Discrete Event Dynamic Systems: Theory and Applications, 10, 251–270, 2000.

Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization

CHUN-HUNG CHEN, JIANWU LIN
Department of Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104-6315

ENVER YÜCESAN
Technology Management Area INSEAD Fontainebleau, France

STEPHEN E. CHICK
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117

Abstract. Ordinal Optimization has emerged as an efficient technique for simulation and optimization. Exponential convergence rates can be achieved in many cases. In this paper, we present a new approach that can further enhance the efficiency of ordinal optimization. Our approach determines a highly efficient number of simulation replications or samples and significantly reduces the total simulation cost. We also compare several different allocation procedures, including a popular two-stage procedure in simulation literature. Numerical testing shows that our approach is much more efficient than all compared methods. The results further indicate that our approach can obtain a speedup factor of higher than 20 above and beyond the speedup achieved by the use of ordinal optimization for a 210-design example.

Keywords: discrete-event simulation, stochastic optimization, ordinal optimisation queuing network

1. Introduction

Discrete-event systems (DES) simulation is a popular tool for analyzing systems and evaluating decision problems since real situations rarely satisfy the assumptions of analytical models. While DES simulation has many advantages for modeling complex systems, efficiency is still a significant concern when conducting simulation experiments (Law and Kelton, 1991). To obtain a good statistical estimate for a design decision, a large number of simulation samples or replications is usually required for each design alternative. This is due to the slow convergence of a performance measure estimator relative to the number of simulation samples or replications. The ultimate accuracy (typically expressed as a confidence interval) of this estimator cannot improve faster than $O(1/\sqrt{N})$, the result of averaging i.i.d. noise, where N is the number of simulation samples or replications (Fabian, 1971; Kushner and Clark, 1978). 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.

Ordinal Optimization has emerged as an efficient technique for simulation and optimization. The underlying philosophy is to obtain good estimates through ordinal comparison while the value of an estimate is still very poor (Ho et al., 1992). If our goal is to find the good designs rather than to find an accurate estimate of the best performance value, which is true in many practical situations, it is advantageous to use ordinal comparison for selecting a good design. Further Dai (1996) shows that the convergence rate for ordinal optimization can be exponential. This idea has been successfully applied to several problems (e.g., Cassandras et al., 1998; Gong et al., 1999; Patsis et al., 1997).

While ordinal optimization could significantly reduce the computational cost for DES simulation, there is potential to further improve its performance by intelligently controlling the simulation experiments, or by determining the best number of simulation samples among different designs as simulation proceeds. The main theme of this paper is to further enhance the efficiency of ordinal optimization in simulation experiments. As we will show in the numerical testing in section 4, the speedup factor can be another order of magnitude above and beyond the exponential convergence of ordinal optimization.

Intuitively, to ensure a high probability of correctly selecting a good design or a high alignment probability in ordinal optimization, a larger portion of the computing budget should be allocated to those designs that are critical in the process of identifying good designs. In other words, a larger number of simulations must be conducted with those critical designs in order to reduce estimator variance. On the other hand, limited computational effort should be expended on non-critical designs that have little effect on identifying the good designs even if they have large variances. Overall simulation efficiency is improved as less computational effort is spent on simulating non-critical designs and more is spent on critical designs. Ideally, one would like to allocate simulation trials to designs in a way that maximizes the probability of selecting the best design within a given computing budget. We present a new optimal computing budget allocation (OCBA) technique to accomplish this goal.

Previous researchers have examined various approaches for efficiently allocating a fixed computing budget across design alternatives. Chen (1995) formulates the procedure of allocating computational efforts as a nonlinear optimization problem. Chen et al. (1996) apply the steepest-ascent method to solve the budget allocation problem. The major drawback of the steepest-ascent method is that an extra computation cost is needed to iteratively search for a solution to the budget allocation problem. Such an extra cost could be significant if the number of iterations is large. Chen et al. (1997) introduce a greedy heuristic to solve the budget allocation problem. This greedy heuristic iteratively determines which design appears to be the most promising for further simulation. However, the budget allocation selected by the greedy heuristic is not necessarily optimal. On the other hand, Chen et al. (2000) replace the objective function with an approximation and the use of Chernoff's bounds, and present an analytical solution to the approximation. The approach in Chen et al. (2000) provides a more efficient allocation than the greedy approach and the steepest-ascent method.

In this paper, we develop a new asymptotically optimal approach for solving the budget allocation problem. The presented approach is even more efficient than the one given by Chen et al. (2000). This is accomplished by replacing the objective function with a better approximation that can be solved analytically. Further, Chernoff's bounds are not used in the derivation and fewer assumptions are imposed. The higher efficiency of this new allocation approach is also shown in the numerical testing.

In addition to presenting a new and more efficient approach to determine the simulation budget allocation, in this paper, we will i) compare several different budget allocation procedures through a series of numerical experiments; ii) demonstrate that our budget allocation approaches are much more efficient than the popular two-stage Rinott procedure; iii) show that our approach is robust and the most efficient in different settings; iv) show that the speedup factor becomes even larger when the number of designs increase; v) demonstrate that additional significant speedup can be achieved above and beyond the exponential convergence of ordinal optimization.

The paper is organized as follows: In the next section, we formulate the optimal computing budget allocation problem. Since our approach is based on the Bayesian model, we also provide a brief discussion of that model for completeness. Section 3 presents an asymptotic allocation rule for OCBA. The performance of the technique is illustrated with a series of numerical examples in Section 4. Section 5 concludes the paper.

2. Problem Statement

Suppose we have a complex discrete event system. A general simulation and optimization problem for such a DES system can be defined as

$$\min_{\theta_i \in \Theta} J(\theta_i) \equiv E[L(\theta_i, \xi)] \tag{1}$$

where Θ, the search space, is an arbitrary, huge, structureless but finite set; θ_i is the system design parameter vector for design i, $i = 1, 2, \ldots, k$; J, the performance criterion which is the expectation of L, the sample performance, as a functional of θ, and ξ, a random vector that represents uncertain factors in the systems. Note that for the complex systems considered in this paper, $L(\theta, \xi)$ is available only in the form of a complex calculation via simulation. The system constraints are implicitly involved in the simulation process, and so are not shown in (1). The standard approach is to estimate $\mathrm{E}[L(\theta_i, \xi)]$ by the sample mean performance measure

$$\bar{J}_i \equiv \frac{1}{N_i} \sum_{j=1}^{N_i} L(\theta_i, \xi_{ij}),$$

where ξ_{ij} represents the j-th sample of ξ and N_i represents the number of simulation samples for design i. Denote by

σ_i^2: the variance for design i, i.e., $\sigma_i^2 = \mathrm{Var}(L(\theta_i, \xi))$. In practice, σ_i^2 is unknown beforehand and so is approximated by sample variance.

b: the design having the smallest sample mean performance measure, i.e., $\bar{J}_b \le \min_i \bar{J}_i$,

$\delta_{b,i} \equiv \bar{J}_b - \bar{J}_i$.

