This article was downloaded by: [Professor Barry Nelson]
On: 12 June 2013, At: 09:24
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

IIE Transactions
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uiie20

Empirical stochastic branch-and-bound for optimization via simulation

Wendy Lu Xu [a] & Barry L. Nelson [a]
[a] Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL, 60208-3119, USA
Published online: 10 Apr 2013.

To cite this article: Wendy Lu Xu & Barry L. Nelson (2013): Empirical stochastic branch-and-bound for optimization via simulation, IIE Transactions, 45:7, 685-698

To link to this article: http://dx.doi.org/10.1080/0740817X.2013.768783

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

IIE Transactions (2013) **45**, 685–698
Copyright © "IIE"
ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2013.768783

Empirical stochastic branch-and-bound for optimization via simulation

WENDY LU XU* and BARRY L. NELSON

Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, IL 60208-3119, USA
E-mail: luxu2009@u.northwestern.edu

Received June 2011 and accepted January 2013

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

This article introduces a new method for discrete decision variable optimization via simulation that combines the nested partitions method and the stochastic branch-and-bound method in the sense that advantage is taken of the partitioning structure of stochastic branch-and-bound, but the bounds are estimated based on the performance of sampled solutions, similar to the nested partitions method. The proposed Empirical Stochastic Branch-and-Bound (ESB&B) algorithm also uses improvement bounds to guide solution sampling for better performance. A convergence proof and empirical evaluation are provided.

[Supplementary materials are available for this article. Go to the publisher's online edition of *IIE Transaction* for datasets, additional tables, detailed proofs, etc.]

Keywords: Stochastic branch-and-bound, optimization via simulation

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 *et al.* (2005) give extensive overviews of the OvS literature.

Most algorithms for Discrete decision variable Optimization via Simulation (DOvS) are based on adaptive random search. Hu *et al.* (2007) classified random search methods as being either model-based or instance-based approaches. Model-based methods, such as model-based annealing random search by Hu and Hu (2010), generate new solutions via an intermediate probabilistic model that is updated or induced from previous solutions. Instance- or sampling-based methods, on the other hand, search for new solutions depending directly on previously sampled solutions. Our algorithm is of the latter type. A partitioning approach, like ours, can assess the promise of a subregion by accounting for how much of the subregion we have explored and how homogeneous and noisy the subregion appears to be, without the need to define a distribution over all solutions in the entire feasible region. A partitioning approach can be used for general DOvS problems without significant amounts of customization but can also exploit a good partitioning scheme if we know it. 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 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 (B&B) algorithms are widely used to solve deterministic integer optimization problems. Norkin, Ermoliev, and Ruszczynski (1998) and Norkin, Pflug, and Ruszczynski (1998) adapted 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 *et al.* (1999, 2000) and

*Corresponding author

0740-817X © 2013 "IIE"

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

Doerner *et al.* (2006). Unfortunately, for complicated DOvS problems there may not be solvable bounding problems.

Our Empirical Stochastic Branch-and-Bound (ESB&B) algorithm combines SB&B and NP in the sense that it takes advantage of the partitioning structure of SB&B but estimates bounds based on the performance of sampled solutions, as in NP. Using sampling-based bounds allows our algorithm to apply even when solvable bounding problems are not available and hence makes SB&B more practical. Although it incurs some computational overhead, maintaining a partition structure has advantages for problems with a very bumpy or noisy response surface. In addition, ESB&B employs Upper Confidence Bounds (UCBs) on the potential of subregions created by branching to actually contain better solutions. To the best of our knowledge, these UCBs are new, and we use them to guide solution sampling for better finite-time performance without sacrificing global convergence. This is a central contribution of the article.

The concept of a UCB is seen in the multi-armed bandit literature. In a multi-armed bandit problem, play of each arm yields a reward with an unknown underlying probability distribution. The goal is to choose which arm to play sequentially to maximize the long-run expected sum of rewards. As described in Lai and Robbins (1985) and Auer *et al.* (2002), a UCB is calculated to represent the knowledge of each arm's expected reward so far, evaluates each arm's potential of being the best, and helps decide which arm to play next or, put in the DOvS context, how to allocate the simulation budget. Our UCB, applied in a partitioning-based DOvS algorithm, is different in the sense that it estimates the potential of a particular subregion of the feasible region to contain better solutions, rather than the potential of a particular solution, and is used to guide solution sampling across subregions.

The article is organized as follows. Section 2 reviews SB&B and NP, presents ESB&B, and states the main convergence theorem. Section 3 describes how ESB&B uses statistical bounds to guide solution sampling. Section 4 analyzes a simple problem to illustrate the behavior of ESB&B, as well as displaying experiment results, and Section 5 concludes the article. Details not found in this article are available in Xu (2009).

2. ESB&B

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

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

where X is defined as $\mathrm{X} = \mathsf{X} \bigcap \mathsf{D}$, X is the intersection of the integer lattice with a hypercube in $\mathbb{R}^q$, given by

$$l_i \le x_i \le u_i, \qquad l_i, x_i, u_i \in \mathbb{Z}, \qquad i = 1, 2, \ldots, q,$$

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

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

We assume that D is convex and that there are only a finite number of feasible solutions.

In this article 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. The observed performance of $\mathbf{x}$ on replication s is 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 for all s, which is true when the index s represents replications. Throughout this article we use the term *sample* to mean randomly choosing solutions $\mathbf{x}$ and *simulate* to mean observing $Y(\mathbf{x})$. Like the NP method, ESB&B can also be applied to deterministic problems; the corresponding adjustments can be found in Xu (2009).

2.1. *SB&B and NP*

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

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

Then clearly the optimal value of Problem (1) equals $\mu^*(\mathrm{X}) = \max_{\mathrm{X}^P \in \mathcal{P}} \mu^*(\mathrm{X}^P)$. Norkin, Ermoliev, and Ruszczynski (1998) make the following assumptions.

Assumption 1. For each subregion $\mathrm{X}^P \subseteq \mathrm{X}$, there exist functions L: $2^{\mathrm{X}} \to \mathbb{R}$ and U: $2^{\mathrm{X}} \to \mathbb{R}$ such that

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

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

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 $\mathrm{X}^P \subseteq \mathrm{X}$, there exist sequences of estimators η^k and ξ^k, $k = 1, 2, \ldots$, such that $\lim_{k\to\infty} \eta^k(\mathrm{X}^P) = U(\mathrm{X}^P)$ a.s. and $\lim_{k\to\infty} \xi^k(\mathrm{X}^P) = L(\mathrm{X}^P)$ a.s.

The SB&B algorithm works as follows: At each iteration from among all of the subregions of the current partition, the subregion with the greatest (estimated) upper bound is selected as the record set. 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. When the algorithm is terminated (by some stopping criterion), the approximate solution of the current iteration is selected as the best solution. 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 $\mathcal{P}_0 = \{\mathrm{X}\}$, and record set $R^0 = \mathrm{X}$ and calculate the bounds $\eta^0(\mathrm{X})$ and $\xi^0(\mathrm{X})$.

Step 1. Partitioning: Select an approximate solution $\mathbf{x}^k \in \mathrm{X}^k = \arg\max\{\xi^k(\mathrm{X}^P) : \mathrm{X}^P \in \mathcal{P}_k\}$. 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, $\mathcal{P}''_k(R^k)$. Define the new full partition by $\mathcal{P}'_k = (\mathcal{P}_k \backslash \{R^k\}) \bigcup \mathcal{P}''_k(R^k)$. Elements of $\mathcal{P}'_k$ will also be denoted by X^P.

Step 2. Bounding: For all subregions $\mathrm{X}^P \in \mathcal{P}'_k$ calculate estimates $\eta^{k+1}(\mathrm{X}^P)$ and $\xi^{k+1}(\mathrm{X}^P)$ for $U(\mathrm{X}^P)$ and $L(\mathrm{X}^P)$, respectively.

Step 3. Updating partition and record set: Update record set $R^{k+1} = \arg\max\{\eta^{k+1}(\mathrm{X}^P) : \mathrm{X}^P \in \mathcal{P}'_k\}$ 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 need 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. For instance, 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 $\mathcal{P}'_k = \{\mathrm{X} \backslash R^k\} \bigcup \mathcal{P}''_k(R^k)$. In other 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).

We use Fig. 1 to illustrate the difference in partitioning of SB&B and NP. Starting from a partition $\mathcal{P}_k = \{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3\}$, suppose the record set is $R^k = \mathrm{X}_3$, and the partition of the record set $\mathcal{P}''_k(R^k) = \{\mathrm{X}_4, \mathrm{X}_5, \mathrm{X}_6\}$. In the SB&B algorithm, the new full partition $\mathcal{P}'_k = (\mathcal{P}_k \backslash \{R^k\}) \bigcup \mathcal{P}''_k(R^k) = \{\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_4, \mathrm{X}_5, \mathrm{X}_6\}$, as shown in Fig. 1(a). In NP, $\mathrm{X} \backslash R^k = \mathrm{X}_1 \bigcup \mathrm{X}_2$, and $\mathcal{P}'_k = \{\mathrm{X}_1 \bigcup \mathrm{X}_2, \mathrm{X}_4, \mathrm{X}_5, \mathrm{X}_6\}$, as shown in Fig. 1(b). SB&B maintains a partition structure for X_1 and X_2, subregions that are not within the record set, but NP combines them.

