Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

A Case Study for Optimal Dynamic Simulation Allocation in Ordinal Optimization[1]

Chun-Hung Chen[2] , Donghai He[3], and Michael Fu[4]

Abstract— **Ordinal Optimization has emerged as an efficient technique for simulation and optimization. Exponential convergence rates can be achieved in many cases. A good allocation of simulation samples across designs can further dramatically improve the efficiency of ordinal optimization by orders of magnitude. However, the allocation problem itself is a big challenge. Most existing methods offer approximations. Assuming the availability of perfect information, we investigate theoretically optimal allocation schemes for some special cases. We compare our theoretically optimal solutions with existing approximation methods using a series of numerical examples. While perfect information is not available in real life, such an optimal solution provides an upper bound for the simulation efficiency we can achieve. The results indicate that the simulation efficiency can still be further improved beyond the existing methods. Also the numerical testing shows that dynamic allocation is much more efficient than static allocation.**

I. 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 simulation has many advantages for modeling complex systems, efficiency is still a significant concern when conducting simulation experiments. 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. If the accuracy requirement is high and 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., Hsieh et al. 2001, Patsis et al. 1997).

While ordinal optimization could significantly reduce the computational cost for DES simulation, it has been shown that the efficiency can be further dramatically improved by intelligently controlling the simulation experiments, or by determining the efficient number of simulation samples among different designs as simulation proceeds (Chen et al. 1997). 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 the best design. In other words, a larger number of simulations must be conducted with those critical alternatives in order to reduce these critical estimators' variances. 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 alternatives and more is spent on critical alternatives. Ideally, one would like to allocate simulation samples to designs in a way that maximizes the probability of selecting the best design within a given computing budget. Chen et al. (2000) formalize this idea and develop a new approach called optimal computing budget allocation (OCBA) algorithm. They demonstrate that the speedup factor can be another order of magnitude above and beyond the exponential convergence of ordinal optimization. In addition, several simulation budget allocation schemes have been developed for various applications or from different perspectives (Chick and Inoue 2001, Lee 2003, Trailovic and Pao 2004). They also show that the efficiency can be significantly enhanced.

1. This work has been supported in part by NSF under Grants DMI-9988867, DMI-0002900, DMI-0049062, DMI-0323220, and IIS-0325074, by NASA Ames Research Center under Grants NAG-2-1565 and NAG-2-1643, by FAA under Grant 00-G-016, by AFOSR under Grant F496200110161, and by George Mason University Research Foundation.

2. Dr. Chun-Hung Chen is with Department of Systems Engineering & Operations Research, George Mason University, 4400 University Drive, MS 4A6, Fairfax, VA 22030, USA (phone: 703-993-3572; email: cchen9@gmu.edu).

3. Mr. Donghai He is with Department of Systems Engineering & Operations Research, George Mason University, 4400 University Drive, MS 4A6 , Fairfax, VA 22030, USA (email: dhe@gmu.edu).

4. Dr. Michael Fu is with the Robert H. Smith School of Business and Institute for Systems Research, University of Maryland College Park, MD, 20742-1815,USA (email: mfu@rhsmith.umd.edu)

In this paper we extensively study the efficiency issue of simulation budget allocation for ordinal optimization. In applying the aforementioned simulation budget allocation methods, one has to determine a good allocation of simulation replications (budget) using some information and then perform simulation accordingly. One challenge is that the objective function we intend to estimate and then optimize is a critical component in determining the simulation budget allocation. Unfortunately a good estimate of the objective function is usually not available until after the simulation is carried out. Without a good estimate, the budget allocation may not be good, which has an impact on the simulation efficiency. There are two possible approaches for handling this issue. One is to perform some preliminary simulation to obtain information for determining budget allocation, and then allocate the remaining simulation budget at once. We call this the *static* allocation. On the other hand, one may allocate only a small portion of the simulation budget each time and utilize the most updated simulated information to determine the new budget allocation iteratively. We refer to the second approach as a *dynamic* allocation. Intuitively, a dynamic allocation should work better than a static allocation.