As N_i increases, $\bar{J}_i$ becomes a better approximation to $J(\theta_i)$ in the sense that its corresponding confidence interval becomes narrower. The ultimate accuracy of this estimate cannot improve faster than $1/\sqrt{N}$. Note that each sample of $L(\theta_i, \xi_{ij})$ requires one simulation run. A large number of required samples of $L(\theta_i, \xi_{ij})$ for all designs may become very time consuming. On the other hand, ordinal optimization (Ho et al., 1992) concentrates on ordinal comparison and achieves a much faster convergence rate. Dai (1996) shows that an alignment probability of ordinal comparison can converge to 1.0 exponentially fast in most cases. Such an alignment probability is also called the probability of correct selection or $P\{CS\}$. One example of $P\{CS\}$ is the probability that design b is actually the best design. To take advantage of such an exponential convergence, our approach is developed under the framework of ordinal comparison. Furthermore, instead of equally simulating all designs, we will further improve the performance of ordinal optimization by determining the best numbers of simulation samples for each design. Stating this more precisely, we wish to choose $N_1, N_2, \ldots, N_k$ such that $P\{CS\}$ is maximized, subject to a limited computing budget T,

$$\begin{aligned} &\max_{N_1,\ldots,N_k} P\{CS\} \\ &\quad s.t.\ N_1 + N_2 + \cdots + N_k = T. \\ &\quad N_i \in N, i = 1, \ldots, k. \end{aligned} \tag{2}$$

Here N is the set of non-negative integers and $N_1 + N_2 + \cdots + N_k$ denotes the total computational cost assuming the simulation times for different designs are roughly the same. To solve problem (2), we must be able to estimate $P\{CS\}$. There exists a large literature on assessing $P\{CS\}$ based on classical statistical models (e.g., Goldsman and Nelson, 1994; Banks, 1998 give an excellent survey on available approaches). However, most of these approaches are only suitable for problems with a small number of designs. Recently, Chen (1996) introduced an estimation technique that approximates $P\{CS\}$ for ordinal comparison when the number of designs is large based on a Bayesian model (Bernardo and Smith, 1994). This technique has the added benefit of providing sensitivity information that is useful in solving problem (2). We will incorporate this technique within our budget allocation approach.

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. In this paper we assume that the simulation output, $L(\theta, \xi)$, is normally distributed. However, we will demonstrate that our approach works equally well when for non-normal distributions.

After the simulation is performed, a posterior distribution of $J(\theta_i)$, $p(J(\theta_i) | L(\theta_i, \xi_{ij}), j = 1, \ldots, N_i)$, can be constructed based on two pieces of information: (i) prior knowledge of the system's performance, and (ii) current simulation output. If we select the observed best design (design b), the probability that we selected the best design is

$$\begin{aligned} P\{CS\} &= P\{\text{design } b \text{ is actually the best design}\} \\ &= P\{J(\theta_b) < J(\theta_i), i \neq b \mid L(\theta_i, \xi_{ij}), j = 1, \ldots, N_i, i = 1, 2, \ldots, k\}. \end{aligned} \tag{3}$$

To simplify the notation used, we rewrite Eq. (3) as $P\{\tilde{J}_b < \tilde{J}_i, i \neq b\}$, where $\tilde{J}_i$ denotes the random variable whose probability distribution is the posterior distribution for design i. Assume that the unknown mean $J(\theta_i)$ has the conjugate normal prior distribution. We consider non-informative prior distributions. This implies that no priori knowledge is given about the performance of any design alternative before conducting the simulation. In that case, DeGroot (1970) shows that the posterior distribution of $J(\theta_i)$ is

$$\tilde{J}_i \sim N\left(\bar{J}_i, \frac{\sigma_i^2}{N_i}\right).$$

After the simulation is performed, $\bar{J}_i$ can be calculated, σ_i^2 can be approximated by the sample variance; $P\{CS\}$ can then be estimated using a Monte Carlo simulation. However, estimating $P\{CS\}$ via Monte Carlo simulation is time-consuming. 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. We adopt a common approximation procedure used in simulation and statistics literature (Brately et al., 1987; Chick, 1997; Law and Kelton, 1991). This approximation is based on the Bonferroni inequality.

Let Y_i be a random variable. According to the Bonferroni inequality, $P\{\cap_{i=1}^k (Y_i < 0)\} \geq 1 - \sum_{i=1}^k [1 - P\{Y_i < 0\}]$. In our case, Y_i is replaced by $(\tilde{J}_b - \tilde{J}_i)$ to provide a lower bound for the probability of correct selection. That is,

$$\begin{aligned} P\{CS\} &= P\left\{\bigcap_{i=1, i\neq b}^{k} \left(\tilde{J}_b - \tilde{J}_i < 0\right)\right\} \geq 1 - \sum_{i=1, i\neq b}^{k} \left[1 - P\left\{\tilde{J}_b - \tilde{J}_i < 0\right\}\right] \\ &= 1 - \sum_{i=1, i\neq b}^{k} P\left\{\tilde{J}_b > \tilde{J}_i\right\} = APCS. \end{aligned}$$

We refer to this lower bound of the correct selection probability as the *Approximate Probability of Correct Selection* (*APCS*). *APCS* can be computed very easily and quickly; no extra Monte Carlo simulation is needed. Numerical tests show that the APCS approximation can still lead to highly efficient procedures (e.g., Chen, 1996; Inoue and Chick, 1998). We therefore use *APCS* to approximate $P\{CS\}$ as the simulation experiment proceeds. More specifically, we consider the following problem:

$$\begin{aligned} &\max_{N_1,\ldots,N_k} 1 - \sum_{i=1, i\neq b}^{k} P\left\{\tilde{J}_b > \tilde{J}_i\right\} \\ &\text{s.t.} \sum_{i=1}^{k} N_i = T \text{ and } N_i \geq 0. \end{aligned} \tag{4}$$

In the next section, an asymptotic allocation rule with respect to the number of simulation replications, N_i will be presented.