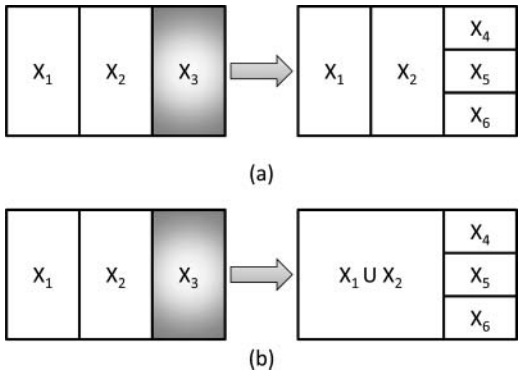

Fig. 1. (a) Partitioning of SB&B and (b) partitioning of NP.

While we adopt the concept of sampling-based bounds, we believe that there is substantial value in retaining the partitions. The sampling-based bounds make SB&B more practical. Also, the partition structure is advantageous especially when the response surface is bumpy or noisy, because an algorithm tends to make more mistakes when selecting the record set by sampling from such a surface, and a partition structure allows the algorithm to quickly recover from mistakes. In the remainder of this article we describe and evaluate our ESB&B algorithm, which combines the partition structure of SB&B with the sample-based bounds of NP. ESB&B also allocates solution sampling by calculating UCBs for subregions. The UCB is another sort of statistical bound that estimates the potential of a subregion to yield even better solutions based on samples seen so far. This guided sampling approach is new for partitioning-based DOvS algorithms and can improve the algorithm's finite-time performance substantially, as shown in our numerical studies.

2.2. *ESB&B*

ESB&B estimates bounds as does NP 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 on subregions using the estimated performance of solutions within the subregion. In the next iteration, ESB&B allocates the number of solutions to be sampled from subregions in the current partition $\mathcal{P}_k$ based on each subregion's estimated potential to contain better 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.

ESB&B algorithm

Step 0. Initialization: Set iteration counter $k = 0$, $\mathfrak{S}^0 = \emptyset$, initial partition $\mathcal{P}_0 = \{\mathrm{X}\}$, record set $R^0 = \mathrm{X}$.

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

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, $\mathcal{P}''_k(R^k)$. Define the new full partition by $\mathcal{P}'_k = (\mathcal{P}_k \backslash \{R^k\}) \bigcup \mathcal{P}''_k(R^k)$. Elements of $\mathcal{P}'_k$ will also be denoted by X^P.

Step 2. Bounding:

2.1. *Solution sampling*: For each subregion $\mathrm{X}^P \in \mathcal{P}''_k(R^k)$, randomly sample ϑ_{R} solutions. If $k > 0$, for subregions $\mathrm{X}^P \in \mathcal{P}_k \backslash \{R^k\}$ sample $\theta(\mathrm{X}^P)$ solutions, where $\theta(\mathrm{X}^P)$ depends on information in $\mathfrak{S}^{k-1}$ and has been computed in Step 2.3 of the last iteration. Aggregate all of the sampled solutions into a set, S^k. Let $\mathfrak{S}^k = \mathfrak{S}^{k-1} \bigcup S^k$.

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 subregions $\mathrm{X}^P \in \mathcal{P}'_k$, calculate estimates $\eta^{k+1}(\mathrm{X}^P)$ and $\xi^{k+1}(\mathrm{X}^P)$ for $U(\mathrm{X}^P)$ and $L(\mathrm{X}^P)$, respectively.

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

Step 3. Updating partition and record set: Update the record set $R^{k+1} = \arg\max\{\eta^{k+1}(\mathrm{X}^P) : \mathrm{X}^P \in \mathcal{P}'_k\}$ 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 the simulation budget is reached), we select as the best solution $\hat{\mathbf{x}}^*$ the one with the maximum cumulative sample average. Notice that this is different from SB&B, which selects the current approximate solution as the best solution. We do not use an approximate solution in ESB&B.

Partitioning divides the record set into disjoint nonempty subregions. In our implementation of ESB&B, solution sampling is done using the MIX-D algorithm (Pichitlamken and Nelson, 2003). See online Appendices A and B for a detailed description of the partitioning and solution sampling schemes.

Bound estimation estimates the upper and lower bounds of all subregions. 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 total number of replications obtained from solution $\mathbf{x}$ through iteration k. As stated in the beginning of Section 2, the observed performance of $\mathbf{x}$ on replication s can be represented as $Y_s(\mathbf{x}) = \mu(\mathbf{x}) + \varepsilon_s(\mathbf{x})$. Let

$$\bar{Y}(\mathbf{x}) = \frac{1}{n(\mathbf{x})} \sum_{s=1}^{n(\mathbf{x})} Y_s(\mathbf{x}) = \mu(\mathbf{x}) + \bar{\varepsilon}(\mathbf{x}) \tag{2}$$

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

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

Notice that $\mathrm{X}^P \cap \mathfrak{S}^k \neq \emptyset$ since whenever a record set is partitioned to generate new subregions, every subregion is sampled. In ESB&B, when the algorithm terminates, we select the solution with the maximum cumulative sample average as the best solution, rather than using the approximate solution from the subregion with the greatest estimated lower bound, as does SB&B. Therefore, the lower bound estimation in ESB&B is not of practical use. While keeping lower bound estimation in ESB&B so that it shares the same structure as SB&B, we adopt the easiest way of estimating the lower bound that satisfies Assumptions 1 and 2, which is setting the lower bound equal to the upper bound and both equal to the greatest sample mean.

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

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

We also define $\eta^* = \max_{\mathrm{X}^P \in \mathcal{P}'_k} \{\eta^{k+1}(\mathrm{X}^P)\}$ as the value of 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 subregion and uses this information to guide sampling. This step improves performance of the algorithm without affecting convergence under some mild assumptions. Before going into the details of the sample allocation step, we state the convergence of the ESB&B algorithm in the following theorem.

Theorem 1. *Assume* $|\mu(\mathbf{x})| < \infty$ *and* $\mathrm{Var}(\varepsilon(\mathbf{x})) < \infty$ *for all* $\mathbf{x} \in \mathrm{X}$ *and* $|\mathrm{X}| < \infty$. *Denote by* X^* *the solution set of Equation* (1). *If at every iteration every subregion has a positive probability bounded away from zero of being sampled, then 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* $R^k \subseteq X^*$.

Proof. We first prove $\lim_{k\to\infty} \eta^{k+1}(\mathrm{X}^P) = U(\mathrm{X}^P)$ *a.s.* for any given subregion X^P.

At each iteration, every subregion has a positive probability of being sampled that is bounded away from zero. The MIX-D algorithm guarantees that whenever a subregion is chosen for sampling, every solution has a non-zero probability of being selected, and that probability is bounded below by $1/|\mathrm{X}| > 0$ (Pichitlamken, 2002). Therefore, with probability one, all solutions are sampled;

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

i.e., $\lim_{k\to\infty} \mathrm{X}^P \cap \mathfrak{S}^k = \mathrm{X}^P$ *a.s.* Since X^P is finite, for almost all sample paths ω, there exists $K_1(\omega) > 0$ such that $\mathrm{X}^P \cap \mathfrak{S}^k = \mathrm{X}^P$ for all $k > K_1(\omega)$.

Since we add more simulations for every sampled solution for every iteration, every solution is simulated infinitely many times as k goes to infinity. Also, since we use the cumulative sample mean to evaluate the value of each solution, by the Strong Law of Large Numbers, for all $\mathbf{x} \in \mathrm{X}$, $\lim_{k\to\infty} \bar{Y}(\mathbf{x}) = \mu(\mathbf{x})$ *a.s.* This means that for almost all sample paths ω, for any $\mathbf{x}$ and any given $\delta > 0$ there exists a $K(\omega, \mathbf{x}, \delta) > 0$ such that $|\bar{Y}(\mathbf{x}) - \mu(\mathbf{x})| < \delta$ for all $k > K(\omega, \mathbf{x}, \delta)$.

