*Proceedings of the 2015 Winter Simulation Conference*
*L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.*

# PARTITION BASED OPTIMIZATION FOR UPDATING SAMPLE ALLOCATION STRATEGY USING LOOKAHEAD

David D. Linz
Hao Huang
Zelda B. Zabinsky

Department of Industrial and Systems Engineering
University of Washington
Seattle, WA 98195-2650 USA

## ABSTRACT

Simulation models typically describe complicated systems with no closed-form analytic expression. To optimize these complex models, general "black-box" optimization techniques must be used. To confront computational limitations, Optimal Computational Budget Allocation (*OCBA*) algorithms have been developed in order to arrive at the best solution relative to a finite amount of resources primarily for a finite design space. In this paper we extend the *OCBA* methodology for partition based random search on a continuous domain using a lookahead approximation on the probability of correct selection. The algorithm uses the approximation to determine the order of dimensional-search and a stopping criterion for each dimension. The numerical experiments indicate that the lookahead OCBA algorithm improves the allocation of computational budget on asymmetrical functions while preserving asymptotic performance of the general algorithm.

## 1 INTRODUCTION

As applications for simulation grow, there is an increased demand for efficient generalized algorithms that can be used to efficiently optimize black-box objective functions. Partition based search methods, such as branch and bound methods, nested partitions, and adaptive random search (Chen et al. 2014) (Chew et al. 2009) (Shi and Ólafsson 1998) (Tang 1994) have been used to optimize objective functions evaluated by black-box simulations.

Optimal Computing Budget Allocation (*OCBA*) algorithms have been applied to simulation optimization with a finite budget constraint. They optimize budget allocation based on maximizing the probability of correctly selecting the optimal design. An extensive discussion of OCBA methods can be found in Chen (2011). *OCBA* algorithms have commonly been applied to a finite set of designs. However, more recent developments have focused specifically on optimizing samples taken within a set of discrete domain categories. Starting with Chen et al. (1997), *OCBA* strategies were outlined for ranking and selection algorithms based on statistical estimates of a lower bound on the probability of correct selection. In Chen et al. (1997) the lower-bound depended on a Bayesian approximation and relied on the independence of the response between designs. An extension of this approach was created for optimal intelligent air traffic management in Chen and He (2005). A similar result was extended to ordinal optimization across different designs in Chen et al. (2000), which develops an allocation of samples to a discrete number of designs that is optimal asymptotically. Brantley et al. (2014) extend the *OCBA* approach from discrete designs to groups of designs by use of regression. More recently, Chen et al. (2014) extended the application of *OCBA* to partition based selection for black box functions on a continuous domain.

When optimizing the allocation of computational budget to a multidimensional space, Chen et al.'s algorithm sequentially optimizes the space one dimension at a time, with identical budget in an arbitrary

order. While this makes the problem tractable computationally, there is no guidance for selecting which dimension should be optimized first. Moreover, despite possible asymmetries in dimensional behavior, each dimension is given an equal computational budget. These assumptions could lead to a less efficient allocation of computational budget than an algorithm that could intelligently determine the order of dimensions to be evaluated and allocate the budget dedicated to each dimension individually.

This paper proposes a look-ahead algorithm that seeks to address these two issues. The algorithm employs a lookahead approximation on the probability of correct selection to determine the next dimension to be optimized as well as a stopping criterion for samples in that dimension.

This paper provides an overview of the *OCBA* approach to partition based search and details the derivation of the lookahead algorithm in Section 2. Here we detail a general strategy for multi-dimensional partition based search in Section 2.1. After, we the develop the formulation for the lookahead metric in Section 2.2 as well as its statistical approximation in Section 2.3. We describe how the lookahead is integrated into an *OCBA* algorithm in Section 2.4 and discuss the algorithm's performance relative to alternatives in 3. Result and future directions are discussed in Section 4.

