending on the problem) bound. Global convergence can be proven for problems with a finite number of feasible solutions. The SB&B algorithm assumes that it is possible to estimate the bounds more and more precisely with increasing simulation effort. Papers applying SB&B include [Gutjahr, Hellmayr, and Pflug \(1999\)](#), [Gutjahr, Strauss, and Wagner \(2000\)](#), and [Doerner et al. \(2006\)](#). Unfortunately, complicated DOvS problems may not have solvable bounding problems.

In our research, we combine SB&B and NP in the sense that we take advantage of the partitioning structure of SB&B, but estimate bounds based on the performance of sampled solutions, as NP does. Our Empirical Stochastic Branch-and-Bound (ESB&B) algorithm also uses bounds to guide solution sampling for better performance. ESB&B is globally convergent.

The paper is organized as follows. Section 2 reviews SB&B and NP, presents ESB&B and states the convergence theorem. Section 3 describes how ESB&B uses bounds to guide solution sampling. Section 4 displays experimental results, while Section 5 concludes the paper. Details not found in this paper are available in Xu (2009).

2 EMPIRICAL STOCHASTIC BRANCH-AND-BOUND

First we define the DOvS problem. Our goal is to find \mathbf{x} that solves

$$\max_{\mathbf{x} \in X \cap D} \mu(\mathbf{x}), \tag{1}$$

where X is a finite set defined as the intersection of the integer lattice with a hypercube in \mathbb{R}^q , given by

$$l_i \leq x_i \leq u_i, \quad l_i, x_i, u_i \in \mathbb{Z}, \quad i = 1, 2, \dots, q;$$

and D is a subregion in \mathbb{R}^q given by the inequalities

$$D = \{\mathbf{x} \in \mathbb{R}^q : g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, p\}.$$

We assume D is convex, and therefore $X \cap D$ is convex and finite.

In this paper we consider stochastic problems where $\mu(\mathbf{x}) = \mathbf{E}[Y(\mathbf{x})]$, and $\mu(\mathbf{x})$ can only be estimated by generating observations of $Y(\mathbf{x})$ via simulation. Like the NP method, ESB&B can also be applied to deterministic problems; adjustments can be found in Xu (2009).

2.1 SB&B and NP

In the classical branch-and-bound algorithm, X is iteratively divided into subregions X^P generating a partition \mathcal{P} . Let $\mu^*(X^P)$ denote the optimal objective function value of the subproblem that is restricted to X^P :

$$\mu^*(X^P) = \max_{\mathbf{x} \in X^P \cap D} \mu(\mathbf{x}), \quad X^P \in \mathcal{P},$$

where $\mu^*(X^P) = -\infty$ if $\mathbf{x} \in X^P \cap D = \emptyset$. Then clearly the optimal value of Problem (1) equals $\mu^*(X) = \max_{X^P \in \mathcal{P}} \mu^*(X^P)$. Norkin, Ermoliev, and Ruszczyński (1998) makes the following assumptions:

Assumption 1. For each subregion $X^P \subset X$, there exists functions $L : 2^X \rightarrow \mathbb{R}$ and $U : 2^X \rightarrow \mathbb{R}$ such that

$$\begin{aligned} L(X^P) &\leq \mu^*(X^P) \leq U(X^P), \\ L(X^P) &= \mu(\mathbf{x}') \quad \text{for some } \mathbf{x}' \in X^P, \end{aligned}$$

and if X^P is a singleton then $L(X^P) = \mu^*(X^P) = U(X^P)$.

In a stochastic problem these bounds can only be calculated exactly in some special cases. Therefore, SB&B is based on the availability of statistical estimates, η and ξ , of U and L , respectively, that can be generated and refined through iterations of the algorithm.

Assumption 2. For each subregion $X^P \subset X$, there exists sequences of estimators η^k and ξ^k , $k = 1, 2, \dots$, such that $\lim_{k \rightarrow \infty} \eta^k(X^P) = U(X^P)$ a.s. and $\lim_{k \rightarrow \infty} \xi^k(X^P) = L(X^P)$ a.s. as $k \rightarrow \infty$.

The SB&B algorithm works as follows: A record set is selected at each iteration as the subregion with the greatest (estimated) upper bound. An approximate solution is chosen as an element of the subregion with the greatest (estimated) lower bound. The record set is partitioned into smaller subregions, and the estimates of the upper and lower bounds are refined. A new record set and approximate solution are selected and the algorithm continues. Below we give a more detailed description of the SB&B algorithm.

