Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

EFFICIENT SIMULATION BUDGET ALLOCATION FOR SELECTING THE BEST SET OF SIMPLEST GOOD ENOUGH DESIGNS

Shen Yan
Enlu Zhou

Department of Industrial and
Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Champaign, IL 61820, USA

Chun-Hung Chen

Department of Systems Engineering
and Operations Research
George Mason University
Fairfax, VA 22030, USA

ABSTRACT

Simple designs have many advantages compared with complex designs, such as requiring less computing and memory resources, and easier to interpret and to implement. Therefore, they are usually more preferable than complex designs in the real world if their performances are within a good enough range. In this paper, we propose an algorithm OCBA-bSG to identify a best subset of m simplest and good enough designs among K ($K > m$) total designs. The numerical results show that our approach allocates the simulation budget efficiently, and outperforms some other approaches on the test problems.

1 INTRODUCTION

Simple designs are preferred compared to complex ones if they have similar performance. One example is the design of node activation rules in the wireless sensor networks (WSNs), as described in Kar, Krishnamurthy, and Jaggi (2006) and Jia (2009). Consider a solar powered WSN that monitors an area of interest. Each node needs to collaborate with its neighbors in order to get enough power to do the monitoring. The problem is to decide which node to active and when to activate such that we can get the highest probability of correct detection. It is clear that a larger communication radius means a more complex node activation rule and requires more power. Conversely, a smaller communication radius gives us a simpler node activation rule and consumes less power. When both large and small communication radii provide close probabilities of correct detection, we prefer the small radius (i.e., simple rule) to the large radius (i.e., complex rule).

In this paper, we use descriptive complexity to represent the degree of complexity of a design, where a simpler design has a smaller descriptive complexity. Since sometimes we may need to select more than one design in practice for robustness of the system, we consider the general problem of selecting the top m ($m \geq 1$) designs that are simplest (with smallest descriptive complexity) and good enough (satisfying a constraint on the performance measure). If the number of such designs is greater than m, the we select the top m ones with the best performance among such designs. The descriptive complexity of a design is a deterministic value, and we know the value once we simulate that design. However, the performance of a design is subject to system noise, and hence, it can only be estimated from simulation. Our goal is to allocate a given simulation budget efficiently to the designs so as to maximize the probability of correctly selecting the best m simplest good enough designs out of a total of K designs.

The above problem is closely related with many known results in the literature on ranking and selection (R&S). Koenig and Law (1985) developed a two-stage procedure for selecting the top m designs with best performance, following the results in Dudewicz and Dalal (1975). Chen et al. (2000) and Chen and Yücesan (2005) developed the optimal computing budget allocation (OCBA) procedure for the selection of one best design, and later Chen et al. (2008) extended that to find m best designs. Development of such a Bayesian approach for selecting the m best designs has wide applications for the simulation-based global optimization methods that require a subset of top designs in each iteration of the algorithms. There is also research focusing on optimizing the primary performance measure subject to the feasibility of a secondary performance measure. Andradóttir et al. (2005) proposed a two-phase approach which identifies all the feasible systems first and then selects the best from them. Szechtman and Yücesan (2008) used large deviation theory to deal with

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

feasibility determination. Most recently, Lee et al. (2009) developed OCBA further to select one single best design under multiple constraints of secondary performance measures.

The problem of considering both complexity and performance evaluation has only been considered recently. Jia (2009) proposed an Adaptive Sampling Algorithm (ASA) for selecting one simplest good enough design with the goal to minimize the Type II error of the chosen design. ASA is not suitable for extension to select multiple designs, because it screens the designs sequentially one by one until the goal is achieved. Therefore, it still remains open how to select multiple best designs that are simplest and good enough. In this paper, we address this problem by developing a method called OCBA-bSG, abbreviated for optimal computing budget allocation for m best simplest good enough designs. Numerical results show that OCBA-bSG achieves a higher probability of correct selection in much less iterations than the Equal Allocation scheme and Levin Search method.

2 PROBLEM STATEMENT

Let θ denote a design, and Θ denote the finite set of all the designs. The performance of the design θ is measured by

$$J(\theta) = E[L(\theta,\zeta)],$$

where ζ is a random vector that represents the uncertainty in the system, and $L(\theta,\zeta)$ can only be obtained through simulation of the complex system. The underlying assumption is that such simulation is expensive. A design is considered better if its performance measure $J(\theta)$ is smaller. A *good enough design* is one that satisfies $J(\theta) < J_0$, where J_0 is a given threshold on the performance. Hence, we define the feasible set (or good enough set) as

$$F = \{\theta | J(\theta) < J_0, \forall \theta \in \Theta\}.$$

The complexity of the design θ is represented by the descriptive complexity $C(\theta)$, which is a deterministic value in the set $\{0,1,\dots,n\}$. The value of $C(\theta)$ in known once we simulate the design θ. Therefore, the *optimal set* S_{mb} of m simplest good enough designs is defined as

$$S_{mb} = \{\theta_{mb_1}, \theta_{mb_2}, \dots, \theta_{mb_m} \in F | C(\theta_{mb_i}) < C(\theta) \text{ OR } J(\theta_{mb_i}) < J(\theta) \text{ if } C(\theta_{mb_i}) = C(\theta), \forall \theta \in F/S_{mb}\}.$$