## 2 LOOKAHEAD ALGORITHM FOR PARTITION BASED OPTIMIZATION

Consider a black-box optimization problem

$$min_{x\in\Theta} f(x) \tag{1}$$

where $\Theta \subset \mathbb{R}^D$ and $f$ is a function such that $f : \Theta \to \mathbb{R}$. We suppose $\Theta$ is a hyper-rectangle such that each dimension is divided into $M$ equal intervals so that the space is partitioned into $M^D$ boxes $\theta_\gamma^{(D)}$ indexed by a $D$ length array of tuples $\gamma = ((d_1, m_{d_1}), (d_2, m_{d_2}), ..., (d_D, m_{d_D}))$ where $d_1, ..., d_D$ is a permutation of the dimensions $\{1, ..., D\}$, and the values $m_{d_1}, ..., m_{d_D}$ range from 1 to $M$ and indicate a specific interval on a dimension. This defines a partition on $\Theta$ such that $\bigcup_{m_1=1}^{M} \cdots \bigcup_{m_D=1}^{M} \theta_\gamma = \Theta$. We consider different levels of aggregation of the $M^D$ boxes, starting with the smallest box of order $D$, $\theta_\gamma^{(D)}$, and aggregating up to the largest box of order 0, $\theta^{(0)} = \Theta$. Define a box of order $k$ as

$$\theta^{(k)}_{(d_1,m_{d_1}),...,(d_k,m_{d_k})} = \bigcup_{m_d=1}^{M} \theta^{(k+1)}_{(d_1,m_{d_1}),...,(d_k,m_{d_k})(d,m_d)}$$

where $d$ is one value in the set of remaining dimensions $\mathbb{D}_r^{(k)} = \{1, ..., D\}/\{d_1, d_2, ..., d_k\}$ and $k = 0, 1, ..., D-1$. For reference, $\theta^{(k)}_{(d_1,m_{d_1}),...,(d_{k-1},m_{d_{k-1}})(d_k,\cdot)}$ indicates the set of boxes $\{\theta^{(k)}_{(d_1,m_{d_1}),...,(d_{k-1},m_{d_{k-1}})(d_k,m)} \ \forall m = 1, ..., M\}$.

An example of a set of aggregated boxes is shown in Figure 1 over a three dimensional ($D = 3$) space with $M = 3$. Starting with the total space $\theta^{(0)} = \Theta$, Figure 1 illustrates a partition on the second dimension with the first order boxes $\theta^{(1)}_{(2,1)}$, $\theta^{(1)}_{(2,2)}$, $\theta^{(1)}_{(2,3)}$. Considering the shaded surface $\theta^{(1)}_{(2,3)}$, the partition of second order boxes consists of $\theta^{(2)}_{(2,3),(3,1)}$, $\theta^{(2)}_{(2,3),(3,2)}$ $\theta^{(2)}_{(2,3),(3,3)}$. Finally, the gray strip, $\theta^{(2)}_{(2,3),(3,3)}$, is divided into three third order boxes $\theta^{(3)}_{(2,3),(3,3),(1,1)}$, $\theta^{(3)}_{(2,3),(3,3)(1,2)}$, $\theta^{(3)}_{(2,3),(3,3),(1,3)}$ where the black box is $\theta^{(3)}_{(2,3),(3,3),(1,3)}$. From this pattern we can see that each $k$ order box can be divided into $M$ equal $k+1$ order boxes, for $k+1 \leq D$.

Let $\theta_{\gamma^*}^{(D)}$ be the smallest box that contains the optimum $x^*$ where $x^* = argmin_{x\in\Theta} f(x)$ and $\gamma^* = ((d_1, m^*_{d_1}), (d_2, m^*_{d_2}), ..., (d_D, m^*_{d_D}))$. Let $\theta_{\gamma^b}^{(D)}$ be the box statistically estimated to contain the optimum based on observed data and $\gamma^b = ((d_1, m^b_{d_1}), (d_2, m^b_{d_2}), ..., (d_D, m^b_{d_D}))$. For use in the algorithm, we denote the first $k$ elements of $\gamma^b$ as $\gamma^{(b,k)} = ((d_1, m^b_{d_1}), (d_2, m^b_{d_2}), ..., (d_k, m^b_{d_k}))$.

