INTELLIGENT SIMULATION FOR ALTERNATIVES COMPARISON AND APPLICATION TO AIR TRAFFIC MANAGEMENT[*]

Chun-Hung CHEN   Donghai HE

Department of Systems Engineering and Operations Research
George Mason University, Fairfax, VA 22030, USA
cchen9@gmu.edu   dhe1@gmu.edu

Abstract

We present a simulation run allocation scheme for improving efficiency in simulation experiments for decision making under uncertainty. This scheme is called Optimal Computing Budget Allocation (OCBA). OCBA advances the state-of-the-art by intelligently allocating a computing budget to the candidate alternatives under evaluation. The basic idea is to spend less computational effort on simulating non-critical alternatives to save computation cost. In particular, OCBA is employed to intelligently provide the smallest number of simulation runs for a desired accuracy. In this paper, we present a new and more general OCBA scheme which can consider cases that users are interested not only the best design, but also any one in a good design set. In addition, this paper also presents the application of our OCBA to a design problem in US air traffic management. The national air traffic system in US is modeled as a large, complex, and stochastic network. The numerical examples show that the computation time can be reduced by 54% to 88% with the use of OCBA.

Keywords: stochastic simulation, stochastic optimization, air traffic management.

1. Introduction

Computer simulation technology has matured over the past decade and is now commonly used to evaluate large-scale real systems with complex stochastic behavior. Simulation allows one to more accurately specify a system through the use of logically complex, and often non-algebraic, variables and constraints. This capability compliments the inherent limitation of traditional optimization. However, the added flexibility often creates models that are computationally intractable. The efficiency issue becomes an even bigger concern when one intends to evaluate and compare a set of alternatives. A decision maker is forced to compromise on simulation accuracy, modeling accuracy, and

the optimality of the selected design. This paper is to address such an efficiency issue.

There exists a large literature on innovative methods for improving the efficiency of simulation experiments. Fishman (1996) provide a comprehensive presentation of recent developments in simulation methodologies. Some methods exploit the fact that the required number of simulation runs decreases when the simulation variance is reduced. Experimental as well as theoretical results for some special cases have shown that inducing some sort of dependence among experiments for different designs can increase the chance of selecting the true best design. In other words, the so-called variance reduction techniques exploit the fact that the required number of simulation runs decreases when the simulation variance is reduced. Glasserman and Yao (1992) show that the schemes of common random numbers and control variates are helpful in obtaining better confidence intervals for various selection procedures when the performance measure is obtained by averaging i.i.d. random variables with normal distributions. Heidelberger (1993) deals with rare event problems by developing an importance sampling scheme. Antithetic Variates induce negative correlation between separate runs (e.g., Cheng 1984). The selection of an adequate variance reduction or correlated sampling technique usually depends on the particular model of interest. Therefore, a thorough understanding of the inner workings of the models is required for proper use of those techniques. Another major limitation of variance reduction techniques is that only the information of each design is locally used to reduce its variance for improving simulation

efficiency. The total simulation cost for a design problem may still be high.

On the other hand, several researchers have shown that allocating simulation runs in an uneven manner can significantly enhance simulation efficiency by reducing the total number of runs required to identify the best candidate design. Previous work such as Rinott (1978) develops two-stage procedures for allocating runs to designs and several papers (Bechhofer et al. 1995) extend the technique to general ranking and selection problems. The major disadvantage of existing approaches is that they utilize only the information of variance to control simulation experiments. Chen et al. (1998, 1999) address this limitation and demonstrate that the use of additional information on relative means among different alternatives can dramatically improve simulation efficiency even if only a simple heuristic is applied. Hyden and Schruben (2000), Chick and Inoue (2001), Lee (2003), Trailovic and Pao (2004) also demonstrate that simulation efficiency can be improved by utilizing more simulation information that is readily available.

The efficient simulation technique introduced in this paper intend to optimally allocate a computing budget to the designs under evaluation. Intuitively, to ensure a high probability of correctly selecting an optimal design, 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. Similarly, 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 simulation efficiency. The technique is called Optimal Computing Budget Allocation (OCBA). Chen et al. (2000) present a computing budget allocation technique to asymptotically maximize the probability of correctly selecting the best design within a given computing budget.

However, the previous work about efficient computing budget allocation by Chen et al. is limited to problems concerned with selecting only the best design. In real-life problems, particularly for those with a large number of alternatives, one may be happy with not only the best but also the second best design, or even any of the top-M designs. The contribution of this paper has two folds: first we extend the OCBA scheme to problems considering selecting any of the top-m designs. Secondly we apply our new OCBA scheme to a design problem in US air traffic system. We show that a good design alternative can be identified with only a small portion of computation time if OCBA is used.

The paper is organized as follows: In the next section, we formulate the new and extended optimal computing budget allocation problem. Section 3 presents the development of our OCBA scheme based a sequential Bayesian model. Section 4 shows numerical testing using two generic examples. The application of OCBA to air traffic management is presented in Section 5. Section 6 concludes the paper.

2. Problem Formulation

Suppose that our goal is to select a design associated with the smallest mean performance measure among k alternative designs with unequal and possibly unknown variances. Consider the following optimization in decision making problems:

$$\min_{i} m_i \equiv E_{\boldsymbol{x}}[L(\boldsymbol{q}_i, \boldsymbol{x})] \qquad (1)$$

where $\boldsymbol{q}_i \in \Theta$, the search space, is an arbitrary finite set; $\boldsymbol{q}_i$ is the system design parameter vector for design i, $i = 1, 2,..., k$; m_i the design i performance measure which is the expectation of L, the sample performance, as a function of $\boldsymbol{q}_i$ and $\boldsymbol{x}$, a random vector that represents the uncertainty in the system.

