OPTIMAL COMPUTING BUDGET ALLOCATION OF INDIFFERENCE-ZONE-SELECTION PROCEDURES

E. JACK CHEN
BASF Corporation, 3000 Continental Drive–North, Mount Olive, New Jersey 07828, USA
chenej@basf.com

CHUN-HUNG CHEN
Dept. of Systems Engineering and Operations Research, George Mason University, Fairfax, Virginia 22030, USA
cchen9@gmu.edu

W. DAVID KELTON
Dept. of Quantitative Analysis and Operations Management, University of Cincinnati, Cincinnati, OH 45221, USA
david.kelton@uc.edu

Indifference-zone-selection procedures have been widely studied and applied to determine the required sample sizes for selecting a good design from k alternatives. However, efficiency is still a key concern for application of simulation to ranking and selection problems. In this paper, we present a new approach that can further enhance the efficiency of indifference-zone-selection procedures. Our approach determines a highly efficient number of simulation replications or samples and significantly reduces the total simulation effort. An experimental performance evaluation demonstrates the efficiency of the new procedure.

Subject classifications: Simulation: Sample size allocation, Simulation: Indifference-zone selection, Simulation: Statistical analysis

---

Discrete-event simulation (DES) has been widely used to estimate some measure of performance defined over a stochastic system and to compare alternative system designs or operating policies. When evaluating k alternative system designs, we would like to select one as the best, while ensuring the probability of correctly selecting the best is sufficiently high. Simulation studies of this kind are often referred as *simulation optimization*, i.e., we use simulation as an aid for optimizing stochastic complex systems where their performance measure cannot be easily obtained through analytical means and therefore must be estimated via stochastic simulation.

While DES has many advantages for modeling complex systems, efficiency is still a key concern when performing simulation experiments (Law and Kelton 2000). Since the objective is to select the good designs rather than to obtain an accurate estimate of the best performance measure, it is advantageous to use ordinal comparison for selecting a good design. The underlying philosophy is to rank estimators through ordinal comparison while the precision of the estimates are still poor, see Ho (1992) for detail of ordinal optimization. It is known that "order" converges exponentially fast (Dai 1996) while "value" converges at rate $1/\sqrt{n}$ (Kushner and Clark 1978), where n is the length or number of replications of the simulation experiments. We achieve this goal by using a class of Ranking and Selection (R&S) procedures.

One crucial element in R&S procedures is the event of "correct selection" (CS) of the true best system. In a stochastic simulation, the possibility of CS, denoted by P(CS), increases as the sample sizes become larger. Most indifference-zone-selection procedures are directly or indirectly based on Dudewicz and Dalal's (1975) or Rinott's (1978) indifference-zone-selection procedures. However, these procedures determine the number of additional replications based on a conservative *least favorable configuration* (LFC) assumption and do not take into account the value of sample means. If the accuracy requirement is high, and if the total number of designs in a decision problem is large, then the total simulation cost can easily become prohibitively high. For an overview of earlier methods of R&S see Law and Kelton (2000).

Some new approaches, such as *Optimal Computing Budget Allocation* (OCBA) (Chen et al. 2000b) and the *Enhanced Two-Stage Selection* (ETSS) procedure (Chen and Kelton 2000) incorporate first-stage sample mean information with sample variance in determining the number of additional replications. In numerical testing, both OCBA and ETSS procedures demonstrate a significant reduction in computing effort compared to Rinott's procedure. The basic idea of those procedures is that to ensure a high probability of correctly selecting a good design, a larger portion of the computing budget should be allocated to those designs that are critical in the process of identifying good designs. Overall simulation efficiency is improved as less computational effort is spent on simulating non-critical designs and more is spent on critical designs. There are several other new approaches aimed at improving the efficiency of ranking and selection, Chick and Inoue (2001) use a Bayesian framework for constructing ranking and selection procedures. Nelson et al. (2001) develop selecting procedures when the number of alternative is large. Goldsman et al. (2002) using all pairwise comparisons to eliminate inferior designs at early iterations to reduce overall sample sizes.

In this paper, we focus on those procedures which intend to allocate simulation trials to designs in a way that maximizes P(CS) within a given computing budget. Previous researches have examined various approaches for efficiently allocating a fixed computing budget across design alternatives, in particular, Chen (1995), Chen et al. (2000a), and Chen et al. (2000b). However, those procedures are developed without including the idea of indifference zone. Until now, traditional indifference-zone-selection procedures and optimal computing budget allocations have been treated as two completely separate approaches. Chen and Kelton (2000) demonstrate that these two classes of approaches obtain every similar results. In this paper, we examine the relationship between the indifference amount and the sample size allocation and develop a optimal approach for solving the budget allocation problem with indifference zone. The proposed approach is simple, general, practical and complementary to other techniques.

The rest of this paper is organized as follows. In Section 1, we provide background for the proposed procedure. In Section 2, we present our methodologies and proposed procedure for computing budget allocation. In Section 3, we give our empirical-experimental results. In Section 4, we give concluding remarks.

1 BACKGROUND

First, some notation:

X_{ij}: the observation from the j^{th} replication or batch of the i^{th} design,

N_i: the final number of replications or batches for design i,

r: the intermediate number of replications or batches for design i,

μ_i: the expected performance measure for design i, i.e., $\mu_i = E(X_{ij})$,

$\bar{X}_i(r)$: the running sample mean for design i, i.e., $\sum_{j=1}^{r} X_{ij}/r$,

$\bar{X}_i$: the final sample mean for design i, i.e., $\sum_{j=1}^{N_i} X_{ij}/N_i$,

σ_i^2: the variance of the observed performance measure of design i from one replication or batch, i.e., $\sigma_i^2 = \mathrm{Var}(X_{ij})$,

$S_i^2(N_i)$: the sample variance of design i with N_i replications or batches, i.e., $S_i^2(N_i) = \sum_{j=1}^{N_i}(X_{ij} - \bar{X}_i)^2/(N_i - 1)$.