3. An Asymptotic Allocation Rule

First, we assume the variables, N_i's, are continuous. Second, our strategy is to tentatively ignore all non-negativity constraints; all N_i's can therefore assume any real value. Let $N_i = \alpha_i T$. Thus, $\sum_{i=1}^k \alpha_i = 1$. Before the end of this section, we will show how all α_i's become positive and hence all N_i's are positive. Based on this idea, we first consider the following:

$$\max_{N_1,\ldots,N_k} 1 - \sum_{i=1, i\neq b}^{k} P\left\{\tilde{J}_b > \tilde{J}_i\right\}$$

$$\text{s.t.} \sum_{i=1}^{k} N_i = T. \tag{5}$$

For the objective function,

$$\begin{aligned}
\sum_{i=1, i\neq b}^{k} P\{\tilde{J}_b > \tilde{J}_i\} &= \sum_{\substack{i=1\\ i\neq b}}^{k} \int_0^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{b,i}}^{\frac{(x-\delta_{b,i})^2}{2\sigma_{b,i}^2}} dx \\
&= \sum_{\substack{i=1\\ i\neq b}}^{k} \int_{-\frac{\delta_{b,i}}{\sigma_{b,i}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{t^2}{2}} dt,
\end{aligned}$$

where a new variable is introduced, $\sigma_{b,i}^2 = \frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}$, for notation simplification ($\delta_{b,i}$ is defined in section 2). Then, let F be the Lagrangian relaxation of (5):

$$F = 1 - \sum_{\substack{i=1\\ i\neq b}}^{k} \int_{-\frac{\delta_{b,i}}{\sigma_{b,i}}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{t^2}{2}} dt - \lambda \left(\sum_{i=1}^{k} N_i - T \right).$$

Furthermore, the Karush-Kuhn-Tucker (KKT) (Walker, 1999) conditions of this problem can be stated as follows.

$$\begin{aligned}
\frac{\partial F}{\partial N_i} &= \frac{\partial F}{\partial\left(-\frac{\delta_{b,i}}{\sigma_{b,i}}\right)} \frac{\partial\left(-\frac{\delta_{b,i}}{\sigma_{b,i}}\right)}{\partial \sigma_{b,i}} \frac{\partial \sigma_{b,i}}{\partial N_i} - \lambda \\
&= \frac{-1}{2\sqrt{2\pi}} \exp\left[\frac{-\delta_{b,i}^2}{2\sigma_{b,i}^2}\right] \frac{\delta_{b,i}\sigma_i^2}{N_i^2(\sigma_{b,i}^2)^{3/2}} - \lambda = 0, \text{ for } i = 1, 2, \ldots, k, \text{ and } i \neq b. \qquad (6) \\
\frac{\partial F}{\partial N_b} &= \frac{-1}{2\sqrt{2\pi}} \sum_{\substack{i=1\\ i\neq b}}^{k} \exp\left[\frac{-\delta_{b,i}^2}{2\sigma_{b,i}^2}\right] \frac{\delta_{b,i}\sigma_b^2}{N_b^2(\sigma_{b,i}^2)^{3/2}} - \lambda = 0, \qquad (7) \\
&\lambda\left(\sum_{i=1}^{k} N_i - T\right) = 0, \text{ and } \lambda \geq 0.
\end{aligned}$$

We now examine the relationship between N_b and N_i for $i = 1, 2, \ldots, k$, and $i \neq b$. From Eq. (6),

$$\frac{-1}{2\sqrt{2\pi}} \exp\left[\frac{-\delta_{b,i}^2}{2\sigma_{b,i}^2}\right] \frac{\delta_{b,i}}{(\sigma_{b,i}^2)^{3/2}} = -\lambda \frac{N_i^2}{\sigma_i^2}, \text{ for } i = 1, 2, \ldots, k, \text{ and } i \neq b. \tag{8}$$

Plugging (8) into (7), we have

$$\sum_{\substack{i=1\\i\neq b}}^{k} \frac{-\lambda N_i^2 \sigma_b^2}{N_b^2 \sigma_i^2} - \lambda = 0.$$

Then

$$N_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{N_i^2}{\sigma_i^2}} \quad \text{or} \quad \alpha_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{\alpha_i^2}{\sigma_i^2}}. \tag{9}$$

We further investigate the relationship between N_i and N_j, for any $i, j \in \{1, 2, \ldots, k\}$, and $i \neq j \neq b$. From Eq. (6),

$$\exp\left(\frac{-\delta_{b,i}^2}{2\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}\right)}\right) \cdot \frac{\delta_{b,i}\sigma_i^2/N_i^2}{\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_i^2}{N_i}\right)^{3/2}} = \exp\left(\frac{-\delta_{b,i}^2}{2\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_j^2}{N_j}\right)}\right) \cdot \frac{\delta_{b,j}\sigma_j^2/N_j^2}{\left(\frac{\sigma_b^2}{N_b} + \frac{\sigma_j^2}{N_j}\right)^{3/2}}. \tag{10}$$

To reduce the total simulation time for identifying a good design, it is worthwhile to concentrate the computational effort on good designs. Namely, N_b should be increased relative to N_i, for $i = 1, 2, \ldots, k$, and $i \neq b$. This is indeed the case in the actual simulation experiments. And this assumption can be supported by considering a special case in Eq. (9): When $\sigma_1 = \sigma_2 = \cdots = \sigma_k$,

$$N_b = \sqrt{\sum_{i=1, i\neq b}^{k} N_i^2}.$$

Therefore, we assume that $N_b \gg N_i$, which enables us to simplify Eq. (10) as

$$\exp\left(\frac{-\delta_{b,i}^2}{2\left(\frac{\sigma_i^2}{N_i}\right)}\right) \cdot \frac{\delta_{b,i}\sigma_i^2/N_i^2}{\left(\frac{\sigma_i^2}{N_i}\right)^{3/2}} = \exp\left(\frac{-\delta_{b,j}^2}{2\left(\frac{\sigma_j^2}{N_j}\right)}\right) \cdot \frac{\delta_{b,j}\sigma_j^2/N_j^2}{\left(\frac{\sigma_j^2}{N_j}\right)^{3/2}}.$$

On rearranging terms, the above equation becomes

$$\exp\left(\frac{1}{2}\left(\frac{\delta_{b,j}^2}{\frac{\sigma_j^2}{N_j}} - \frac{\delta_{b,i}^2}{\frac{\sigma_i^2}{N_i}}\right)\right) \frac{N_j^{1/2}}{N_i^{1/2}} = \frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}.$$