In this paper we consider a small but general problem. We present theoretical optimal allocation schemes for both static and dynamic allocation. We also test some existing budget allocation methods against the presented theoretically optimal allocation using three numerical examples. We find that dynamic allocation is indeed much more efficient than static allocation. It is interesting to see that OCBA performs better than the optimal static allocation. However, there is still room for OCBA to improve its performance, because we observe that the optimal dynamic allocation performs much better than OCBA.

The paper is organized as follows: In the next section, we define the notation and the simulation run allocation problem for ordinal optimization. Section 3 gives a solution to a theoretically optimal allocation problem. Numerical experiments are given in Section 4. Section 5 concludes the paper.

II. Efficient Simulation Allocation for Ordinal Optimization

Suppose we have a complex discrete-event system. A general simulation and optimization problem with finite number of designs can be defined as

$$\min_i \mu_i \equiv E_\xi[L(\theta_i, \xi)] \qquad (1)$$

where $\theta_i \in \Theta$ the search space is an arbitrary, huge, structureless but finite set; θ_i is the system design parameter vector for design i, $i = 1, 2, ..., k$; μ, 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). In simulation approach, multiple simulation samples/replications are taken and then $E[L(\theta_i, \xi)]$ is estimated by the sample mean performance measure:

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

where ξ_{ij} represents the j-th sample of ξ and N_i represents the number of simulation samples for design i. For notational simplicity, define

$$X_{ij} \equiv L(\theta_i, \xi_{ij})$$

which is the j-th sample of the performance measure from design i. In this paper, we assume that the simulation output is independent from replication to replication. The sampling across designs is also independent. Also we assume X_{ij} is normally distributed. The normality assumption is usually not a problem, because typical simulation output is obtained from an average performance or batch mean. Our goal is to select a design associated with the smallest mean performance measure among k alternative designs. Denote by

- $\bar{X}_i$: the sample average of the simulation output for design i; $\bar{X}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} X_{ij}$,
- S_i^2 : the sample variance of the simulation output for design i,
- σ_i^2 : the variance for design i, i.e., $\sigma_i^2 = \mathrm{Var}(X_{ij})$. In practice, σ_i^2 is unknown beforehand and so is approximated by sample variance.
- b: the design with the smallest sample mean performance; $b = \arg\min_i \{\bar{X}_i\}$.

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

With the above notations and assumption,

$$X_{ij} \sim N(\mu_i, \sigma_i^2).$$

As N_i increases, $\bar{X}_i$ becomes a better approximation to μ_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 X_{ij} requires one simulation run. A large number of required samples of X_{ij} for all designs may become very time consuming. On the other hand, 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. With the advantage of such an exponential convergence, instead of equally simulating all designs, Chen et al. (2000) further improve the performance of ordinal optimization by determining the best numbers of simulation samples for each design. Assume that the computation cost for each run is roughly the same across different designs. The computation cost can then be approximated by $N_1 + N_2 + \ldots + N_k$, the total number of samples. We wish to choose $N_1, N_2, \ldots, N_k$ such that $P\{\text{CS}\}$ is maximized, subject to a limited computing budget T, i.e.,

$$\max_{N_1,\cdots,N_k} P\{\text{CS}\}$$
$$\text{s.t. } N_1 + N_2 + \cdots + N_k = T. \qquad (2)$$

However, solving such an optimal sample allocation problem is a big challenge because i) there is no closed-form expression for $P\{\text{CS}\}$ in general; ii) $P\{\text{CS}\}$ is a function of the means and variances of all designs which are unknown; and iii) a solution should be found efficiently. Otherwise the benefit of efficient run allocation will be lost.

III. Theoretically Optimal Computing Budget Allocation - Three Designs

It is difficult to solve problem (2). However, for smaller problems with some assumptions, it is possible to find the optimal solution. In this paper, first we limit our scope within problems having only three designs competing for simulation budget allocation. Second, we assume we have perfect information until the point of determining simulation allocation. Namely, we assume we know the means, variances, and all the samples we have obtained up to the point.