Let μ_{i_l} be the l^{th} smallest of the μ_i's, so that $\mu_{i_1} \leq \mu_{i_2} \leq \ldots \leq \mu_{i_k}$. Our goal is to select a design with the smallest expected response μ_{i_1} using stochastic simulation.

1.1 Optimal Computing Budget Allocation (OCBA)

Let design b denote the design having the smallest sample mean performance measure, i.e, $\bar{X}_b = \min_{1\leq i\leq k} \bar{X}_i$, the probability of correct selection

$$\mathrm{P(CS)} = \mathrm{P}[\text{design } b \text{ is actually the best design}].$$

Chen et al. (2000b) propose OCBA that is based on a fixed total computing budget $T = \sum_{i=1}^k N_i$ and attempts to maximize P(CS). The sequential procedure utilizes the information of both the means and variances obtained from each iteration to allocate incremental sample size for each design.

Based on a Bayesian model, they develop an *Approximate Probability of Correct Selection* (APCS) as a lower bound of P(CS). That is,

$$\mathrm{P(CS)} \geq 1 - \sum_{i=1, i\neq b}^{k} \mathrm{P}[\bar{X}_b > \bar{X}_i].$$

The right hand side of the above equation is the APCS. They show that for a fixed number of replications or batches, the APCS can be asymptotically maximized when

$$\frac{N_i}{N_j} = \left(\frac{\sigma_i/\delta_{i,b}}{\sigma_j/\delta_{j,b}}\right)^2, i, j \in \{1, 2, \ldots, k\}, \text{ and } i \neq j \neq b, \tag{1}$$

$$N_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{N_i^2}{\sigma_i^2}}, \tag{2}$$

where $\delta_{i,b} = \bar{X}_i - \bar{X}_b$, and σ_i is the standard deviation of the response of design i.

1.2 Indifference-Zone-Selection Procedures

Even though our goal is to select a design with the smallest expected response μ_{i_1}, in practice however, if μ_{i_1} and μ_{i_2} are very close together, we might not care if we mistakenly choose design i_2, whose expected response is μ_{i_2}. The "practically significant" difference d^* (a positive real number) between a best and a satisfactory design is called the indifference zone in the statistical literature and represents the smallest difference about which we care. Therefore, we want a procedure that avoids making a large number of replications or batches to resolve differences less than d^*. That is, we want P(CS) $\geq P^*$ provided that $\mu_{i_2} - \mu_{i_1} \geq d^*$, where the minimal CS probability P^* and the "indifference" amount d^* are both specified by the user.

In the ordinal optimization literature, a broader sense of P(CS) is defined as the *alignment probability*. Briefly, the "good enough" subset G of the design space Θ is the set of top-g designs and the selected subset S is a subset of Θ in which the members are selected by the designer using certain evaluation criteria as the outcome for certain designs. The alignment probability is the probability that among the set S we have at least $g'(\leq g)$ of the member of G, i.e., $P[|G \cap S| \geq g']$, where $|S|$ is the cardinality of the set S and g' is called the *alignment level*. Note that $1 \leq g' \leq \min(|G|, |S|)$. The set of designs i such that $\mu_i < \mu_{i_1} + d^*$ can be viewed as the "good enough" subset G in ordinal optimization. As we increase the indifference amount, we relax our selection criteria, i.e., goal softening in ordinal optimization literature, and can significantly reduce our computation burden.

1.3 An Enhanced Two-Stage Selection (ETSS) Procedure

Chen and Kelton (2000) propose an ETSS procedure based on the assumption that we know the true mean of designs under consideration and takes into account not only the sample variances but also the difference between sample means across designs when determining sample sizes. While the observed P(CS)'s of the ETSS procedure are slightly lower than Rinott's procedure, in most cases, they are still higher than the specified P^*.

Let n_0 be the number of initial replications or batches and $\bar{X}_b(n_0) = \min_{1 \leq i \leq k} \bar{X}_i(n_0)$. The first-stage sample means $\bar{X}_i(n_0) = \sum_{j=1}^{n_0} X_{ij}/n_0$, marginal sample variances

$$S_i^2(n_0) = \frac{\sum_{j=1}^{n_0}(X_{ij} - \bar{X}_i(n_0))^2}{n_0 - 1},$$

and the estimated-controlled distance

$$\hat{d}_i = \max(d^*, \bar{X}_i(n_0) - \bar{X}_b(n_0)), \tag{3}$$

for $i = 1, 2, \ldots, k$ are computed. Based on the number of initial replications or batches n_0, the sample variance estimate $S_i^2(n_0)$ and the difference of means estimator $\hat{d}_i$ obtained from the first stage, the number of additional simulation replications or batches for each design in the second stage is $N_i - n_0$, where

$$N_i = \max(n_0, \lceil (hS_i(n_0)/\hat{d}_i)^2 \rceil), \text{ for } i = 1, 2, \ldots, k, \tag{4}$$

where $\lceil z \rceil$ is the smallest integer that is greater than or equal to the real number z, and h (which depends on k, P^*, and n_0) is a constant which solves Rinott's (1978) integral (h can be calculated by the FORTRAN program *rinott* in Bechhofer et al. (1995), or can be found from the tables in Wilcox (1984)). We then compute the overall sample means $\bar{X}_i = \sum_{j=1}^{N_i} X_{ij}/N_i$, and select the design with the smallest $\bar{X}_i$ as the best one.

The difference between ETSS and Rinott's procedure is that ETSS uses $\hat{d}_i$ instead of a fixed indifference-amount d^* in equation (4). This is because Ronott's procedure computes sample sizes based on the LFC, while ETSS assumes the mean for all designs are known and takes into account the information of sample means when computing sample sizes. Since $\hat{d}_i \geq d^*$, the sample size N_i allocated by ETSS will be no more than by Rinott's procedure.