Define the Probability of Correct Selection ($PCS$) as $P(m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_D} = m^*_D)$ for any permutation $d_1, ..., d_D$ of $\{1, ..., D\}$. The Optimal Computational Budget Allocation ($OCBA$) problem can be defined as

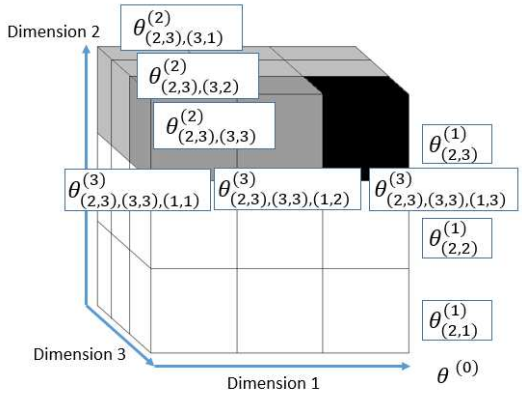


Figure 1: An example of boxes of different order in a three dimensional domain, with M=3.

follows. Let $N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}$ be the number of samples in box $\theta^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}$. Our goal is to select the allocation of samples to maximize the probability of correct selection relative to some budget $T$. The optimization problem therefore becomes:

$$\begin{aligned} &max_{N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}} \; PCS|N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})} \\ &\text{subject to} \quad \sum_{m_{d_1}=1}^{M} \cdots \sum_{m_{d_D}=1}^{M} N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})} = T, \text{ for any permutation } d_1,\ldots,d_D \text{ of } \{1,\ldots,D\} \end{aligned} \tag{2}$$

where $PCS|N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})} = P(m^b_{d_1} = m^*_{d_1},\ldots,m^b_{d_D} = m^*_{d_D}|N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})})$. However, the problem can be intractable if $M^D$ is large, since it requires optimizing over $M^D$ variables, $N^{(D)}_{(d_1,m_{d_1}),\ldots,(d_D,m_{d_D})}$. Our approach is to simplify the problem by optimizing the probability of selecting the correct interval on each dimension sequentially.

The proposed sequential optimization has a two-stage form. First, the algorithm computes an approximation of the probability of correct selection conditioned on which dimension is optimized first in the sequence. After selecting the dimension with the highest approximate PCS, the algorithm starts sampling points. The second stage is to decide whether to continue sampling along the current dimension, or to proceed to the next dimension. The decision is based on calculating the approximate probability of correct selection conditioned on the budget allocation on the current dimension. Either more points are sampled on the current dimension, or the algorithm stops sampling on the current dimension, determines the best interval on the current dimension, and proceeds to selecting the next dimension. The details of the full sequential optimization algorithm are described in Section 2.4.

### 2.1 Sequential Optimization Approach

One solution to reducing the size of problem (2) is to sequentially optimize the probabilities of correct interval selection on each dimension. In sequence, each probability can be conditioned on selecting previous dimensions' interval correctly,

$$P(m^b_{d_1} = m^*_{d_1}), \quad P(m^b_{d_2} = m^*_{d_2}|m^b_{d_1} = m^*_{d_1}), \quad \ldots \quad ,P(m^b_{d_D} = m^*_{d_D}|m^b_{d_1} = m^*_{d_1},\ldots,m^b_{d_{D-1}} = m^*_{d_{D-1}}). \tag{3}$$

Here each conditional probability $P(m^b_{d_k} = m^*_{d_k}|m^b_{d_1} = m^*_{d_1},\ldots,m^b_{d_{k-1}} = m^*_{d_{k-1}})$ can be estimated given the allocation of samples to be taken on the $M$ $k$th order boxes $\theta^{(k)}_{(d_1,m^b_{d_1}),\ldots,(d_{k-1},m^b_{d_{k-1}})(d_k,\cdot)}$. In addition an established