We consider two possible approaches of allocating computing budget to designs: static vs. dynamic. In the static approach, we solve problem (2) and determine its optimal solution $N^* \equiv [\ N_1^*, N_2^*, \ldots, N_k^*]$. Then we perform N_i^* simulation samples for design i for all i. For the dynamic approach, instead of allocating all of the T simulation samples at the beginning, we dynamically allocate only a small number of simulation samples at each iteration. In the dynamic approach, the computing budget allocation is determined iteratively using the most updated simulation information. Details of these two approaches are presented in the following subsections.

A. *Theoretically Optimal Static Allocation (TOSA) Procedure*

Note that $P\{\text{CS}\}$ is a function of means, variances, and $(N_1, N_2, \ldots, N_k)$. When the means and variances for all designs are known, $P\{\text{CS}\}$ can be calculated (or estimated through Monte Carlo simulation) if the values of $(N_1, N_2, \ldots, N_k)$ are given. Since the total computing budget, T, considered in this paper is not big, we can evaluate $P\{\text{CS}\}$ for all possible combinations of $(N_1, N_2, \ldots, N_k)$ with a constraint that $N_1 + N_2 + \ldots + N_k = T$. Then the maximum $P\{\text{CS}\}$ and the corresponding $(N_1, N_2, \ldots, N_k)$ can be determined. For example, the three designs in the second example given in Section 4 are:

$X_{1j} \sim \text{N}(0,\ 0^2)$,
$X_{2j} \sim \text{N}(0.4,\ 1.5^2)$, and
$X_{3j} \sim \text{N}(0.4,\ 3^2)$.

Suppose that the total computing budget $T = N_1 + N_2 + N_3 = 120$. In this case, design 1 is the best design, and

$$\begin{aligned} P\{\text{CS}\} &= \Pr\{\ \bar{X}_1(N_1) < \bar{X}_2(N_2) \text{ and } \bar{X}_1(N_1) < \bar{X}_3(N_3)\ \} \\ &= \Pr\{\ \bar{X}_2(N_2) > 0\ \}\ \Pr\{\ \bar{X}_3(N_3) > 0\ \} \\ &= \Phi\Big(\frac{0.4}{1.5/\sqrt{N_2}}\Big)\ \Phi\Big(\frac{0.4}{3.0/\sqrt{N_3}}\Big) \end{aligned}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal random variable. Since the variance of design 1 is zero, we know that in the theoretically optimal allocation, we should not allocate any sample to design 1 as it will not further reduce its estimation variance. We should allocate the limited computing budget to Designs 2 and 3 only. Thus we can easily evaluate $P\{\text{CS}\}$ for all 121 combinations with the constraint $N_2 + N_3 = 120$ and find the best allocation. Once the optimal solution (N_1^*, N_2^*, N_3^*) is found, the TOSA allocates all the computing budget at a time and performs N_i^* simulation samples for each design i accordingly.

B. *Theoretically Optimal Dynamic Allocation (TODA) Procedure*

In the dynamic approach, we allocate only one additional simulation sample at each iteration. By utilizing the sampled information, we determine the new allocation dynamically. Again, we assume we have perfect information of knowing the means, variances, and all the samples we have taken for all designs up to the decision point. For notational simplicity, we assume $\mu_1 < \mu_2 < \mu_3$ in the following discussion. Suppose we have conducted N_1, N_2, N_3 simulation samples for the three designs and obtained the sample means: $\bar{X}_1$, $\bar{X}_2$, and $\bar{X}_3$. The decision problem is which design we should choose to have one more simulation sample such that the $P\{\text{CS}\}$ can be maximized given the information we have. Denote a as the decision variable that gives the index of design which we will simulate at the next iteration and y_a as the new sample obtained after we perform this additional simulation on design a. Thus

$$a^* = \arg\max_a P\{\text{CS} \mid \bar{X}_1, \bar{X}_2, \bar{X}_3, a\}$$
$$\text{where } a \in \{1, 2, 3\} \qquad (3)$$