2 METHODOLOGIES

In this section we examine a sample size allocation strategy to obtain the APCSIZ (*approximate P(CS) with indifference-zone*). The APCSIZ is derived with the assumption that we know the true mean and variance of the performance measure. As with most indifference-zone-selection procedures, we require the input data are i.i.d. normal. Many performance measures of interest are taken over some averages of a sample path or a batch of samples. Thus, the simulation output tends to be normally distributed in many applications. If the nonnormality of the samples is a concern, users can use batch means (see Law and Kelton 2000) to obtain sample means that are essentially i.i.d. normal. Moreover, Chen et al. (2000b) demonstrate that OCBA is robust to the normality assumption.

2.1 Problem Statement

We wish to choose $N_1, N_2, \ldots, N_k$ such that P(CS) is maximized, subject to a limited computing budget T, i.e.,

$$\begin{aligned} &\max_{N_1,\ldots,N_k} \mathrm{P(CS)} \\ &\text{s.t. } \sum_{i=1}^{k} N_i = T. \\ &N_i \in N, i = 1, 2, \ldots, k. \end{aligned}$$

Here N is the set of non-negative integers and $N_1 + N_2 + \ldots + N_K$ denotes the total computational cost assuming the simulation times for different designs are roughly the same. To solve the problem, we must be able to estimate P(CS). There exists a large literature on assessing P(CS) based on classical statistical models (e.g. Banks 1998 gives an excellent survey on available approaches). However, estimating P(CS) via Monte Carlo simulation is time-consuming, consequently, most P(CS) assessment procedures are mainly developed for problems with small number of designs.

To facilitate the derivation of our approximation of P(CS), we assume the means and variances are known. Let $\phi(x)$ and $\Phi(x)$ denote the probability density and distribution function, respectively, of the standard normal distribution. Without lost of generality, assume $\mu_{i_1} + d^* \leq \mu_{i_2} \leq \ldots \leq \mu_{i_k}$, the corresponding variances are $\sigma^2_{i_l}$. Furthermore, let $\delta_{i_l} = \mu_{i_l} - \mu_{i_1}$. Then

$$\begin{aligned}
\mathrm{P(CS)} &= \mathrm{P}[\bar{X}_{i_1} < \bar{X}_{i_l}, \text{ for } l = 2, 3, \ldots, k] \\
&= \mathrm{P}[\bar{X}_{i_1} - \bar{X}_{i_l} + \delta_{i_l} < \delta_{i_l}, \text{ for } l = 2, 3, \ldots, k] \\
&\geq \Pi_{l=2}^{k} \mathrm{P}[\bar{X}_{i_1} - \bar{X}_{i_l} + \delta_{i_l} < \delta_{i_l}] \\
&= \Pi_{l=2}^{k} \Phi(\delta_{i_l} / \sqrt{\sigma^2_{i_l}/N_{i_l} + \sigma^2_{i_1}/N_{i_1}}).
\end{aligned}$$

The inequality follows from Slepian's inequality (Tong 1980) since the values $X_{i_1} - X_{i_l}$ are positively correlated. The last equality follows from that the variate

$$Z_{i_l} = \frac{\bar{X}_{i_1} - \bar{X}_{i_l} + \delta_{i_l}}{\sqrt{\sigma^2_{i_l}/N_{i_l} + \sigma^2_{i_1}/N_{i_1}}}$$

has a $\mathcal{N}(0, 1)$ distribution, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2.

Let

$$Y_{i_l} = \delta_{i_l} / \sqrt{\sigma^2_{i_l}/N_{i_l} + \sigma^2_{i_1}/N_{i_1}}.$$

Theoretically, the following optimization problem can provide a tighter bound of P(CS):

$$\begin{aligned}
&\max_{N_1, \ldots, N_k} \Pi_{l=2}^{k} \Phi\left(Y_{i_l}\right) \\
&\text{s.t.} \sum_{i=1}^{k} N_i = T. \\
&N_i \in N, i = 1, 2, \ldots, k.
\end{aligned}$$

However, because of its complexity there is no known analytical solutions of this optimization problem. The OCBA of Chen et al. (2000b) consider the following optimization problem:

$$\begin{aligned}\max_{N_1,\ldots,N_k} \sum_{l=2}^{k} \Phi\left(Y_{i_l}\right)\\ \text{s.t.} \sum_{i=1}^{k} N_i = T.\\ N_i \in N, i = 1, 2, \ldots, k.\end{aligned}$$

They solved this optimization problem asymptotically with some assumptions. Section 1.1 summarized their results.

Furthermore, the ratios obtained by OCBA will stay the same if we consider the following optimization problem:

$$\begin{aligned}\min_{N_1,\ldots,N_k} \sum_{i=1}^{k} N_i\\ \text{s.t.} \sum_{l=2}^{k} \Phi\left(Y_{i_l}\right) = k - 2 + P^*.\\ N_i \in N, i = 1, 2, \ldots, k.\end{aligned}$$

That is, we want to minimize the sample sizes that achieve the specified minimal P(CS), P^*.

2.2 A Heuristic Indifference-Zone Allocation Rule

Indifference-zone procedures seek to avoid allocating extra samples to rank designs differ less than d^*. Let $d_{i_l} = \max(d^*, |\mu_{i_l} - \min_{j=1,j\neq l}^{k} \mu_{i_j}|)$ for $l = 1, 2, \ldots, k$, and let

$$D_{i_l} = d_{i_l} / \sqrt{\sigma_{i_l}^2 / N_{i_l} + \sigma_{i_1}^2 / N_{i_1}}. \tag{5}$$

By the *Bonferroni* inequality (Law and Kelton 2000),

$$\begin{aligned}\text{P(CS)} &= \text{P}[\bar{X}_{i_1} - \bar{X}_{i_l} + d_{i_l} < d_{i_l},\ \text{for } l = 2, 3, \ldots, k]\\ &\geq 1 - \sum_{l=2}^{k}(1 - \text{P}[\bar{X}_{i_1} - \bar{X}_{i_l} + d_{i_l} < d_{i_l}])\\ &= 1 - \sum_{l=2}^{k}(1 - \Phi(D_{i_l}))\\ &= 2 - k + \sum_{l=2}^{k} \Phi(D_{i_l}).\end{aligned}$$