SB&B Algorithm

- Step 0. Initialization:* Set iteration counter $k = 0$, initial partition and record set $\mathcal{P}_0 = R^0 = \{X\}$, and calculate the bounds $\eta^0(X)$ and $\xi^0(X)$.
- Step 1. Partitioning:* Select an approximate solution $\mathbf{x}^k \in X^k = \arg \max\{\xi^k(X^P) : X^P \in \mathcal{P}_k\}$. If the record set R^k is a singleton, then set $\mathcal{P}'_k = \mathcal{P}_k$ and go to Step 2. Otherwise construct a partition of the record set, $\mathcal{P}''_k(R^k)$, from which infeasible subregions have been cleaned. Define the new full partition by $\mathcal{P}'_k = (\mathcal{P}_k \setminus R^k) \cup \mathcal{P}''_k(R^k)$. Elements of \mathcal{P}'_k will also be denoted by X^P .
- Step 2. Bounding:* For all regions $X^P \in \mathcal{P}'_k$ calculate estimates $\eta^{k+1}(X^P)$ and $\xi^{k+1}(X^P)$ for $U(X^P)$ and $L(X^P)$, respectively.
- Step 3. Updating partition and record set:* Update record set $R^{k+1} = \arg \max\{\eta^{k+1}(X^P) : X^P \in \mathcal{P}'_k\}$, and partition $\mathcal{P}_{k+1} = \mathcal{P}'_k$. Set $k = k + 1$ and go to Step 1.

There are two potential drawbacks to the direct application of SB&B. First, there needs to be bounding functions L and U and convergent estimators of them. Second, there is overhead needed to retain and refine a larger and larger partition structure as the algorithm progresses since no partition is ever eliminated from consideration as in deterministic branch and bound.

The first drawback can be addressed using a sampling-based upper bound. We can simply choose the solution with the greatest accumulated sample average through the current iteration and use this average as the estimate of the upper bound. Then to avoid the need to carry along information on an increasing number of partitions, we could modify the definition of the new partition to be $\mathcal{P}'_k = (\mathcal{P}_0 \setminus R^k) \cup \mathcal{P}''_k(R^k)$. In words, we only maintain the most recently refined partition, and aggregate all other solutions into a single “surrounding region.” With these two refinements we have a version of the NP method similar to [Pichitlamken and Nelson \(2003\)](#).

While we adopt the concept of sampling-based bounds, we believe that there is substantial value in retaining the partitioning. In the remainder of the paper we describe and evaluate our ESB&B algorithm, which combines the partitioning structure of SB&B with the sample-based bounds of NP. ESB&B also allocates solution sampling by looking at the potential of the regions to yield better solutions using another sort of bound, and this is different from either SB&B or NP.

2.2 ESB&B

ESB&B estimates bounds as NP does using the estimated objective function values of the solutions that have been simulated. Specifically, at each iteration k , ESB&B randomly samples a number of feasible solutions; call this set S^k . It also maintains a set \mathfrak{S}^k of all solutions that have been sampled through iteration k . ESB&B simulates the solutions in S^k and computes bounds using their estimated performance. In the next iteration, ESB&B allocates a number of solutions to be sampled from each region in the current partition based on the region’s estimated potential to contain good solutions. This expands Step 2 of SB&B into three sub-steps. And at each iteration, ESB&B chooses the solution with the best estimated performance as the current best solution.

Algorithm ESB&B

- Step 0. Initialization:* Set iteration counter $k = 0$, initial partition and record set $\mathcal{P}_0 = R^0 = \{X\}$, and calculate the bounds $\eta^0(X)$ and $\xi^0(X)$.
- Step 1. Partitioning:* If the record set R^k is a singleton, then set $\mathcal{P}'_k = \mathcal{P}_k$ and go to Step 2. Otherwise construct a partition of the record set, $\mathcal{P}''_k(R^k)$, from which infeasible subregions have been cleaned. Define the new full partition by $\mathcal{P}'_k = (\mathcal{P}_k \setminus R^k) \cup \mathcal{P}''_k(R^k)$. Elements of \mathcal{P}'_k will also be denoted by X^P .
- Step 2. Bounding:*
 - Step 2.1. Solution sampling:* For each region $X^P \in \mathcal{P}''_k(R^k)$, randomly sample ϑ_R solutions. For regions $X^P \in \mathcal{P}_k \setminus R^k$, sample $\theta(X^P)$ solutions (the function $\theta(\cdot)$ is described below). Aggregate all the sampled solutions into a set, S^k . Let $\mathfrak{S}^k = \mathfrak{S}^{k-1} \cup S^k$.
 - Step 2.2. Bound estimation:* Simulate Δn_F observations from each solution in S^k that has not been encountered before, and simulate Δn_A additional observations from each solution that has been encountered before. For all regions $X^P \in \mathcal{P}'_k$, calculate estimates $\eta^{k+1}(X^P)$ and $\xi^{k+1}(X^P)$ for $U(X^P)$ and $L(X^P)$, respectively.

Step 2.3. Sample allocation: Compute the number of solutions to be sampled, $\theta(X^P)$ for all $X^P \in \mathcal{P}'_k$, for the next iteration, based on information in \mathfrak{S}^k .

Step 3. Updating partition and record set: Update the record set $R^{k+1} = \arg \max\{\eta^{k+1}(X^P) : X^P \in \mathcal{P}'_k\}$, and partition $\mathcal{P}_{k+1} = \mathcal{P}'_k$. Set $k = k + 1$ and go to Step 1.

Whenever we terminate the algorithm (usually when some simulation budget is reached), we select as the best solution $\hat{\mathbf{x}}^*$ the one with the maximum cumulative sample average.

Partitioning divides the record set, which is convex and finite, into disjoint subregions, each of which is convex. In our implementation of ESB&B, solution sampling is done using the MIX-D algorithm. See Xu (2009) for a detailed description of the partitioning and solution sampling schemes.