Taking the natural log on both sides, we have

$$\frac{\delta_{b,j}^2}{\sigma_j^2} N_j + \log(N_j) = \frac{\delta_{b,i}^2}{\sigma_i^2} N_i + \log(N_i) + 2\log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right),$$

or

$$\frac{\delta_{b,j}^2}{\sigma_j^2}\alpha_j T + \log(\alpha_j T) = \frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i T + \log(\alpha_i T) + 2\log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right),$$

which yields

$$\frac{\delta_{b,j}^2}{\sigma_j^2}\alpha_j T + \log(\alpha_j) = \frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i T + \log(\alpha_i) + 2\log\left(\frac{\delta_{b,j}\sigma_i}{\delta_{b,i}\sigma_j}\right). \tag{11}$$

To further facilitate the computations, we intend to find an asymptotic allocation rule. Namely, we consider the case that $T \to \infty$. While it is impossible to have an infinite computing budget in real life, our allocation rule provides a simple means for allocating simulation budget in a way that the efficiency can be significantly improved, as we will demonstrate in numerical testing later. As $T \to \infty$, all the log terms become much smaller than the other terms and are negligible. This implies

$$\frac{\delta_{b,j}^2}{\sigma_j^2}\alpha_j = \frac{\delta_{b,i}^2}{\sigma_i^2}\alpha_i$$

Therefore, we obtain the ratio between α_i and α_j or between N_i and N_j as:

$$\frac{N_i}{N_j} = \frac{\alpha_i}{\alpha_j} = \left(\frac{\sigma_i/\delta_{b,i}}{\sigma_j/\delta_{b,j}}\right)^2 \text{ for } i = 1, 2, \ldots, k, \text{ and } i \neq j \neq b. \tag{12}$$

Now we return to the issue of nonnegative constraint for N_i, which we temporarily ignored. Note that from Eq. (9) and Eq. (12), all α_i's have the same sign. Since $\sum_{i=1}^{k} \alpha_i = 1$ and $N_i = \alpha_i T$, it implies that all α_i's ≥ 0, and hence N_i's ≥ 0, where $i = 1, 2, \ldots, k$.

In conclusion, if a solution satisfies Eq. (9) and Eq. (12), then KKT conditions must hold. According to the KKT Sufficient Condition, this solution is a local optimal solution to Eq. (4). We therefore have the following result:

THEOREM 1 *Given a total number of simulation samples T to be allocated to k competing designs whose performance is depicted by random variables with means $J(\theta_1), J(\theta_2), \ldots, J(\theta_k)$, and finite variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2$ respectively, as $T \to \infty$, the Approximate Probability of Correct Selection (APCS) can be asymptotically maximized when*

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

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

where N_i is the number of samples allocated to design i, $\delta_{b,i} = \bar{J}_b - \bar{J}_i$, and $\bar{J}_b \leq \min_i \bar{J}_i$.

Remark 1. In the case of $k = 2$ and $b = 1$, Theorem 1 yields

$$N_1 = \sigma_1 \sqrt{\frac{N_2^2}{\sigma_2^2}},$$

Therefore,

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

This evaluated result is identical to the well-known optimal allocation solution for $k = 2$.

Remark 2. To gain better insight into this approach, consider another case where $k = 3$ and $b = 1$, which yields

$$\frac{N_2}{N_3} = \frac{\sigma_2^2 \delta_{1,3}^2}{\sigma_3^2 \delta_{1,2}^1}.$$

We can see how the number of simulation samples for design 2, N_2, is affected by different factors. For a minimization problem, when $\bar{J}_3$ increases (we are more confident of the difference between design 1 and design 3) or σ_2 increases (we are less certain about design 2), N_2 increases as well. On the other hand, when $\bar{J}_2$ increases (we are more confident of the difference between design 1 and design 2) or σ_3 increases (we are less certain about design 3), N_2 decreases. The above relationship between N_2 and other factors is consistent with our intuition.

With Theorem 1, we now present a cost-effective sequential approach based on OCBA to select the best design from k alternatives with a given computing budget. Initially, n_0 simulation replications for each of 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, Δ, is allocated based on Theorem 1 at each stage. Ideally, each new replication should bring us closer to the optimal solution. This procedure is continued until the total budget T is exhausted. The algorithm is summarized as follows.

A Sequential Algorithm for Optimal Computing Budget Allocation (OCBA)

Step 0. Perform n_0 simulation replications for all designs; $l \leftarrow 0$; $N_1^l = N_2^l = \cdots = N_k^l = n_0$.

Step 1. If $\sum_{i=1}^{k} N_i^l \geq T$, stop.

Step 2. Increase the computing budget (i.e., number of additional simulations) by Δ and compute the new budget allocation, $N^{l+1}1$, $N_2^{l+1}, \ldots, N_k^{l+1}$, using Theorem 1.

Step 3. Perform additional $\max(0, N_i^{l+1} - N_i^l)$ simulations for design i, $i = 1, \ldots, k$.

$l \leftarrow l + 1$. Go to Step 1.

In the above algorithm, l is the iteration number. As simulation evolves, design b, which is the design with the largest sample mean, may change from iteration to iteration, although it will converge to the optimal design as the l goes to infinity. When b changes, Theorem 1 is directly applied in step 2. However, the older design b may not be simulated at all in this iteration in step 3 due to extra allocation to this design in earlier iterations.

In addition, we need to select the initial number of simulations, n_0, and the one-time increment, Δ. Chen et al. (1999) offers detailed discussions on the selection. It is well understood that n_0 cannot be too small as the estimates of the mean and the variance may be very poor, resulting in premature termination of the comparison. A suitable choice for n_0 is between 5 and 20 (Law and Kelton, 1991; Bechhofer et al., 1995). Also, a large Δ can result in waste of computation time to obtain an unnecessarily high confidence level. On the other hand, if Δ is small, we need to the computation procedure in step 2 many times. A suggested choice for Δ is a number bigger than 5 but smaller than 10% of the simulated designs.

4. Numerical Testing and Comparison with Other Allocation Procedures

In this section, we test our OCBA algorithm and compare it with several different allocation procedures by performing a series of numerical experiments. We also apply our OCBA to a buffer resource allocation problem, which has 210 design alternatives.

4.1. Different Allocation Procedures

In addition to the OCBA algorithm, we test several procedures and compare their performances. Among them, equal allocation represents the sole use of ordinal optimization, the greedy allocation and CCY are developed based a same Bayesian framework given in section 2, and Rinott is highly popular in simulation literature. We briefly summarize the compared allocation procedures as follows.

Equal Allocation

This is the simplest way to conduct simulation experiments and has been widely applied. The simulation budget is equally allocated to all designs. Namely, all designs are equally simulated and then we focus on ordinal comparison. Such a way is equivalent to the sole use of ordinal optimization. As we noted in previous sections, ordinal optimization can ensure that $P\{CS\}$ converges to 1.0 exponentially fast even if we simulate all design alternatives equally, that is, $N_i = T/k$ for each i. The performance of equal allocation will serve as a benchmark for comparison.