We use the approximate probability of correct selection with indifference zone (APCSIZ) to estimate P(CS). That is,

$$APCSIZ = 2 - k + \sum_{l=2}^{k} \Phi(D_{i_l}) = P^*.$$

Thus, to achieve P^*, we need to have $\sum_{l=2}^{k} \Phi(D_{i_l}) = P^* + k - 2$. Since both OCBA and OCBAIZ are based on maximizing the lower bound of P(CS), which is derived from the Bonferroni inequality, we can use common random numbers to increase $\Phi(D_{i_l})$ for $l = 2, 3, \ldots, k$ and APCSIZ. We will derive our allocation rule heuristically based on minimizing the sample sizes. Since the purpose of budget allocation is to improve simulation efficiency, we need a relatively fast and inexpensive way of estimating P(CS) within the budget allocation procedure. Efficiency is more crucial than estimation accuracy in this setting.

Let

$$P = 1 - \frac{1 - P^*}{k - 1}.$$

We *assume* that the optimal is obtained when $\Phi(D_{i_l}) = P,$ for $l = 2, 3, \ldots, k$. Or similarly, $D_{i_l} = z_P,$ for $l = 2, 3, \ldots, k$, where z_P is the P quantile of the standard normal distribution. It follows from (**??**) that

$$d_{i_l} = z_P \sqrt{\sigma_{i_l}^2 / N_{i_l} + \sigma_{i_1}^2 / N_{i_1}} = w_{i_l}, \tag{6}$$

where w_{i_l} is the one-tailed P confidence interval half-width and is depended on the confidence level, variance and sample sizes. Hence, if we want to achieve $\Phi(D_{i_l}) = P$, the sample sizes N_{i_l} and N_{i_1} should be large enough such that $w_{i_l} = d_{i_l}$. For any $l \neq 1$, (5) holds if $N_{i_l} = 2(z_P \sigma_{i_l} / d_{i_l})^2$ and $N_{i_1} = 2(z_P \sigma_{i_1} / d_{i_l})^2$. We conveniently set

$$N_i = \lceil 2(z_P \sigma_i / d_i)^2 \rceil, \text{ for } i = 1, 2, \ldots, k. \tag{7}$$

Note that $d_{i_l} \geq d_{i_2} = d_{i_1}$, for $l = 3, 4, \ldots, k$. So $d_{i_l} \leq w_{i_l}$, and APCSIZ $\geq P^*$.

We summary the result as follows:

Proposition 1. Given a total number of simulation samples T to be allocated to k competing designs and their known means and variances are $\mu_1, \mu_2, \ldots, \mu_k$, and $\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2$ respectively, the Approximate Probability of Correct Selection with Indifference Zone (APCSIZ) will be near optimal when

$$\frac{N_i}{N_j} = \left(\frac{d_j}{d_i}\right)^2 \left(\frac{\sigma_i}{\sigma_j}\right)^2, \; i, j \in \{1, 2, \ldots, k\}. \tag{8}$$

Where N_i is the number of samples allocated to design i and $d_i = \max(d^*, |\mu_i - \min_{j=1,j\neq i}^{k} \mu_j|)$.

Remark 1. We assume the optimal sample sizes to achieve P(CS) $\geq P^*$ is when $\Phi(D_{i_l}) = P$ and satisfy (6). These two assumptions may not be true, however, they are reasonable approaches and can simplify the computation effort.

Remark 2. If $D_{i_l} > D_{i_2}$, then the allocated sample sizes are larger than necessary. That is, if $d_{i_l} > d_{i_2}$, then fewer samples than those computed by (6) will be enough to achieve $\Phi(D_{i_l}) = P$. Even though we assume that optimal is reached when $D_{i_l} = D_{i_2}$ for $l = 2, 3, \ldots, k$, the allocated sample size will result in $D_{i_l} \geq D_{i_2}$ for $l \neq 2$, which may be closer to the true optimal.

Remark 3. Since $d_{i_1} = d_{i_2}$,

$$\frac{N_1}{N_2} = \left(\frac{\sigma_1}{\sigma_2}\right)^2$$

regardless of the number of alternatives, k. This result is different from OCBA of Chen et al. (2000b), which is not based on the indifference-zone, and when $k = 2$ will allocate sample sizes to each design so that

$$\frac{N_1}{N_2} = \frac{\sigma_1}{\sigma_2}.$$

Remark 4. The sample sizes allocated by the ETSS procedure approximately satisfy the ratio of (7).

With Proposition 1, we now present a cost-effective sequential approach based on the concept described earlier to select the best design from k alternatives with a given computing budget. In our procedure, we use mean and variances estimators $\bar{X}_i(r)$ and $S_i(r)$ to compute the ratio of equation (7) and the estimator of d_i is $\hat{d}_i = \max(d^*, |\bar{X}_i(r) - \min_{j=1,j\neq i}^{k} \bar{X}_j(r)|)$. We use the equation $S_i^2(r) = (\sum_j^r X_{ij}^2/r - \bar{X}_i(r)^2)r/(r-1)$ to compute the variance estimator, therefore, we are only required to store the triple $(r, \sum_{j=1}^r X_{ij}, \sum_j^r X_{ij}^2)$, instead of the entire sequences $(X_{i1}, X_{i2}, \ldots, X_{ir})$.

Initially, n_0 simulation replications for each k design are conducted to get some information about the performance of each design during the first stage. As simulation proceeds, the sample means and sample variances of each design are computed from the data already collected up to that stage. According to this collected simulation output, an incremental computing budget for each iteration, Δ_l is distributed to each design based on the ratio of equation (7), where l is the iteration number. Ideally, each new replication should bring us closer to the optimal solution. The procedure will be iterated repeatedly until we have exhausted the pre-determined computing budget T. The algorithm is summarized as follows.

A Sequential Algorithm for Optimal Computing Budget Allocation with Indifference Zone (OCBAIZ):

1. Simulate n_0 replications or batches for each design. Set $l = 0$, $N_1^l = N_2^l = \ldots = N_k^l = n_0$, and $T = T - kn_0$.