*OCBA* method can determine the sampling allocation on the $k$th order boxes to optimize $P(m^b_{d_k} = m^*_{d_k} | m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_{k-1}} = m^*_{d_{k-1}})$. Thus, our algorithm follows Chen et al. (2014) to determine the sample allocations $N^{(1)}_{(d_1,\cdot)}, N^{(2)}_{(d_1,m^b_{d_1}),(d_2,\cdot)}, \ldots, N^{(D)}_{(d_1,m^b_{d_1}),,...,(d_{D-1},m^b_{d_{D-1}})(d_D,\cdot)}$ that optimize the values of $P(m^*_{d_1} = m^b_{d_1} | N^{(1)}_{(d_1,\cdot)})$, $P(m^b_{d_2} = m^*_{d_2} | m^b_{d_1} = m^*_{d_1}, N^{(2)}_{(d_1,m^b_{d_1}),(d_2,\cdot)})$ ,..., $P(m^b_{d_D} = m^*_{d_D} | m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_{D-1}} = m^*_{d_{D-1}}, N^{(D)}_{(d_1,m^b_{d_1}),...,(d_{D-1},m^b_{d_{D-1}})(d_D,\cdot)})$ sequentially. The result is a series of $D$ optimization problems across $M$ variables corresponding to each dimension where the $k$th iteration determines the optimal $m^b_{d_k}$ which is used to condition the next $k+1$st order problem.

The total budget $T$ implies a constraint on the number of samples allocated to all $D$ problems. For our sequential approach we define $T_{d_k}$ to be the number of samples taken from the $k$th order boxes such that $\sum_{k=1}^{D} T_{d_k} = T$. For the $k$th problem, $T_{d_k}$ constrains the number of samples taken from the $k$th order boxes such that $\sum_{m=1}^{M} N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{D-1},m^b_{d_{k-1}})(d_k,m)} = T_{d_k}$. Therefore the $k$th order problem determining $N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{D-1},m^b_{d_{k-1}})(d_k,\cdot)}$ follows

$$\begin{aligned} &max_{N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{k-1},m^b_{d_{k-1}})(d_k,\cdot)}} P(m^b_{d_k} = m^*_{d_k} | m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_{k-1}} = m^*_{d_{k-1}}, N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{k-1},m^b_{d_{k-1}})(d_k,\cdot)}) \\ &\text{subject to } \sum_{m=1}^{M} N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{k-1},m^b_{d_{k-1}})(d_k,m)} = T_{d_k}. \end{aligned} \tag{4}$$

Before the $k$th order problem in (4) can be optimized, values for $T_{d_1}, ..., T_{d_D}$ as well as the order of dimension $d_1, ..., d_D$ need to be determined. Ideally an algorithm would choose values of $T_{d_1}, ..., T_{d_D}$ and $d_k$ to optimize the probability of selecting the correct intervals in each remaining dimension. We define the Remaining Probability of Correct Selection (*RPCS*) for the $k$th order problem as

$$\begin{aligned} &RPCS(k) \\ &= P\left(m^b_{d_k} = m^*_{d_k}, ..., m^b_{d_D} = m^*_{d_D} | d_k, ..., d_D, T_{d_k}, ...., T_{d_D}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, ..., N^{(D)}_{\gamma^{(b,D-1)}(d_D,\cdot)}\right). \end{aligned} \tag{5}$$

However, at a given $k$th order problem in the sequential optimization process, the value of $RPCS(k)$ depends on values of $d_{k+1}, ..., d_D$, $T_{d_k}, ...., T_{d_D}$ and $N^{(k+1)}_{(d_1,m^b_{d_1}),...,(d_{k+1},\cdot)}$ ,..., $N^{(D)}_{(d_1,m^b_{d_1}),...,(d_D,\cdot)}$ that cannot be determined with the current information.