Greedy Allocation

The greedy procedure offered by Chen et al. (1997) is developed based on the Bayesian framework presented in section 2. They introduce an approach for approximately estimating the gradient of $P\{CS\}$ with respect to N_i. Since we intend to maximize the resulting $P\{CS\}$, a greedy approach selects and simulates a subset of promising designs in each iteration, then repeats the process until the total budget is exhausted. Since the gradient of $P\{CS\}$ with respect to N_i is an indication of how much $P\{CS\}$ can be improved if we perform additional simulations on design i, the promising designs are defined as those which have large gradients. More specifically, in each iteration, we select the set of designs with top-m largest gradients. Then the computing budget for this iteration is equally allocated to these m designs. Obviously, the budget allocation selected by the greedy heuristic is not necessarily optimal.

Chen, Chen and Yücesan Procedure (CCY)

This is also developed based on the Bayesian framework presented in section 2 and is proposed by Chen et al. (2000). Similar to the OCBA algorithm, CCY is an asymptotic solution to an approximation problem. However, Chernoff's bounds are used and further assumptions are imposed in the development of CCY. At a theoretical level, the OCBA algorithm is superior to CCY. As we show later in the numerical experiments, the OCBA algorithm indeed performs better than the CCY procedure, although both outperform other compared procedures. CCY allocates simulation budget according to:

$$(1)\quad \frac{N_i}{N_s} = \left(\frac{\sigma_i/\delta_{b,i}}{\sigma_s/\delta_{b,s}}\right)^2 \text{ for } i = 1, \ldots, k \text{ and } i \neq s \neq b,$$

$$(2)\quad \frac{N_b}{N_s} = \frac{\sigma_b}{\sigma_s}\left[\sum_{\substack{i=1\\ i\neq b}}^{k}\left(\frac{\delta_{b,s}^2}{\delta_{b,i}^2}\right)\right]^{1/2},$$

where s is the design having the second smallest sample mean performance measure.

Two-Stage Rinott Procedure

The two-stage procedure of Rinott (1978) has been widely applied in the simulation literature (Law and Kelton, 1991). Unlike the OCBA approach, the two-stage procedures are developed based on the classical statistical model. See Bechhofer et al. (1995) for a systematic discussion of two-stage procedures. In the first stage, all designs are simulated for n_0 samples. Based on the sample variance estimate (S_i^2) obtained from the first stage, the number of additional simulation samples for each design in the second stage is determined by:

$$N_i = \max\left(0, \lceil (S_i^2 h^2/d^2 \rceil - n_0\right), \text{ for } i = 1, 2, \ldots, k,$$

where $\lceil \bullet \rceil$ is the integer "round-up" function, d is the indifference zone, h is a constant which solves Rinott's integral (h can also be found from the tables in Wilcox, 1984). In short, the computing budget is allocated proportionally to the estimated sample variances.

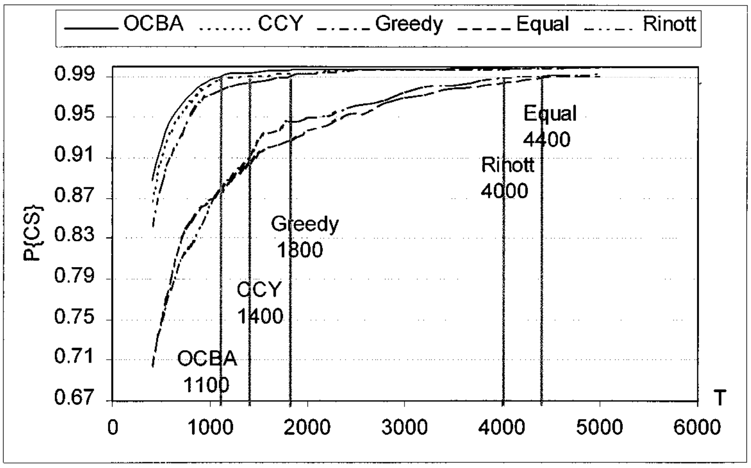

Figure 1. $P\{CS\}$ vs. T using five different allocation procedures for experiment 1. Normal distributions with 10 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

The major drawback is that only the information on variances is used when determining the simulation allocation, while the OCBA algorithm, the CCY procedure and the greedy approach utilize the information on both means and variances. As a result, the performance of Rinott's procedure is not as good as others. We do, however, include it in our testing due to its popularity in the simulation literature.

4.2. Numerical Experiments

The numerical experiments include a series of generic tests plus a test on the buffer allocation problem. In all the numerical illustrations, we estimate $P\{CS\}$ by counting the number of times we successfully find the true best design (design 0 in this example) out of 10,000 independent applications of each selection procedure. $P\{CS\}$ is then obtained by dividing this number by 10,000, representing the correct selection frequency.

Experiment 1. Normal Distribution

There are ten design alternatives. Suppose $L(\theta_i, \xi)) \sim N(i, 6^2)$, $i = 0, 1, \ldots, 9$. We want to find a design with the minimum mean. It is obvious that design 0 is the actual best design. In the numerical experiment, we compare the convergence of $P\{CS\}$ for different allocation procedures. We have $n_0 = 10$ and $\Delta = 20$.

Different computing budgets are tested. Figure 1 shows the test results using OCBA and the other four different procedures discussed in section 4.1. Note that the simulation variance of each design is 36, while the difference of two adjacent designs' means is only 1.

Given such a high noise ratio, we see that with the total computation cost as low as 700 simulation samples, the probability of correctly selecting the best design is already higher than 80% even using the simple equal allocation. This demonstrates the advantage of applying ordinal optimization.

We see that all procedures obtain a higher $P\{CS\}$ as the available computing budget increases. However, OCBA achieves a same $P\{CS\}$ with a lower amount of computing budget than other procedures. In particular, we indicate the computation costs in order to attain $P\{CS\} = 99\%$ for different procedures in Figure 1. While ordinal optimization is efficient, our OCBA can further reduce the simulation time by 75% for $P\{CS\} = 99\%$.

It is worth noting that Rinott's procedure does not perform much better than the simple equal allocation. This is because Rinott's procedure determines the number of simulation samples for all designs using only the information of sample variances. On the hand, Rinott's procedure is much slower than the greedy allocation, CCY and OCBA. This is because when determining budget allocation, the latter three procedures exploit the information of both sample means and variances, while Rinott's procedure does not utilize the information of sample means. The sample means can provide the valuable information of relative differences across the design space.