Assume that all $\mathbf{x} \in \mathrm{X}^P$ have in total $N \le |\mathrm{X}^P|$ distinct $\mu(\mathbf{x})$ values. Group all $\mathbf{x} \in \mathrm{X}^P$ by their $\mu(\mathbf{x})$ values into N sets $X^1, X^2, \ldots, X^N$ so that $\mu(\mathbf{x}) = \mu(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in X^n$, $n = 1, 2, \ldots, N$. Let $\mu(X^n)$ denote $\mu(\mathbf{x})$ for all $\mathbf{x} \in X^n$. Order these sets in sequence $X^{(1)}, X^{(2)}, \ldots, X^{(N)}$ so that $\mu(X^{(1)}) > \mu(X^{(2)}) > \cdots > \mu(X^{(N)})$. Let $d = 0.5 \min_{n=1,2,\ldots,N-1} \{\mu(X^{(n)}) - \mu(X^{(n+1)})\}$ and $K_2(\omega) = \max_{\mathbf{x}\in\mathrm{X}^P} K(\omega, \mathbf{x}, d)$. Then $\max_{\mathbf{x}\in\mathrm{X}^P} \{\bar{Y}(\mathbf{x})\} = \max_{\mathbf{x}\in X^{(1)}} \bar{Y}(\mathbf{x})$ for all $k > K_2(\omega)$.

For any given $\epsilon > 0$, let $K_3(\omega, \epsilon) = \max_{\mathbf{x}\in\mathrm{X}^P} K(\omega, \mathbf{x}, \epsilon)$. Then for all $k > K_3(\omega, \epsilon)$, $|\bar{Y}(\mathbf{x}) - \mu(\mathbf{x})| < \epsilon$ for all $x \in \mathrm{X}^P$.

Now for any given ϵ and almost all sample paths ω, for all $k > \max\{K_1(\omega), K_2(\omega), K_3(\omega, \epsilon)\}$ we have

$$
\begin{aligned}
&|\eta^{k+1}(\mathrm{X}^P) - U(\mathrm{X}^P)| \\
&\quad = \left| \max_{\mathbf{x}\in\mathrm{X}^P\cap\mathfrak{S}^k} \{\bar{Y}(\mathbf{x})\} - \max_{\mathbf{x}\in\mathrm{X}^P} \mu(\mathbf{x}) \right| \\
&\quad = \left| \max_{\mathbf{x}\in\mathrm{X}^P} \{\bar{Y}(\mathbf{x})\} - \max_{\mathbf{x}\in\mathrm{X}^P} \mu(\mathbf{x}) \right| \qquad \because \mathrm{X}^P \cap \mathfrak{S}^k = \mathrm{X}^P, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k > K_1(\omega) \\
&\quad = \left| \max_{\mathbf{x}\in X^{(1)}} \bar{Y}(\mathbf{x}) - \mu(X^{(1)}) \right| \qquad \because \max_{\mathbf{x}\in\mathrm{X}^P} \{\bar{Y}(\mathbf{x})\} = \max_{\mathbf{x}\in X^{(1)}} \bar{Y}(\mathbf{x}), \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k > K_2(\omega) \\
&\quad < \epsilon. \qquad\qquad\qquad\qquad \because |\bar{Y}(\mathbf{x}) - \mu(\mathbf{x})| < \epsilon,\ k > K_3(\omega, \epsilon)
\end{aligned}
$$

That is, we have

$$\lim_{k\to\infty} \eta^{k+1}(\mathrm{X}^P) = U(\mathrm{X}^P) \text{ a.s.} \tag{2}$$

Next, notice that since $|\mathrm{X}| < \infty$, there can only be a finite number of iterations with partitioning. Therefore, there exists a $K_4(\omega)$ such that for $k \ge K_4(\omega)$ the partition remains unchanged. Denote it as $\mathcal{P}^\infty$. Within this partition all record sets are singletons; otherwise, they would have been further partitioned. Since there are a finite number of record sets, there exists a $K_5(\omega)$ such that for $k \ge K_5(\omega)$ all record sets are recurrent in the sense that they are record sets for infinitely many k. Note that both $K_4(\omega)$ and $K_5(\omega)$ depend on the sample path. Let R denote one such record set, and let $\{k'_j\}$ be the iteration indices on which R is chosen as record set. By definition we have

$$\eta^{k'_j}(R) \ge \eta^{k'_j}(\mathrm{X}^P) \quad \forall\, \mathrm{X}^P \in \mathcal{P}^\infty,\ k'_j \ge \max\{K_4(\omega), K_5(\omega)\}. \tag{3}$$

From Equation (2) we have

$$\lim_{k'_j\to\infty} \eta^{k'_j}(R) = U(R) = \mu(R),$$

$$\lim_{k'_j\to\infty} \eta^{k'_j}(\mathrm{X}^P) = U(\mathrm{X}^P),$$

since R is a singleton. Passing to the limit in Inequality (3) we obtain $\mu(R) \ge U(\mathrm{X}^P)$ for all $\mathrm{X}^P \in \mathcal{P}^\infty$, which completes the proof. ■

3. Estimating the partition potential for a sample allocation

In Step 2.3 of ESB&B we compute $\theta(\mathrm{X}^P)$, the number of solutions to be sampled from each subregion that is not a record set $\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}$. In our implementation, $\boldsymbol{\theta} = \{\theta(\mathrm{X}^P),\ \mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}\}$ is a sample from a multinomial distribution with ϑ_O trials and success probabilities $\boldsymbol{\varphi} = \{\varphi(\mathrm{X}^P),\ \mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}\}$. The values $\varphi(\mathrm{X}^P)$ reflect our assessment of the potential of subregion X^P to contain better solutions. This step improves algorithm performance by assigning more samples on average to a subregion with better potential. The particular scheme used in this step does not affect convergence as long as each subregion has a non-zero and bounded probability of being sampled.

To see how we assess the potential of a subregion X^P, suppose that we have sampled $m < |\mathrm{X}^P|$ solutions from this subregion, denoted $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ for convenience. Furthermore, suppose for the moment that they could be evaluated without noise, giving $\mu(\mathbf{x}_1), \mu(\mathbf{x}_2), \ldots, \mu(\mathbf{x}_m)$. Let,

$$\bar{\mu} = \frac{1}{m} \sum_{i=1}^{m} \mu(\mathbf{x}_i)$$

$$S^2 = \frac{1}{m-1} \sum_{i=1}^{m} (\mu(\mathbf{x}_i) - \bar{\mu})^2.$$

Now consider another randomly sampled solution, denoted $\mathbf{x}_{m+1}$, from this subregion. Our assessment of the potential of the subregion is based on being able to claim, either exactly or approximately, that

$$\Pr\left\{ \mu(\mathbf{x}_{m+1}) \ge \bar{\mu} + \lambda(\alpha_0) S \sqrt{1 + \frac{1}{m}} \right\} \le \alpha_0. \tag{4}$$

We compare subregions based either on the relative values of their bounds $\bar{\mu} + \lambda(\alpha_0) S\sqrt{1 + 1/m}$ (with a larger bound meaning more potential) or by fixing their bounds at $\bar{\mu} + \lambda(\alpha_0) S\sqrt{1 + 1/m} = \eta^*$, the estimated objective function value of the current best solution, and solving for α_0 (with a larger α_0 meaning more potential). The empirical Chebyshev inequality of Saw *et al.* (1984) provides a value of $\lambda(\alpha_0)$ that guarantees that Equation (4) holds, while assuming that the values $\mu(\mathbf{x}_i)$ are normally distributed provides an approximate (and tighter) bound where $\lambda(\alpha_0)$

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

is based on the t distribution. It is also possible to use Hoeffding's inequality or Bernstein's inequality to develop a UCB, assuming some kind of bounds for $\mu(\mathbf{x}_i)$ (see, for instance, Li (2009)), which we do not use due to the difficulty of estimating parameters.

In our context, we cannot apply Equation (4) directly since we do not observe $\mu(\mathbf{x}_i)$ but instead observe an estimate $\bar{Y}(\mathbf{x}_i)$ based on $n(\mathbf{x}_i)$ replications. Therefore, the bounds we use are adjusted by assuming that the simulation noise is normally distributed with equal variances ($\mathrm{Var}(\varepsilon_s(\mathbf{x})) = \sigma^2 \,\forall\, \mathbf{x}$); this makes all of our bounds approximate. In this article we simply state how the bounds are calculated and used; detailed derivations can be found in Xu (2009).

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

We compute $\theta(\mathrm{X}^P)$, the number of solution samples allocated to subregion X^P, using the two methods described in the following subsections. These methods apply to stochastic problems; readers interested in deterministic problems can refer to Xu (2009) for corresponding adjustments.

3.1. *UCB-based sample allocation*

In this approach the subregions with greater UCB tend to be assigned more solution samples. The outline of the algorithm is as follows. Let $0 < \epsilon \ll 1$ be a small positive constant we choose to guarantee that the probability of each subregion to be sampled is bounded away from zero.

Algorithm SA-UCB

Step 1. For each $\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}$, compute the UCB $\zeta(\mathrm{X}^P)$, which will be described below.

Step 2. Let $\zeta_{\min} = \min_{\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}} \zeta(\mathrm{X}^P)$, $\mathrm{X}^P_{\min} = \arg\min \{\zeta(\mathrm{X}^P) : \mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}\}$, and $T_\zeta = \sum_{\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}} (\zeta(\mathrm{X}^P) - \zeta_{\min})$.