The setting of this paper assumes that $L(\boldsymbol{q}, \boldsymbol{x})$ is available only via simulation (or sampling). The system constraints are implicitly involved in the simulation process, and so are not shown explicitly in (1). To estimate performance of a given design, multiple simulation samples/replications are taken, and $E[L(\boldsymbol{q}_i, \boldsymbol{x})]$ is estimated by the sample mean:

$$\frac{1}{N_i} \sum_{j=1}^{N_i} L(\boldsymbol{q}_i, \boldsymbol{x}_{ij})$$

where $\boldsymbol{x}_{ij}$ represents the j-th sample of $\boldsymbol{x}$ for design i and N_i represents the number of simulation samples for design i. For notational simplicity, define

$$X_{ij} \equiv L(\boldsymbol{q}_i, \boldsymbol{x}_j)$$

which is the j-th sample of the performance measure from design i. Denote by

$\overline{X}_i$: the sample mean of $L(\mathbf{q}_i, \mathbf{x})$, the simulation output for design i;

$$\overline{X}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} X_{ij}$$

$\mathbf{X}_i$: the vector representing the simulation output for design i; $\mathbf{X}_i = \{X_{ij} : j=1,2,...,N_i \}$

S_i^2: the sample variance of $L(\mathbf{q}_i, \mathbf{x})$, the simulation output for design i

$\mathbf{s}_i^2$: the variance for design i, i.e., $\mathbf{s}_i^2 = \text{Var}(X_{ij})$. In practice, $\mathbf{s}_i^2$ is unknown beforehand and so is estimated by the sample variance

b: the design with the smallest sample mean performance; $b = \arg\min_i \{ \overline{X}_i \}$.

We assume that the simulation output is independent from replication to replication and that the sampling across designs is also independent. Furthermore, we assume that X_{ij} is normally distributed, and write

$$X_{ij} \sim \text{N}(\mathbf{m}, \mathbf{s}_i^2)$$

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.

In this paper, we are concerned with the selection of a good design among k alternatives. While the design with the smallest sample mean (design b) is usually selected, design b is not necessarily the one with the smallest unknown mean performance. Correct selection is usually defined as the event that design b is actually the best design (i.e., with the smallest population mean, hence the true best design). For problems with a large number of alternatives, (i.e., when k is large), the performances of the top few designs are quite close so that a decision maker may be willing to accept any of the top-M designs. In this paper, *correct selection* (CS) herein is defined as the event that design b *is* actually one of the top-M designs. We wish to maximize the probability of correctly identifying one of the top-M designs, $P\{\text{CS}\}$ in simulation experiment. Assume that the computation cost for each replication is roughly the same across different designs. The computation cost can then be approximated by $N_1 + N_2 + \cdots + N_k$, the total number of samples. We wish to choose N_1, N_2, $\cdots$, N_k such that $P\{\text{CS}\}$ is maximized, subject to a limited computing budget T, i.e.,

$$\max_{N_1,\cdots,N_k} P\{\text{CS}\}$$

$$\text{s.t. } N_1 + N_2 + \cdots + N_k = T. \tag{2}$$

Some difficulties in solving (2) include the following:

- ◆ There is no closed-form expression for the confidence level, $P\{\text{CS}\}$.
- ◆ $P\{\text{CS}\}$ can not be computed before conducting all the simulations if the mean and variance of each design are unknown.
- ◆ $N_1, N_2,...,N_k$ *are integers and the* number of combinations satisfying the constraint in (2) for $N_1, N_2, ..., N_k$ is large even for moderate k.
- ◆ Most optimization techniques require sensitivity information.

Due to these problems, solving (2) can be very difficult, especially when k is not small. Since the purpose of solving (2) is to improve overall computational efficiency, we need a relatively fast and inexpensive way of approximately solving (2) during the simulation experiment.

3. A Sequential Bayesian Approach

To solve problem (2), we must be able to first estimate $P\{CS\}$. We follow the Bayesian model introduced in Chen (1996) to develop an effective approach to estimate the general $P\{CS\}$ considered in this paper.

After the simulation is performed, a posterior distribution of $\boldsymbol{m}_i$, $p(\boldsymbol{m}_i \mid \mathbf{X}_i)$, can be constructed based on prior knowledge on the system's performance and the simulation output. The probability of correctly selecting one of the top-*M* designs can then be estimated by

$P\{CS\} \equiv P\{$ Design b is actually one of the

top-*M* designs $\}$

$= \displaystyle\sum_{m=1}^{M} P\{$ Design b is actually the *m*-th

smallest design $\}$

$= \displaystyle\sum_{m=1}^{M} P\{$ There are exactly *m*-1

designs which are smaller design b $\}$

$= \displaystyle\sum_{m=1}^{M} \{ \sum_{r_1=1}^{k} \sum_{r_2>r_1}^{k} \cdots \sum_{r_{m-1}>r_{m-2}}^{k} P\{ (\bigcap_{i=1}^{m-1}$

$\boldsymbol{m}_{r_i} < \boldsymbol{m}_b) \bigcap (\bigcap_{i \neq \{r_1, \cdots r_{m-1}\}} \boldsymbol{m}_b < \boldsymbol{m}_i) \mid \mathbf{X}_i,$

$i = 1, 2, ..., k \} \}$

In this paper, we consider non-informative prior distributions. This implies that no prior knowledge is available about the performance of any design alternative before conducting the simulation.

Lemma 1. If the simulation output is normally distributed and if no prior knowledge is available about the performance before conducting the simulation, then the posterior

distribution $p(\boldsymbol{m}_i \mid \mathbf{X}_i)$ is (See DeGroot 1970)

$$N(\overline{X}_i, \ \frac{\boldsymbol{s}_i^2}{N_i}), \quad \text{for} \quad i = 1, ..., k.$$

After the simulation is performed, $\overline{X}_i$ and $\boldsymbol{s}_i^2$ can be estimated, and $P\{CS\}$ can be estimated using Monte Carlo simulation. However, estimating $P\{CS\}$ via Monte Carlo simulation is time-consuming. Our budget allocation approach, as will be presented later, needs to estimate $P\{CS\}$ several times. We therefore need a relatively fast and inexpensive way of estimating $P\{CS\}$ within the budget allocation procedure. Based on Lemma 2 in Chen (1996),