Bound estimation estimates the upper and lower bounds of all regions. Rather than solving bounding problems, as SB&B does, ESB&B uses the estimated objective function values of solutions that have been simulated. To describe how this is done, we first define some notation. Let $n(\mathbf{x})$ denote the number of replications obtained from solution \mathbf{x} through iteration k . The observed performance of \mathbf{x} can be represented as $Y_s(\mathbf{x}) = \mu(\mathbf{x}) + \varepsilon_s(\mathbf{x})$. For fixed \mathbf{x} , we assume that the stochastic noise $\varepsilon_s(\mathbf{x})$ is independent and identically distributed (i.i.d.) for all s , which is true if the index s represents replications. Let

$$\bar{Y}(\mathbf{x}) = \frac{1}{n(\mathbf{x})} \sum_{s=1}^{n(\mathbf{x})} Y_s(\mathbf{x}) = \mu(\mathbf{x}) + \bar{\varepsilon}(\mathbf{x})$$

be the accumulated sample mean of all observations of solution \mathbf{x} for $n(\mathbf{x}) > 0$. For each region $X^P \in \mathcal{P}'_k$, we select the solution in S^k with the greatest accumulated sample average through the current iteration, and use this average as the estimate of the upper and lower bounds:

$$\eta^{k+1}(X^P) = \xi^{k+1}(X^P) = \max_{\mathbf{x} \in X^P \cap S^k} \{\bar{Y}(\mathbf{x})\}.$$

We then choose the region with the greatest estimated upper bound as the record set for next iteration:

$$R^{k+1} = \arg \max_{X^P \in \mathcal{P}'_k} \{\eta^{k+1}(X^P)\}.$$

We also define $\eta^* = \max_{X^P \in \mathcal{P}'_k} \{\eta^{k+1}(X^P)\}$ as the greatest estimated upper bound, which is also the estimated objective function value of the current best solution. Notice that these simple sampling-based bounds can be used for any problem. However, problem-specific bounds that satisfy Assumption 2 can be used to improve performance.

Step 2.3 in ESB&B assesses the potential of each region and uses this information to guide sampling. This step improves performance of the algorithm without affecting convergence. Before going into the details of the sample allocation step, we state the convergence of ESB&B algorithm in the following theorem (refer to Xu (2009) for proof).

Theorem 1. Denote by X^* the solution set of (1). With probability one there exists an iteration number k_0 such that for all $k \geq k_0$, the record sets R^k are singletons and $R^k \subset X^*$.

3 ESTIMATING PARTITION POTENTIAL FOR SAMPLE ALLOCATION

In step 2.3 of ESB&B we assess the potential of each region $X^P \in \mathcal{P}'_k \setminus R^{k+1}$, and allocate a total of ϑ_O solution samples among these regions based on their potential, giving more solutions, on average, to regions with greater potential. This step improves algorithm performance. The particular scheme used in this step does not affect convergence as long as each solution has a non-zero probability of being sampled.

We assess “potential” using one of two approaches: either by estimating an upper confidence bound (UCB) on the true value of the best solution in the region, or by estimating the probability that a solution with value better than the current sample best solution is in the region. For each of the two approaches, we also have two ways to produce the estimate: Using either an extension of the empirical Chebyshev bound of Saw, Yang, and Mo (1984), or by assuming the objective function values in the region are approximately normally distributed. The empirical

Chebyshev bound is provably correct for deterministic problems and was extended by us to noisy problems under certain restrictions on the distribution of the noise. The normal-distribution bound is always an approximation. In this paper we simply state how the bounds are calculated and used; derivations can be found in [Xu \(2009\)](#).

First we define some additional notation. For a generic region $X^P \in \mathcal{D}'_k$, let $m = |X^P \cap \mathcal{G}^k|$ be the total number of solutions in the region that have been sampled and simulated through iteration k . Let $\bar{Y} = m^{-1} \sum_{\mathbf{x} \in X^P \cap \mathcal{G}^k} \bar{Y}(\mathbf{x})$ denote the sample mean of all solutions in the region, $S_Y^2 = (m-1)^{-1} \sum_{\mathbf{x} \in X^P \cap \mathcal{G}^k} (\bar{Y}(\mathbf{x}) - \bar{Y})^2$ denote the sample variance of the estimates, and let $S_P^2 = v^{-1} \sum_{\mathbf{x} \in X^P \cap \mathcal{G}^k} \sum_{s=1}^{n(\mathbf{x})} (Y_s(\mathbf{x}) - \bar{Y}(\mathbf{x}))^2$ be the pooled sample variance within the region, where $v = \sum_{\mathbf{x} \in X^P \cap \mathcal{G}^k} (n(\mathbf{x}) - 1)$. Also define the effective degrees of freedom n^* such that $1/n^* = m^{-2} \sum_{\mathbf{x} \in X^P \cap \mathcal{G}^k} 1/n(\mathbf{x})$. Notice that m , \bar{Y} , S_Y^2 , S_P^2 , v , and n^* are statistical measures of a specific region X^P at iteration k . We omit k for simplicity.

We compute $\theta(X^P)$, the number of solution samples allocated to region X^P , using the two methods described in the following paragraphs. These methods apply to stochastic problems; readers interested in deterministic problems can refer to [Xu \(2009\)](#) for adjustments.