When $a = 1$,

$$\begin{aligned} P_1 &\equiv P\{\text{CS} \mid \bar{X}_1, \bar{X}_2, \bar{X}_3, a{=}1\} \\ &= P\Big\{\frac{n_1\bar{X}_1 + y_1}{n_1 + 1} < \min(\bar{X}_2, \bar{X}_3)\Big\} \end{aligned}$$

$$= \Phi\left(\frac{(n_1+1)\bullet\min\{\bar{X}_2,\bar{X}_3\}-n_1\bar{X}_1-\mu_1}{\sigma_1}\right)$$

Similarly, if we decide to simulate design 2, i.e., $a = 2$, then

$$P_2 \equiv P\{\text{CS} \mid \bar{X}_1, \bar{X}_2, \bar{X}_3, a{=}2\}$$
$$= P\{\bar{X}_1 < \frac{n_2\bar{X}_2 + y_2}{n_2+1} \cap \bar{X}_1 < \bar{X}_3\}$$
$$= \text{I}(\bar{X}_1 < \bar{X}_3)\,\Phi\left(\frac{n_2\bar{X}_2 - (n_2+1)\bar{X}_1 + \mu_2}{\sigma_2}\right)$$

where I(·) is an identity function. When $a = 3$,

$$P_3 \equiv P\{\text{CS} \mid \bar{X}_1, \bar{X}_2, \bar{X}_3, a{=}3\}$$
$$= P\{\bar{X}_1 < \frac{n_3\bar{X}_3 + y_3}{n_3+1} \cap \bar{X}_1 < \bar{X}_2\}$$
$$= \text{I}(\bar{X}_1 < \bar{X}_2)\,\Phi\left(\frac{n_3\bar{X}_3 - (n_3+1)\bar{X}_1 + \mu_3}{\sigma_3}\right)$$

Thus, problem (3) is equivalent to

$$a^* = \arg\max_a P_a \tag{4}$$

To solve the optimal decision problem in (4), we need to consider different cases based on the order of $\bar{X}_1$, $\bar{X}_2$, and $\bar{X}_3$ as follows.

Case 1. When $\bar{X}_1$ is the smallest,

$$P_1 = \Phi\left(\frac{(n_1+1)\bullet\min\{\bar{X}_2,\bar{X}_3\}-n_1\bar{X}_1-\mu_1}{\sigma_1}\right)$$
$$P_2 = \Phi\left(\frac{n_2\bar{X}_2 - (n_2+1)\bar{X}_1 + \mu_2}{\sigma_2}\right)$$
$$P_3 = \Phi\left(\frac{n_3\bar{X}_3 - (n_3+1)\bar{X}_1 + \mu_3}{\sigma_3}\right)$$

Case 2. When $\bar{X}_3 < \bar{X}_1 < \bar{X}_2$,

$$P_1 = \Phi\left(\frac{(n_1+1)\bar{X}_3 - n_1\bar{X}_1 - \mu_1}{\sigma_1}\right)$$
$$P_2 = 0$$
$$P_3 = \Phi\left(\frac{n_3\bar{X}_3 - (n_3+1)\bar{X}_1 + \mu_3}{\sigma_3}\right).$$

Case 3. When $\bar{X}_2 < \bar{X}_1 < \bar{X}_3$,

$$P_1 = \Phi\left(\frac{(n_1+1)\bar{X}_2 - n_1\bar{X}_1 - \mu_1}{\sigma_1}\right)$$
$$P_2 = \Phi\left(\frac{n_2\bar{X}_2 - (n_2+1)\bar{X}_1 + \mu_2}{\sigma_2}\right)$$
$$P_3 = 0$$

Case 4. When $\bar{X}_1$ is the largest,

$$P_1 = \Phi\left(\frac{(n_1+1)\bullet\min\{\bar{X}_2,\bar{X}_3\}-n_1\bar{X}_1-\mu_1}{\sigma_1}\right)$$
$$P_2 = 0;\ P_3 = 0.$$

Note that Φ is a monotonically increasing function and $\Phi(-\infty) = 0$. To maximize P_a, it is equivalent to maximize Q_a, which is defined as the parameter value in the Φ function as follows:

Case 1. When $\bar{X}_1$ is the smallest,

$$Q_1 = \frac{(n_1+1)\bullet\min\{\bar{X}_2,\bar{X}_3\}-n_1\bar{X}_1-\mu_1}{\sigma_1}$$
$$Q_2 = \frac{n_2\bar{X}_2 - (n_2+1)\bar{X}_1 + \mu_2}{\sigma_2}$$
$$Q_3 = \frac{n_3\bar{X}_3 - (n_3+1)\bar{X}_1 + \mu_3}{\sigma_3}$$

Case 2. When $\bar{X}_3 < \bar{X}_1 < \bar{X}_2$,

$$Q_1 = \frac{(n_1+1)\bar{X}_3 - n_1\bar{X}_1 - \mu_1}{\sigma_1}$$
$$Q_2 = (-\infty)$$
$$Q_3 = \frac{n_3\bar{X}_3 - (n_3+1)\bar{X}_1 + \mu_3}{\sigma_3}.$$

Case 3. When $\bar{X}_2 < \bar{X}_1 < \bar{X}_3$,

$$Q_1 = \frac{(n_1+1)\bar{X}_2 - n_1\bar{X}_1 - \mu_1}{\sigma_1}$$
$$Q_2 = \frac{n_2\bar{X}_2 - (n_2+1)\bar{X}_1 + \mu_2}{\sigma_2}$$
$$Q_3 = (-\infty)$$

Case 4. When $\bar{X}_1$ is the largest,

$$Q_1 = \frac{(n_1+1)\bullet\min\{\bar{X}_2,\bar{X}_3\}-n_1\bar{X}_1-\mu_1}{\sigma_1}$$
$$Q_2 = (-\infty);\ Q_3 = (-\infty).$$

Now the action a can be easily determined by maximizing Q_i. Namely,

$$a^* = \arg\max_a Q_a \tag{5}$$

In summary, at each iteration, we calculate the most updated sample means, $\bar{X}_1$, $\bar{X}_2$, and $\bar{X}_3$. Then Q_a is calculated and the action a can be determined. We allocate the additional computing budget to design a. After design a is simulated for one more sample, $\bar{X}_a$ is updated, and the whole procedure is repeated until the computing budget T is exhausted.

IV. Numerical Testing and Evaluation on Practical Allocation Procedures

In this section, we test and compare the theoretically optimal allocation schemes in Section 3 with some practical allocation procedures using three numerical examples.

A. Practical Allocation Procedures

In practice, the means and variances of all designs are unknown prior to performing simulations. Instead of using the real means and variances, practical allocation procedures apply sample means and sample variances obtained from simulation samples to determine additional simulation allocation. Three representative allocation procedures in ordinal optimization are considered in this paper. They are briefly summarized as follows.

1) Equal Allocation (Equal)

This has been widely applied. The simulation budget is equally allocated to all designs.

2) Proportional To Variance (PTV)

This is based on well-known two-stage Rinott procedure (Rinott 1978). The idea is to allocate computing budget in a way that N_i is proportional to the estimated sample variances, S_i^2.

3) OCBA by Chen el al. (2000)

Under a Bayesian model, OCBA approximates $P\{\text{CS}\}$ using the Bonferroni inequality and offers an asymptotic solution to this approximation. While the run allocation given by OCBA is not an optimal allocation when the simulation budget is finite, the numerical testing demonstrates that OCBA is a very efficient approach and can dramatically reduce simulation time. In particular, OCBA allocates simulation runs according to:

- $$\frac{N_i}{N_j} = \left(\frac{\sigma_i / \delta_{b,i}}{\sigma_j / \delta_{b,j}}\right)^2, \; i,j \in \{1, 2, ..., k\}, \text{ and } i \neq j \neq b, \quad (6)$$
- $$N_b = \sigma_b \sqrt{\sum_{i=1, i\neq b}^{k} \frac{N_i^2}{\sigma_i^2}}. \quad (7)$$

Note that in OCBA, the allocation is a function of the differences in sample means and the variances, which are approximated by sample variances.