The objective is to determine an efficient simulation budget allocation to maximize the probability of correctly selecting the set S_{mb}. The above definition of S_{mb} implies that we start choosing the designs from the good enough designs in the smallest complexity set until we reach a complexity set that there are more good enough designs than we need, and then from this complexity set we select the deigns with smaller performance until the number of selected designs reaches m. To proceed, we introduce the following notations and assumptions.

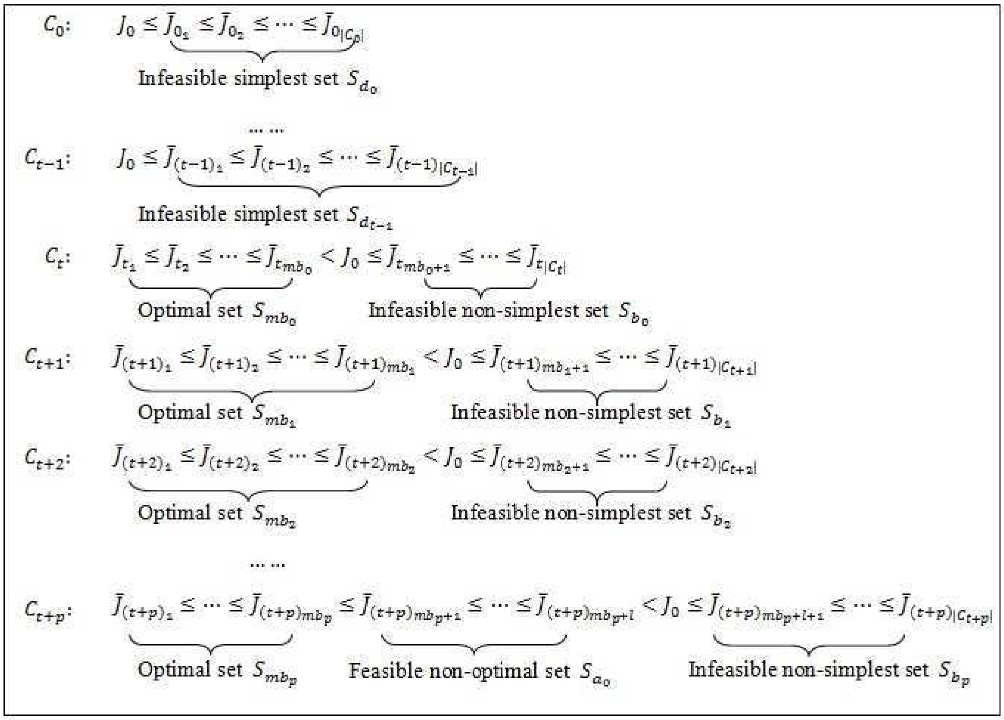
$P(CS)$: the probability of correct selection of S_{mb}.
K: the total number of designs, where $K = |\Theta|$.
T: the total simulation budget.
C_i: the set of designs with descriptive complexity i, i.e., $C(\theta) = i$, $\forall \theta \in C_i$.
$|C_i|$: the total number of designs in C_i.
θ_{ij}: the j^{th} design after ordering the designs in C_i from the smallest sample mean to the largest, $j = 1,2,\dots,|C_i|$.
$J(\theta_{ij})$ or J_{ij}: the performance measure for design θ_{ij}. For simplicity of the notation, we write $J(\theta_{ij})$ as J_{ij} in the following.
σ_{ij}^2: variance of design θ_{ij}. Since the true variance is unknown, we use the sample variance to estimate it.
N_{ij}: simulation budget for design θ_{ij}.
X_{ij}^k: the k^{th} simulation replication for design θ_{ij}, $k = 1,\dots,N_{ij}$. We assume that X_{ij}^k's are independent inside the design (i.e., with respect to k) and across the designs (i.e., with respect to ij).

$\bar{J}_{ij}$: sample mean of design θ_{ij}, $\bar{J}_{ij} = (1/N_{ij}) \sum_{k=1}^{N_{ij}} X_{ij}^k$.