The key idea for our approach is to approximate $RPCS(k)$ with a function that only depends on $d_k, T_{d_k}, ..., T_{d_D}$ and variables determined up to $k-1$. This function is called

$$APCS(k, d_k, T_{d_k}, ..., T_{d_D}). \tag{6}$$

We develop *APCS* by defining aggregations of $k+1$ order boxes and associated sample allocations over remaining dimensions based on current observations and *OCBA* methodology. The approximation is looking ahead to future probabilities of correctly selecting the intervals on each remaining dimension. This is an important approximation since it reduces the complexity of computing the *RPCS* from roughly $M^D$ to roughly $M \times D$.

We propose a general strategy that sequentially optimizes each dimension with lookahead based on *APCS* to determine values of $d_k$, $T_{d_k}$, and $N^{(k)}_{(d_1,m^b_{d_1}),...,(d_{k-1},m^b_{d_{k-1}})(d_k,\cdot)}$ for sampling on the $k$th iteration. The $k$th iteration consists of three steps. For a given iteration $k$ in the sequential optimization algorithm, the $k-1$ previous iterations will have determined $d_1$, ... , $d_{k-1}$ , $m^b_{d_1}$, ..., $m^b_{d_{k-1}}$ , $T_{d_1}, ..., T_{d_{k-1}}$. The $k$th iteration

of the sequential method for choosing sample allocations is

- Determine $d_k$, and $T_{d_k},...,T_{d_D}$ that maximizes $APCS(k,d_k,T_{d_k},...,T_{d_D})$ such that $\sum_{l=1}^{D} T_{d_l} = T$
- Determine sample allocations that maximize (4)
- Determine $m^b_{d_k}$ based on observed samples.

What remains is to develop an expression for $APCS(k,d_k,T_{d_k},...,T_{d_D})$, that can be easily computed given $d_k,T_{d_k},...,T_{d_D}$.

### 2.2 Developing an Approximate Lower Bound on RPCS

The primary difficulty with computing $RPCS(k)$ is that it depends on unspecified values of $N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$, $\ldots$, $N^{(D)}_{\gamma^{(b,D-1)}(d_D,\cdot)}$. Our approximation to *RPCS* is developed by looking ahead to the $k+1$ order boxes that are aggregated in different orientations. For example suppose the first iteration has determined $d_1 = 2$ and $m^b_{d_1} = 3$, selecting the shaded surface $\theta^{(1)}_{(2,3)}$ in Figure 1. The second iteration could choose $d_2 = 1$ or $d_2 = 3$ so that the set of remaining dimensions would be $\mathbb{D}^{(k)}_r = \{3\}$ or $\mathbb{D}^{(k)}_r = \{1\}$ respectively. The lookahead strategy develops future sampling allocations for these different possibilities. In the approximation we assume points are optimally allocated to the $k+1$ order boxes with *OCBA* methodology from Chen et al. (2014). Assuming an interval $m^b_{d_k}$, we then consider $D-k$ sets of $k+1$ order boxes by considering different possibilities for $d_{k+1}$. We therefore write this approximation to *RPCS* as

$$P\left(m^b_{d_k} = m^*_{d_k},...,m^b_{d_D} = m^*_{d_D} | d_k, T_{d_k}, T_{d_{k+1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)} \forall d_{k+1} \in \mathbb{D}^{(k)}_r\right). \tag{7}$$

To further simplify, we apply Bonferroni's inequality to achieve a lower bound on (7)

$$\begin{aligned} &\geq 1 - \sum_{d_{k+1} \in \mathbb{D}^{(k-1)}_r} \Big(1 - P(m^b_d = m^*_d | \\ &d_k, T_{d_k}, T_{d_{k+1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)} \forall d_{k+1} \in \mathbb{D}^{(k)}_r)\Big). \end{aligned} \tag{8}$$

A further lower bound is created by estimating $P(m^b_d = m^*_d)$ using observed samples in the $k+1$ order boxes where $d_{k+1} = d$. Separating the $k$ order term in (8), we get another lower bound for (8)

$$\begin{aligned} &\geq 1 - \left(1 - P(m^b_{d_k} = m^*_{d_k} | d_k, T_{d_k}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)})\right) \\ &- \sum_{d_{k+1} \in \mathbb{D}^{(k)}_r} \left(1 - \left(P(m^b_{d_{k+1}} = m^*_{d_{k+1}} | d_k, T_{d_{k+1}}, N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)})\right)\right). \end{aligned} \tag{9}$$

We next estimate the conditional probabilities in (9) using the *OCBA* methodology to determine $N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$ and $N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)} \forall d_{k+1} \in \mathbb{D}^{(k)}_r$.

