Efficient Dynamic Simulation Allocation in Ordinal Optimization[+]

Chun-Hung Chen
Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan, and
Department of Systems Engineering & Operations Research
George Mason University, Fairfax, VA 22030, USA

Donghai He
Department of Systems Engineering & Operations Research
George Mason University, Fairfax, VA 22030, USA

Michael Fu
The Robert H. Smith School of Business
and Institute for Systems Research
University of Maryland
College Park, MD 20742-1815

Abstract

Ordinal Optimization has emerged as an efficient technique for simulation optimization. A good allocation of simulation samples across designs can further dramatically improve the efficiency of ordinal optimization. We investigate the efficiency gains of using *dynamic* simulation allocation for ordinal optimization by comparing the *sequential* version of the optimal computing budget allocation (OCBA) method with optimal static and one-step look-ahead dynamic allocation schemes with "perfect information" on the sampling distribution. Computational results indicate that this sequential version of OCBA, which is based on *estimated* performance, can easily outperform the optimal static allocation derived using the *true* sampling distribution. These results imply that the advantage of sequential allocation often outweighs having accurate estimates of the means and variances in determining a good simulation budget allocation. Furthermore, the performance of the perfect information dynamic scheme can be viewed as an approximate upper bound on the performance of different sequential schemes, thus providing a target for further achievable efficiency improvements using dynamic allocations.

[+] This work has been supported in part by the National Science Council of the Republic of China under Grant NSC 95-2811-E-002-009, by NSF under Grants IIS-0325074 and DMI-0323220, by NASA Ames Research Center under Grants NAG-2-1643 and NNA05CV26G, by FAA under Grant 00-G-016, and by AFOSR under Grant FA95500410210.

1. Introduction

Simulation is commonly used to model and analyze complex discrete-event systems, but because of its computational demands, efficiency is an important concern (Law and Kelton 2000). We consider the problem of selecting the best design from a given set of alternatives, where the performance of each design must be estimated using simulation. To obtain a good statistical estimate for a given design, many simulation samples or replications are usually necessary. If the accuracy requirement is high and the number of alternatives is large, then the total simulation cost can easily become prohibitively high. *Ordinal Optimization* (OO) has emerged as an effective approach in this setting, where the primary objective is to pick out designs that perform *relatively* better than others, rather than accurately estimating performance metrics for all designs under consideration (Ho et al. 2000).

The efficiency of OO can be further improved through intelligent allocation of the total simulation sampling budget among different designs as the simulation proceeds (Chen 1996). 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. Chen et al. (2000) formalize this idea and develop a new approach called optimal computing budget allocation (OCBA), which can attain a speedup of an additional order of magnitude faster than base algorithms that can be proven to achieve exponential convergence for ordinal optimization. Similar budget allocation ideas have been extended to various applications (e.g., Lee 2003, Trailovic and Pao 2004, Chen and Yücesan 2005). Other simulation budget allocation schemes developed from a different perspective include the expected value of information procedure (VIP, Chick and Inoue 1998 and 2001) and the indifference zone procedure (IZ, Kim and Nelson 2006). Based on extensive numerical testing, Branke et al. (2006) show that OCBA and VIP (and their variations) are among the top performers; this note focuses on further characterizing the advantages of dynamic allocation in sequential OCBA (SOCBA). We consider only fully sequential schemes, as Inoue et al. (1999) show that they dominate two-stage procedures.

To study the efficiency issue of simulation budget allocation for ordinal optimization, we take the probability of correctly selecting the best design ($P\{\text{CS}\}$) as the measure of effectiveness of an allocation, and maximize it subject to a limited computing budget. A good allocation procedure relies on accurate information about design means and variances, but in practice, the mean and variance for each design must be estimated using simulation output. Such a setting would benefit from dynamic allocation of the computing budget, whereby only a small portion of the simulation budget is allocated at each stage, utilizing the updated parameter estimates to determine the new budget allocation iteratively.

We test the performance of the fully sequential version of OCBA using estimated means and variances. To serve as benchmarks, we consider the optimal static (one-time) allocation under the assumption of perfect information (SAPI), where the sampling distributions for all designs are known and can be used for simulation allocation, and also develop a dynamic allocation under perfect information (DAPI) based on myopic one-step look-ahead optimization. We find that the sequential allocation with perfect information is indeed much more efficient than static allocation, and even SOCBA outperforms the static allocation with perfect information. In other words, the advantage of sequential allocation often outweighs having accurate estimates of the means and variances in determining a good simulation budget allocation. However, dynamic allocation with perfect information still substantially outperforms SOCBA, indicating that there is still room for improvement in developing sequential schemes based on sample information.