2. Set $l = l + 1$. Increase the computing budget (i.e., number of additional simulations) by Δ_l and compute the new budget allocation, $N_1^l, N_2^l, \ldots, N_k^l$, using Proposition 1.

3. Simulate additional $\max(0, N_i^l - N_i^{l-1})$ replications or batches for each design i, $i = 1, 2, \ldots, k$.

4. $T = T - \Delta_l$. If $T > 0$, go to step 2.

5. Return the values b and $\bar{X}_b$, where $\bar{X}_b = \min_{1 \le i \le k} \bar{X}_i$.

As simulation evolves, design b, which is the design with the smaller sample mean, may change from iteration to iteration, although it will converge to the optimal design as the l goes to infinity. In addition, we need to select the initial number of simulations, n_0, and the one-time increment, Δ_l. A suitable choice for n_0 is between 5 and 20 (Law and Kelton 2000, Bechhofer et al. 1995). Also, with a small Δ_l, we need to iterate the computation procedure in step 2 many times. On the other hand, with a large Δ_l, we are putting too much confidence on the mean and variance estimators of early iterations and can result in waste of computation time to obtain an unnecessarily high confidence level of non-critical designs. Instead of using a fixed Δ_l for every iteration, we suggest computing Δ_l dynamically at each iteration

$$\Delta_l = \min(T, \max(k, \lceil T/2 \rceil)).$$

Thus, the sequential procedure allocates incremental sample sizes aggressively at earlier iterations and become less aggressive as the procedure proceeds and the computing budget becomes scarce. This way we will be able to reduce the number of iterations of step 2 without the risk of putting too much resources to simulate non-critical designs. If users do not enter the indifference amount d^*, we will set $d^* = \bar{X}_s(r) - \bar{X}_b(r)$, where $\bar{X}_b(r) = \min_{i=1}^{k} \bar{X}_i(r)$, and $\bar{X}_s(r) = \min_{i=1, i \neq b}^{k} \bar{X}_i(r)$. Note that in this case d^* becomes a random variable since $\bar{X}_i(r)$ changed from iteration to iteration. Furthermore, if the user does not specify the value of d^* and $\bar{X}_s(r) - \bar{X}_b(r)$ is small (less than the unknown d^*), then the OCBAIZ procedure will not be able to fully reduce the sample size of design i whose mean $\mu_i - \mu_{i_1} < d^*$. In general, OCBAIZ will allocate a greater proportion of the simulation budget to the estimated best design with a smaller d^*. An example implementation of this algorithm is listed in the Appendix A.

3 EMPIRICAL EXPERIMENTS

Chen et al. (2000b) present the numerical results of OCBA and other commonly used R&S procedures. They demonstrate that OCBA is more efficient in terms of sample size allocation. In this section we present some empirical results obtained from simulations using the OCBA and OCBAIZ. Since it has been shown that OCBA is a very efficient procedure for selecting the best design, we focus only on the comparison of OCBA and OCBAIZ procedures in this paper. In particular, we compare the level of empirical P(CS) which can be obtained by applying OCBA and OCBAIZ. Because OCBA does not use indifference-amount, the indifference amount d^* in OCBAIZ is set to $\bar{X}_s(r) - \bar{X}_b(r)$ at each iteration for all cases so that we can compare these two procedures fairly.

3.1 Experiment 1 Equal Variances

There are ten alternative designs in the selection subset. Suppose $X_{ij} \sim \mathcal{N}(i, 6^2)$, $i = 1, 2, \ldots, 10$. We want to select a design with the minimum mean: design 1. We set the number of initial replications $n_0 = 20$. The computing budgets are ranged from 400 to 1200 with increment size 100. Furthermore, 10,000 independent experiments are performed to estimate the actual P(CS) by $\hat{P}$(CS): the proportion of the 10,000 experiments in which we obtained the correct selection.

Figure 1 lists the results of experiment 1. OCBA obtains slightly higher $\hat{P}$(CS) than OCBAIZ does when the computing budgets are 500, 600, 700, and 1200. Tables 1 and 2 list the detailed simulation replications allocated for each design under different computing budgets. Note that the sum of the average computing budget of each design may not equal to the total computing budget T in these tables because of rounding. The number of additional simulation replications decreases as the differences $\delta_{i,b} = \bar{X}_i - \bar{X}_b (> 0)$ increase. This makes sense because as $\delta_{i,b}$ increases, it is more likely that we will conclude $\mu_i > \mu_b$. In other words, as the observed difference of sample means across alternatives $\delta_{i,b}$ increases, it is less likely that we will conclude $\mu_i < \mu_b$. In this setting, the OCBA algorithm allocates more