### 2.3 Statistically Estimating Approximate Lower Bound on RPCS

Given observed samples taken up to the $k$th iteration, we next statistically estimate each term $P(m_{d_k}^b = m_{d_k}^*|d_k, T_{d_k}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)})$ in (9) and the objective function in (4).

Chen et al. (2014) applied extreme value theory to estimate the minimum in a box based on observed samples. The observed sample minimum on a box can be modeled by a three-parameter Weibull distribution, $Weibull(\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}, \beta^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}, \gamma^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})$. The lower threshold parameter, $\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ estimates the minimum function value in its corresponding box.

Let $\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ be the Maximum Likelihood Estimate for $\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$. If the shape parameter $\beta^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} > 2$, then $(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})^2$ is the variance of $\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$, otherwise $(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})^2$ is not able to be estimated.

Given $T_{d_k}, \hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ and $(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})^2$, the asymptotically optimal values for $N^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ can be determined by solving the following equations (see Chen et al. (2014))

$$\frac{N^{(k)}_{\gamma^{(b,k-1)}(d_k,i)}}{N^{(k)}_{\gamma^{(b,k-1)}(d_k,j)}} = \left( \frac{\frac{\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,i)}}{\delta^{(k)}_{\gamma^{(b,k-1)}(d_k,i)}}}{\frac{\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,j)}}{\delta^{(k)}_{\gamma^{(b,k-1)}(d_k,j)}}} \right)^2 \qquad i,j \neq m_d^b, \text{ where } \delta^{(k)}_{\gamma^{(b,k-1)}(d_k,i)} = \hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,i)} - \hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})} \tag{10}$$

$$N^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})} = \hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})} \cdot \sqrt{\sum_{i=1,i\neq m_d^b}^{M} \frac{(N^{(k)}_{\gamma^{(b,k-1)}(d_k,i)})^2}{(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,i)})^2}} \quad \text{such that } \sum_{m=1}^{M} N^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} = T_{d_k}.$$

Since the lower threshold of the minimum is normally distributed around the MLE estimate $\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ we have

$$\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} \sim Normal\left( \hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}, \frac{(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})^2}{N^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}} \right).$$

The normal distribution on $\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$ can be used to estimate the conditional probabilities in (4) and in (9) since $\bigcap_{m=1,m\neq m^b_{d_k}}^{M} \alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} > \alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})}$ is equivalent to the event $m^b_{d_k} = m^*_{d_k}$.

Chen et al. (2014) provides a good lower bound for the objective function in (4) that can be approximated with the estimated parameters, and applying Bonferroni's inequality yields

$$P\Big( \bigcap_{m=1,m\neq m^b_{d_k}}^{M} \alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} > \alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})} | m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_{k-1}} = m^*_{d_{k-1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)} \Big) \tag{11}$$

$$\geq 1 - \sum_{m=1}^{M} P(\alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m)} < \alpha^{(k)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})} | m^b_{d_1} = m^*_{d_1}, ..., m^b_{d_{k-1}} = m^*_{d_{k-1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}).$$

Therefore determining the estimates $\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$, $(\hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)})^2$ based on observed samples and $N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$ from (10) a lower bound for each term in (9) can be written as

$$P(m^b_{d_k} = m^*_{d_k} | d_k, T_{d_k}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}) \geq 1 - \sum_{m=1}^{M} P(\alpha^{(k)}_{\gamma^{(b,k-1)},(d,m)} < \alpha^{(k)}_{\gamma^{(b,k-1)},(d,m^b_d)} | d_k, T_{d_k}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}). \quad (12)$$