CCY is more efficient than the greedy allocation since CCY intends to optimize the simulation efficiency. Finally, our OCBA is even faster than CCY; the computation costs for attaining $P\{CS\} = 99\%$ are 1,100 vs. 1,400 samples.

Experiment 2. Uniform Distribution

We consider a non-normal distribution for the performance measure: $L(\theta_i, \xi)) \sim$ Uniform $(i - 10.5, i + 10.5)$, $i = 0, 1, \ldots, 9$. The endpoints of the uniform distribution are chosen such that the corresponding variance is close to that in experiment 1. Again, we want to find a design with the minimum mean; design 0 is therefore the actual best design. All other settings are identical to experiment 1. Figure 2 contains the simulation results for the five allocation procedures. We can see that the relative performances of the different procedures are very similar to what we saw in experiment 1. OCBA is the fastest and is more than three times faster than Rinott and equal allocation.

Experiment 3. Normal Distribution with Larger Variance

This is a variant of experiment 1. All settings are preserved except that the variance of each design is doubled. Namely, $L(\theta_i, \xi)) \sim N(i, 2 \cdot 6^2)$, $i = 0, 1, \ldots, 9$. Figure 3 contains the simulation results for the five allocation procedures. We can see that the relative performances of different procedures are very similar with what we see in previous experiments, except that bigger computing budgets are needed in order to obtain the same $P\{CS\}$, due to larger variance. Also, OCBA is more than four times faster than equal allocation.

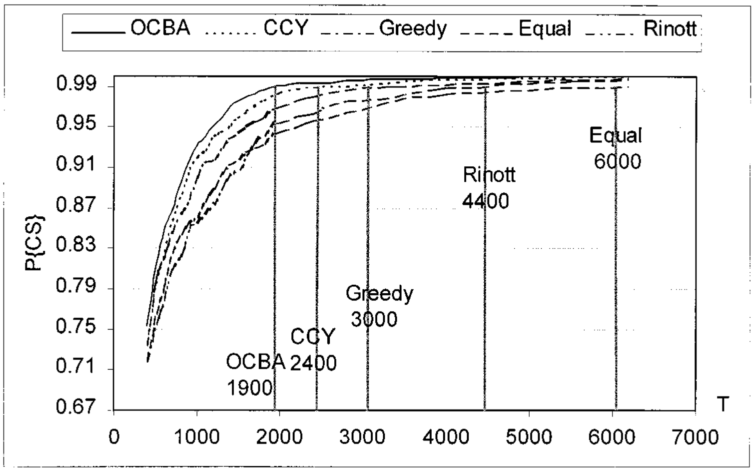

Figure 2. $P\{CS\}$ vs. T using five different allocation procedures for experiment 2. Uniform distributions with 10 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

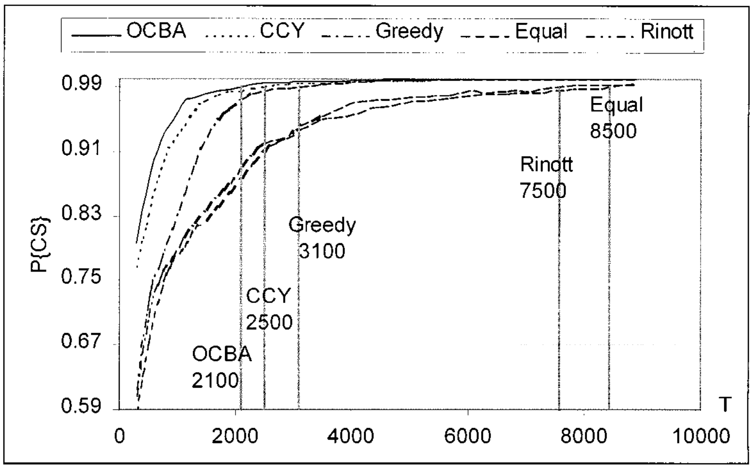

Figure 3. $P\{CS\}$ vs. T using three different allocation procedures for experiment 2. Normal distributions with 10 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

Experiment 4. Flat & Steep Case

This is another variant of experiment 1. We consider three generic cases illustrated in Figure 4.1 (also shown in Ho et al., 1992): neutral, flat, and steep. The neutral case is already presented in experiment 1. In the flat case, $L(\theta_i, \xi)) \sim N(9-3\sqrt{9-i}, 6^2)$, $i = 0, 1, \ldots, 9$;

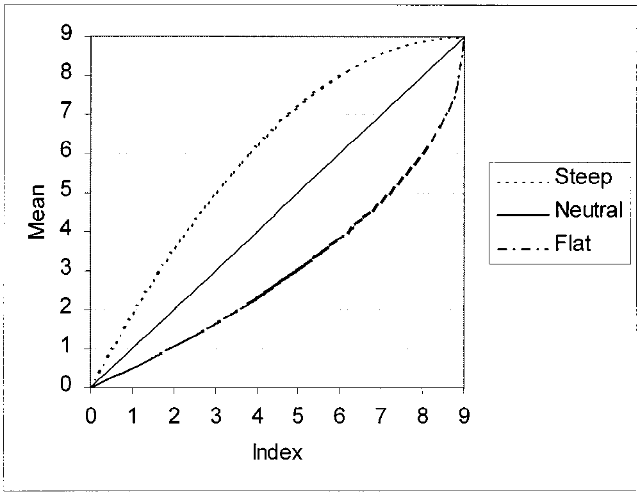

Figure 4.1. Illustration of three generic cases: neutral, flat, steep.

and in the steep case $L(\theta_i, \xi)) \sim N\left(9 - \left(\frac{9-i}{3}\right)^2, 6^2\right), i = 0, 1, \ldots, 9$. In the flat case, good designs are closer; a larger computing budget is therefore needed to identify the best design given the same simulation estimation noise. On the other hand, it is easier to correctly select the best design in the steep case since the good designs are further spread out. The numerical results in Figures 4.2 and 4.3 support this conjecture. In either case, OCBA is the fastest and is more than three times faster than equal allocation.

Experiment 5. Bigger Design Space

This is also a variant of experiment 1. To see the performance of the OCBA algorithm within a bigger design space, we increase the number of designs to 100. $L(\theta_i, \xi)) \sim N(i/10, 1^2), i = 0, 1, 2, \ldots, 98, 99$. Note that we have the range of the means for these 100 designs the same as those in earlier 10-design experiments, which is from 0 to 10. Since the performances of Rinott's procedure and equal allocation are very close, and the performances of CCY and the greedy allocation are also close, we will test and compare only OCBA, greedy and the equal allocation in the remaining experiments.

Figure 5 depicts the simulation results. The speedup factor of using OCBA is increased to 22 in these experiments. This is because a larger design space gives the OCBA algorithm more flexibility in allocating the computing budget.