This technical note is organized as follows: In the next section, we define the notation and the problem setting. In Section 3, we derive the dynamic allocation scheme under perfect information. Numerical experiments are given in Section 4. Section 5 concludes the paper.

2. Problem Setting

Consider the following optimization problem:

$$\min_i \mu_i \equiv \mathrm{E}_\xi[\, L_i(\xi)\,],$$

where i is the design index, i = 1, 2, ..., k; μ_i the design i performance measure which is the expectation of L, the sample performance, as a function of i and ξ, a random vector that represents the uncertainty in the system. Without loss of generality, we assume $\mu_1 < \mu_2 < \ldots < \mu_k$ throughout, so design 1 is the best (but this is unknown a priori), because it has the smallest mean. The setting of this paper assumes that $L_i(\xi)$ is available only via simulation (or sampling). For notational simplicity, define $X_{ij} \equiv L_i(\xi_{ij})$, which is the j-th sample of the performance measure from design i, where ξ_{ij} represents the j-th sample of ξ for design i. To estimate performance of a given design, multiple simulation samples/replications are taken, and E[$L_i(\xi)$] is estimated by the sample mean $\bar{X}_i \equiv \frac{1}{N_i}\sum_{j=1}^{N_i} X_{ij}$,where N_i represents the number of simulation samples for design i. Our goal is to select a design associated with the smallest mean performance measure among k alternative designs. Throughout, we assume that the simulation output is independent from sample to sample. Furthermore, we **assume that X_{ij} is normally distributed**, and write X_{ij} ~ N(μ_i, σ_i^2), where σ_i^2 is the variance for design i. The normality assumption is generally not a problem, because typical simulation output is obtained from an average performance or batch means, so that Central Limit Theorem effects usually hold. As N_i increases, $\bar{X}_i \sim$ N(μ_i, σ_i^2 / N_i) becomes a better estimate of μ_i in the sense that its corresponding confidence interval becomes narrower at the canonical rate of $1/\sqrt{N}$.

Ordinal optimization concentrates on ordinal comparison, achieving an exponential convergence rate for the probability of correct selection or P{CS}, where

$$\mathrm{CS} = \{\ \bar{X}_1(N_1) < \bar{X}_2(N_2)\ ,\ \bar{X}_1(N_1) < \bar{X}_3(N_3),\ \ldots,\ \bar{X}_1(N_1) < \bar{X}_k(N_k)\ \},$$

i.e., the design with the best sample mean is the design with the best theoretical mean. The objective of this paper is to study how the overall simulation quality, measured by $P\{\text{CS}\}$, can be further improved within the same computing budget. Assume that the computation cost for each sample is roughly the same across different designs. The computation cost can then be approximated by the total number of samples. To improve efficiency, ideally, we wish to choose the simulation allocation such that $P\{\text{CS}\}$ is maximized, subject to a limited computing budget T, i.e.,

$$\max_{N_1,\cdots,N_k} P\{\text{CS}\} \qquad \text{s.t. } N_1 + N_2 + \cdots + N_k = T. \tag{1}$$

However, solving such an optimal sample allocation problem is challenging, because i) there is no closed-form expression for $P\{\text{CS}\}$ in general; ii) $P\{\text{CS}\}$ is a function of the unknown design means and variances.

Our first benchmark is the static allocation under perfect information (SAPI), which solves (1) under the assumption that the means and variances are known, yielding a one-time (single) allocation (N_1^*, N_2^*, ..., N_k^*) that specifies that N_i^* simulation samples are allocated to design i. In our numerical experiments, we consider small enough T so that the optimal solution can be found using a search algorithm or exhaustive search, and the $P\{\text{CS}\}$ is generally estimated through Monte Carlo simulation, although in the special case of Example 2 in Section 4, the estimation of $P\{\text{CS}\}$ can be calculated analytically.