Step 3. Let ρ be a small positive number and compute preliminary probabilities $p(\mathrm{X}^P_{\min}) = \rho$ and $p(\mathrm{X}^P) = \zeta(\mathrm{X}^P) - \zeta_{\min} \backslash T_\zeta (1-\rho)$ for all other X^P. The value of ρ is chosen dynamically so that the preliminary probability assigned to subregion $\mathrm{X}^P_{\min}$, $p(\mathrm{X}^P_{\min})$, is half the preliminary probability assigned to the subregion with the second-smallest UCB. Let $T_p = \sum_{\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}} \max\{\epsilon, p(\mathrm{X}^P)\}$. Assign a probability $\varphi(\mathrm{X}^P) = \max\{\epsilon, p(\mathrm{X}^P)\}/T_p$ for all X^P.

Step 4. Let $\boldsymbol{\varphi} = \{\varphi(\mathrm{X}^P)\}$ be the vector of $\varphi(\mathrm{X}^P)$. Draw a sample, $\boldsymbol{\theta} = \{\theta(\mathrm{X}^P)\}$, from a multinomial distribution with parameters ϑ_O and $\boldsymbol{\varphi}$. Assign $\theta(\mathrm{X}^P)$ samples to subregion X^P.

Notice that $T_p < |\mathrm{X}| < \infty$; therefore, $\varphi(\mathrm{X}^P) \geq \epsilon / T_p$ is bounded away from zero.

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

3.1.1. *Chebyshev bound*

The UCB is

$$\zeta(\mathrm{X}^P) = \bar{Y} + \lambda S_Y \sqrt{1 + \frac{1}{m}} + \frac{t_{1-\alpha,\nu} S_p}{\sqrt{n^*}},$$

where λ is computed from the empirical Chebyshev Inequality of Saw *et al.* (1984), which implies that λ solves

$$\frac{\lfloor ((m+1)(m-1+\lambda^2))/m\lambda^2 \rfloor}{m+1} = \alpha_0.$$

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

3.1.2. *Normal bound*

The UCB is

$$\zeta(\mathrm{X}^P) = \bar{Y} + t_{1-\alpha/2,m-1} S_Y \sqrt{1 + \frac{1}{m}} + \frac{t_{1-\alpha/2,\nu} S_p}{\sqrt{n^*}}.$$

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

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

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

3.2. *Probability-based sample allocation*

In this approach subregions with a larger probability of containing a solution with an objective function value better than η^*, the estimated objective function value of the current best solution, tend to be assigned more solution samples. The outline of the algorithm is as follows. Let $0 < \epsilon \ll 1$ be a small positive constant we choose to guarantee that the probability of each subregion to be sampled is bounded away from zero.

Algorithm SA-Pr

Step 1. For all $\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}$, compute a probability $p(\mathrm{X}^P) = \Pr\{\mu(\mathbf{x}_{m+1}) > \eta^*\}$.

Step 2. Let $T_p = \sum_{\mathrm{X}^P \in \mathcal{P}'_k \backslash \{R^{k+1}\}} \max\{\epsilon, p(\mathrm{X}^P)\}$. Assign a probability $\varphi(\mathrm{X}^P) = \max\{\epsilon, p(\mathrm{X}^P)\}/T_p$ for all X^P.

Step 3. Let $\boldsymbol{\varphi} = \{\varphi(\mathrm{X}^P)\}$ be the vector of $\varphi(\mathrm{X}^P)$. Draw a sample, $\Theta = \{\theta(\mathrm{X}^P)\}$, from a multinomial distribution with parameters ϑ_O and $\boldsymbol{\varphi}$. Assign $\theta(\mathrm{X}^P)$ samples to subregion X^P.

Notice that $T_p < |\mathrm{X}| < \infty$, therefore $\varphi(\mathrm{X}^P) \geq \epsilon/T_p$ is bounded away from zero.

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 subregion X^P at iteration k, which we omit for simplicity. In each case we find p that solves the given equation.

3.2.1. *Chebyshev probability*

Let

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

The conclusion $\Pr\{\mu(\mathbf{x}_{m+1}) > \eta^*\} \leq p$ follows directly from the empirical Chebyshev Inequality for the deterministic case. We use this probability as an approximation for the stochastic case.

3.2.2. *Normal probability*

We need to find α_1 and α_2 so that

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

To facilitate this we also require $t_{1-\alpha_1,m-1} = t_{1-\alpha_2,\nu}$, and therefore

$$t_{1-\alpha_1,m-1} = t_{1-\alpha_2,\nu} = \frac{\eta^* - \overline{Y}}{S_Y\sqrt{1+(1/m)} + (S_p/\sqrt{n^*})}.$$

Then $p(\mathrm{X}^P) = \alpha_1 + \alpha_2$.

Similar to the bound-based sample allocation, the probabilities above do not account for the situation where all of the solutions in the subregion have been sampled. In such cases we compute the probability as follows: for all $\mathbf{x} \in \mathrm{X}^P \cap \mathfrak{S}^k$, 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 $p(\mathrm{X}^P) = \max\{\alpha(\mathbf{x}) : \mathbf{x} \in \mathrm{X}^P \cap \mathfrak{S}^k\}$.

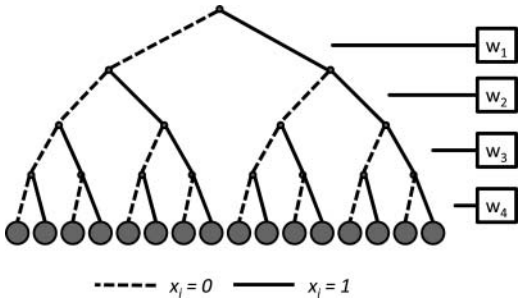

Fig. 2. T-level weighted binary tree ($T = 4$).

4. Evaluation

In this section, we first analyze a stylized optimization problem to shed some light on the behavior of the ESB&B algorithm relative to NP, focusing on the impact of maintaining the partition structure. We then present some experimental results.

4.1. *Illustration: weighted binary tree problem*

Consider the following deterministic problem:

$$\max_{\mathbf{x}} \mu(\mathbf{x}) = \max_{\mathbf{x}} \sum_{i=1}^{T} w_i x_i,$$
$$\text{s.t.} \qquad x_i \in \{0,1\} \qquad 1 \leq i \leq T,$$

where $w_i > 0$ for all i. The optimal solution $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_T^*) = (1, 1, \ldots, 1)$. We can think of solving the problem as searching a weighted T-level binary tree for $\mathbf{x}^*$, shown in Fig. 2. The levels correspond to the indices of decision variables, i. A left branch (dashed line) at level i represents $x_i = 0$, and a right branch (solid lines) $x_i = 1$. One can think of a leaf node as a solution $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, determined by its location, and a non-leaf node at level t as a subregion, denoted by $(x_1, x_2, \ldots, x_t)$, where $x_i \in \{0, 1\}$ is fixed for $1 \leq i \leq t$. The root node at level 0, with all x_i unassigned, is denoted by (). A weight w_i is imposed on branches at level i, $i = 1, 2, \ldots, T$, to determine the value of leaf nodes (solutions).

We compare the performance of ESB&B on this problem with that of NP, both using the same partitioning and sampling scheme. The relative performance thus shows the value of the partition structure of ESB&B. We study how the problem structure determined by the weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_T)$ impacts this value.

The ESB&B algorithm we implement for the weighted binary tree problem (ESB&B-WBT) is as follows:

Algorithm ESB&B-WBT

Step 0. Set $t = 0$, let the record set be the root node, (), and the initial partition be {()}.

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

Step 1. Partition the record set, $(x_1, x_2, \ldots, x_t)$, by x_{t+1} into two subregions. Delete the record set from the partition and add the two subregions.
Step 2. Take one solution sample from all the subregions in the partition.
Step 3. Compute the upper bound of each subregion as the objective value of the sample taken from the subregion.
Step 4. Choose the subregion with the greatest upper bound to be the new record set, update $t = t + 1$, and go to Step 1.
Step 5. Repeat Steps 1 to 4 until the optimal solution $\mathbf{x}^*$ is found.

We implement the NP algorithm of Shi and Olafsson (2000) with the same partitioning and sampling scheme.

Algorithm NP-WBT

Step 0. Set $t = 0$ and let the root node, (), be the initial most-promising region.
Step 1. Partition the most-promising region, $(x_1, x_2, \ldots, x_t)$, by x_{t+1} into two subregions. Aggregate all of the other subregions into one surrounding region.
Step 2. Take one solution sample from all of the subregions.
Step 3. If the sample with the largest objective value is in the most promising region, then the subset that contains it becomes the new most promising region. Otherwise, backtrack to the root node. Update $t = t + 1$ and go to Step 1.
Step 4. Repeat Steps 1 to 3 until the optimal solution $\mathbf{x}^*$ is found.