$\tilde{J}_{ij}$: posterior distribution of J_{ij}. Using noninformative prior distribution for J_{ij} and assuming that the J_{ij} has a conjugate normal prior distribution, the posterior distribution of J_{ij} is shown to be

$$\tilde{J}_{ij} \sim N(\bar{J}_{ij}, \frac{{\sigma_{ij}}^2}{N_{ij}}).$$

Now we propose a procedure to maximize $P(CS)$ with a fixed simulation budget. First, we group the designs according to the descriptive complexity $C(\theta)$. Second, we order the designs according to their current sample means of performance measure $J(\theta)$ within the set C_i, and denote the ordered designs as θ_{ij}, $i = 0,1,\ldots,n,\ j = 1,2,\ldots,|C_i|$. Suppose that all the designs in the sets C_0, C_1, …, C_{t-1} are infeasible and the first feasible design appears in the set C_t. If the set C_t has m feasible designs, then the optimal set S_{mb} is in C_t. If there are less than m feasible designs in C_t, then we continue searching the sets C_{t+1}, C_{t+2}, …, until we find m feasible designs.



$$C_0: \quad J_0 \le \underbrace{\bar{J}_{0_1} \le \bar{J}_{0_2} \le \cdots \le \bar{J}_{0_{|C_0|}}}_{\text{Infeasible simplest set } S_{d_0}}$$

$$\cdots\ \cdots$$

$$C_{t-1}: \quad J_0 \le \underbrace{\bar{J}_{(t-1)_1} \le \bar{J}_{(t-1)_2} \le \cdots \le \bar{J}_{(t-1)_{|C_{t-1}|}}}_{\text{Infeasible simplest set } S_{d_{t-1}}}$$

$$C_t: \quad \underbrace{\bar{J}_{t_1} \le \bar{J}_{t_2} \le \cdots \le \bar{J}_{t_{mb_0}}}_{\text{Optimal set } S_{mb_0}} < J_0 \le \underbrace{\bar{J}_{t_{mb_0+1}} \le \cdots \le \bar{J}_{t_{|C_t|}}}_{\text{Infeasible non-simplest set } S_{b_0}}$$

$$C_{t+1}: \quad \underbrace{\bar{J}_{(t+1)_1} \le \bar{J}_{(t+1)_2} \le \cdots \le \bar{J}_{(t+1)_{mb_1}}}_{\text{Optimal set } S_{mb_1}} < J_0 \le \underbrace{\bar{J}_{(t+1)_{mb_1+1}} \le \cdots \le \bar{J}_{(t+1)_{|C_{t+1}|}}}_{\text{Infeasible non-simplest set } S_{b_1}}$$

$$C_{t+2}: \quad \underbrace{\bar{J}_{(t+2)_1} \le \bar{J}_{(t+2)_2} \le \cdots \le \bar{J}_{(t+2)_{mb_2}}}_{\text{Optimal set } S_{mb_2}} < J_0 \le \underbrace{\bar{J}_{(t+2)_{mb_2+1}} \le \cdots \le \bar{J}_{(t+2)_{|C_{t+2}|}}}_{\text{Infeasible non-simplest set } S_{b_2}}$$

$$\cdots\ \cdots$$

$$C_{t+p}: \quad \underbrace{\bar{J}_{(t+p)_1} \le \cdots \le \bar{J}_{(t+p)_{mb_p}}}_{\text{Optimal set } S_{mb_p}} \le \underbrace{\bar{J}_{(t+p)_{mb_p+1}} \le \cdots \le \bar{J}_{(t+p)_{mb_p+l}}}_{\text{Feasible non-optimal set } S_{a_0}} < J_0 \le \underbrace{\bar{J}_{(t+p)_{mb_p+l+1}} \le \cdots \le \bar{J}_{(t+p)_{|C_{t+p}|}}}_{\text{Infeasible non-simplest set } S_{b_p}}$$

Figure 1: Relationship between subsets and designs.

Figure 1 gives a pictorial view of the above procedure, and also shows that in general there are four sets we need to consider: optimal subset S_{mb_i}, feasible non-simplest subset S_{a_i}, infeasible simplest subset S_{d_i} and infeasible non-simplest subset S_{b_i}. As we have assumed that the first feasible design with the smallest descriptive complexity appears in the set C_t, all the designs in the sets $C_0, C_1, \ldots, C_{t-1}$ belong to the infeasible simplest subsets S_{d_i}. The optimal subsets are $S_{mb_0}, S_{mb_1}, \ldots, S_{mb_p}$, which satisfy

$$S_{mb} = \bigcup_{i=0}^{p} S_{mb_i}, \quad m = \sum_{i=0}^{p} |S_{mb_i}|.$$

The sets from C_t to C_{t+p-1} do not contain feasible non-simplest sets S_{a_i}, since all the feasible designs in these sets should be included in the optimal sets. However, the set C_{t+p} may contain both the optimal subset S_{mb_p} and the feasible non-simplest set S_{a_0}, depending on the number of m. So far we have already selected m optimal designs, so there is no need to consider the sets $C_{t+p+1}, C_{t+p+2}, \ldots, C_n$. But as the iteration continues, we may need to reconsider them if we find that the number of optimal designs in the previous sets are less than m.