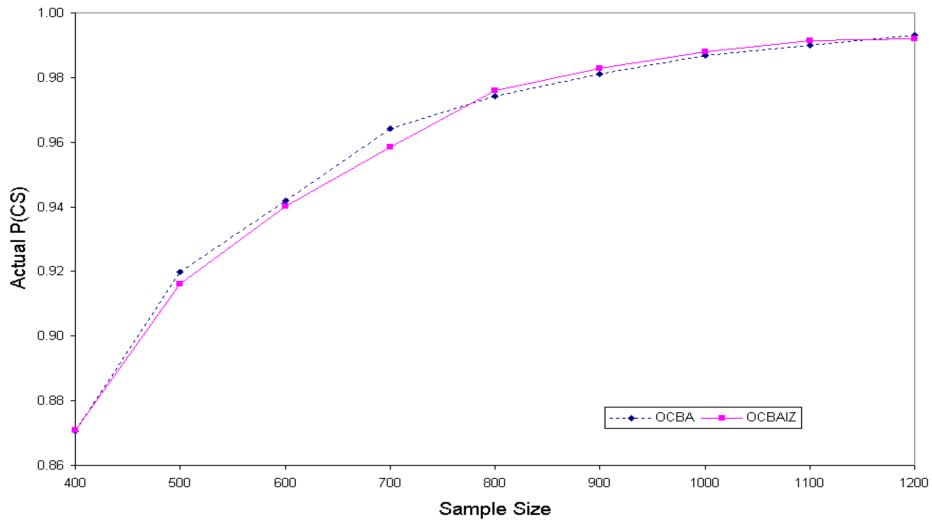

Figure 1
$\hat{P}$(CS) and Sample Sizes for Experiment 1

Table 1
Detailed Sample Sizes of OCBA of Experiment 1

Design	400	500	600	700	800	900	1000	1100	1200
1	104	144	183	221	260	298	335	373	410
2	90	127	164	201	238	275	311	348	383
3	48	61	72	84	96	107	119	130	142
4	30	36	41	47	53	59	65	70	76
5	23	26	29	33	36	39	42	46	49
6	21	22	24	26	28	30	32	34	36
7	20	21	21	22	24	25	26	28	29
8	20	20	20	21	21	22	23	24	25
9	20	20	20	20	20	21	21	22	22
10	20	20	20	20	20	20	20	21	21

Table 2
Detailed Sample Sizes of OCBAIZ of Experiment 1

Design	400	500	600	700	800	900	1000	1100	1200
1	99	137	174	211	247	284	321	356	393
2	92	129	167	204	241	278	315	351	388
3	49	62	74	87	98	110	122	135	146
4	31	37	43	49	55	61	67	73	79
5	24	27	31	34	37	41	44	48	51
6	21	23	25	27	29	31	33	35	38
7	20	21	22	23	24	25	27	28	30
8	20	20	21	21	22	23	24	24	26
9	20	20	20	20	21	21	22	22	23
10	20	20	20	20	20	20	21	21	21

Table 3
Detailed Sample Sizes of OCBA of Experiment 2

Design	400	500	600	700	800	900	1000	1100	1200
1	93	127	161	194	228	261	293	326	358
2	83	116	150	182	217	251	284	318	351
3	50	63	75	88	100	111	123	136	147
4	34	41	47	54	61	67	74	81	88
5	27	32	36	41	44	49	54	58	62
6	24	27	30	33	36	39	43	46	49
7	22	24	26	29	31	34	36	39	42
8	21	23	24	26	28	30	32	34	36
9	21	22	23	24	26	28	29	31	33
10	20	21	22	23	24	26	27	29	30

computing budget to the best design, i.e., design 1, than OCBAIZ does. In both procedures, inferior designs, for instance designs 8 through 10, are almost always excluded from further simulation, i.e., $N_i \approx n_0$.

3.2 Experiment 2 Increasing Variances

This is a variation of experiment 1. All settings are preserved except that the variance of each design increases as the mean increases. Namely, $X_{ij} \sim \mathcal{N}(i, (6+(i-1)/2)^2)$, $i = 1, 2, \ldots, 10$.

The results are in Figure 2 and Tables 3 and 4. Since most designs have larger variances than in experiment 1, $\hat{P}(\text{CS})$ are not as good. OCBA obtains slightly higher $\hat{P}(\text{CS})$ than OCBAIZ when the computing budget $T = 800$, and 1100, Both OCBA and OCBAIZ

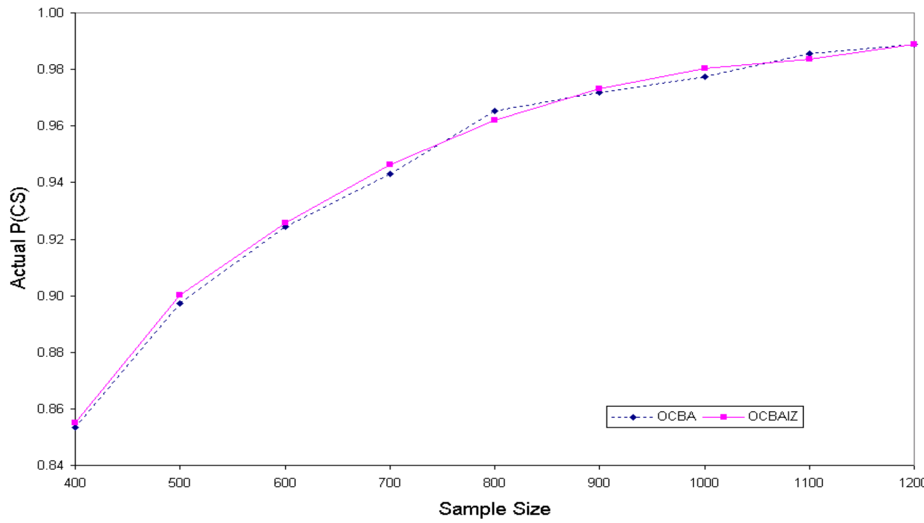

Figure 2

$\hat{P}$(CS) and Sample Sizes for Experiment 1

Table 4

Detailed Sample Sizes of OCBAIZ of Experiment 2

Design	400	500	600	700	800	900	1000	1100	1200
1	86	117	148	178	209	238	267	297	326
2	84	119	153	187	223	256	290	325	358
3	51	65	78	92	103	115	128	140	152
4	36	42	49	56	63	71	78	84	92
5	28	33	37	42	47	52	56	61	66
6	25	28	31	34	38	41	45	48	52
7	23	25	27	30	32	35	38	41	44
8	22	23	25	27	29	31	34	36	38
9	21	22	24	25	27	29	30	32	34
10	20	21	23	24	25	27	28	30	32

procedures allocate relatively more additional simulation replications for designs with larger variances. Both procedures take into consideration the difference between sample means, so $N_i < N_j$ even when $S_i(n_0) > S_j(n_0)$. Note that in this setting, OCBAIZ generally allocates more samples to design 2 than design 1, since $d_1 = d_2$ and design 2 has a larger variance. Chen and Kelton (2000) indicate that procedures that take into account the difference between sample means have the most significant reduction in the number of replications or batches (compared to Rinott's procedure) when the inferior alternatives have larger variances. In such a case, Rinott's procedure tends to allocate most computing resource to inferior designs and so is inefficient.