It is worth mentioning that since the problem is deterministic, once the algorithm finds the optimal solution $\mathbf{x}^*$ it stays there. Therefore, we design both algorithms so that they stop when the optimum is found. The measure we use to compare the two algorithms is the expected number of steps it takes to find the optimal solution.

Both algorithms can be modeled as Markov chains. For ESB&B-WBT, the state is the current partition, whereas for NP-WBT the state is the current most promising region. For both Markov chains we compute the expected number of steps before finding the optimal solution.

The performance of both algorithms is affected by the weight vector $\mathbf{w}$. We use $\mathbf{w}' = (1, 2, 4)$ and $\mathbf{w}'' = (4, 2, 1)$ to explain how the weights affect the problem structure under our partitioning scheme; we use a variety of weights in our evaluation.

When $\mathbf{w} = \mathbf{w}'$, the objective value of solutions, from left to right, is {0, 1, 2, 3, 4, 5, 6, 7}. Partitioning at level 1 gives two subregions: {0, 1, 2, 3} and {4, 5, 6, 7}. Further partitioning of the first subregion at level 2 gives {0, 1}, {2, 3}, and {4, 5, 6, 7}. By contrast, when $\mathbf{w} = \mathbf{w}''$, the objective value of solutions, from left to right, is {0, 4, 2, 6, 1, 5, 3, 7}. Partitioning at level 1 gives two subregions: {0, 4, 2, 6} and {1, 5, 3, 7}. Further partitioning of the first subregion at level 2 gives {0, 4}, {2, 6}, and {1, 5, 3, 7}. Therefore, a weight vector with $w_1 > w_2 > \cdots > w_T$ gives a perfectly ordered list of solutions, or a "smooth" response surface, whereas a weight vector with $w_T > w_{T-1} > \cdots > w_1$ gives a shuffled list of solutions, or a "rough" response surface, which leads to more mistakes for both algorithms when selecting the record set (most promising region) by sampling. It is intuitive that ESB&B-WBT is going to have more of an advantage over NP-WBT when the solutions are less well-ordered, as ESB&B-WBT allows a quick jump to a better subregion when a mistake is realized, whereas NP-WBT has to backtrack to the root node and start to search all over again.

Table 1. Comparison of ESB&B-WBT and NP-WBT: four-level tree

(w_1, w_2, w_3, w_4)	*ESB&B-WBT*	*NP-WBT*	*Difference*	*% Difference*
(8, 4, 2, 1)	4.0000	4.0000	0	0
(4, 8, 2, 1)	4.5000	5.2421	−0.7421	−14
(4, 2, 8, 1)	5.2188	9.7508	−4.5320	−46
(2, 4, 8, 1)	5.9961	11.4414	−5.4453	−48
(2, 4, 1, 8)	6.8164	22.3561	−15.5398	−70
(2, 1, 4, 8)	8.0285	29.9445	−21.9160	−73
(1, 2, 4, 8)	9.1060	32.6701	−23.5641	−72

We list the expected number of steps before finding the optimal solution for ESB&B-WBT, NP-WBT, the difference between the two, and the percentage difference in Table 1 ($T = 4$) and Table 2 ($T = 5$). The percentage difference is computed as the expected number of steps of ESB&B-WBT minus that of NP-WBT over NP-WBT, which is the percentage savings in computing time of ESB&B-WBT over NP-WBT.

In both tables, the weight vectors are listed in a way that, as we go down the table, solutions are less well-ordered. The results show that ESB&B-WBT outperforms

Table 2. Comparison of ESB&B-WBT and NP-WBT: five-level tree

(w_1, w_2, w_3, w_4)	*ESB&B-WBT*	*NP-WBT*	*Difference*	*% Difference*
(16, 8, 4, 2, 1)	5.0000	5.0000	0	0
(8, 16, 4, 2, 1)	5.4808	6.2493	−0.7685	−12
(8, 4, 16, 2, 1)	6.2183	10.8195	−4.6012	−43
(8, 4, 2, 16, 1)	7.0558	21.1377	−14.0819	−67
(8, 4, 2, 1, 16)	8.0347	42.6536	−34.6189	−81
(4, 8, 2, 1, 16)	8.7437	46.4530	−37.7093	−81
(4, 2, 8, 1, 16)	9.8170	57.5061	−47.6891	−83
(4, 2, 1, 8, 16)	11.4054	76.8094	−65.4040	−85
(2, 4, 1, 8, 16)	12.4707	82.0483	−69.5776	−85
(2, 1, 4, 8, 16)	14.3449	95.0562	−80.7113	−85
(1, 2, 4, 8, 16)	15.9234	99.6354	−83.7111	−84

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

Table 3. Parameters used in the numerical experiments

Experiment	*Number of trials*	ω	Δn_F	Δn_A	ϑ_R	ϑ_O	ϑ
Miller and Shaw Fig. 3	200	3	1	1	10	10	10
Miller and Shaw Fig. 4	200	3	1	1	10	20	10
Miller and Shaw Fig. 5	200	3	1	1	20	20	10
Miller and Shaw Fig. 6	200	3	1	1	10	20	10
Miller and Shaw Fig. 7	200	3	10	2	10	20	10
Bowl Table 4	100	3	1	1	30	30	15
Buffer allocation Fig. 8	100	2	4	1	10	10	5

NP-WBT under all weight vectors. When the solutions are less well-ordered, ESB&B-WBT has more advantage over NP-WBT.

4.2. *Empirical evaluation*

Now we compare the performance of ESB&B against the NP algorithm of Pichitlamken and Nelson (2003) on three test problems: Miller and Shaw, bowl with flexible dimensions, 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 3 (number of trials, number of subregions per partition (ω), number of replications for newly sampled solutions (Δn_F), number of replications for previously sampled solutions (Δn_A), number of solution samples for subregions of the record set for ESB&B (ϑ_R), total number of solution samples for all of the other subregions for ESB&B (ϑ_O), and number of solutions per subregion for NP (ϑ)).

4.2.1. *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 two copies of it to make the problem two-dimensional:

$$\mu_1(x_1, x_2) = \frac{\sin^6(0.05\pi x_1)}{2^{2(x_1-10/80)^2}} + \frac{\sin^6(0.05\pi x_2)}{2^{2(x_2-10/80)^2}} \qquad (5)$$

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

This problem has a global optimum (10, 10) with an objective value of 2. The response surface is bumpy, with 25 local optima. To make it a DOvS problem, normally distributed noise with zero mean and standard deviation 0.3 is added to Equation (5). We study both the deterministic and stochastic versions of the problem.

First, we examine the effect of the number of solutions per subregion (ϑ_R solutions for each subregion of record set and a total of ϑ_O solutions for all the other subregions) on the relative performance of ESB&B compared with NP for the deterministic problem. We consider two sample allocation schemes here, Cpr and Npr. We fix the number of samples per subregion at 10 for NP. Figure 3 shows the objective function value of the estimated optimal solution at each time point (averaged over 200 searches) for $\vartheta_R = 10$ and $\vartheta_O = 10$; Fig. 4 is for $\vartheta_R = 10$ and $\vartheta_O = 20$, and Fig. 5 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

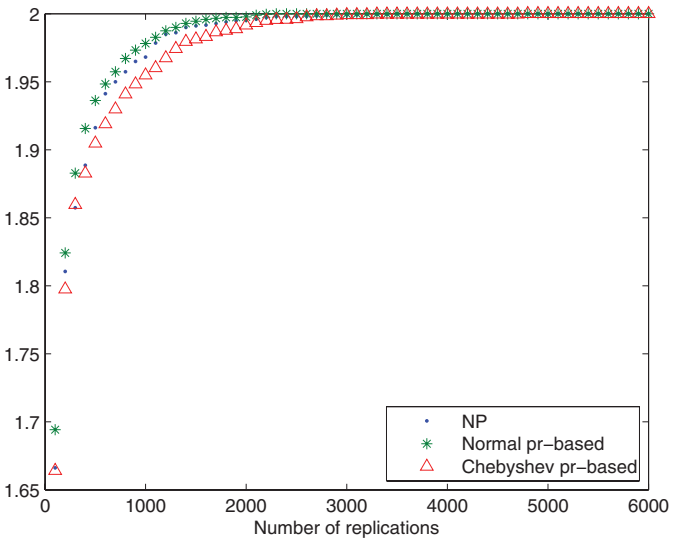

Fig. 3. Objective value of current optimal estimate at each time point for the Miller and Shaw problem: $\vartheta_R = 10$ and $\vartheta_O = 10$ (color figure provided online).