After defining and analyzing the subsets, the problem of maximizing the probability of correct selection can be formulated as follows.

$$\begin{aligned} &max \quad P(CS) = P\{\tilde{J}_i < J_0 \,\&\, \tilde{J}_j \leqslant \tilde{J}_k < J_0 \,\&\, \tilde{J}_s \geqslant J_0, \forall i \in \cup_{l=0}^{p-1} S_{mb_l}, \forall j \in S_{mb_p}, \forall k \in S_{a_0}, \forall s \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}\} \\ &s.t. \quad N_1 + N_2 + \ldots + N_K = T, \end{aligned} \tag{1}$$

where N_i is the simulation budget for design θ_i.

3 MAIN RESULTS

For the problem of selecting m best simplest good enough designs, it is time-consuming to estimate $P(CS)$ directly by Monte Carlo simulation. Thus, we simplify the maximization problem by finding a lower bound for $P(CS)$, and then calculate the answer analytically. This approximation is often referred to as Approximate Probability of Correct Selection ($APCS$), such as in Chen et al. (2008).

$$\begin{aligned}
P(CS) &= P\{\tilde{J}_i < J_0 \,\&\, \tilde{J}_j \leqslant \tilde{J}_k < J_0 \,\&\, \tilde{J}_s \geqslant J_0, \forall i \in \cup_{l=0}^{p-1} S_{mb_l}, \forall j \in S_{mb_p}, \forall k \in S_{a_0}, \forall s \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}\} \\
&\geqslant P\{\tilde{J}_i < J_0 \,\&\, \tilde{J}_j \leqslant \mu \,\&\, \mu \leqslant \tilde{J}_k < J_0 \,\&\, \tilde{J}_s \geqslant J_0, \forall i \in \cup_{l=0}^{p-1} S_{mb_l}, \forall j \in S_{mb_p}, \forall k \in S_{a_0}, \forall s \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}\} \\
&= \prod_{i \in \cup_{l=0}^{p-1} S_{mb_l}} P\{\tilde{J}_i < J_0\} \prod_{i \in S_{mb_p}} P\{\tilde{J}_i \leqslant \mu\} \prod_{i \in S_{a_0}} P\{\mu \leqslant \tilde{J}_i < J_0\} \prod_{i \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}} P\{\tilde{J}_i \geqslant J_0\} \\
&\triangleq APCS
\end{aligned}$$

The value of μ will be discussed in Corollary 1. Because of the independence property we assumed before, we can write $APCS$ into a product form. Hence, problem (1) is approximated by

$$\begin{aligned}
\max \quad & APCS = \prod_{i \in \cup_{l=0}^{p-1} S_{mb_l}} P\{\tilde{J}_i < J_0\} \prod_{i \in S_{mb_p}} P\{\tilde{J}_i \leqslant \mu\} \prod_{i \in S_{a_0}} P\{\mu \leqslant \tilde{J}_i < J_0\} \prod_{i \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}} P\{\tilde{J}_i \geqslant J_0\} \\
s.t. \quad & N_1 + N_2 + \ldots + N_K = T.
\end{aligned}$$

The main results are stated in Corollary 1 and Theorem 2. Their detailed proofs are given in (Yan, Zhou, and Chen 2010).

Corollary 1. *The μ value introduced in APCS is determined by*

$$\mu = \frac{\hat{\sigma}_{(t+p)_{mp+1}} \bar{J}_{(t+p)_{mp}} + \hat{\sigma}_{(t+p)_{mp}} \bar{J}_{(t+p)_{mp+1}}}{\hat{\sigma}_{(t+p)_{mp}} + \hat{\sigma}_{(t+p)_{mp+1}}}, \text{ where } \hat{\sigma}_{(t+p)_i} = \frac{\sigma_{(t+p)_i}}{\sqrt{N_{(t+p)_i}}}. \tag{2}$$

Theorem 2. *APCS is asymptotically (as $T \to \infty$) maximized by the following allocation scheme.*

Case 1: If the number of feasible designs in C_t is less than m and the number of feasible designs in C_{t+p} is more than mb_p, then we allocate N_i according to the following relationship.