$P\{CS\} \geq$

$$\sum_{m=1}^{M} \{ \sum_{r_1=1}^{k} \sum_{r_2>r_1}^{k} \cdots \sum_{r_{m-1}>r_{m-2}}^{k} [\prod_{i=1}^{m-1} P\{ \boldsymbol{m}_{r_i} < \boldsymbol{m}_b \mid$$

$$\mathbf{X}_b, \mathbf{X}_{r_i} \} \prod_{i \neq \{r_1, ..., r_{m-1}\}} P\{ \boldsymbol{m}_b < \boldsymbol{m}_i \mid \mathbf{X}_b, \mathbf{X}_i \}] \}$$

$$\equiv APCS. \tag{3}$$

We refer to the lower bound of the correct selection probability in (3) as the *Approximate Probability of Correct Selection* (*APCS*). The computation of *APCS* involves only summarizing some products of pair wise comparison probabilities, which are simple to compute. Specifically,

$$P\{ \boldsymbol{m}_b < \boldsymbol{m}_i \mid \mathbf{X}_b, \mathbf{X}_i \} = \Phi \left(\frac{\overline{X}_i - \overline{X}_b}{\sqrt{\dfrac{\boldsymbol{s}_i^2}{N_i} + \dfrac{\boldsymbol{s}_b^2}{N_b}}} \right)$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. Indeed, no extra Monte Carlo simulation is needed. Numerical testing in Chen (1996)

shows that such pairwise comparison probabilities provide reasonably good approximations to the probabilities we intend to approximate. In addition, *APCS* is asymptotically close to $P\{CS\}$. Another advantage for *APCS* is that the sensitivity information on the $P\{CS\}$ with respect to the number of simulation replications, N_i – which is central to the allocation of the computing budget – can be easily obtained. *APCS* is therefore used to approximate $P\{CS\}$ within the budget allocation procedure.

We now present a method to estimate the sensitivity information about how *P{CS}* changes as N_i changes. Let Δ_i be a non-negative integer denoting the number of additional simulation replications allocated to design i. We are interested in assessing how $P\{CS\}$ or *APCS* would be affected if design i is simulated for Δ_i additional replications. Note that this assessment must be made *before* actually conducting Δ_i simulation replications.

Based on the Law of Large Numbers,

$$\frac{1}{N_i} \sum_{j=1}^{N_i} X_{ij} \to m_i \quad \text{as} \quad N_i \to \infty$$

For large N_i, an approximate posterior distribution for the performance measure is

$$N(m_i, \; \frac{s_i^2}{N_i})$$

If we conduct Δ_i additional replications on design i, the posterior distribution for design i is approximately

$$N(m_i, \; \frac{s_i^2}{N_i + \Delta_i}).$$

For a finite N_i, define

$$\tilde{m}_i \sim N(\frac{1}{N_i} \sum_{j=1}^{N_i} X_{ij}, \; \frac{s_i^2}{N_i + \Delta_i})$$

for design i. $\hspace{2em}$ (4)

which is a good approximation to the predictive posterior distribution when Δ_i is relatively small compared to N_i, because we know that the posterior distribution, after $N_i + \Delta_i$ replications are actually performed, will become

$$N(\frac{1}{N_i + \Delta_i} \sum_{j=1}^{N_i + \Delta_i} X_{ij}, \; \frac{s_i^2}{N_i + \Delta_i}).$$

The difference between

$$\frac{1}{N_i} \sum_{j=1}^{N_i} X_{ij} \text{ and } \frac{1}{N_i + \Delta_i} \sum_{j=1}^{N_i + \Delta_i} X_{ij} \text{ will be small}$$

if Δ_i is small (and/or N_i is large). The estimated *APCS* can then be calculated by plugging (4) into the *APCS* formula in (3). We refer to this approximation as the *EAPCS* (*Estimated Approximate Probability of Correct Selection*). *EAPCS* provides sensitivity information about how *APCS* will change if additional Δ_i simulations are performed on design i.

EAPCS can be computed very easily. To give an expression for *EAPCS*, we first rewrite

$$APCS = \sum_{m=1}^{M} \{ \sum_{\substack{r_1=1; r_1 \neq i}}^{k} \sum_{\substack{r_2 > r_1; r_2 \neq i}}^{k} \cdots \sum_{\substack{r_{m-1} > r_{m-2} \\ r_{m-1} \neq i}}^{k}$$

$$\{ [\prod_{j=1}^{m-1} P\{ m_{r_j} < m_b \mid \mathbf{X}_b, \mathbf{X}_{r_j} \}]$$

$$[P\{ m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_{r_j} \}] *$$

$$[\prod_{\substack{j \neq i; \\ j \neq \{r_1, \ldots, r_{m-1}\}}} P\{ m_b < m_j \mid \mathbf{X}_b, \mathbf{X}_j \}]\} +$$

$$\sum_{\substack{r_1=1; r_1 \neq i}}^{k} \sum_{\substack{r_2 > r_1; r_2 \neq i}}^{k} \cdots \sum_{\substack{r_{m-2} > r_{m-3} \\ r_{m-2} \neq i}}^{k}$$

$$\{[\prod_{j=1}^{m-2} P\{\,\boldsymbol{m}_{r_j} < \boldsymbol{m}_b \mid \mathbf{X}_b, \mathbf{X}_{r_j}\,\}]\ *$$

$$[P\{\,\boldsymbol{m}_i < \boldsymbol{m}_b \mid \mathbf{X}_b, \mathbf{X}_i\,\}]$$

$$[\prod_{\substack{j\neq i;\\ j\neq\{r_1,\dots,r_{m-2}\}}} P\{\,\boldsymbol{m}_b < \boldsymbol{m}_j \mid \mathbf{X}_b, \mathbf{X}_j\,\}]\}\}$$