3. Dynamic Computing Budget Allocation with Perfect Information

The second benchmark we consider is a *dynamic* allocation scheme again derived under the assumption of perfect information (DAPI). For the dynamic allocation, instead of allocating all of the T simulation samples at the beginning, we sequentially allocate one sample at each iteration, using both the known means and variances, along with the sample information obtained to that point. Suppose we have conducted N_1, N_2, …, N_k simulation samples for the k designs and obtained the sample means: $\bar{X}_1$, $\bar{X}_2$, …, and $\bar{X}_k$. The decision problem is which design to allocate the next simulation sample in order to maximize the $P\{\text{CS}\}$ *after the additional allocation*, given the current sample information. A truly optimal dynamic solution should maximize the P{CS} after *all* of T has been allocated, and would involve solving a stochastic dynamic programming problem. Because obtaining such a solution is intractable even for small problems, we derive a heuristic solution based on a *myopic (greedy) one-step look-ahead* optimization problem.

Let $a \in \{1, 2, …, k\}$ denote the index of the design to which the next sample is allocated and $Y_a \sim \text{N}(\mu_a, \; \sigma_a^2)$ denote the corresponding new sample which will be obtained. Define $\bar{X}_i' \equiv \dfrac{1}{N_i + I_{\{i=a\}}}\left(\sum_{j=1}^{N_i} X_{ij} + Y_a I\{i=a\}\right)$, $i = 1, 2, ..., k$, where $I(\cdot)$ is the indicator function, and the correct selection *after the additional allocation* is defined by

$$\text{CS}' = \{\ \overline{X}_1' < \overline{X}_2',\ \overline{X}_1' < \overline{X}_3',\ \ldots,\ \overline{X}_1' < \overline{X}_k'\ \}.$$

Then we have the following single-stage optimization problem:

$$\max_{a \in \{1,2,\ldots,k\}} P_a \equiv P\{\text{CS}' \mid \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k, a\}. \tag{2}$$

When $a = 1$, $Y_a \sim \text{N}(\mu_1, \sigma_1^2)$, so

$$P_1 \equiv P\{\text{CS}' \mid \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k, a{=}1\}$$

$$= P\{\frac{N_1\overline{X}_1 + Y_1}{N_1 + 1} < \min(\overline{X}_2, \overline{X}_3, \ldots, \overline{X}_k) \mid \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k\}$$

$$= \Phi\left(\frac{(N_1+1)\min\{\overline{X}_2, \ldots, \overline{X}_k\} - N_1\overline{X}_1 - \mu_1}{\sigma_1}\right),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. On the other hand, if we decide to simulate design $a \neq 1$, then

$$P_a \equiv P\{\text{CS}' \mid \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k, a \neq 1\}$$

$$= P\{\overline{X}_1 < \frac{N_a\overline{X}_a + Y_a}{N_a + 1} \text{ and } \overline{X}_1 < \overline{X}_j, \text{ for } \forall\, j \neq a \neq 1 \mid \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_k\}$$

$$= \bigcap_{\substack{j=2 \\ j \neq a}}^{k} I(\overline{X}_1 < \overline{X}_j)\ \Phi\left(\frac{N_a\overline{X}_a - (N_a+1)\overline{X}_1 + \mu_a}{\sigma_a}\right).$$

Define

$$s \equiv \arg\min_{j \neq 1} \overline{X}_j, \text{ and } q \equiv \arg\min_{j \neq 1, j \neq s} \overline{X}_j,$$

which correspond to the designs with the lowest and second lowest sample means, respectively, excluding the true best design. To solve the optimal decision problem in (2), we have to consider different cases based on the order of $\overline{X}_1$, $\overline{X}_2$, …, and $\overline{X}_k$ as follows.

<u>Case 1</u>. $\overline{X}_1 < \min\{\overline{X}_2, \ldots, \overline{X}_k\}$, i.e., $\overline{X}_1 < \overline{X}_s$,

$$P_1 = \Phi\left(\frac{(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1}{\sigma_1}\right),$$

$$P_i = \Phi\left(\frac{N_i\overline{X}_i - (N_i+1)\overline{X}_1 + \mu_i}{\sigma_i}\right), \quad i = 2, 3, \ldots, k.$$

<u>Case 2</u>: $\overline{X}_s < \overline{X}_1 < \overline{X}_q$,

$$P_1 = \Phi\left(\frac{(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1}{\sigma_1}\right),$$

$$P_s = \mathrm{P}\{\overline{X_1} < \frac{N_s\overline{X_s} + Y_s}{N_s+1} \mid \overline{X}_1,\ \overline{X}_s, a = s\} = \Phi\left(\frac{N_s\overline{X}_s - (N_s+1)\overline{X}_1 + \mu_s}{\sigma_s}\right),$$

$P_i = 0$, for $i \neq s \neq 1$.