3.3 Experiment 3 Decreasing Variances

This is another variation of experiment 1. All settings are preserved except that the variance of each design decreases as the mean increases. Namely, $X_{ij} \sim \mathcal{N}(i, (6-(i-1)/2)^2)$, $i = 1, 2, \ldots, 10$.

The results are in Figure 3 and Tables 5 and 6. Since most designs have smaller variance, $\hat{P}(\text{CS})$ are better than in setting 1. When the computing budget $T = 600, 700, 800, 900$, and 1100, OCBAIZ obtains slightly higher $\hat{P}(\text{CS})$ than OCBA does. Both OCBA and OCBAIZ procedures allocate less additional simulation replications for designs that are clearly inferior in this setting, i.e., large sample means with small variances. Since inferior designs have smaller variances, we are confident to exclude those designs from further simulations. For instance, designs 7 through 10, are always excluded from further simulation, i.e., $N_i = n_0$. This suggests that we should use a smaller initial sample size. In this setting, the OCBAIZ algorithm allocates more computing budget to the best design, i.e., design 1, than OCBA does.

The experiments indicate that the computing budget also experience the effect of diminishing returns. For example, $\hat{P}(\text{CS})$ increases by more than 0.03 in these experiments when the computing budget is increased from 400 to 500, while the increase in $\hat{P}(\text{CS})$ is no more than 0.006 when the computing budget is increased from 1100 to 1200. Thus, if the objective is to minimize sample sizes and the given P^* is a small value, then analysts should consider a higher P^* since the marginal cost is small.

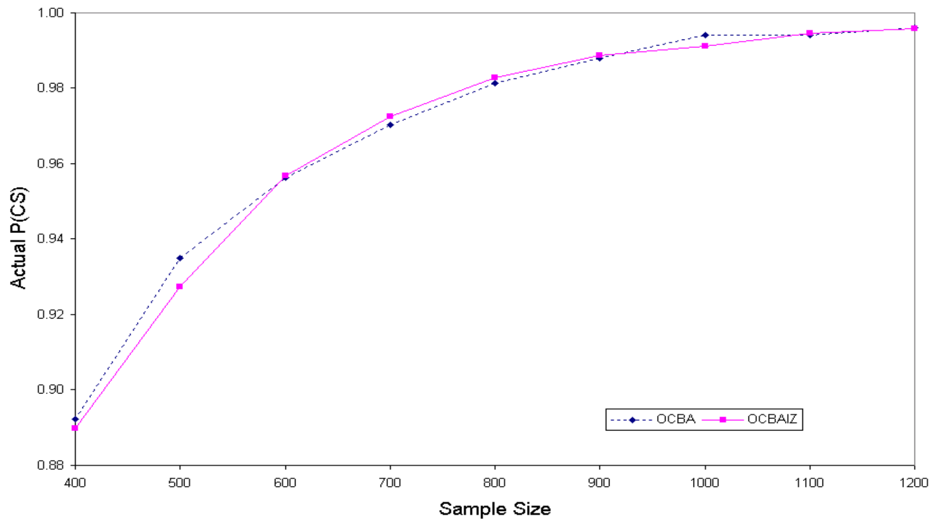

Figure 3
$\hat{P}$(CS) and Sample Sizes for Experiment 3

Table 5
Detailed Sample Sizes of OCBA of Experiment 3

Design	400	500	600	700	800	900	1000	1100	1200
1	114	159	203	247	291	335	378	422	465
2	95	134	174	213	253	292	330	370	409
3	43	54	65	76	86	96	109	117	128
4	25	29	33	37	41	45	50	54	58
5	20	21	22	24	26	27	29	31	33
6	20	20	20	20	21	21	22	22	23
7	20	20	20	20	20	20	20	20	20
8	20	20	20	20	20	20	20	20	20
9	20	20	20	20	20	20	20	20	20
10	20	20	20	20	20	20	20	20	20

Table 6
Detailed Sample Sizes of OCBAIZ of Experiment 3

Design	400	500	600	700	800	900	1000	1100	1200
1	115	160	204	249	293	338	382	426	471
2	94	132	171	210	248	286	324	362	400
3	44	55	66	76	87	98	108	119	129
4	25	29	33	38	42	46	51	55	59
5	20	21	23	24	26	28	30	32	34
6	20	20	20	20	21	21	22	23	23
7	20	20	20	20	20	20	20	20	20
8	20	20	20	20	20	20	20	20	20
9	20	20	20	20	20	20	20	20	20
10	20	20	20	20	20	20	20	20	20

4 CONCLUSIONS

We have developed a highly efficient procedure to identify a good design out of k alternatives. The purpose of this technique is to further enhance the efficiency of ranking and selection in simulation experiments. The objective is to maximize the simulation efficiency, expressed as P(CS) within a given computing budget. The incremental sample sizes at an iteration and for each design are computed dynamically according to the sample means, the sample variances, and the available computing budget at each iteration. Our procedure allocates replications in such a way that optimally improves P(CS).

The performance difference between OCBA and OCBAIZ is minor. However, the incremental sample sizes for each design at each iteration are easier to compute in OCBAIZ than OCBA. Moreover, OCBAIZ is able to explore the information of the indifference amount and estimate the total sample size for each design when the objective is to minimize computing budget given a required minimal P(CS), which can improve the computation efficiency. Both OCBA and OCBAIZ are based on the Bonferroni inequality and are valid with the variance reduction technique of common random numbers. While ordinal optimization can converge exponentially fast, our simulation budget allocation procedure provides a way to further improve overall simulation efficiency. The techniques presented in this paper can be considered as a pre-processing step that precedes any other optimization or search techniques.