3.1 UCB-Based Sample Allocation

The number of solutions to be sampled for all regions, $\theta = \{\theta(X^P)\}$, is the outcome of a multinomial distribution with parameters ϑ_O and φ . The probability vector φ is generated by assigning a probability $\varphi(X^P)$ to each region X^P . This probability is in turn determined by the UCB of the region X^P . A region with greater UCB tends to be assigned more solution samples. The outline of the algorithm is as follows.

Algorithm SA-UCB

- Step 1* For each $X^P \in \mathcal{D}'_k \setminus R^{k+1}$, compute the UCB $\zeta(X^P)$.
- Step 2* Let $\zeta_{\min} = \min_{X^P \in \mathcal{D}'_k \setminus R^{k+1}} \zeta(X^P)$, $X^P_{\min} = \arg \min \{\zeta(X^P) : X^P \in \mathcal{D}'_k \setminus R^{k+1}\}$, and $T_\zeta = \sum_{X^P \in \mathcal{D}'_k \setminus R^{k+1}} (\zeta(X^P) - \zeta_{\min})$.
- Step 3* Assign a probability $\varphi(X^P_{\min}) = \varepsilon$, where ε is a small positive number, and $\varphi(X^P) = \frac{\zeta(X^P) - \zeta_{\min}}{T_\zeta} (1 - \varepsilon)$ for all other X^P .
- Step 4* Let $\varphi = \{\varphi(X^P)\}$ be the vector of $\varphi(X^P)$. Draw a sample, $\theta = \{\theta(X^P)\}$, from a multinomial distribution with parameters ϑ_O and φ . Assign $\theta(X^P)$ samples to region X^P .

We choose ε such that the probability assigned to region X^P_{\min} is half the probability assigned to the region with the second smallest UCB.

We implement the following two different types of UCB, and compare their performance in the numerical study section. Notice that the bounds and statistical measures are all for a specific region X^P at iteration k , which we omit for simplicity.

3.1.1 Chebyshev Bound

The UCB is

$$\zeta = \bar{Y} + \lambda S_Y \sqrt{1 + \frac{1}{m}} + \frac{t_{1-\alpha, v} S_P}{\sqrt{n^*}},$$

where λ is computed from the empirical Chebyshev Inequality of [Saw, Yang, and Mo \(1984\)](#), which implies λ solves

$$\left\lfloor \frac{(n+1)(n-1+\lambda^2)}{n\lambda^2} \right\rfloor = \alpha_0.$$

Also $t_{1-\alpha, v}$ is the $(1 - \alpha)$ -quantile of the t distribution with v degrees of freedom.

3.1.2 Normal Bound

The UCB is

$$\zeta = \bar{Y} + t_{1-\alpha/2, m-1} S_Y \sqrt{1 + \frac{1}{m} + \frac{t_{1-\alpha/2, v} S_p}{\sqrt{n^*}}}.$$

It is worth mentioning that the bounds above do not account for the situation where all the solutions in the region have been sampled. In such cases we compute the bound as follows:

$$\zeta = \max_{\mathbf{x} \in \mathcal{X}^P \cap \mathcal{S}^k} \left\{ \bar{Y}(\mathbf{x}) + t_{1-\alpha, n(\mathbf{x})-1} \frac{S_p}{\sqrt{n(\mathbf{x})}} \right\}.$$

We use $\alpha = 0.05$ and $\alpha_0 = 0.2$ in the implementation.

3.2 Probability-Based Sample Allocation

We compute the number of samples for all regions, $\theta = \{\theta(\mathcal{X}^P)\}$, as the outcome of sampling a multinomial distribution with parameters ϑ_O and φ . The probability vector φ is generated by assigning a probability $\varphi(\mathcal{X}^P)$ to each region \mathcal{X}^P . This probability is a function of the estimated probability that \mathcal{X}^P contains a solution with objective function value better than the estimated objective function value of current best solution, η^* . A region with higher probability of containing better solution tends to be assigned more solution samples. The outline of the algorithm is as follows.

Algorithm SA-Pr

- Step 1 For all $\mathcal{X}^P \in \mathcal{D}'_k \setminus R^{k+1}$, compute a probability $p(\mathcal{X}^P) = P[\mu(\mathbf{x}_{m+1}) > \eta^*]$.
- Step 2 Let $T_p = \sum_{\mathcal{X}^P \in \mathcal{D}'_k \setminus R^{k+1}} p(\mathcal{X}^P)$. Assign a probability $\varphi(\mathcal{X}^P) = p(\mathcal{X}^P)/T_p$ for all \mathcal{X}^P .
- Step 3 Let $\varphi = \{\varphi(\mathcal{X}^P)\}$ be the vector of $\varphi(\mathcal{X}^P)$. Draw a sample, $\theta = \{\theta(\mathcal{X}^P)\}$, from a multinomial distribution with parameters ϑ_O and φ . Assign $\theta(\mathcal{X}^P)$ samples to region \mathcal{X}^P .

We implement the following two probability estimates and compare their performance in the numerical study section. Notice that the probabilities and statistical measures are all for a specific region \mathcal{X}^P at iteration k , which we omit for simplicity. In each case we find p that solves the equation.