Hence, the best action is either 1 or s in this case.

<u>Case 3</u>: $\overline{X}_s < \overline{X}_q < \overline{X}_1$.

In this case, there are at least two designs whose sample means are less than design 1's sample mean. Note that our objective is to allocate this additional one simulation sample to change the order of sample means so that we can have a correct selection, i.e., to have design 1's sample mean become the smallest one. To achieve this goal with the simulation budget of only one sample, the only way is to allocate this simulation sample to design 1, because

$$P_1 = \Phi\left(\frac{(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1}{\sigma_1}\right),$$

$P_i = 0$, for $i \neq 1$.

Efficiency can become an issue when σ_1 is small, since P_1 will then be near zero, as $(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1$ is less than zero. Intuitively speaking, small σ_1 implies that $\overline{X}_1$ is not going to change quickly, but such a change is needed to correct the order of sample means and obtain a correct selection. One alternative for handling this case is to allocate multiple simulation samples at each iteration, which would allow additional allocations outside of design 1. However, solving the problem of allocating multiple samples to multiple designs can quickly lead to computational intractability. Instead, we adopt a simple but effective heuristic, which chooses either design 1 or design s to maximize the probability that the order of $\overline{X}_s$ and $\overline{X}_1$ is reversed. Define

$$P_1' \equiv P\{\ \overline{X}_1' < \overline{X}_s' \mid \overline{X}_1,\ \overline{X}_s, a = 1\}$$

$$= P\{\frac{N_1\overline{X}_1 + Y_1}{N_1+1} < \overline{X}_s \mid \overline{X}_1,\ \overline{X}_s, a = 1\} = \Phi\left(\frac{(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1}{\sigma_1}\right),$$

$$P_s' \equiv P\{\ \overline{X}_1' < \overline{X}_s' \mid \overline{X}_1,\ \overline{X}_s, a = s\}$$

$$= \mathrm{P}\{\overline{X_1} < \frac{N_s\overline{X_s} + Y_s}{N_s+1} \mid \overline{X}_1,\ \overline{X}_s, a = s\} = \Phi\left(\frac{N_s\overline{X}_s - (N_s+1)\overline{X}_1 + \mu_s}{\sigma_s}\right),$$

which are equal to P_1 and P_2 in Case 2, respectively. Therefore the decision problem is the same as that in Case 2.

Since Φ is a monotonically increasing function bounded from below ($\Phi(-\infty) = 0$), maximizing P_a is equivalent to maximizing the argument in Φ, which leads to the following algorithm for DAPI:

INITIALIZE Perform n_0 simulation replications for all designs; $N_1 = N_2 = \ldots = N_k = n_0$.

LOOP While $\sum_{i=1}^{k} N_i < T$ do

SELECT If $\overline{X}_1 \leq \overline{X}_s$, then $a^* = \arg \max_{a \in \{1,2,\ldots,k\}} Q_a$;

Else, $a^* = \arg \max_{a \in \{1,s\}} Q_a$,

where $Q_1 = \dfrac{(N_1+1)\overline{X}_s - N_1\overline{X}_1 - \mu_1}{\sigma_1}$ and $Q_i = \dfrac{N_i\overline{X}_i - (N_i+1)\overline{X}_1 + \mu_i}{\sigma_i}$, for $i \neq 1$.

SIMULATE Perform one replication for design a^*; update $\overline{X}_{a^*}$; $N_{a^*} = N_{a^*} + 1$.

END OF LOOP

Initially, n_0 simulation runs for each of k designs are performed to obtain sample statistics such as sample means and variances. Then the allocation procedure is applied to determine how to allocate the additional simulation budget and the selected design is simulated for one replication. The procedure is continued until the total budget T is exhausted.

4. Numerical Testing and Evaluation on Practical Allocation Procedures

In this section, we use four examples to numerically test and compare SOCBA with the benchmarks of SAPI and DAPI, as well as with equal allocation.

4.1 Practical Allocation Procedures

Equal Allocation (Equal)

This is a simple way to conduct simulation experiments and has been widely applied. The simulation budget is equally allocated to all designs, i.e., $N_i = T/k$ for all $i = 1, 2, \ldots, k$.