When the simulation is executed for $(N_1, N_2, ..., N_k)$ replications,

$EAPCS(N_1, N_2, ..., N_{i-1}, N_i+\Delta_i, N_{i+1}, ..., N_k)$

$$= \sum_{m=1}^{M} \{\ \sum_{r_1=1;r_1\neq i}^{k} \sum_{r_2>r_1;r_2\neq i}^{k} \cdots \sum_{\substack{r_{m-1}>r_{m-2}\\ r_{m-1}\neq i}}^{k}$$

$$\{[\prod_{j=1}^{m-1} P\{\,\boldsymbol{m}_{r_j} < \boldsymbol{m}_b \mid \mathbf{X}_b, \mathbf{X}_{r_j}\,\}]$$

$$[P\{\,\boldsymbol{m}_b < \tilde{\boldsymbol{m}}_i \mid \mathbf{X}_b, \mathbf{X}_{r_j}\,\}]\ *$$

$$[\prod_{\substack{j\neq i;\\ j\neq\{r_1,\dots,r_{m-1}\}}} P\{\,\boldsymbol{m}_b < \boldsymbol{m}_j \mid \mathbf{X}_b, \mathbf{X}_j\,\}]\}+$$

$$\sum_{r_1=1;r_1\neq i}^{k} \sum_{r_2>r_1;r_2\neq i}^{k} \cdots \sum_{\substack{r_{m-2}>r_{m-3}\\ r_{m-2}\neq i}}^{k}$$

$$\{[\prod_{j=1}^{m-2} P\{\,\boldsymbol{m}_{r_j} < \boldsymbol{m}_b \mid \mathbf{X}_b, \mathbf{X}_{r_j}\,\}]\ *$$

$$[P\{\,\tilde{\boldsymbol{m}}_i < \boldsymbol{m}_b \mid \mathbf{X}_b, \mathbf{X}_i\,\}]$$

$$[\prod_{\substack{j\neq i;\\ j\neq\{r_1,\dots,r_{m-2}\}}} P\{\,\boldsymbol{m}_b < \boldsymbol{m}_j \mid \mathbf{X}_b, \mathbf{X}_j\,\}]\}\}$$

for $i \neq b$, and

$EAPCS(N_1, N_2, ..., N_{b-1}, N_b+\Delta_b, N_{b+1}, ..., N_k)$

$$= \sum_{m=1}^{M} \{\ \sum_{r_1=1}^{k} \sum_{r_2>r_1}^{k} \cdots \sum_{r_{m-1}>r_{m-2}}^{k} \{\prod_{i=1}^{m-1} P\{\,\boldsymbol{m}_{r_i} < \tilde{\boldsymbol{m}}_b$$

$$\mid \mathbf{X}_b, \mathbf{X}_{r_i}\,\} \prod_{i\neq\{r_1,\dots,r_{m-1}\}} P\{\,\tilde{\boldsymbol{m}}_b < \boldsymbol{m}_i \mid \mathbf{X}_b, \mathbf{X}_i\,\}\}\ \}$$

$$(5)$$

To ensure that $\Delta_1, \Delta_2, ..., \Delta_k$ are small as the simulation proceeds, we develop the following sequential algorithm. Instead of finding the best $N_1, N_2, ..., N_k$ at the beginning of a simulation experiment, we sequentially add to the computing budget and determine the number of simulation replications such that the improvement of *EAPCS* is maximized. In doing so, $\Delta_1, \Delta_2, ..., \Delta_k$ should not be too large, which can be ensured by allocating the computing budget conservatively (i.e., in small increments).

Let $\Delta \equiv \sum_{i=1}^{k} \Delta_i$. More specifically, Δ is a one-time computing budget increase in our sequential algorithm. We sequentially increase the computing budget by Δ until the total budget T is exhausted. At step l, $l = 1, 2, ...,$ we therefore have

$$\max_{\Delta_1^l,\dots,\Delta_k^l} EAPCS(N_1^l + \Delta_1^l, N_2^l + \Delta_2^l, \cdots, N_k^l + \Delta_k^l)$$

s.t. $\sum_{i=1}^{k}\Delta_i^l = \Delta$ and $\Delta_i^l \geq 0$ for all i. $\quad(6)$

Solving the optimization problem in (6) can be time-consuming. In this paper, we consider a simple greedy approach, where q most *promising* designs are chosen and the computing budget is distributed equally among them (each design receives Δ/q). Since we want to maximize the *EAPCS*, *promising* designs can be defined as those designs that have the highest potential to increase the *EAPCS* if they are simulated further. For each design, we calculate the anticipated increase in the *EAPCS* if a computing budget of Δ/q is allocated to it. Then a design is chosen if its anticipated *EAPCS* increase is among the top-q.

Note that before conducting the simulation,

neither the *APCS* nor a good way to allocate the simulation budget is known. Therefore, all designs are initially simulated with n_0 replications in the first stage. In summary, we have the following algorithm (assuming T-kn_0 is a multiple of Δ):

Sequential Algorithm for Optimal Computing Budget Allocation (OCBA)

Step 0. Perform n_0 simulation replications for each of the k designs,

$l \leftarrow 0$,

$N_1^l = N_2^l = ... = N_k^l = n_0$,

$T \leftarrow T - kn_0$.

Choose a positive integer q and an appropriate Δ such that $t = \Delta / q$ is an integer

Step 1. If $T = 0$, stop, otherwise, go to Step 2.

Step 2. For $i = 1, ..., n$, calculate

$D_i \equiv EAPCS*$

$(N_1^l, N_2^l,..,N_{i-1}^l, N_i^l + t, N_{i+1}^l,..,N_k^l)$

$- APCS (N_1^l, N_2^l,..,N_{i-1}^l, N_i^l, N_{i+1}^l,..,N_k^l)$.

Step 3 Find the set $S(q) \equiv \{i : D_i$ is among the q highest values$\}$

Step 4 $\Delta_i^l = \tau$, for all $i \in S(q)$; otherwise,

$\Delta_i^l = 0$.

Step 5. Perform Δ_i^l additional simulation replications for design i, $i = 1, ..., k$.

$N_i^{l+1} = N_i^l + \Delta_i^l$, for $i = 1, ..., k$.

$l \leftarrow l + 1$.

update $\dfrac{1}{N_i^l} \sum\limits_{j=1}^{N_i^l} X_{ij}$ (and S_i^2 if the variance is unknown), for $i = 1, ...,k$.

$T \leftarrow T - \Delta$.

go to step 1.

While the *EAPCS* formula in (5) looks quite lengthy, the computational complexity is not much more complex than that for *APCS* in (3). It mainly includes summations of some products of pairwise comparison probabilities, which are simple to compute. In finding the designs with high D_i in Steps 2 and 3, the computation can be further simplified as follows. Define

$$C \equiv \prod_{i=1,i\neq b}^{k} P(m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i)$$

which remains as a constant when Δ_i changes. Therefore, the ranking of D_i is the same as D_i/C.