B. Numerical Testing

We also consider both dynamic and two-stage allocation for both PTV and OCBA. Initially, n_0 simulation runs for each of k designs are performed to get some information such as sample mean and variance of each design during the first stage. Then different allocation procedures are applied to determine how to allocate the remaining simulation budget. In the two-stage allocation, all the remaining budget is allocated at once after the first-stage simulation. They are called PTV-2 and OCBA-2. On the other hand, for dynamic allocation, only an incremental computing budget, Δ, is allocated at each iteration after the first stage. As simulation proceeds, the sample means and sample variances of all designs are computed from the data already collected up to that stage. The simulation budget allocation is determined dynamically using the most updated sampling information. The procedure is continued until the total budget T is exhausted. We denote them as PTV-D and OCBA-D. The algorithm for PTV-D and OCBA-D is summarized as follows.

A Sequential Algorithm for OCBA or PTV

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, $N_1^{l+1}, N_2^{l+1}, \cdots, N_k^{l+1}$, using (6) and (7) for OCBA.

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

In the above algorithm, l is the iteration number. As simulation evolves, design b, which is the design with the smallest 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.

The $P\{\text{CS}\}$ for each procedure is estimated by counting the number of times the procedure successfully finds the true best design out of 1,000,000 independent applications, and then dividing this number by 1,000,000. The choice of 1,000,000 macro replications leads to a standard error for the $P\{\text{CS}\}$ estimate of under 0.001 or 0.1%. The $P\{\text{CS}\}$ estimate for each procedure will serve as a measurement of its effectiveness for comparison purposes. In all of the examples, there are three design alternatives, the total computing budget is $T = N_1 + N_2 + N_3 = 120$, and we have set $n_0 = 10$ and $\Delta = 5$.

1) Example 1.

This is a special case where the best design has zero variance and the two inferior designs have the same performance. The three design alternatives are:

$X_{1j} \sim \text{N}(0,\ 0^2)$,
$X_{2j} \sim \text{N}(0.4,\ 3^2)$, and
$X_{3j} \sim \text{N}(0.4,\ 3^2)$.

Practically speaking, this case implies that design 1 has no estimation uncertainty, while design 2 and design 3 are extremely close but with uncertainty. It is obvious that for TOSA, we should not allocate any simulation budget to design 1 , but should equally divide the budget to designs 2 and 3. Thus, $N_1 = 0$ and $N_2 = N_3 = 60$.

Table I shows the test results using different allocation procedures. We see that TODA performs much better than TOSA. This shows the benefit of dynamic allocation. We also see that OCBA performs better than Equal and PTV. However, it is interesting to observe that OCBA-D outperforms TOSA by a very big margin. This once again demonstrates the benefit of dynamic allocation.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT SIMULATION RUN ALLOCATION PROCEDURES IN EXAMPLE 1. THE TOTAL COMPUTING BUDGET IS 120

Procedures	TOSA	TODA	Equal	PTV-2	PTV-D	OCBA-2	OCBA-D
P{CS}	0.720	0.964	0.640	0.691	0.711	0.764	0.826

2) Example 2.

This is the same as Example 1, except the variances of the two inferior designs differ:

$X_{1j} \sim N(0,\ 0^2)$,
$X_{2j} \sim N(0.4,\ 1.5^2)$, and
$X_{3j} \sim N(0.4,\ 3^2)$.

In this case,

$$P\{CS\} = \Pr\{\ \bar{X}_1(N_1) < \bar{X}_2(N_2)\ \text{and}\ \bar{X}_1(N_1) < \bar{X}_3(N_3)\ \}$$
$$= \Pr\{\ \bar{X}_2(N_2) > 0\ \}\ \Pr\{\ \bar{X}_3(N_3) > 0\ \}$$
$$= \Phi\left(\frac{0.4}{1.5/\sqrt{N_2}}\right)\Phi\left(\frac{0.4}{3.0/\sqrt{N_3}}\right)$$

Since the variance of design 1 is zero, we should allocate the limited computing budget to Designs 2 and 3 only. Thus we can easily evaluate $P\{CS\}$ for all 121 combinations with the constraint $N_2 + N_3 = 120$ to find the optimal static allocation. Table II shows the test results using different allocation procedures.