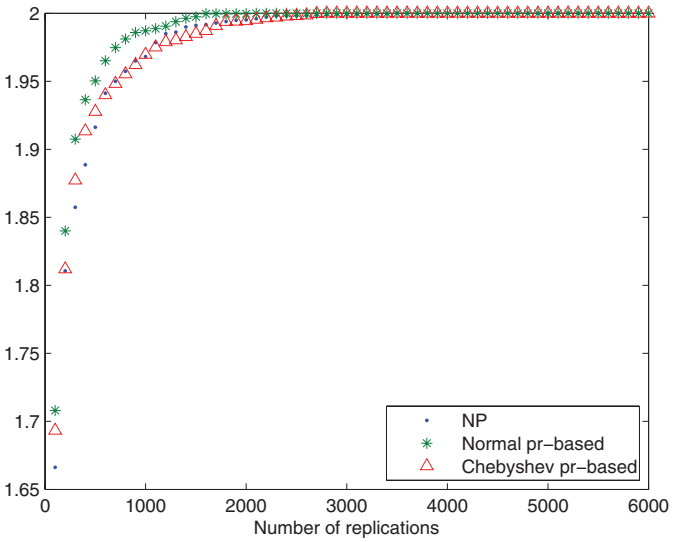

Fig. 4. Objective value of current optimal estimate at each time point for the Miller and Shaw problem: $\vartheta_R = 10$ and $\vartheta_O = 20$ (color figure provided online).

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

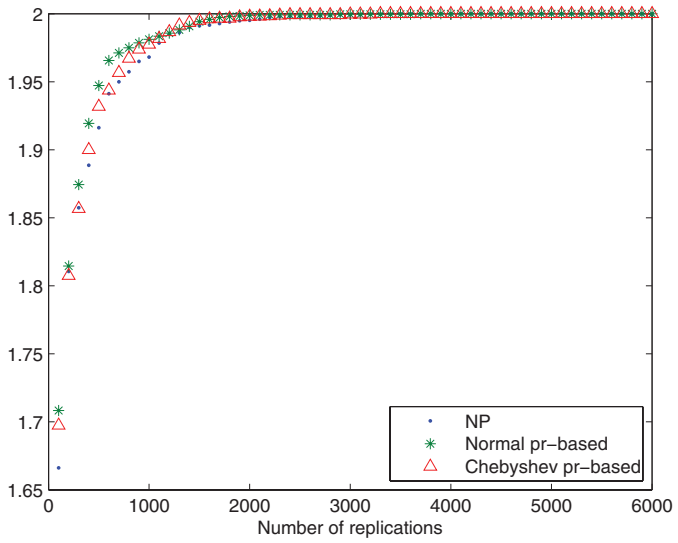

Fig. 5. Objective value of current optimal estimate at each time point for the Miller and Shaw problem: $\vartheta_R = 20$ and $\vartheta_O = 20$ (color figure provided online).

NP as ϑ_O increases. The intuition is that, when a "good" subregion (which contains good solutions) happens to appear inferior because of bad samples, enough samples from the subregion for the following iterations allows the algorithm to correct the mistake.

Next, we study the performance of different sample allocation methods. Figure 6 shows the performance of ESB&B compared with NP for Npr, Cpr, Nbd, and Cbd. It suggests that the normal probability-based sample allocation is better than all of the other methods.

We then examine the performance of different sample allocation methods with a noisy objective function. Figure 7 demonstrates the advantage of the normal probability-based method over the others and that ESB&B has an even greater advantage over NP when the problem is stochastic, for all types of sample allocation schemes.

4.2.2. *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:

$$\mu_2(x_1, x_2, \ldots, x_d) = 1000 \exp\{-0.001(x_1, x_2, \ldots, x_d)' \boldsymbol{\Sigma}^{-1}(x_1, x_2, \ldots, x_d)\}, \quad -\frac{m^{1/d}}{2} \le x_i \le \frac{m^{1/d}}{2}, \quad x_i \in \mathbb{Z}, \quad i = 1, 2, \ldots, d, \quad (6)$$

where $m = 20\,000$ is the total number of feasible solutions, d is dimension, and $\boldsymbol{\Sigma}$ is a $d \times d$ matrix that determines the correlation between different dimensions of the decision variable. This problem has a surface that is shaped like one for the probability density function of a multivariate normal distribution. It has a single global optimum $(0, 0, \ldots, 0)$ 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 the dimension changes, allowing us to isolate the impact of

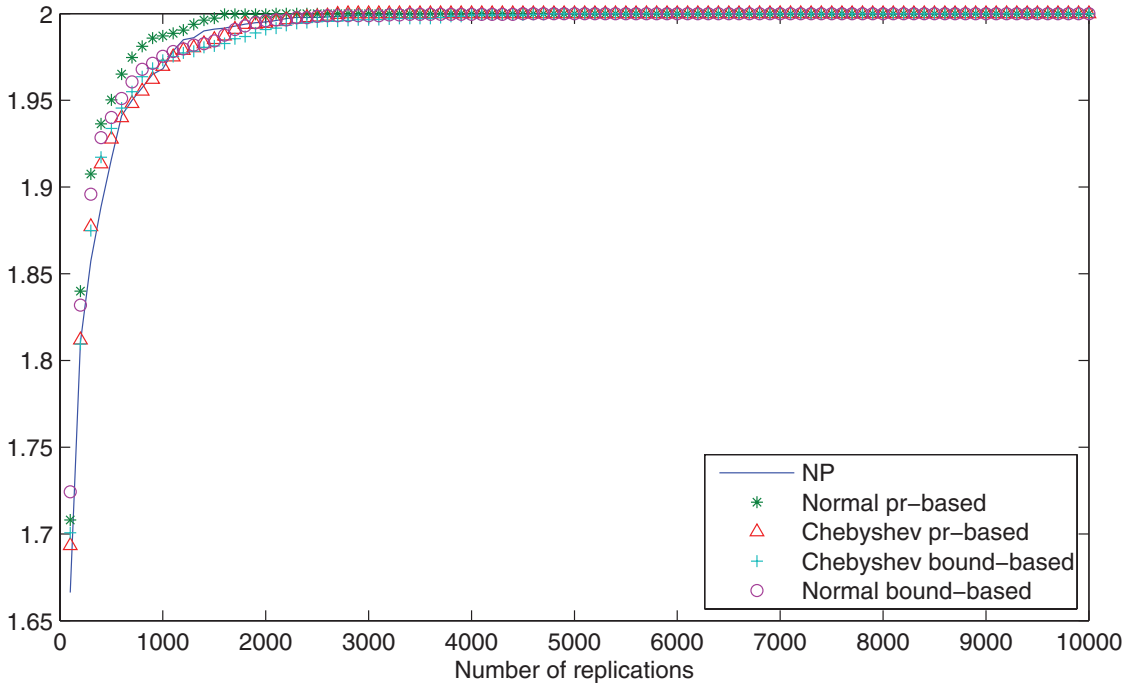

Fig. 6. Objective value of current optimal estimate at each time point for the Miller and Shaw problem: all sample allocation methods, deterministic case (color figure provided online).

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

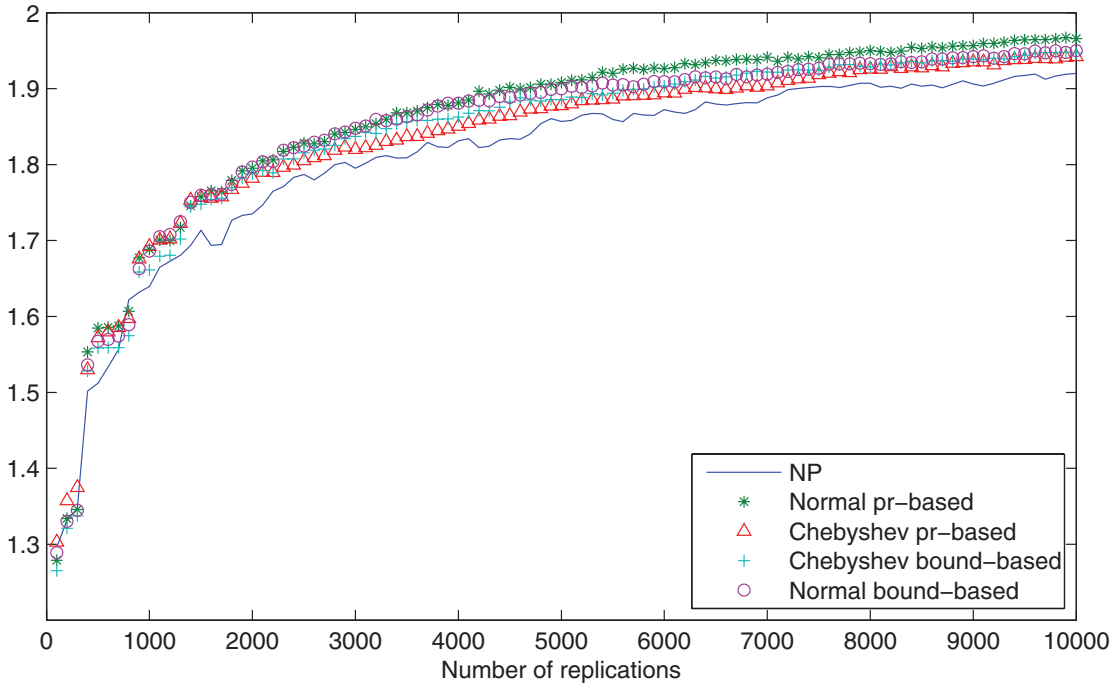

Fig. 7. True value of current optimal estimate at each time point for the Miller and Shaw problem: all sample allocation methods, stochastic case (color figure provided online).