3.2.1 Chebyshev Probability

Let

$$p = \frac{\lfloor \frac{(n+1)(n-1+\lambda^2)}{n\lambda^2} \rfloor}{n+1}, \quad \text{where} \quad \lambda = \frac{\eta^* - \bar{Y}}{S_Y \sqrt{1 + \frac{1}{m}}}.$$

3.2.2 Normal Probability

Find α_1, α_2 that solve

$$t_{1-\alpha_1, m-1} = t_{1-\alpha_2, v} = \frac{\eta^* - \bar{Y}}{S_Y \sqrt{1 + \frac{1}{m} + \frac{S_p}{\sqrt{n^*}}}},$$

and let $p = \alpha_1 + \alpha_2$.

Similar to the bound-based sample allocation, the probabilities above do not account for the situation where all the solutions in the region have been sampled. In such cases we compute the probability as follows: for all

Table 1: Parameters used in the numerical experiments.

Experiment	# of Trials	ω	Δn_F	Δn_A	ϑ_R	ϑ_O	ϑ
Miller & Shaw Figure 1	200	3	1	1	10	10	10
Miller & Shaw Figure 2	200	3	1	1	10	20	10
Miller & Shaw Figure 3	200	3	1	1	20	20	10
Miller & Shaw Figure 4	200	3	1	1	10	20	10
Miller & Shaw Figure 5	200	3	10	2	10	20	10
Bowl Table 2	100	3	1	1	30	30	15
Buffer Allocation Figure 6	100	2	4	1	10	10	5

$\mathbf{x} \in X^P \cap \mathcal{G}^k$, compute $\alpha(\mathbf{x})$ that solves

$$t_{1-\alpha(\mathbf{x}),n(\mathbf{x})-1} = \frac{\eta^* - \bar{Y}(\mathbf{x})}{S_P / \sqrt{n(\mathbf{x})}},$$

and let $p = \max\{\alpha(\mathbf{x}) : \mathbf{x} \in X^P \cap \mathcal{G}^k\}$.

4 EVALUATION

In this section, we compare the performance of ESB&B with the NP algorithm of [Pichitlamken and Nelson \(2003\)](#) on three test problems: Miller and Shaw, bowl with flexible dimension, and buffer allocation. We consider different sample allocation schemes for ESB&B: Chebyshev probability-based (Cpr), normal probability-based (Npr), Chebyshev bound-based (Cbd), and normal bound-based (Nbd). To compare different algorithms, we measure their performance against simulation effort (the number of simulation replications), rather than iterations. In each of the following subsections, we first characterize the test problem, and then give results and observations. Parameters used in the numerical experiments are summarized in Table 1 (number of trials, number of subregions for partition (ω), number of simulations for newly sampled solutions (Δn_F), number of simulations for previously sampled solutions (Δn_A), number of samples for subregions of record set for ESB&B (ϑ_R), total number of samples for all the other regions for ESB&B (ϑ_O), and number of solutions per region for NP (ϑ)).

4.1 Miller and Shaw Problem

This test problem is a modification of the multimodal function F_2 used in [Miller and Shaw \(1995\)](#). We rescale F_2 and add up two copies of it to make the problem 2-dimensional:

$$\mu_1(x_1, x_2) = \frac{\sin^6(0.05\pi x_1)}{2^2 \left(\frac{x_1-10}{80}\right)^2} + \frac{\sin^6(0.05\pi x_2)}{2^2 \left(\frac{x_2-10}{80}\right)^2} \tag{2}$$

$$0 \leq x_i \leq 100, \quad x_i \in \mathbb{Z}, \quad i = 1, 2.$$

This problem has a global optimum (10, 10) with objective value 2. The response surface is bumpy, with 25 local optima. Normally distributed noise with zero mean and standard deviation 0.3 is added to (2) to make the problem stochastic. We study both the deterministic and stochastic versions of the problem.

First we examine the effect of the number of solutions per region (ϑ_R solutions for each subregion of record set, and a total of ϑ_O solutions for all the other regions) on the relative performance of ESB&B compared with NP for the deterministic problem. We look at two sample allocation schemes here, Cpr and Npr. We fix the number of samples per region at 10 for NP. Figure 1 shows the objective function value of the estimated optimal solution at each time point (averaged over 200 searches) for $\vartheta_R = 10$ and $\vartheta_O = 10$. Figure 2 is for $\vartheta_R = 10$ and $\vartheta_O = 20$, and Figure 3 is for $\vartheta_R = 20$ and $\vartheta_O = 20$. From these three figures we can see that ESB&B with both Cpr and Npr sample allocation schemes has better performance compared with NP as ϑ_O increases. The intuition is that,

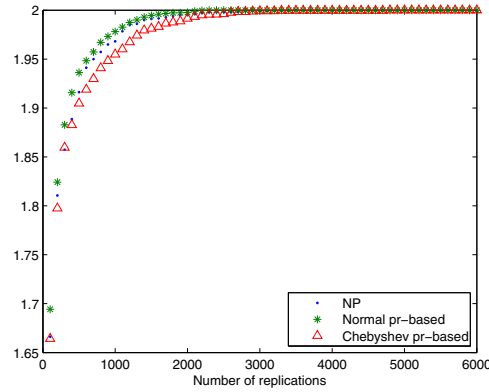


Figure 1: Objective value of current optimal estimate at each time point for the Miller & Shaw problem: $\vartheta_R = 10$ and $\vartheta_O = 10$.