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT SIMULATION RUN ALLOCATION PROCEDURES IN EXAMPLE 2. THE TOTAL COMPUTING BUDGET IS 120

Procedures	TOSA	TODA	Equal	PTV-2	PTV-D	OCBA-2	OCBA-D
P{CS}	0.843	0.982	0.764	0.791	0.799	0.852	0.899

3) Example 3.

This is a more general case where all three designs have non-zero variances and different means:

$X_{1j} \sim N(0,\ 3^2)$,
$X_{2j} \sim N(0.4,\ 1.5^2)$, and
$X_{3j} \sim N(1.5,\ 3^2)$.

In this case, we evaluate $P\{CS\}$ using Monte Carlo simulation for all combinations. Table III shows the test results using different allocation procedures.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT SIMULATION RUN ALLOCATION PROCEDURES IN EXAMPLE 3. THE TOTAL COMPUTING BUDGET IS 120

Procedures	TOSA	TODA	Equal	PTV-2	PTV-D	OCBA-2	OCBA-D
P{CS}	0.800	0.962	0.772	0.750	0.787	0.778	0.808

V. CONCLUSION

This paper examines the efficiency issue of simulation budget allocation for ordinal optimization, which has emerged as an efficient technique for simulation and optimization. By intelligently allocating simulation samples across designs, we can further dramatically improve the efficiency of ordinal optimization. Under some assumptions, we develop theoretically optimal simulation budget allocation schemes for both version of static and dynamic sampling. We show that dynamic simulation allocation is more efficient than static allocation notwithstanding that in general problems a theoretically optimal allocation may not be at hand. Instead some work should be done on the dynamic OCBA algorithm to reduce the simulation time.

REFERENCES

[1] Chen, H. C., C. H. Chen, L. Dai, E. Yücesan, "New Development of Optimal Computing Budget Allocation for Discrete Event Simulation," Proceedings of the 1997 Winter Simulation Conference, 334-341, 1997.

[2] Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick, "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization," Journal of Discrete Event Dynamic Systems: Theory and Applications, Vol. 10, pp. 251-270, 2000.

[3] Chen, C. H., K. Donohue, E. Yücesan, and J. Lin, "Optimal Computing Budget Allocation for Monte Carlo Simulation with Application to Product Design," Journal of Simulation Practice and Theory, Vol. 11, No. 1, pp. 57-74, March 2003.

[4] Chick, S.E. and K. Inoue, "New Two-Stage and Sequential Procedures for Selecting the Best Simulated System," Operations Research, Vol. 49, pp. 732-743, 2001.

[5] Dai, L., "Convergence Properties of Ordinal Comparison in the Simulation of Discrete Event Dynamic Systems," Journal of Optimization Theory and Applications, Vol. 91, No.2, pp. 363-388, 1996.

[6] Ho, Y. C., R. S. Sreenivas, and P. Vakili, "Ordinal Optimization of DEDS," Journal of Discrete Event Dynamic Systems, 2, #2, 61-88, 1992.

[7] Hsieh, B. W., C. H. Chen, and S. C. Chang, "Scheduling Semiconductor Wafer Fabrication by Using Ordinal Optimization-Based Simulation," IEEE Transactions on Robotics and Automation, Vol. 17, No. 5, pp. 599-608, October 2001.

[8] Lee, L. H., " A simulation Study on Sampling and Selecting under Fixed Computing Budget," Proceedings of 2003 Winter Simulation Conference, forthcoming, December 2003.

[9] Patsis, N. T., C. H. Chen, and M. E. Larson, "SIMD Parallel Discrete Event Dynamic System Simulation," IEEE Transactions on Control Systems Technology, Vol. 5, No. 3, pp. 30-41, January, 1997.

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

[11] Trailovic, L. and L. Y. Pao, "Computing Budget Allocation for Efficient Ranking and Selection of Variances with Application to Target Tracking Algorithms," to appear in IEEE Transactions on Automatic Control, 2004.