For $i \neq b$,

$$D_i / C = \sum_{m=1}^{M} \{ \sum_{r_1=1;r_1 \neq i}^{k} \sum_{\substack{r_2>r_1;r_2 \neq i}}^{k} \cdots \sum_{\substack{r_{m-1}>r_{m-2} \\ r_{m-1}\neq i}}^{k}$$

$$[\prod_{j=1}^{m-1} \frac{P\{m_{r_j} < m_b \mid \mathbf{X}_{r_j}, \mathbf{X}_b\}}{P\{m_b < m_{r_j} \mid \mathbf{X}_b, \mathbf{X}_{r_j}\}}]$$

$$- \sum_{r_1=1;r_1 \neq i}^{k} \sum_{r_2>r_1;r_2 \neq i}^{k} \cdots \sum_{\substack{r_{m-2}>r_{m-3} \\ r_{m-2}\neq i}}^{k}$$

$$[\prod_{j=1}^{m-2} \frac{P\{m_{r_j} < m_b \mid \mathbf{X}_{r_j}, \mathbf{X}_b\}}{P\{m_b < m_{r_j} \mid \mathbf{X}_b, \mathbf{X}_{r_j}\}}] \} *$$

$$\frac{P\{m_b < \tilde{m}_i \mid \mathbf{X}_b, \mathbf{X}_i\} - P\{m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}}{P\{m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}}$$

$$= \sum_{r_1=1;r_1 \neq i}^{k} \sum_{r_2>r_1;r_2 \neq i}^{k} \cdots \sum_{\substack{r_{M-1}>r_{M-2} \\ r_{M-1}\neq i}}^{k}$$

$$\left[\prod_{j=1}^{M-1} \frac{P\{m_{r_j} < m_b \mid \mathbf{X}_{r_j}, \mathbf{X}_b\}}{P\{m_b < m_{r_j} \mid \mathbf{X}_b, \mathbf{X}_{r_j}\}} \right] \quad * $$

$$\frac{P\{m_b < \tilde{m}_i \mid \mathbf{X}_b, \mathbf{X}_i\} - P\{m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}}{P\{m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}}$$

$$(7)$$

and

$$D_b / C = \sum_{m=1}^{M} \sum_{r_1=1}^{k} \sum_{r_2 > r_1}^{k} \cdots \sum_{r_{m-1} > r_{m-2}}^{k}$$

$$\left\{ \left[\prod_{i=1}^{m-1} \frac{P\{m_{r_i} < \tilde{m}_b \mid \mathbf{X}_{r_i}, \mathbf{X}_b\}}{P\{m_b < m_{r_i} \mid \mathbf{X}_b, \mathbf{X}_{r_i}\}} \right. \right.$$

$$\left. \prod_{i \neq \{r_1, \dots, r_{m-1}\}} \frac{P\{\tilde{m}_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}}{P\{m_b < m_i \mid \mathbf{X}_b, \mathbf{X}_i\}} \right]$$

$$- \prod_{i=1}^{m-1} \frac{P\{m_{r_i} < m_b \mid \mathbf{X}_{r_i}, \mathbf{X}_b\}}{P\{m_b < m_{r_i} \mid \mathbf{X}_b, \mathbf{X}_{r_i}\}} \right\} .$$

$$(8)$$

Hence, Steps 2 and 3 can be replaced with the follows.

Step 2. For $i = 1, ..., n$, calculate D_i / C

Step 3. Find the set $S(q) \equiv \{i : D_i/C$ is among the q highest values$\}$

4. Numerical Testing on Generic Examples

In this section, we use two generic examples to test our OCBA approach. Example 2 is a larger problem. We compare the numerical performance for cases with and without using OCBA.
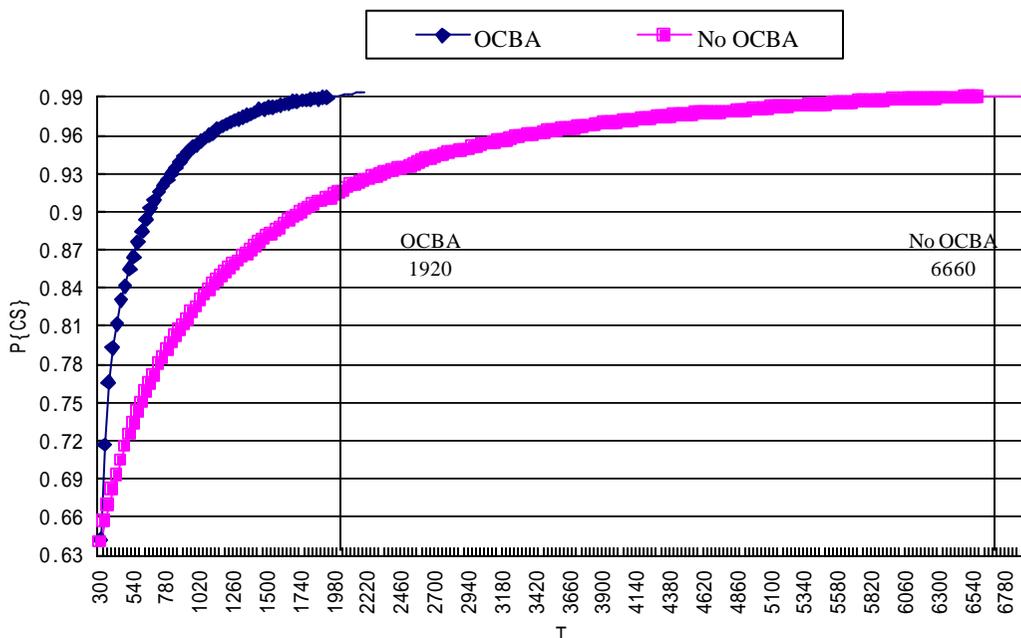