$$\frac{N_a}{\sigma_a^2/(\bar{J}_a - J_0)^2} = \frac{N_b}{\sigma_b^2/(\bar{J}_b - \mu)^2} = \frac{N_c}{\sigma_c^2/(\bar{J}_c - J_0)^2} = \frac{N_x}{\sigma_x^2/(\bar{J}_x - \mu)^2} = \frac{N_y}{\sigma_y^2/(\bar{J}_y - J_0)^2}, \tag{3}$$

for all $a \in \cup_{l=0}^{p-1} S_{mb_l}$, $b \in S_{mb_t}$, $c \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}$, $x \in S_{a_0}$ and $\bar{J}_x \leqslant \frac{\mu + J_0}{2}$, $y \in S_{a_0}$ and $\bar{J}_y > \frac{\mu + J_0}{2}$.

Case 2: If the number of feasible designs in C_t is less than m and the number of feasible designs in C_{t+p} is exactly mb_p, then we allocate N_i according to the following relationship.

$$\frac{N_a}{\sigma_a^2/(\bar{J}_a - J_0)^2} = \frac{N_c}{\sigma_c^2/(\bar{J}_c - J_0)^2} \tag{4}$$

for all $a \in \cup_{l=0}^{p} S_{mb_l}$ and $c \in \{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}$.

Case 3: If there are less than m feasible designs in the given K designs, then we allocate N_i according to the following relationship.

$$\frac{N_a}{\sigma_a^2/(\bar{J}_a - J_0)^2} = \frac{N_c}{\sigma_c^2/(\bar{J}_c - J_0)^2}, \tag{5}$$

for all $a \in \cup_{l=0}^{n-t} S_{mb_l}$ and $c \in \{\cup_{l=0}^{n-t} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}$.

Theorem 2 provides some very intuitive results. The simulation budget is proportional to the sample variance for each design. If a design has a larger sample variance, more simulation budget will be allocated to it in order to find a more accurate estimate of the performance mean. We also notice that at the two critical points μ and J_0: μ is the critical point for the optimality, J_0 is the critical point for the feasibility. The designs closer to these two points will be assigned more simulation budget among all the designs in $C_0, \ldots, C_{t+p}$, the complexity sets under consideration. For the subsets $\cup_{l=0}^{p-1} S_{mb_l}$ and $\{\cup_{l=0}^{p} S_{b_l}\} \cup \{\cup_{r=0}^{t-1} S_{d_r}\}$, we are only interested in determining wether the designs are good enough, and indeed more simulation budget is assigned to the designs near J_0. Similarly, for the subset S_{mb_p}, we are only interested in comparing the performance of the designs, and more simulation budget is assigned to the designs around μ. For the subset S_{a_0} where both μ and J_0 are critical points, the last two terms in (3) imply that we should divide the set into two parts by the midpoint $\frac{\mu+J_0}{2}$: the designs with sample means in the range $\mu \leqslant \bar{J}_x \leqslant \frac{\mu+J_0}{2}$ will be compared with μ, and the ones nearer to μ will get more simulation budget; the designs falling into the range $\mu > \bar{J}_y > \frac{\mu+J_0}{2}$ will be compared with J_0, and be assigned more simulation budget if closer to J_0.

3.1 Allocation Procedure

Recall the following notations:
K: the total number of the designs.
m: the number of designs we want to choose.
T: the total simulation budget.
Δ: the increased simulation budget each iteration.
n_0: the initial simulation budget for every design.
J_0: the good enough performance constraint.