The inequality (12) can now be substituted in the expression (9) to develop our *APCS* function since the new term only depends on $d_k, T_d \in \mathbb{D}^{(k)}_r$. The following provides our zero-level *APCS* function,

$$\begin{aligned} APCS_0(d_k, T_{d_k}, ..., T_{d_D}) &= 1 - \sum_{m=1}^{M} P(\alpha^{(k)}_{\gamma^{(b,k-1)},(d_k,m)} < \alpha^{(k)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})} | T_{d_k}) \\ &- \sum_{d \in \mathbb{D}^{(k)}_r} \sum_{m=1}^{M} P(\alpha^{(k+1)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})(d,m)} < \alpha^{(k+1)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})(d,m^b_d)} | T_d). \end{aligned} \quad (13)$$

Alternatively, we have developed a slightly different *APCS* function derived from conditioning (7) on the correct selection $m^b_{d_k}$, yielding

$$\begin{aligned} &P\left( m^b_{d_k} = m^*_{d_k} | d_k, T_{d_k}, T_{d_{k+1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)} \forall d_{k+1} \in \mathbb{D}^{(k)}_r \right) \\ &\cdot P\left( m^b_{d_{k+1}} = m^*_{d_{k+1}}, ..., m^b_{d_D} = m^*_{d_D} | m^b_{d_k} = m^*_{d_k}, d_k, T_{d_k}, T_{d_{k+1}}, N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d_{k+1},\cdot)} \forall d_{k+1} \in \mathbb{D}^{(k)}_r \right). \end{aligned} \quad (14)$$

Following the same steps from Sections 2.2 and 2.3 we write an alternative expression for *APCS* called one-level *APCS*

$$\begin{aligned} APCS_1(d_k, T_{d_k}, ..., T_{d_D}) &= (1 - \sum_{m=1}^{M} P(\alpha^{(k)}_{\gamma^{(b,k-1)},(d_k,m)} < \alpha^{(k)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})} | T_{d_k}) \\ &\cdot ( \sum_{d \in \mathbb{D}^{(k)}_r} \sum_{m=1}^{M} P(\alpha^{(k+1)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})(d,m)} < \alpha^{(k+1)}_{\gamma^{(b,k-1)},(d_k,m^b_{d_k})(d,m^b_d)} | T_d)). \end{aligned} \quad (15)$$

Either $APCS_0$ or $APCS_1$ can provide an approximate lower bound for *RPCS* and can subsequently be used in the lookahead procedure.

### 2.4 Partition Based Lookahead Algorithm

The specification for *APCS* can be further incorporated into the sequential optimization strategy outlined in Section 2.1 providing a look-ahead algorithm. Because Chen et al. (2014) derived the optimal sample size (10) based on an asymptotic analysis, a one time determination of the total budget $T_{d_k}$ based on a single estimation could lead to a sub-optimal allocation of $N_{\gamma^{(b,k-1)}(d_k,\cdot)}$.

Hence, instead of performing the optimization of a dimension via one large sample, the algorithm instead proceeds iteratively by allocating a small pre-defined number of samples, $\Delta$, from the budget $T$ and determining the allocation of samples by solving (10) with $T_{d_k}$ replaced by $\Delta$. This process of sampling continues relative to dimension $d_k$ based on a stopping condition derived from the *APCS* function.

Let $T_r(k,i) = T - \sum_{l=1}^{k-1} T_{d_l} - i * \Delta$ be the remaining budget after $i \times \Delta$ samples have been taken along dimension $d_k$. Let $D_r(k) = \|\mathbb{D}^{(k)}_r\| = D - k$ be the number of remaining dimensions after $k$. Before the algorithm iteratively allocates $\Delta$ samples to the $k$th problem, we determine whether it is preferable to continue allocating an additional portion of the budget to dimension $d_k$ or to save the remaining budget for the remaining dimensions. Here we propose comparing *APCS* under two different scenarios.