Figure 1 P{CS} vs. T with and without using OCBA for example 4.1. The computation costs in order to attain P{CS} = 99% are indicated.
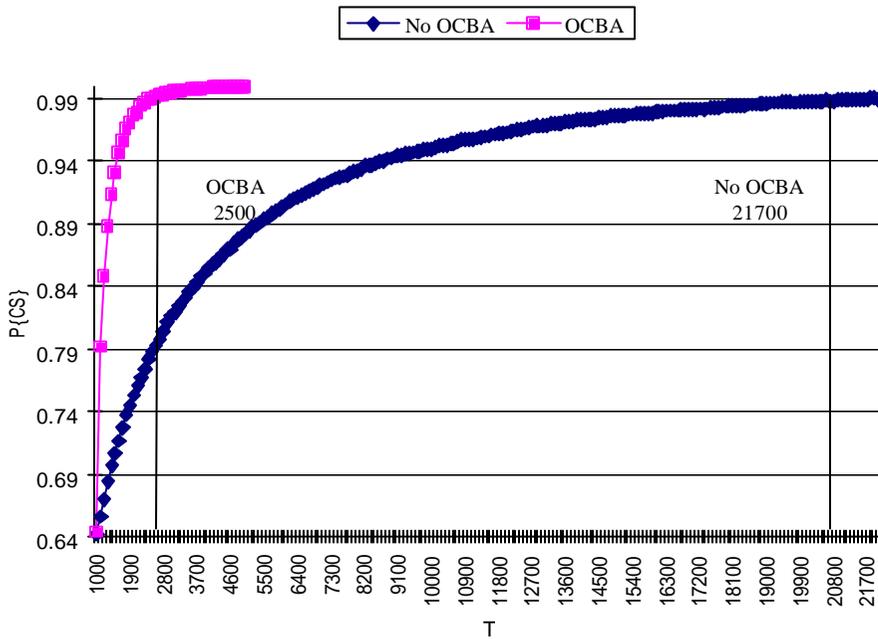
Figure 2 P{CS} vs. T with and without using OCBA for example 4.2. The computation costs in order to attain P{CS} = 99% are indicated.
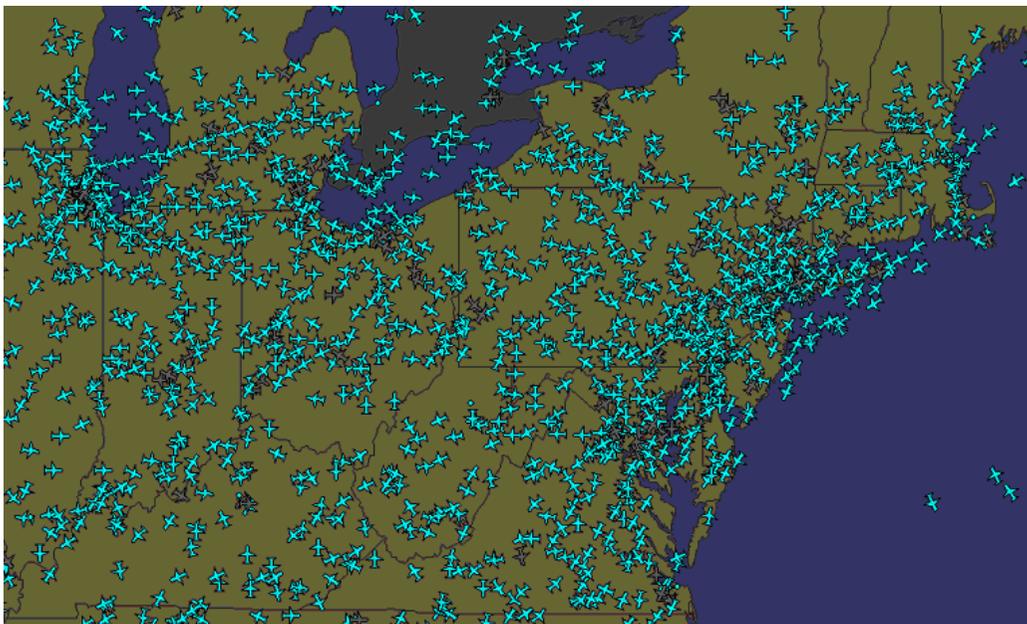

Figure 3 Snapshot of real air traffic in the Northeast USA on a typical day

In all the numerical illustrations, the true $P\{CS\}$ must be estimated, which is done for each procedure by counting the number of times the procedure successfully finds one of the true top-M designs out of 1,000,000 independent "macro" replications, and then dividing this number by 1,000,000. The choice of 1,000,000 macro replications leads to a standard error for the estimated $P\{CS\}$ of under 0.001 or 0.1%. The $P\{CS\}$ estimate for each procedure will serve as a measurement of its effectiveness for comparison purposes. In all of the examples, we have set $n_0 = 10$, $\Delta = 2$ and $q = 2$. More detailed information about the selection of these parameters in the procedure can be found in Chen et al. (1999).

Example 4.1

There are 30 design alternatives. Suppose $L(q_i, x) \sim N(i, 16^2)$, $i = 1, 2, .., 30$. We are satisfactory with any of the top-3 designs. It is obvious that designs 1, 2, and 3 are the actual top-3 designs. Note that the information about designs 1~3 being the good designs is used only in the calculation of the resulting $P\{CS\}$, but not used in any of the simulation procedures.