OCBA-bSG Allocation Procedure

```
Input: $K$, $m$, $T$, $\Delta$, $n_0$, $J_0$.
Initialize: $r = 0$.
    • Perform $n_0$ simulation replications for all designs and generate samples $X_i^k,\ k = 1,2,\dots,n_0,\ i = 1,2,\dots,K$.
    • Group the designs according to their descriptive complexities $C(\theta)$ to obtain the sets $C_i,\ i = 0,1,...,n.$, and order them according to their sample mean. Denote the $j^{th}$ design in the complexity set $C_i$ as $\theta_{ij}$, its samples as $X_{ij}^k$, and its current simulation budget as $N_{ij}^r = n_0$. Set $N^r = Kn_0$.
Loop: while $N^r < T$, do
    1. Update:
        • Calculate sample mean $\bar{J}_{ij} = \frac{1}{N_{ij}^r}\sum_{k=1}^{N_{ij}^r} X_{ij}^k$, for $i = 0,\dots,n,\ j = 1,\dots,|C_i|$.
        • Calculate sample standard deviation $\sqrt{\frac{\sum_{k=1}^{N_{ij}^r}(X_{ij}^k - \bar{J}_{ij})^2}{N_{ij}^r - 1}}$ for estimating $\sigma_{ij},\ i = 0,\dots,n,\ j = 1,\dots,|C_i|$.
        • Calculate $\mu$ according to (2) if there exist $\bar{J}_{(t+p)_{mb_p}}$ and $\bar{J}_{(t+p)_{mb_p+1}}$ in the set $C_{(t+p)}$ (c.f. Figure 1).
        • Increase the computing budget $N^{r+1} = \min\{N^r + \Delta, T\}$.
    2. Allocate: calculate the new budget allocation $N_{ij}^{r+1},\ i = 0,\dots,n,\ j = 1,\dots,|C_i|$ as follows.
        • If the number of feasible designs in $C_t$ is less than $m$ and the number of feasible designs in $C_{t+p}$ is more than $mb_p$, then we compute $N_{ij}^{r+1}$ according to (3).
        • If the number of feasible designs in $C_t$ is less than $m$ and the number of feasible designs in $C_{t+p}$ is exactly $mb_p$, then we compute $N_{ij}^{r+1}$ according to (4).
        • If there are less than $m$ feasible designs in the given $K$ designs, then we compute $N_{ij}^{r+1}$ according to (5).
    3. Simulate:
        • If $N_{ij}^{r+1} \geqslant 0.1T$ or $N_{ij}^{r+1} < N_{ij}^r$, we let $N_{ij}^{r+1} = N_{ij}^r$ (that means not to do additional simulation in this iteration). Subtract these $N_{ij}^{r+1}$ from the current total budget and go back to step 2 to recalculate the remaining $N_{ij}^{r+1}$.
        • If $N_{ij}^r \leqslant N_{ij}^{r+1} < 0.1T$, perform additional $(N_{ij}^{r+1} - N_{ij}^r)$ simulations for the $j^{th}$ design in the complexity set $C_i$ to generate more samples $X_{ij}^k,\ k = N_{ij}^r + 1, N_{ij}^r + 2, \dots, N_{ij}^{r+1}$.
    4. Update: $r \to r + 1$.
End of Loop
```

Since we obtain the simulation budget allocation formula under the asymptotic limit $T \to \infty$, the scheme needs some modifications when the total simulation budget T is not infinity. To avoid that one simulation budget N_{ij} is too large such that other designs are not assigned enough simulation budget, we put a constraint on N_{ij}: if $N_{ij} \geqslant 0.1T$, we stop allocating new simulation budget to that design. The constraint $0.1T$ works well in our numerical experiments. However, there is no formal guideline on how to choose that constraint. On the other hand, the simulation budget cannot be taken back once the simulation is done, so if $N_{ij}^{r+1} < N_{ij}^{r}$, then we let $N_{ij}^{r+1} = N_{ij}^{r}$ and do not do additional simulations.

4 NUMERICAL EXPERIMENTS

In this section, we will demonstrate OCBA-bSG by some simple examples with comparison to two other methods - Equal Allocation and Levin Search methods.

Equal Allocation (EA) scheme allocates the simulation budget equally among all the designs and do not use any information of the mean, the variance or the descriptive complexity of the underlying designs. Levin Search (LS) method (Levin 1973) allocates simulation budget to designs with smallest descriptive complexity first to obtain an estimate for the performance with certain accuracy, and then continues to the next simplest design until it finds m simplest good enough designs.

In the numerical experiments, we use $P(CS)$ as the efficiency measurement to compare the three allocation methods: Equal Allocation (EA), Levin Search (LS) and our method OCBA-bSG. For a given total simulation budget, the faster the $P(CS)$ converges, the better the corresponding method is. Here we estimate $P(CS)$ by calculating the ratio of the number of simulation runs that get correct selection to the total number of simulation runs. In addition, for convenience, we assume that design θ_i has descriptive complexity $\lfloor 1+\log_2 i \rfloor$, so the descriptive complexity is non-decreasing for i.

4.1 Example 1 (Mean increases as descriptive complexity increases)

There are 20 designs in total, each with the underlying distribution $N(i,(0.5i)^2)$, $i = 1,2,...,20$. We want to find 5 simplest good enough designs with good enough constraint J_0 = 7.3.