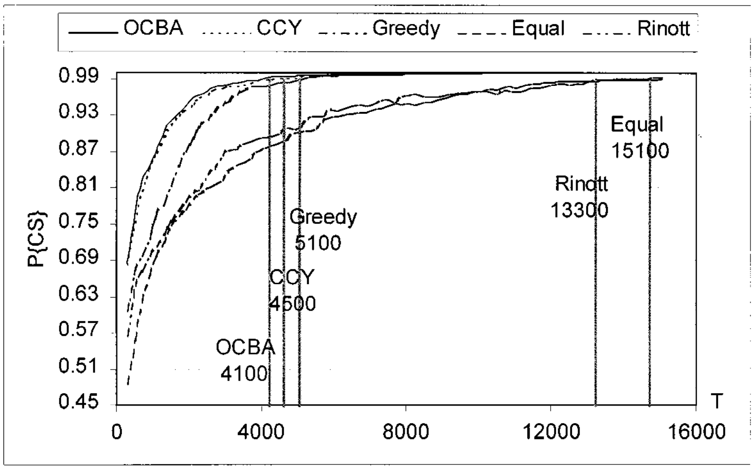

Figure 4.2. $P\{CS\}$ vs. T using three different allocation procedures for the flat case in experiment 4. Normal distributions with 10 designs.

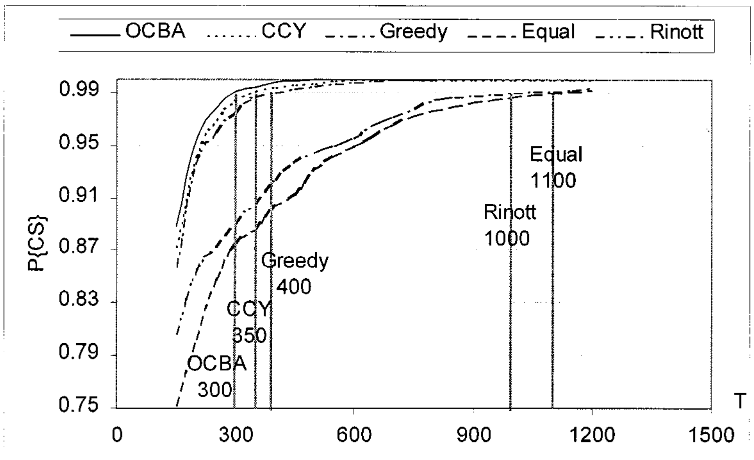

Figure 4.3. $P\{CS\}$ vs. T using three different allocation procedures for the steep case in experiment 4. Normal distributions with 10 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

Experiment 6. A Buffer Resource Allocation Problem

A 10-node network shown in Figure 6.1 is used to test different buffer allocation procedures including our OCBA. Details about the network can be found in Chen and Ho (1995). There are 10 servers and 10 buffers that are connected as a switching network. There are two classes

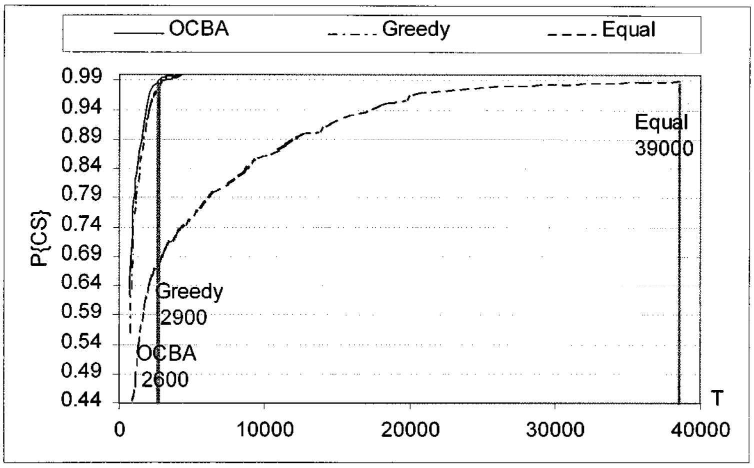

Figure 5. $P\{CS\}$ vs. T using three different allocation procedures for experiment 5. Normal distributions with 100 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

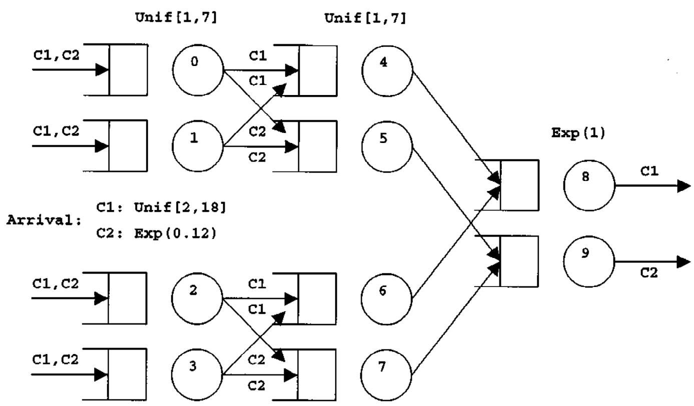

Figure 6.1. A 10-node network.

of customers with different arrival distributions, but with the same service requirements. We consider both exponential and non-exponential distributions (uniform) in the network. Both classes arrive at any of nodes 0 through 3, and leave the network after having gone through three different stages of service. The routing is class dependent. Such a network could be the model for a large number of real-world systems, such as a manufacturing system, a communication or a traffic network.

Finite buffer sizes at all nodes are assumed. In this design problem, we are interested in distributing optimally limited buffer spaces to different nodes. Specifically, we consider the problem of allocating 12 buffer units, among the 10 different nodes numbered from 0

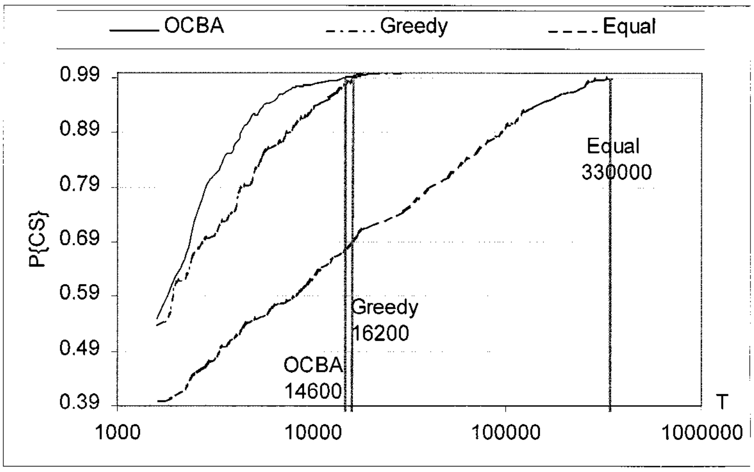

Figure 6.2. $P\{CS\}$ vs. T using three different allocation procedures for experiment 6. Note the x-axis is in log scale. There are 210 designs. The computation costs in order to attain $P\{CS\} = 99\%$ are indicated.