In the numerical experiment, we estimate the probability that b is 1, 2 or 3 after the simulation is performed with and without using OCBA. Different computing budgets are assigned and simulated. Figure 1 shows the test results with and without using OCBA. We see that in both cases we obtain a higher $P\{CS\}$ as the available computing budget
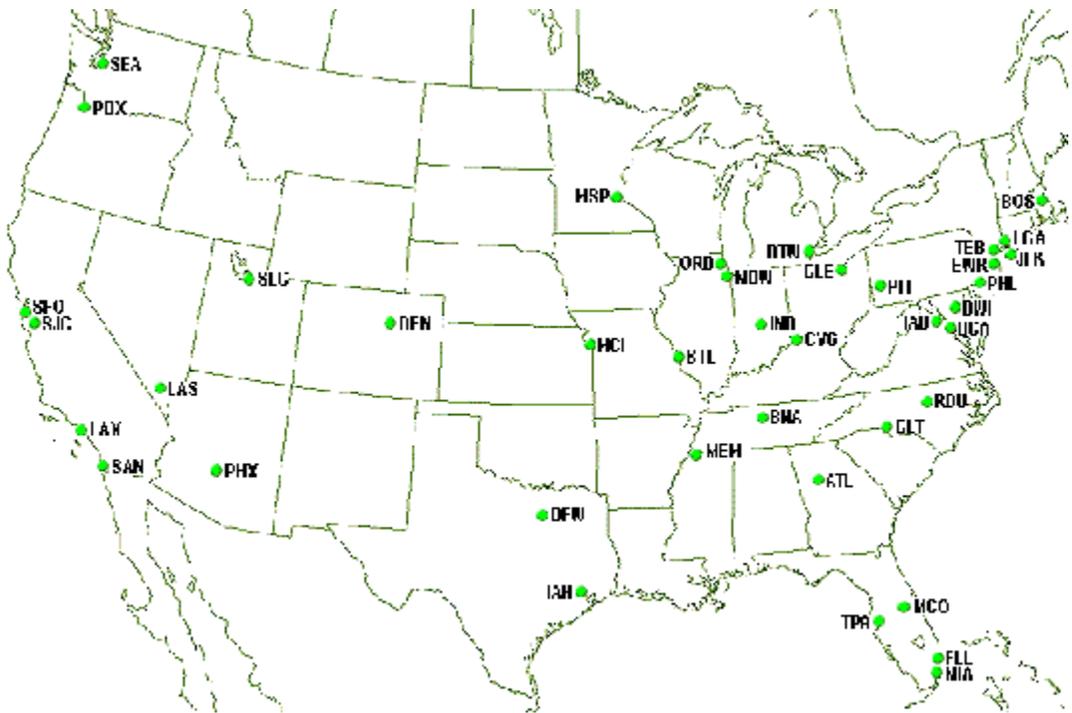

Figure 4 The airports included in our simulation model and their geographic locations

increases. However, OCBA achieves the same $P\{CS\}$ with a lower amount of computing budget. In particular, Figure 1 indicates the computation costs in order to attain $P\{CS\} = 99\%$ for both cases. In this setting, OCBA reduces the simulation time by 71%.

Example 4.2

This is a variant of example 4.1. To see the performance of the OCBA algorithm within a bigger design space, we increase the number of designs to 100. $L(\mathbf{q}_i, \mathbf{x}) \sim N(i, 16^2)$, $i = 1, 2, ..,$ 100. Figure 2 depicts the simulation results. The time savings factor of using OCBA is increased to 88% in this example. This is because a larger design space gives the OCBA algorithm more flexibility in allocating the computing budget.

5. Application to US Air Traffic System

In this section, we apply our OCBA algorithm to two design and management problems for the US air traffic system. Congestion in the US air transportation system is a major problem and is getting worse. Figure 3 gives a snapshot of the typical air traffic in the Northeast USA, which shows that the air traffic is very busy. This system is so complex that simulation is the only effective to evaluate such a system.

We have developed a simulation model for the US air traffic system. Our model includes major airports in US (see Figure 4). Some details about the simulation model can also be found at Le et al. (2004). To relieve the congestion in the air system, suppose the
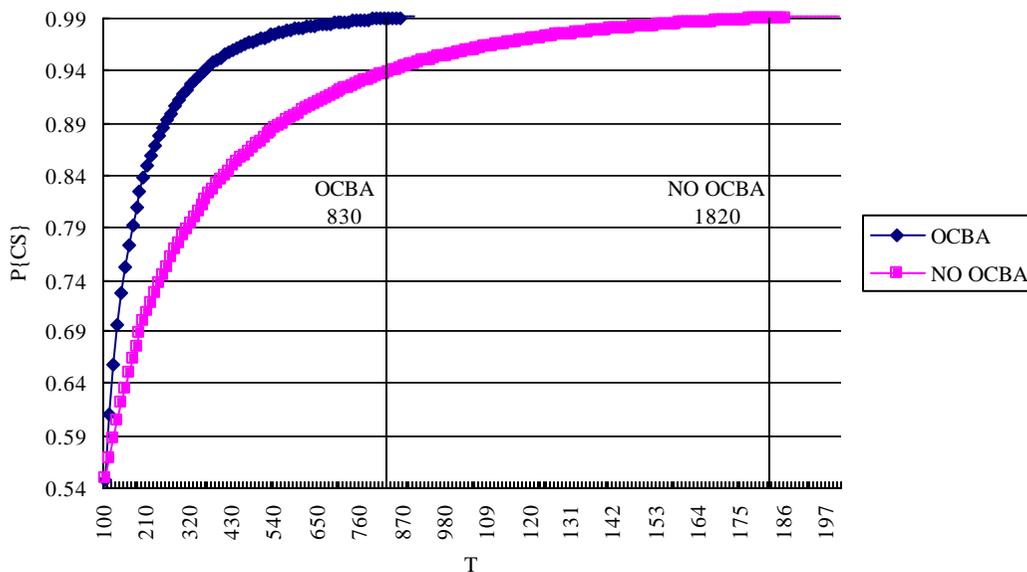

Figure 5 P{CS} vs. T with and without using OCBA for example 5.1. The computation costs in order to attain P{CS} = 99% are indicated.

decision maker is considering to increase the airport capacity by adding runways. Due to the limit of expenditure budget, however, a critical decision question is which airports we should add runways. In this paper, we consider two case studies and show how OCBA can reduce the simulation time for finding a good decision.

Example 5.1

We consider a design problem that we want to construct new runways in some most congested airports. The objective is to reduce flight delay. In particular, we consider to add runways to the following five congested airports: Atlanta (ATL), Chicago O'Hare (ORD), Dallas/Fort worth (DFW), Denver (DEN), and Washington Dulles (IAD) airports. Suppose we have a budget to construct a total

of two new runways among these five airports. The design problem is to determine which two airports will be added by one more runway. There are totally 10 alternative designs. Our goal here is to find the best design with minimum expected system flight delay if two runways are added. Thus $M = 1$. Figure 5 shows the test results with and without using OCBA. OCBA reduces the simulation time by 54.3%.