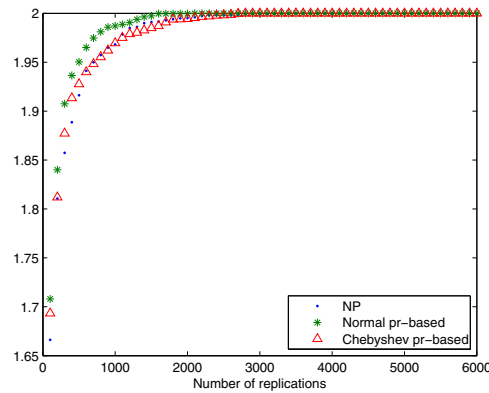


Figure 2: Objective value of current optimal estimate at each time point for the Miller & Shaw problem: $\vartheta_R = 10$ and $\vartheta_O = 20$.

when a “good” region (which contains good solutions) happens to appear inferior because of bad samples, enough samples from the region for the following iterations allows the algorithm to correct the mistake.

Next, we study the performance of different sample allocation methods. Figure 4 shows the performance of ESB&B compared with NP for Npr, Cpr, Nbd, and Cbd. It suggests that normal probability-based sample allocation tops all the other methods.

We then examine the performance of different sample allocation methods with a noisy objective function. Figure 5 demonstrates the advantage of the normal probability-based method over the others. It can also be seen that ESB&B has an even greater advantage over NP when the problem is stochastic, for all types of sample allocation schemes.

4.2 Bowl Problem with Flexible Dimension

This test problem is designed to illustrate the impact of dimension, and the interaction between dimensions. It is formulated as follows:

$$\begin{aligned} \mu_2(x_1, x_2, \dots, x_d) &= 1000 \exp \{ -0.001 (x_1, x_2, \dots, x_d)' \Sigma^{-1} (x_1, x_2, \dots, x_d) \} \\ -\frac{m^{1/d}}{2} &\leq x_i \leq \frac{m^{1/d}}{2}, \quad x_i \in \mathbb{Z}, \quad i = 1, 2, \dots, d, \end{aligned} \tag{3}$$

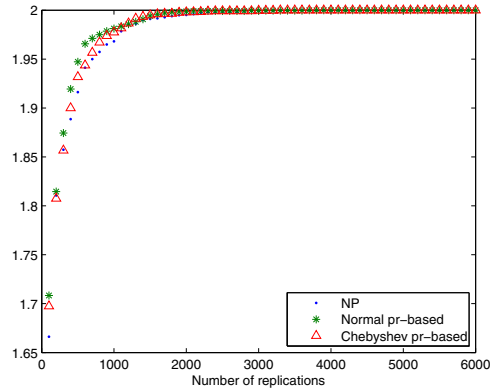


Figure 3: Objective value of current optimal estimate at each time point for the Miller & Shaw problem: $\vartheta_R = 20$ and $\vartheta_O = 20$.

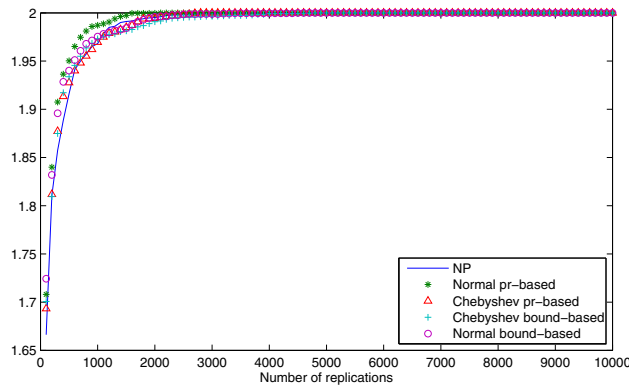


Figure 4: Objective value of current optimal estimate at each time point for the Miller & Shaw problem: all sample allocation methods, deterministic case.

where $m = 20,000$ is the total number of feasible solutions, d is dimension, and Σ is a d -by- d matrix that determines the correlation between different dimensions of decision variable. This problem has a surface shaped like the probability density function of a multivariate normal distribution. It has a single global optimum $(0, 0, \dots, 0)$ with objective value 1000. The feasible region is a hyperbox, where the bounds are rounded to the nearest integer if necessary. Defining the feasible region this way keeps the number of feasible solutions (nearly) the same as dimension changes, allowing us to isolate the impact of dimension from that of the number of feasible solutions. To make the problem stochastic a normally distributed noise with standard deviation $0.3 \times |\mu_2(\mathbf{x})|$ is added to (3). This problem is similar to the test problem in [Xu, Hong, and Nelson \(2010\)](#).