OCBA (Chen el al. 2000)

Under a Bayesian model, OCBA intends to maximize a $P\{\text{CS}\}$ approximation and offers an asymptotic solution to this approximation:

- $\dfrac{N_i}{N_j} = \left(\dfrac{\sigma_i/\delta_{b,i}}{\sigma_j/\delta_{b,j}}\right)^2$, $i, j \in \{1, 2, \ldots, k\}$, and $i \neq j \neq b$, (3.1)

- $N_b = \sigma_b \sqrt{\sum_{i=1, i \neq b}^{k} \frac{N_i^2}{\sigma_i^2}}$, (3.2)

where $\delta_{b,i} \equiv \bar{X}_b - \bar{X}_i$ and $b = \arg\min_i \{\bar{X}_i\}$. Note that design b is the design has smallest sample mean, which is estimated using simulation output. Neither OCBA nor SOCA uses knowledge of which design is the true best design. In implementation, variances are approximated by sample variances. The SOCBA algorithm simply replaces the **SELECT** step in DAPI algorithm by the following:

- Calculate $T' = N_1 + N_2 + \ldots + N_k + 1$.
- Find N_i', for $i = 1, 2, ..., k$, such that N_i' satisfies (3.1), (3.2), and $N_1' + N_2' + \ldots + N_k' = T'$.
- $a^* = \arg\max_{a \in \{1,2,\ldots,k\}} \{N_a' - N_a\}$.

4.2. Numerical Testing

In comparing the procedures, the measurement of effectiveness used is the $P\{\text{CS}\}$ estimated by the fraction of times the procedure successfully finds the true best design out of 1,000,000 independent experiments. In all of the examples, $n_0 = 10$.

Example 1.

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

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

and the total computing budget $T = 121$. 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 SAPI, we should allocate only one sample to design 1, but should equally divide the remaining budget between designs 2 and 3. Thus, $N_1 = 1$ and $N_2 = N_3 = 60$.

Table 1 shows the test results using different allocation procedures. We see that DAPI performs much better than SAPI, providing an indication of the benefit of dynamic allocation; however, it is interesting to observe that sequential OCBA outperforms SAPI, indicating that suboptimal dynamic allocation can outperform the optimal static allocation.

Table 1. Comparison of estimated $P\{\text{CS}\}$ based on 1,000,000 replications for Example 1.

Procedures	SAPI	DAPI	Equal	SOCBA
$\boldsymbol{P}$**{CS}**	0.720	0.964	0.640	0.828

Example 2.

There are five designs:

$X_{1j} \sim \text{N}(0,\ 0^2)$, $X_{2j} \sim \text{N}(0.4,\ 1.5^2)$, $X_{3j} \sim \text{N}(0.4,\ 3^2)$, $X_{4j} \sim \text{N}(1,\ 3^2)$, and $X_{5j} \sim \text{N}(2,\ 3^2)$.

In this case, design 1 is the best design, and because design 1's variance is zero and the assumption of independence and normal distribution, the P{CS} can be calculated analytically:

$$P\{\text{CS}\} = \Pr\{\ 0 < \overline{X}_2(N_2)\ ,\ 0 < \overline{X}_3(N_3),\ 0 < \overline{X}_4(N_4),\ 0 < \overline{X}_5(N_5)\ \}$$

$$= \Phi\left(\frac{0.4}{1.5/\sqrt{N_2}}\right)\Phi\left(\frac{0.4}{3.0/\sqrt{N_3}}\right)\Phi\left(\frac{1.0}{3.0/\sqrt{N_4}}\right)\Phi\left(\frac{2.0}{3.0/\sqrt{N_5}}\right).$$

To investigate the performance for different procedures as a function of the available computing budget, we vary T between 50 and 150, with the achieved $P\{\text{CS}\}$ shown in Figure 1. All procedures except SAPI start at the same $P\{\text{CS}\}$ when $T = 50$ because of the n_0 (=10) initial samples in the first stage, and then $P\{\text{CS}\}$ increases monotonically with increasing T. However, DAPI achieves the highest $P\{\text{CS}\}$ for the same amount of computing budget. SOCBA quickly overtakes SAPI as more computing budget is available for dynamic allocation at later stage.