APPENDIX A: An example implementation of OCBA

```
void ocba(float* s_mean,float* s_var,int nd,int* n,int add_budget,int* an,int
iz, float d)
/* s_mean[i]: sample mean of design i, i=0,1,..,nd-1
   s_var[i]: sample variance of design i, i=0,1,..,nd-1
   nd: the number of designs
   n[i]: number of simulation replication of design i, i=0,1,..,nd-1
   add_budget: the additional simulation budget
   an[i]: additional number of simulation replication assigned to design i,
          i=0,1,..,nd-1
   iz: a boolean to indicate whether to use indifference zone
   d: the indifference amount                                              */

{
int i,j;
int b, s;
int t_budget,t1_budget;
int more_alloc; /* 1:Yes; 0:No */
int *morerun;
float ratio_s, temp, temp1;
float *ratio;

      morerun  = (int   *) calloc(nd, sizeof(int));
      ratio    = (float *) calloc(nd, sizeof(float));

      t_budget=add_budget;
      for (i=0; i<nd; i++) t_budget+=n[i];
      b=0;
      for (i=1;i<nd;i++)  /* search the best design */
        if (s_mean[i] < s_mean[b]) b=i;

      if (b==0) s=1; else s=0;
      for (i=0;i<nd;i++) /* search the second best design */
        if (s_mean[i] < s_mean[s] && i!=b)
          s=i;
      ratio[s]=1.0;
      for (i=0;i<nd;i++)
        if (i!=s && i!=b) {
          if (iz) { /* use indifference zone */
                  temp=s_mean[s]-s_mean[b];
                  if (temp < d) temp=d;
                  temp1=s_mean[i]-s_mean[b];
                  if (temp1 < d) temp1=d;
                  temp/=temp1;
          } else  /* ocba */
            temp=(s_mean[s]-s_mean[b])/(s_mean[i]-s_mean[b]);
          ratio[i]=temp*temp*s_var[i]/s_var[s];
        }   /* calculate ratio of Ni/Ns*/

      if (iz) { /* use indifference zone */
         ratio[b] = s_var[b]/s_var[s];
      } else {  /* ocba */
         temp=0;
         for(i=0;i<nd;i++) if(i!=b) temp+=(ratio[i]*ratio[i]/s_var[i]);
         ratio[b]=sqrt(s_var[b]*temp); /* calculate Nb/Ns */
      };

      for(i=0;i<nd;i++) morerun[i]=1;
      t1_budget=t_budget;
```

```
      do{
        more_alloc=0;
        ratio_s=0.0;
        for(i=0;i<nd;i++) if(morerun[i]) ratio_s+=ratio[i];
        for(i=0;i<nd;i++) if(morerun[i]) {
            an[i]=(int)(t1_budget/ratio_s*ratio[i]);
            /* disable thoese design which have been run too much */
            if(an[i]<n[i]) {
               an[i]=n[i];
               morerun[i]=0;
               more_alloc=1;
            }
         }
         if (more_alloc) {
            t1_budget=t_budget;
            for(i=0;i<nd;i++) if(!morerun[i]) t1_budget-=an[i];
         }
       } while(more_alloc); /* end of WHILE */
                                           /* calculate the difference */
       t1_budget=an[0];
       for(i=1;i<nd;i++) t1_budget+=an[i];
       an[b]+=(t_budget-t1_budget);       /* give the difference to design b */
       for(i=0;i<nd;i++) an[i]-=n[i];
} /* ocba */
```

REFERENCES

Banks, J. 1998. *Handbook of Simulation*. New York: John Wiley & Sons, Inc.

Bechhofer, R. E., T. J. Santner, and D. M. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. New York: John Wiley & Sons, Inc.

Chen, C. H. 1995. An Effective Approach to Smartly Allocate Computing Budget for Discrete Event Simulation. *Proceedings of the 34th IEEE Conference on Decision and Control*, 2598–2605.

Chen, H. C., C. H. Chen, and E. Yücesan. 2000a. Computing Efforts Allocation for Ordinal Optimization and Discrete Event Simulation. *IEEE Transactions on Automatic Control*, **45**(5), 960-964.

Chen, E. J., W. D. Kelton. 2000. An Enhanced Two-Stage Selection Procedure. *Proceedings of the 2000 Winter Simulation Conference*, 727–735.

Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick. 2000b. Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization. *Journal of Discrete Event Dynamic Systems*, **10**(3), 251–270.

Chick, S. E., K. Inoue. 2001. New Two-Stage and Sequential Procedures for Selecting The

Best System. *Operations Research*, **49**, 732-743.

Dai, L., 1996. Convergence Properties of Ordinal Comparison in the Simulation of Discrete Even Dynamic Systems. *Journal of Opt. Theory and Applications*, **91**(2), 363–388.

Dudewicz, E. J., S. R. Dalal. 1975. Allocation of Observations in Ranking and Selection with Unequal Variances. *Sankhya*, **B37**, 28–78.

Goldsman, D., S-H. Kim, W. S. Marshall, and B. L. Nelson. 2002. Ranking and Selection for Steady-State Simulation: Procedures and Perspectives, *INFORMS Journal on Computing* **4** (1), 2–19.

Ho, Y. C., R. S. Sreenivas, and P. Vakili. 1992. Ordinal Optimization of DEDS. *Journal of Discrete Event Dynamic Systems*, **2**, 61–68.

Kushner, H. J., D. S. Clark. 1978. *Stochastic Approximation for Constrained and Unconstrained Systems*. New York: Springer-Verlag.

Law, A. M, W. D. Kelton. 2000. *Simulation Modeling and Analysis*. Third ed. New York: McGraw-Hill.

Nelson, B. L, J. Swann, D. Goldsman, and W. Song. 2001. Simple Procedures for Selecting the Best Simulated System when the Number of Alternative is Large. *Operations Research*, **49**, 950-963.

Rinott, Y., 1978. On Two-stage Selection Procedures and Related Probability Inequalities. *Communications in Statistics*, **A7**, 799-811.

Tong, Y. L. 1980. *Probability Inequalities in Multivariate Distributions.* New York: Academic Press.

Wilcox, R. R., 1984. A Table for Rinott's Selection Procedure. *Journal of Quality Technology*, **16**(2), 97-100.