We use the correlation matrix

$$\Sigma = \begin{bmatrix} 1 & \rho & \cdots & \rho & \rho \\ \rho & 1 & \cdots & \rho & \rho \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho & \rho & \cdots & 1 & \rho \\ \rho & \rho & \cdots & \rho & 1 \end{bmatrix},$$

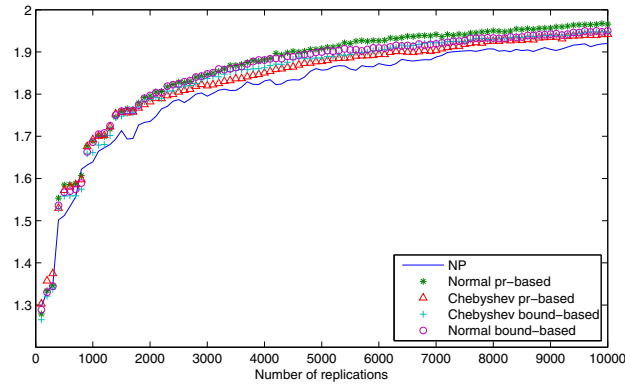


Figure 5: True value of current optimal estimate at each time point for the Miller & Shaw problem: all sample allocation methods, stochastic case.

Table 2: Number of simulation replications to find the optimal solution for the bowl problem with flexible dimension.

	$d = 2$			$d = 3$			$d = 4$		
	$\rho = 0$	$\rho = 0.5$	$\rho = 0.9$	$\rho = 0$	$\rho = 0.5$	$\rho = 0.9$	$\rho = 0$	$\rho = 0.5$	$\rho = 0.9$
NP	3200	3900	9600	3400	20100	18300	2600	4100	25900
NPr	1100	1800	1000	2500	3000	7000	1700	3000	11500
NPr – NP	-2100	-2100	-8600	-900	-17100	-11300	-900	-1100	14400
% difference	-66%	-54%	-90%	-26%	-85%	-62%	-35%	-27%	-56%
CPr	1500	2100	1200	3700	4100	5900	2100	4100	15800
CPr – NP	-1700	-1800	-8400	300	-16000	-12400	-500	0	-10100
% difference	-53%	-46%	-88%	9%	-80%	-68%	-19%	0%	-39%

where ρ is the common correlation coefficient. A larger ρ indicates higher correlation among dimensions, which in this case makes the problem less separable. This makes it harder to search for the optimum for both ESB&B and NP, since both algorithms partition by dimension.

Here we only report results for the deterministic problem, for which both ESB&B and NP stay at the optimum once it is found. We thus use the number of iterations to find the optimum as the performance measure (averaged over 100 searches, recorded every 100 iterations). Table 2 lists the performance measure for NP, ESB&B with Npr and ESB&B with Cpr, the absolute and percentage difference between NP and ESB&B-Npr, and NP and ESB&B-Cpr. The numbers reveal two trends: 1) The advantage of ESB&B over NP increases in ρ , but 2) this increasing trend becomes unclear as dimension increases. We can explain these trends as follows: under higher correlation, both algorithms tend to make more mistakes in selecting a good region. ESB&B hence outperforms NP as it is able to jump directly from region to region, while NP has to backtrack and then search all the way down. However, this trend gets dominated by dimension as dimension increases.

4.3 Three-Stage Buffer Allocation Problem

This test problem is to find the optimal design of a three-stage flow line with finite buffer storage spaces in front of stations 2 and 3. Each station h has a single server, whose service time is exponentially distributed with rate μ_h , $h = 1, 2, 3$. There are an infinite number of jobs in front of station 1. If the buffer in front of station h is full, station $h - 1$ is blocked. The goal is to find the service rate for all stations, as well as the buffer space before stations 2 (b_2) and 3 (b_3), to maximize the throughput of the line. The total buffer spaces and service rate is limited by the

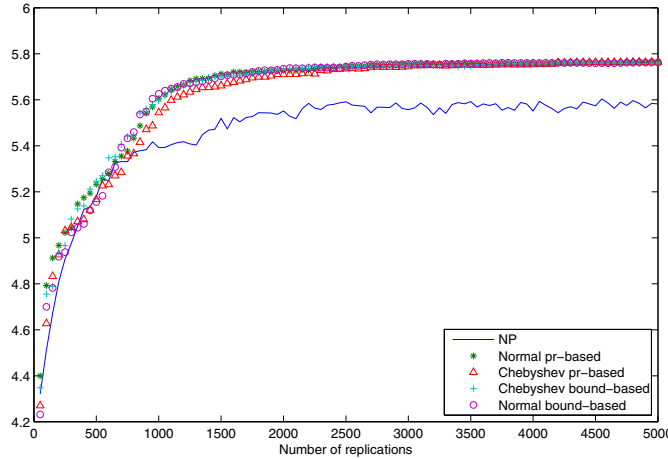


Figure 6: True value of current optimal estimate at each time point for 3-stage buffer allocation problem.

following constraints:

$$\begin{aligned}
 \mu_1 + \mu_2 + \mu_3 &\leq 20 \\
 b_2 + b_3 &\leq 20 \\
 -b_2 - b_3 &\leq -20 \\
 1 \leq \mu_h &\leq 20, \quad h = 1, 2, 3 \\
 1 \leq b_h &\leq 20, \quad h = 2, 3 \\
 \mu_h, b_h &\in \mathbb{Z}.
 \end{aligned}$$