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

dimension from that of the number of feasible solutions. This problem is similar to the test problem in Xu *et al.* (2010).

We use the correlation matrix

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

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 4 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: (i) the advantage of ESB&B over NP increases in ρ, but (ii) this trend becomes unclear as the dimension increases. We can explain these results as follows: Under higher correlation, both algorithms tend to make more mistakes in selecting a good subregion. ESB&B hence outperforms NP as it is able to jump directly from subregion to subregion, while NP has to backtrack and then search all the way down. However, this trend gets dominated by dimension as dimension increases.

Table 4. 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	20 100	18 300	2600	4100	25 900
ESB&B NPr	1100	1800	1000	2500	3000	7000	1700	3000	11 500
NPr − NP	−2100	−2100	−8600	−900	−17 100	−11 300	−900	−1100	−14 400
% Difference	−66	−54	−90	−26	−85	−62	−35	−27	−56
ESB&B CPr	1500	2100	1200	3700	4100	5900	2100	4100	15 800
CPr − NP	−1700	−1800	−8400	300	−16 000	−12 400	−500	0	−10 100
% Difference	−53	−46	−88	9	−80	−68	−19	0	−39

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

4.2.3. *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 τ_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 following constraints:

$$\begin{aligned} \tau_1 + \tau_2 + \tau_3 &\le 20, \\ b_2 + b_3 &\le 20, \\ -b_2 - b_3 &\le -20, \\ 1 \le \tau_h &\le 20, \quad h = 1, 2, 3, \\ 1 \le b_h &\le 20, \quad h = 2, 3, \\ \tau_h, b_h &\in \mathbb{Z}. \end{aligned}$$

The number of feasible solutions is 21 660. There are two optimal solutions: $(\tau_1, \tau_2, \tau_3, b_2, b_3) = (6, 7, 7, 12, 8)$ 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 sampled five solutions per subregion for NP and set $\vartheta_R = \vartheta_O = 10$ for ESB&B. Figure 8 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.

4.3. *UCB on region vs. UCB on solution*

One of the central contributions of this research is to use a UCB to estimate the potential of solution subregions and guide sampling. As described in Section 1, a UCB can also be applied to solutions, rather than regions, to guide allocation of simulation budget as in the multi-arm bandit literature. To assess the value of maintaining partitions and estimating UCBs on regions, relative to applying UCBs directly to solutions without partitioning, we developed a random search algorithm RS-UCB. Algorithm RS-UCB is designed so that at each iteration, $\omega \times \vartheta_R + \vartheta_O$ solutions are sampled and evaluated (some could be repeats), which matches with what happens in ESB&B. Recall that

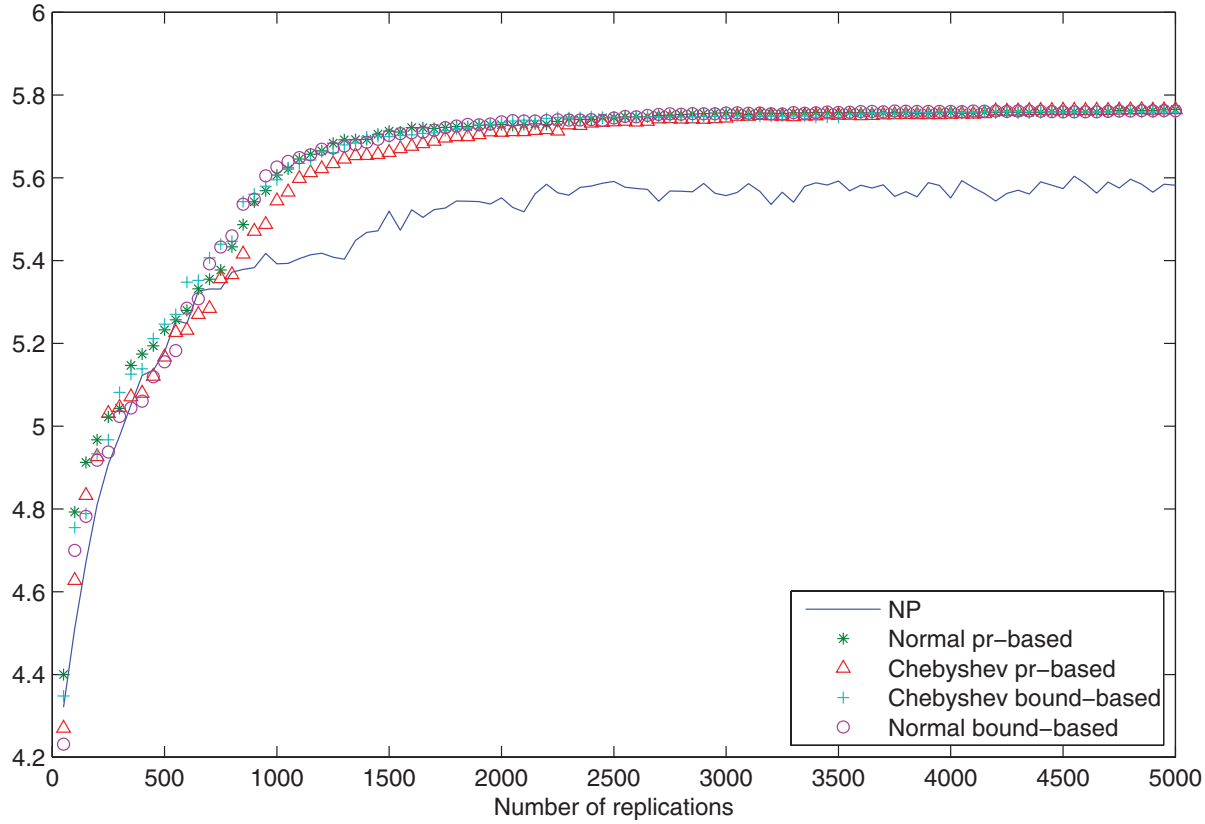

Fig. 8. True value of current optimal estimate at each time point for three-stage buffer allocation problem (color figure provided online).

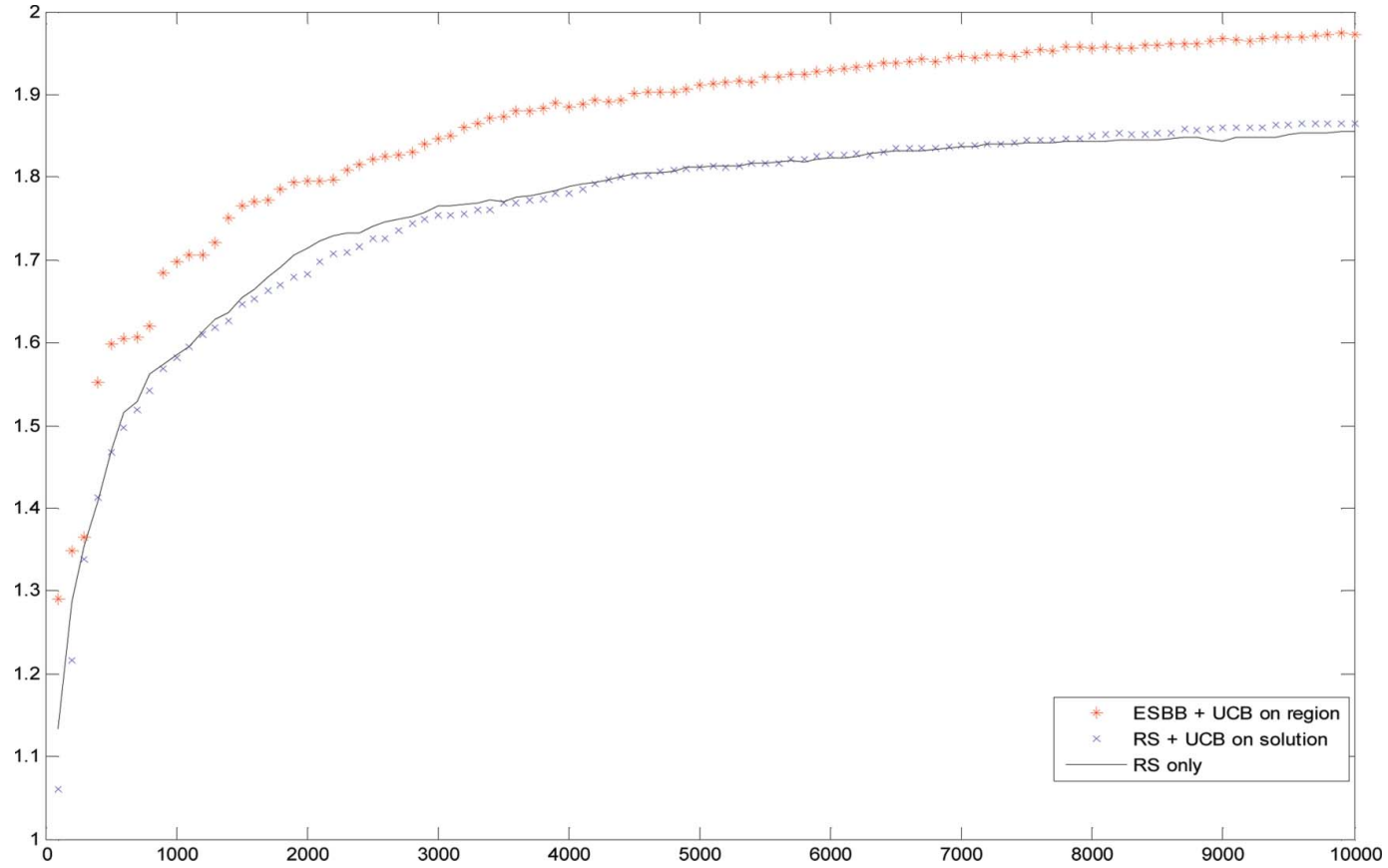

Fig. 9. True value of current optimal estimate at each time point for the stochastic Miller and Shaw problem: ESB&B, RS-UCB, and pure random search (RS) (color figure provided online).