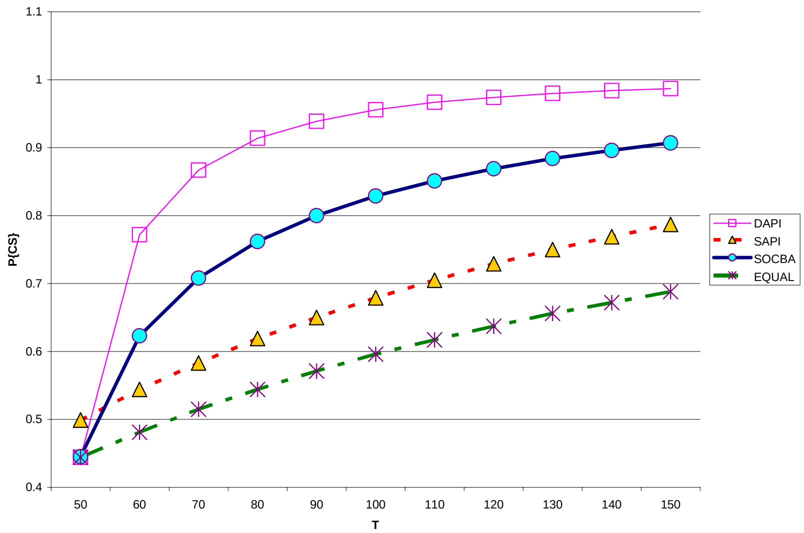

Figure 1. $P\{\text{CS}\}$ vs. T using different allocation procedures in example 2.

Example 3.

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

$X_{ij} \sim \text{N}(i,\ 6^2)$, for $i = 1, 2, ..., 10$. In this case, we evaluate $P\{\text{CS}\}$ for SAPI using Monte Carlo simulation for all combinations, which requires a huge amount of computation time. Again, SOCBA overtakes SAPI as more computing budget is available for dynamic allocation at later stage. The difference in performance between DAPI and SOCBA also indicates that there is further room for improvement in finding a good dynamic allocation procedure.

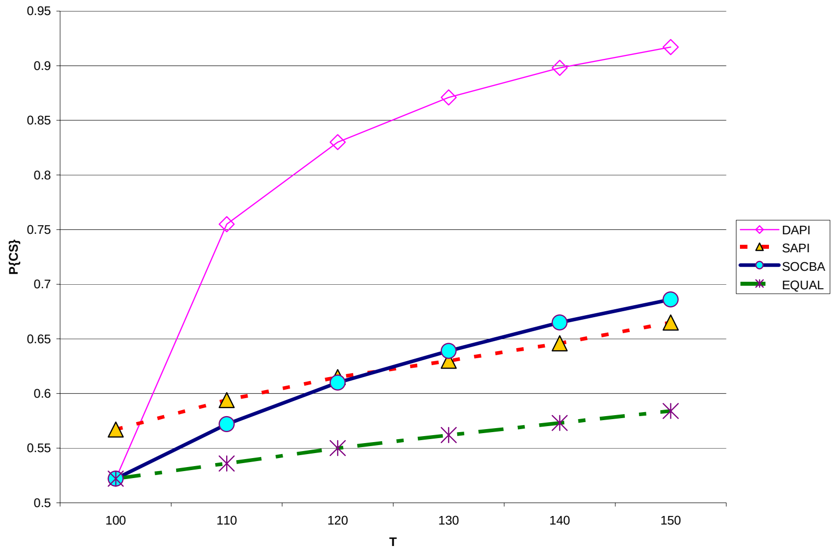

Figure 2. $P\{\text{CS}\}$ vs. T using different allocation procedures in example 3.

Example 4.

This experiment investigates sensitivity to non-normality, since recall that DAPI and OCBA are derived under the assumption of normality. The parameters are chosen so that they have the same means and variances as those in Example 3. Specifically, there are again ten designs, but the samples are uniformly distributed:

$X_{ij} \sim \text{Uniform}[i- 6\sqrt{3}\ , i+6\sqrt{3}\ ]$, for $i = 1, 2, ..., 10$.