The number of feasible solutions is 21,660. There are two optimal solutions: $(\mu_1, \mu_2, \mu_3, b_2, b_3) = (6, 7, 7, 12, 8)$ and $(7, 7, 6, 8, 12)$ with expected throughput of 5.776 (the optima are obtained from the balance equations of the underlying Markov chain; see [Buzacott and Shantikumar \(1993\)](#)). To reduce the initial condition bias, the throughput is estimated after the first 2000 units have been produced, and it is averaged over the subsequent 50 units released. We sample 5 solutions per region for NP, and set $\vartheta_R = \vartheta_O = 10$ for ESB&B. Figure 6 depicts the true performance of the current optimal estimate at each time point, averaged over 100 searches. It indicates that ESB&B has an advantage over NP, with normal probability-based and Chebyshev bound-based sample allocation methods performing the best.

5 CONCLUSIONS

In this paper we proposed an Empirical Stochastic Branch-and-Bound algorithm, which keeps the partition structure of the Stochastic Branch-and-Bound algorithm, while estimating bounds based on sampling. The algorithm uses the estimated performance of the observed best solution to guide searching; and computes empirical bounds, which indicate the potential of regions, to control sampling. This research provides a framework to apply SB&B where there is no solvable bounding problem.

The ESB&B algorithm converges asymptotically to the global optimum. Numerical study shows that ESB&B outperforms NP in general. The advantage is maximized when the problem is noisy, or there is significant interaction between different decision variables. A normal probability-based sample allocation scheme exhibits the most potential.

To make this framework more adaptive, we can balance the effort in sampling and simulation by adjusting the following four parameters: number of samples for current best region, number of samples for other regions, initial number of simulations for new samples, and incremental number of simulations for re-sampled solutions. We can also adjust the number of subregions and the number of samples as the algorithms goes along and regions become

smaller. A combination of different empirical bounds, at different stages of searching, is a possible direction, as well as designing new empirical bounds to guide sampling.

ACKNOWLEDGMENTS

This work was partially supported by a grant from General Motors R&D.

REFERENCES

- Andradottir, S. 1998. Simulation optimization. In *Handbook of simulation*, ed. J. Banks, Chapter 9. New York: Wiley-Interscience.
- Buzacott, J. A., and J. G. Shantikumar. 1993. *Stochastic models of manufacturing systems*. New Jersey: Princeton Hall.
- Doerner, K., W. J. Gutjahr, G. Kotsis, M. Polaschek, and C. Strauss. 2006. Enriched workflow modeling and stochastic branch-and-bound. *European Journal of Operations Research* 175:1798–1817.
- Fu, M. C. 2002. Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing* 14 (3): 192–215.
- Fu, M. C., F. W. Glover, and J. April. 2005. Simulation optimization: a review, new developments, and applications. In *Proceedings of the 2005 Winter Simulation Conference*, 83–95. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Gutjahr, W. J., A. Hellmayr, and G. C. Pflug. 1999. Optimal stochastic single-machine-tardiness scheduling by stochastic branch-and-bound. *European Journal of Operations Research* 117:396–413.
- Gutjahr, W. J., C. Strauss, and E. Wagner. 2000. A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing* 12 (2): 125–135.
- Miller, B. L., and M. J. Shaw. 1995. Generic algorithms with dynamic niche sharing for multimodal function optimization. Technical Report No. 95010, Illinois Genetic Algorithms Laboratory, University of Illinois of Champaign-Urbana, Champaign, Illinois.
- Norkin, V. I., Y. M. Ermoliev, and A. Ruszczyński. 1998. On optimal allocation of indivisibles under uncertainty. *Operations Research* 46 (3): 381–395.
- Norkin, V. I., G. C. Pflug, and A. Ruszczyński. 1998. A branch and bound method for stochastic global optimization. *Mathematical Programming* 83:425–450.
- Pichitlamken, J., and B. L. Nelson. 2003. A combined procedure for optimization via simulation. *The ACM Transactions on Modeling and Computer Simulation* 13:155–179.
- Saw, J. G., M. C. K. Yang, and T. C. Mo. 1984. Chebyshev inequality with estimated mean and variance. *The American Statistician* 38 (2): 130–132.
- Shi, L., and S. Olafsson. 2000. Nested partition method for global optimization. *Operations Research* 48 (3): 390–407.
- Xu, J., L. J. Hong, and B. L. Nelson. 2010. Industrial strength compass: a comprehensive algorithm and software for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 20:1–29.
- Xu, W. L. 2009. *Flexibility, lifecycle planning and simulation-based optimization in integrated supply chains*. Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois.

AUTHOR BIOGRAPHIES

WENDY LU XU is currently a post-doctoral research fellow at ExxonMobil Corporate Strategic Research. Her research interest includes product portfolio management, supply chain management, and simulation optimization. Her address is 1545 US Route 22E, EMRE-CLN-LC272, Annandale, NJ 08801, USA. Her email address is lu.xu@exxonmobil.com.

BARRY L. NELSON is the Charles Deering McCormick Professor and Chair of the Department of Industrial Engineering and Management Sciences at Northwestern University. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems. His e-mail and web addresses are nelsonb@northwestern.edu and www.iems.northwestern.edu/~nelson/.