to 9. We denote the buffer size of node i by B_i. Thus,

$$B_0 + B_1 + B_2 + \cdots + B_9 = 12.$$

There are three constraints for symmetry reasons:

$$B_0 = B_1 = B_2 = B_3,\ B_4 = B_6,\ \text{and } B_5 = B_7.$$

Totally there are 210 design alternatives for consideration. The objective is to select a design with minimum expected time to process the first 100 jobs from a same initial state (that is $[B_0, B_1, B_2, \ldots, B_9] = [1, 1, 1, 1, 2, 1, 2, 1, 1, 1]$).

Figure 6.2 depicts the simulation results for the three allocation procedures. Once again, we can see that the relative performances of different procedures are very similar to what we saw in the previous experiments, except that bigger computing budgets are needed in order to obtain the same $P\{CS\}$, due to the larger design space. The speedup factor of using OCBA is about 23, which is even bigger than that in experiment 5. Note that in Figure 6.2, the x-axis is in log scale since the differences of the computation costs using different approaches are very large.

In addition, we can see that OCBA is much more efficient even when T is small, despite that our algorithm is developed based on asymptotic allocation. This is also true in earlier numerical experiments.

5. Conclusions

We present a highly efficient procedure to identify the best design out of k (simulated) competing designs. The purpose of this technique is to further enhance the efficiency of ordinal optimization in simulation experiments. The objective is to maximize the simulation efficiency, expressed as the probability of correct selection within a given computing budget. Our procedure allocates replications in a way that optimally improves an asymptotic approximation to the probability of correct selection. We also compare several different allocation procedures, including a popular two-stage procedure in simulation literature. Numerical testing shows that our approach is much more efficient than all compared methods. Comparisons with the crude ordinal optimization show that our approach can achieve a speedup factor of $3 \sim 4$ for a 10-design example. The speedup factor is even higher with the problems having a larger number of designs. For a buffer resource allocation problem, in which there are 210 designs, our approach is more than 20 times faster than crude ordinal optimization. Although our procedure allocates the available computing budget based on an asymptotic derivation, all of our numerical results indicate that our procedure is highly effective when the computing budget is small. While ordinal optimization can converge exponentially fast, our simulation budget allocation procedure provides a way to further significantly improve overall simulation efficiency.

Acknowledgements

This work has been supported in part by NSF under Grant DMI-9732173, by the U.S. Department of Transportation uner a grant from the University Transportation Centers Program through the Mid-Atlantic Transportation Consortium, by Sandia National Laboratories under Contract BD-0618, and by the University of Pennsylvania Research Foundation.

References

Banks, J. 1998. *Handbook of Simulation*. John Wiley.

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

Bernardo, J. M., and Smith, A. F. M. 1994. *Bayesian Theory*. Wiley.

Bratley, P., Fox, B. L., and Schrage, L. E. 1987. *A Guide to Simulation*, 2nd ed. Springer-Verlag.

Cassandras, C. G., Dai, L., and Panayiotou, C. G. 1998. Ordinal optimization for deterministic and stochastic discrete resource allocation. *IEEE Trans. on Automatic Control* 43(7): 881–900.

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*, pp. 2598–2605.

Chen, C. H. 1996. A lower bound for the correct subset-selection probability and its application to discrete event system simulations. *IEEE Transactions on Automatic Control* 41(8): 1227–1231.

Chen, C. H., and Ho, Y. C. 1995. An approximation approach of the standard clock method for general discrete event simulation. *IEEE Transactions on Control Systems Technology* 3(3): 309–317.

Chen, H. C., Chen, C. H., Dai, L., and Yücesan, E. 1997. New development of optimal computing budget allocation for discrete event simulation. *Proceedings of the 1997 Winter Simulation Conference*, pp. 334–341.

Chen, C. H., Wu, S. D., and Dai, L. 1999. Ordinal comparison of heuristic algorithms using stochastic optimization. *IEEE Transactions on Robotics and Automation* 15(1): 44–56.

Chen, H. C., Chen, C. H., and Yücesan, E. 2000. Computing efforts allocation for ordinal optimization and discrete event simulation. To appear in *IEEE Transactions on Automatic Control*.

Chick, S. E. 1997. Bayesian analysis for simulation input and output. *Proceedings of the 1997 Winter Simulation Conference*, pp. 253–260.

Dai, L. 1996. Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems. *Journal of Optimization Theory and Applications* 91(2): 363–388.

DeGroot, M. H. 1970. *Optimal Statistical Decisions*. McGraw-Hill, Inc.

Fabian, V. 1971. *Stochastic Approximation, Optimization Methods in Statistics*. J. S. Rustagi (ed.), Academic Press, New York.

Goldsman, G., and Nelson, B. L. 1994. Ranking, selection, and multiple comparison in computer simulation. *Proceedings of the 1994 Winter Simulation Conference*, pp. 192–199.

Gong, W. B., Ho, Y. C., and Zhai, W. 1999. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*.

Ho, Y. C., Sreenivas, R. S., and Vakili, P. 1992. Ordinal optimization of DEDS. *Journal of Discrete Event Dynamic Systems* 2(2): 61–88.

Inoue, K., and Chick, S. 1998. Comparison of Bayesian and Frequentist assessments of uncertainty for selecting the best system. *Proceedings of the 1998 Winter Simulation Conference*, pp. 727–734.

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

Law, A. M., and Kelton, W. D. 1991. *Simulation Modeling & Analysis*. McGraw-Hill, Inc.

Patsis, N. T., Chen, C. H., and Larson, M. E. 1997. SIMD parallel discrete event dynamic system simulation. *IEEE Transactions on Control Systems Technology* 5(3): 30–41.

Rinott, Y. 1978. On two-stage selection procedures and related probability inequalities. *Communications in Statistics* A7: 799–811.

Walker, R. C. 1999. *Introduction to Mathematical Programming*. Prentice Hall.

Wilcox, R. R. 1984. A table for Rinott's selection procedure. *Journal of Quality Technology* 16(2): 97–100.

Yan, D., and Mukai, H. 1993. Optimization algorithm with probabilistic estimation. *Journal of Optimization Theory and Applications* 79: 345–371.