In this example, $m = 5$, $K = 20$, $J_0 = 7.3$, initial simulation budget $n_0 = 20$, simulation budget increment $\Delta = 200$, total simulation budget $T = 8000$, and the total number of simulation runs is 50000. If we group the designs according to their descriptive complexity, we get the design sets $\{\theta_1\}$, $\{\theta_2,\theta_3\}$, $\{\theta_4,\theta_5,\theta_6,\theta_7\}$, $\{\theta_8,\ldots,\theta_{15}\}$ and $\{\theta_{16},\ldots,\theta_{20}\}$ with descriptive complexity 1, 2, 3, 4 and 5 respectively. In example 1, the mean increases as the descriptive complexity increases, and the variance increases as the mean increases. The correct selection is the set $\{\theta_1,\theta_2,\theta_3,\theta_4,\theta_5\}$. Figure 2 shows the simulation result. OCBA-bSG converges faster than EA and LS. LS searches from the simplest design sets $\{\theta_1\}$, $\{\theta_2,\theta_3\},\ldots$, and in this example the correct selections are design set $\{\theta_1,\theta_2,\theta_3,\theta_4,\theta_5\}$, so LS converges in about 5 iterations.

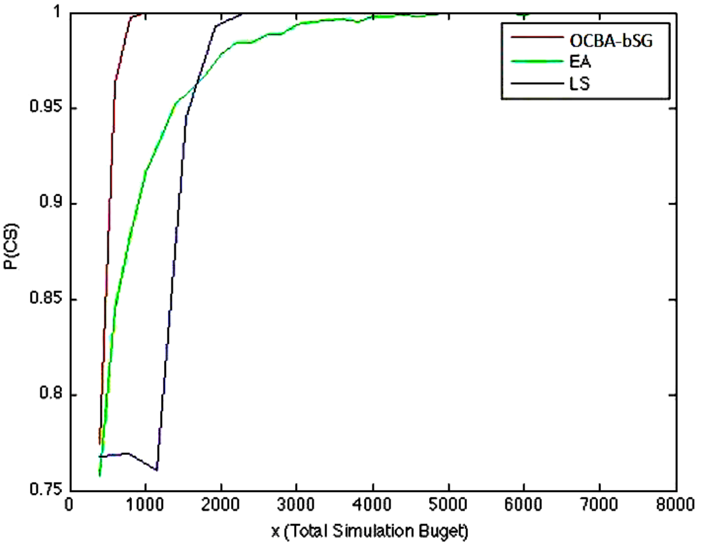


Figure 2: Selecting 5 best simplest good enough designs from 20 designs with distribution $N(i,(0.5i)^2)$ and $J_0 = 7.3$.

4.2 Example 2 (Mean decreases as descriptive complexity increases)

There are 20 designs, each with the underlying distribution $N((21-i),(0.5i)^2)$, $i = 1,2,...,20$. We want to find 5 simplest good enough designs with good enough constraint J_0 = 7.3.

The parameter setting is the same as Example 1 except that the total number of simulation runs is 5000. If we group the designs according to their descriptive complexity, we get the design sets $\{\theta_1\}$, $\{\theta_2,\theta_3\}$, $\{\theta_4,\theta_5,\theta_6,\theta_7\}$, $\{\theta_8,\ldots,\theta_{15}\}$ and $\{\theta_{16},\ldots,\theta_{20}\}$ with descriptive complexity 1, 2, 3, 4 and 5 respectively. In example 2, the descriptive complexity is non-decreasing as the mean decreases, and the variance increases as the mean decreases. Hence, the correct selection is the set $\{\theta_{14},\theta_{15},\theta_{18},\theta_{19},\theta_{20}\}$. Figure 3 shows the simulation result. Since in this example designs with smaller means have larger variances and the correct selections $\{\theta_{14},\theta_{15},\theta_{18},\theta_{19},\theta_{20}\}$ have relatively large variances compared to other designs, OCBA-bSG converges slower than that of example 1 but still faster than EA and LS. Moreover, since LS searches from the simplest sets $\{\theta_1\}$, $\{\theta_2,\theta_3\}$,..., it has the worst performance.

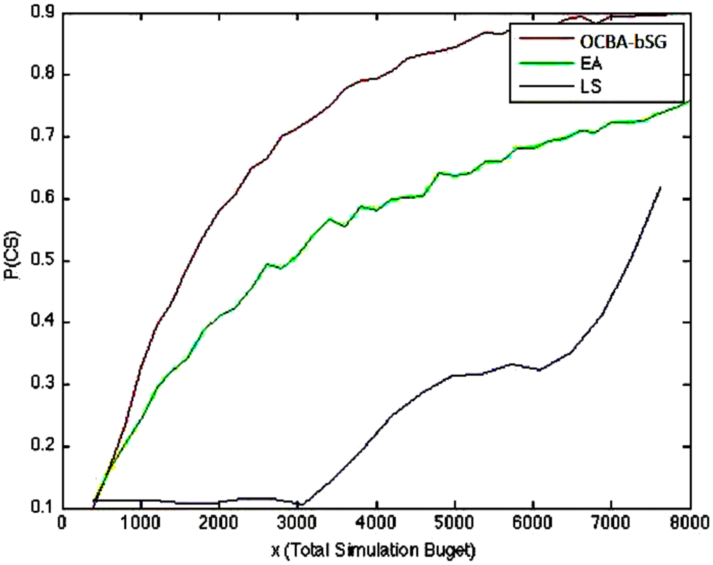


Figure 3: Select 5 best simplest good enough designs from 20 designs with distribution $N((21-i),(0.5i)^2)$ and $J_0 = 7.3$.