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

for ESB&B ω is the number of subregions per partition, ϑ_{R} is the number of solution samples for subregions of the record set, ϑ_{O} is the total number of solution samples for all of the other subregions, Δn_{F} is the number of replications for newly sampled solutions, and Δn_{A} is the number of replications for previously sampled solutions for ESB&B.

Algorithm RS-UCB

Step 0. Initialization: Set iteration counter $k = 0$, $\mathfrak{S}^0 = \emptyset$.
Step 1. Solution sampling and evaluation: Randomly sample $\omega \times \vartheta_{\mathrm{R}}$ solutions and call the set of sampled solutions S^k. Let $\mathfrak{S}^k = S^{k-1} \bigcup S^k$. 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.
Step 2. Additional evaluation based on UCB: Compute UCB of all samples in S^k. Simulate Δn_{A} additional observations for ϑ_{O} solutions with the largest UCB.
Step 3. Updating iteration number: Set $k = k + 1$ and go to Step 1.

Whenever we terminate the algorithm (usually when the simulation budget is reached), we select as the best solution $\hat{\mathbf{x}}^*$ from $\mathfrak{S}^k$ the one with the maximum cumulative sample average. The calculation of UCB on a solution is straightforward given the assumptions that simulation noise is normally distributed with equal variances, and the UCB basically gives an upper confidence interval.

We study the performance of ESB&B and RS-UCB on the stochastic version of the Miller and Shaw problem described in Section 4.2.1. We also study pure random search, which samples and evaluates $\omega \times \vartheta_{\mathrm{R}} + \vartheta_{\mathrm{O}}$ solutions at each iteration as a baseline. All of the parameters are as in the fifth row of Table 3. Figure 9 shows that ESB&B clearly outperforms RS-UCB and pure random search. This demonstrates the value of UCB combined with partitioning over just using UCB on solutions. Interestingly, RS-UCB only shows a slight advantage over pure random search at later iterations. The intuition is that in the early iterations, using UCB on solutions to assign additional simulation budget is not as efficient as using the budget to explore more new solutions; however, as the number of iterations increases, using UCB on solutions does help identify good solutions among those that have already been sampled.

5. Conclusions

In this article we proposed the ESB&B algorithm that keeps the partition structure of the SB&B algorithm, while

estimating bounds based on sampling. The algorithm uses the estimated performance of the observed best solution to guide searching and computes statistical bounds that indicate the potential of subregions to control solution 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. A numerical study shows that ESB&B outperforms NP in the considered test problems. The advantage is greater 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 solutions sampled for the current best subregion, number of solutions sampled for the other subregions, initial number of replications for newly sampled solutions, and incremental number of replications for re-sampled solutions. We can also adaptively adjust the number of subregions and the number of solutions sampled as the algorithm progresses and subregions become smaller. A combination of different statistical bounds, at different stages of searching, is a possible direction, as well as designing new statistical bounds to guide sampling.

Downloaded by [Professor Barry Nelson] at 09:24 12 June 2013

Acknowledgement

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

References

Andradottir, S. (1998) Simulation optimization, in *Handbook of Simulation*, Banks, J. (ed), Wiley-Interscience, New York, NY, ch. 9.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002) Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, **47**, 235–256.

Buzacott, J.A. and Shantikumar, J.G. (1993) *Stochastic Models of Manufacturing Systems*, Prentice Hall, Englewood Cliffs, NJ.

Doerner, K., Gutjahr, W.J., Kotsis, G., Polaschek, M. and Strauss, C. (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., Glover, F.W. and April, J. (2005) Simulation optimization: a review, new developments, and applications, in *Proceedings of the 2005 Winter Simulation*, Kuhl, M.E., Steiger, N.M., Armstrong, F.B. and Joines, J.A. (eds), IEEE Press, Piscataway, NJ, pp. 83–95.

Gutjahr, W.J., Hellmayr, A. and Pflug, G.C. (1999) Optimal stochastic single-machine-tardiness scheduling by stochastic branch-and-bound. *European Journal of Operations Research*, **117**, 396–413.

Gutjahr, W.J., Strauss, C. and Wagner, E. (2000) A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing*, **12**(2), 125–135.

Hu, J., Fu, M.C. and Marcus, S.I. (2007) A model reference adaptive search method for global optimization. *Operations Research*, **55**(3), 549–568.

Hu, J. and Hu, P. (2010) An approximate annealing search algorithm to global optimization and its connection to stochastic approximation, in *Proceedings of the 2010 Winter Simulation Conference*, Johansson, B., Jain, S., Montoya-Torres, J., Hugan, J. and Yucesan, E. (eds), IEEE Press, Piscataway, NJ, pp. 1223–1234.

Lai, T.L. and Robbins, H. (1985) Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, **6**, 4–22.

Li, L. (2009) A unifying framework for computational reinforcement learning theory. Ph.D. thesis, Department of Computer Science, The State University of New Jersey, New Brunswick, NJ.

Miller, B.L. and Shaw, M.J. (1995) Genetic algorithms with dynamic niche sharing for multimodal function optimization. Technical Report no. 95010, Illinois Genetic Algorithms Laboratory, University of Illinois of Champaign–Urbana, Champaign, IL.

Norkin, V.I., Ermoliev, Y.M. and Ruszczynski, A. (1998) On optimal allocation of indivisibles under uncertainty. *Operations Research*, **46**(3), 381–395.

Norkin, V.I., Pflug, G.C. and Ruszczynski, A. (1998) A branch and bound method for stochastic global optimization. *Mathematical Programming*, **83**, 425–450.

Pichitlamken, J. (2002) A combined procedure for optimization via simulation. Ph.D. thesis, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Pichitlamken, J. and Nelson, B.L. 2003 A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation*, **13**, 155–179.

Saw, J.G., Yang, M.C.K. and Mo, T. C. (1984) Chebyshev inequality with estimated mean and variance. *American Statistician*, **38**(2), 130–132.

Shi, L. and Olafsson, S. (2000) Nested partition method for global optimization. *Operations Research*, **48**(3), 390–407.

Xu, J., Hong, L.J. and Nelson, B. L. (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, IL.

Biographies

Wendy Lu Xu is a senior researcher at Corporate Strategic Research, ExxonMobil Research and Engineering Company. She received her Ph.D. in Industrial Engineering and Management Sciences from Northwestern University in 2009. Her research focus is on modeling and analysis of complex system with uncertainty, including simulation optimization, metamodeling, and robust optimization, with application areas in oil and gas development planning and production optimization.

Barry L. Nelson is the Walter P. Murphy Professor and Chair of the Department of Industrial Engineering and Management Sciences at Northwestern University. His research focus is on the design and analysis of computer simulation experiments on models of discrete-event, stochastic systems, including methodology for simulation optimization, variance reduction, output analysis, metamodeling, and multivariate input modeling. His application areas are manufacturing, services, financial engineering, and transportation. He has published numerous papers and three books, including *Discrete-Event System Simulation* (fifth edition, Prentice Hall, 2010) and *Foundations and Methods of Stochastic Simulation: A First Course* (Springer, 2013). He is a Fellow of INFORMS and IIE. In 2006 he received the Outstanding Simulation Publication Award from the INFORMS Simulation Society for his work on simulation optimization, and in 2007 and 2010 he was awarded the Best Paper—Operations Award from *IIE Transactions*. His teaching has been acknowledged by a Northwestern University Alumni Association Excellence in Teaching Award, a McCormick School of Engineering & Applied Science Teacher of the Year Award, and the IIE Operations Research Division Award for Excellence in the Teaching of Operations Research. Further information can be found at http://www.iems.northwestern.edu/∼nelsonb/.