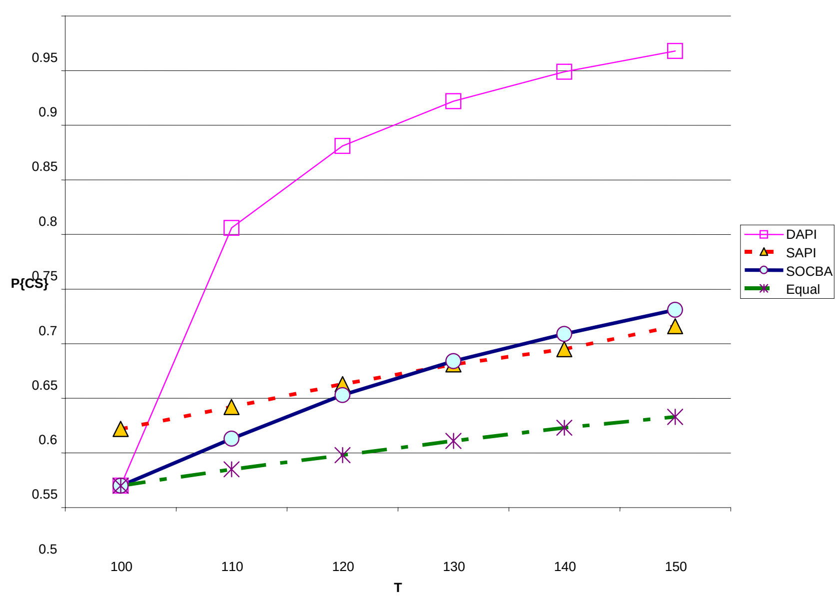

Figure 3. $P\{\text{CS}\}$ vs. T using different allocation procedures in example 4.

The test results shown in Figure 3 are very similar to those in Example 3 (Figure 2), indicating a robustness to the normality assumption.

5. Conclusion

This paper investigates the efficiency gains in using dynamic simulation budget allocation for ordinal optimization by comparing with benchmark allocations for optimal static and one-step look-ahead dynamic sequential schemes with perfect information. DAPI can be viewed as a closed-loop controller that adjusts to realization of the output samples for each particular application of the optimization procedure, and therefore improves upon SAPI, which can be considered to be an optimal open-loop system. Numerical experiments indicate that DAPI outperforms SAPI, and even sequential OCBA outperforms the optimal static allocation, which allocates the computing budget in a way that one would intuitively think is optimal. Results from the sequential solution with perfect information indicate that there is further room for improvement in developing efficient dynamic simulation allocation schemes, and research along this avenue is ongoing.

References

1. Branke, J., S. E. Chick, and C. Schmidt, "Selecting a Selection Procedure," submitted to *Management Science*, 2006.
2. Chen, C. H. "A Lower Bound for the Correct Subset-Selection Probability and Its Application to Discrete Event System Simulations," *IEEE Transactions on Automatic Control*, Vol. 41, No. 8, pp. 1227-1231, August 1996.
3. 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.
4. Chen, C. H., and E. Yücesan, "An Alternative Simulation Budget Allocation Scheme for Efficient Simulation," *International Journal of Simulation and Process Modeling*, Vol. 1, Nos. 1/2, pp. 49-57, 2005.
5. Chick, S. E., K. Inoue, "Sequential Allocations That Reduce Risk For Multiple Comparisons," *Proceedings of 1998 Winter Simulation Conference*, pp. 669-676, 1998.
6. 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.
7. Ho, Y. C., C. G. Cassandras, C. H. Chen, and L. Dai, "Ordinal Optimization and Simulation," *Journal of Operational Research Society*, Vol. 51, No. 4, pp. 490-500, April 2000.
8. Inoue, K., S. E. Chick, and C. H. Chen, "An Empirical Evaluation of Several Methods to Select the Best System," *ACM Transitions on Modeling and Computer Simulation*, Vol. 9, No. 4, pp. 381-407, October 1999.

9. Kim, S.-H. and B. L. Nelson. "Selecting the Best System," in S. G. Henderson and B. L. Nelson (Eds.), Handbook in Operations Research and Management Science: Simulation. Elsevier. (to appear). 2006.

10. Law, A.M., W.D. Kelton, *"Simulation Modeling and Analysis"*, 3rd edition, McGraw-Hill, 2000.

11. Lee, L. H., "A Simulation Study on Sampling and Selecting under Fixed Computing Budget," *Proceedings of 2003 Winter Simulation Conference*, pp. 535-542, December 2003.

12. Trailovic, L. and L. Y. Pao, "Computing Budget Allocation for Efficient Ranking and Selection of Variances with Application to Target Tracking Algorithms," *IEEE Transactions on Automatic Control*, 49(1): 58-67, January 2004.