5 CONCLUSION

Motivated by real-world applications, we consider the problem of selecting multiple designs that are both simple and good enough. We provide an algorithm called OCBA-bSG to allocate the simulation budget for all the designs in order to maximize the probability of correctly finding the optimal set of *m* simplest good enough designs out of a total number of *K* designs. Numerical results show that OCBA-bSG converges fast in all situations considered here, which indicates OCBA-bSG indeed allocates simulation budget efficiently.

REFERENCES

Andradóttir, S., D. Goldsman, B. W. Schmeiser, L. W. Schruben, and E. Yücesan. 2005. Analysis methodology: Are we done? In *Proceedings of the 37th conference on Winter simulation*, ed. M. E. Kuhl, N. Steiger, F. Armstrong, and J. Joines: IEEE.

Chen, C. H., D. H. He, M. Fu, and L. H. Lee. 2008. Efficient simulation budget allocation for selecting an optimal subset. *INFORMS Journal on Computing* 20 (4): 579–595.

Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick. 2000. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Journal of Discrete Event Dynamic Systems: Theory and Applications* 10:251–270.

Chen, C. H., and E. Yücesan. 2005. An alternative simulation budget allocation scheme for efficient simulation. *International Journal of Simulation and Process Modeling* 1 (1/2): 49–57.

Dudewicz, E. J., and S. R. Dalal. 1975. Allocation of observation in ranking and selection with unequal variances. *Sankhya: The Indian Journal of Statistics* 37:28–78.

Jia, Q. S. 2009. An adaptive sampling algorithm for simulation-based optimization with descriptive complexity preference. under review.

Kar, K., A. Krishnamurthy, and N. Jaggi. 2006. Dynamic node activation in networks of rechargeable sensors. *IEEE/ACM Transactions on Networking* 14 (1): 15–26.

Koenig, L. W., and A. M. Law. 1985. A procedure for selecting a subset of size m containing the l best of k independent normal populations. *Communications in Statistics - SImulation and Computation* 14:719–734.

Lee, H. L., N. A. Pujowidianto, L. W. Li, C. H. Chen, and C. M. Yap. 2009. Asymptotic simulation budget allocation for selecting the best design in the present of stochastic constraints. working paper.

Levin, L. A. 1973. Universal sequential search problems. *Problems of Information Transmission* 9 (3): 265–266.

Szechtman, R., and E. Yücesan. 2008. A new perspective on feasibility determination. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. Mason, R. Hill, O. R. L. Monch, T. Jefferson, and J. Fowler, 273–280: IEEE.

Yan, S., E. Zhou, and C. H. Chen. 2010. Efficient simulation budget allocation for selecting an optimal good enough subset with descriptive complexity preference. working paper.

AUTHOR BIOGRAPHIES

Shen Yan is a master student in the Department of Industrial and Enterprise Systems Engineering, the University of Illinois at Urbana-Champaign. She received her Bachelors of Engineering with highest honors from Chinese University of Hong Kong, Hong Kong in 2009. Her research interests are in simulation optimization. Her email address is <shenyan1@illinois.edu>.

Enlu Zhou is an Assistant Professor in the Department of Industrial and Enterprise Systems Engineering, at the University of Illinois at Urbana-Champaign. She received the B.S. degree with highest honors in electrical engineering from Zhejiang University, China, in 2004, and received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 2009. Her research interests include stochastic control, nonlinear filtering, and simulation optimization. Her email address is <enluzhou@illinois.edu>.

Chun-Hung Chen is a Professor of Systems Engineering and Operations Research at George Mason University. He received his Ph.D. from Harvard University in 1994. His research interests are mainly in development of very efficient methodology for simulation and optimization and its applications. Dr. Chen has served as Co-Editor of the Proceedings of the 2002 Winter Simulation Conference and Program Co-Chair for 2007 INFORMS Simulation Society Workshop. He is currently an associate editor of IEEE Transactions on Automatic Control, area editor of Journal of Simulation Modeling Practice and Theory, associate editor of International Journal of Simulation and Process Modeling, and simulation department editor for IIE Transactions. His email address is <cchen9@gmu.edu>.