Example 5.2

We consider a bigger design problem by adding one more airport to our consideration list for adding runways. The extra airport is Minneapolis - St. Paul (MSP). In this case study, suppose that we have a budget to construct a total of three new runways among these six airports. The design question is to
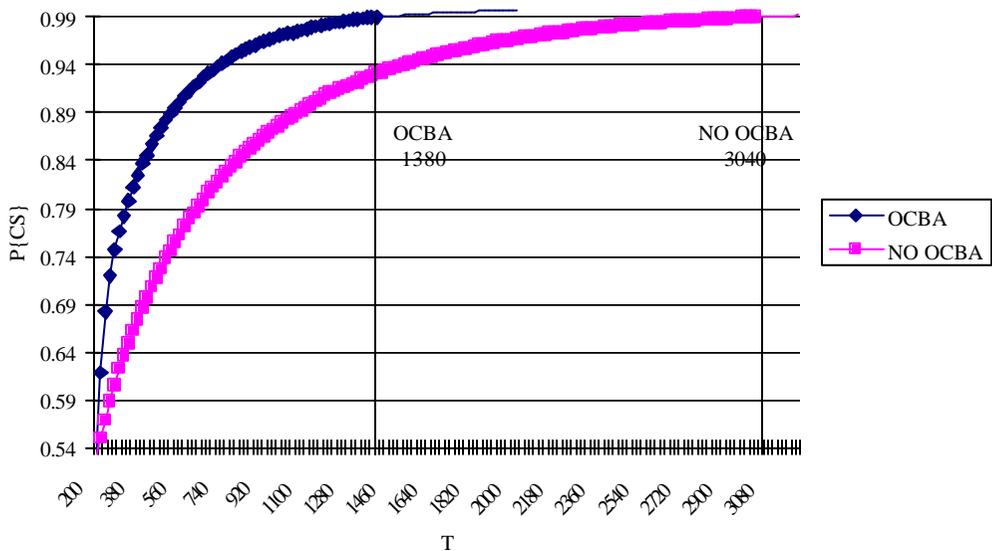

Figure 6 P{CS} vs. T with and without using OCBA for example 5.2. The computation costs in order to attain P{CS} = 99% are indicated.

select three airports, each of them will be added one more runway. There are totally 20 alternative designs. Again, the objective is to find the design with minimum expected system flight delay. Suppose we are satisfactory with either the best design or the second best design, i.e., $M = 2$. Figure 2 depicts the simulation results. The time savings factor of using OCBA is 54.6% in this example.

6. Conclusions

We present a new procedure to enhance the efficiency of simulation experiments for decision making under uncertainty. The objective is to efficiently identify one of the top-M designs. We also apply our efficient simulation approach to air traffic system design problems. Numerical results show that the total computation time can be dramatically reduced.

References

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

[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. Yü cesan, 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., S. D. Wu, and L. Dai, "Ordinal comparison of heuristic algorithms using stochastic optimization", *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 1, pp. 44-56, February 1999.

[5] Chen, C. H., E. Yü cesan, Y. Yuan, H. C. Chen and L. Dai, "Computing budget allocation for simulation experiments with different system structures", *Proceedings of the 1998 Winter Simulation Conference*, pp. 735-741, December 1998.

[6] Cheng, R. C. H. "The use of antithetic variates in computer simulations", *journal of operational research society*, 15, pp. 227-236, 1984.

[7] Chick, S. and K. Inoue. "New two-stage and sequential procedures for selecting the best simulated system", *Operations Research*, Vol. 49, pp. 1609–1624, 2001.

[8] DeGroot, M. H., *Optimal Statistical Decisions.* New York, NY: McGraw-Hill Book Co., 1970.

[9] Fishman, G. *Monte Carlo: Concepts, Algorithms, and Applications*, Springer-Verlag, 1996.

[10] Glasserman, P. and D. D. Yao, "some guidelines and guarantees for common random numbers", *Management Science*, Vol. 38, No. 6, pp. 884-908, 1992.

[11] Heidelberger, P. "Fast simulation of rare events in queueing and reliability models", *In Performance Evaluation of Computer and Communication Systems*, ed. L. Donatiello and R. Nelson, pp. 165-202, Springer Verlag, 1993.

[12] Hyden, P. and L. Schruben, "Improved decision processes through simultaneous simulation and time dilation", *Proceedings of the 2000 Winter Simulation Conference*, pp. 743-748, 2000.

[13] Le, L., G. L. Donohue, and C. H. Chen, "Using auction-based slot allocation for traffic demand management at hartsfield atlanta international airport: a case study", to appear in *Journal of the Transportation Research Board*, 2004.

[14] Lee, L. H., and E. P. Chew " A simulation study on sampling and selecting under fixed computing budget", *Proceedings of 2003 Winter Simulation Conference*, pp. 535-542, December 2003.

[15] Rinott, Y., "On two-stage selection procedures and related probability inequalities", *Communications in Statistics*, A7, 799-811, 1978.

[16] 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.

Chun-Hung Chen Dr. Chen received his Ph.D. degree in Engineering Sciences from Harvard University, Cambridge, MA, in 1994. Dr. Chen is an Associate Professor of Systems Engineering & Operations Research at George Mason University, Fairfax, VA. He was an Assistant Professor of Systems Engineering at the University of Pennsylvania, Philadelphia, PA, before joining GMU. His interests cover a wide range of areas in discrete event systems modeling and simulation, optimization, network management, and systems design under uncertainty. Sponsored by NSF, NASA, and FAA, Dr. Chen has recently worked on several critical issues, such as runway slots auction, in the analysis and management of the US air traffic network. Dr. Chen won the 1994 Harvard University Eliahu I. Jury Award for the Best Thesis in the field of Control. He is one of the recipients of the 1992 MasPar Parallel Computer Challenge Award. Dr. Chen is an IEEE senior member and is listed in Who'sWho in America, Who'sWho in Science and Engineering, and Who'sWho in Engineering Education.

Donghai He is a Ph.D. candidate at the Department of Systems Engineering & Operations Research at George Mason University. He received his BS degree in Control Engineering at Dalian University of Technology in China. His research interests include optimization, object-oriented modeling and Bayesian networks.