The "leave" scenario allocates no additional budget to the current dimension $d_k$ and allocates the remaining budget equally to remaining dimensions. The "stay" scenario $APCS_{stay}$ allocates equal portions of the remaining budget among the current dimension $d_k$ and remaining dimensions

$$APCS_{leave} = APCS\left(d_k, T_{d_k} = 0, T_{d_{k+1}} = \frac{T_r(k,i)}{D_r(k)}, ..., T_{d_D} = \frac{T_r(k,i)}{D_r(k)}\right)$$

$$APCS_{stay} = APCS\left(d_k, T_{d_k} = \frac{T_r(k,i)}{D_r(k-1)}, T_{d_{k+1}} = \frac{T_r(k,i)}{D_r(k-1)}, ..., T_{d_D} = \frac{T_r(k,i)}{D_r(k-1)}\right).$$

If $APCS_{leave} \leq APCS_{stay}$ we allocate another $\Delta$ samples along dimension $d_k$. If $APCS_{leave} > APCS_{stay}$ we stop the $k$th iteration and proceed to the $k+1$st iteration.

Moving on the $k+1$st iteration we decide the dimension for $d_{k+1}$ by comparing $APCS$ for all remaining dimensions, that is

$$APCS\left(d_{k+1} = d, T_{d_{k+1}} = \frac{T_r(k+1,i)}{D_r(k)}, T_{d_{k+2}} = \frac{T_r(k+1,i)}{D_r(k)}, ..., T_{d_D} = \frac{T_r(k+1,i)}{D_r(k)}\right).$$

for $\forall D \in \mathbb{D}_r^k$. The value $d$ that maximizes the $APCS$ is set to $d_{k+1}$

Thus the complete algorithm for sequentially allocating samples to optimize the probability of correct interval selection can be written as the following.

**PARTITION BASED LOOKAHEAD ALGORITHM**

*INPUT :* Domain $\Theta$, function $f$, partition size $M$, budget $T$, initial sample size $N_0$ and increment size $\Delta$.

1. *INITIALIZE:*
   Uniformly sample $N_0$ independent points throughout the domain $\Theta$. Set $k = 1$, $T_r = T$, $\gamma^{(b,0)} = \emptyset$.
2. *DIMENSION DETERMINATION:*
   Compute
   $$\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d'_k,\cdot)}, \hat{s}^{(k)}_{\gamma^{(b,k-1)}(d'_k,\cdot)} \text{ for each } d'_k \in \mathbb{D}_r^{(k-1)}$$
   and
   $$\hat{\alpha}^{(k+1)}_{\gamma^{(b,k-1)}(d'_k,m^b_{d'_k})(d,\cdot)}, \hat{s}^{(k+1)}_{\gamma^{(b,k-1)}(d'_k,m^b_{d'_k})(d,\cdot)} \text{ for each } d'_k \in \mathbb{D}_r^{(k-1)} \text{ and } d \in \mathbb{D}_r^{(k-1)} \setminus d'_k$$
   by *MLE* on points sampled from the corresponding boxes.
   Based on (10) determine
   $$N^{(k)}_{\gamma^{(b,k-1)}(d'_k,\cdot)} \text{ such that } T_{d'_k} = \frac{T_r}{D_r(k-1)} \text{ for each } d'_k \in \mathbb{D}_r^{(k-1)}$$
   and
   $$N^{(k+1)}_{\gamma^{(b,k-1)}(d'_k,m^b_{d'_k})(d,\cdot)} \text{ such that } T_d = \frac{T_r}{D_r(k-1)} \text{ for all } d \in \mathbb{D}_r^{(k-1)} \setminus d'_k.$$
   Calculate $APCS(d_k = d'_k, T_{d_k} = \frac{T_r}{D_r(k-1)}, T_{d_{k+1}} = \frac{T_r}{D_r(k-1)}, ..., T_{d_D} = \frac{T_r}{D_r(k-1)}) \forall d'_k \in (D)_r^{(k-1)}$.
   Set $d_k$ to $d'_k$ that maximizes the above $APCS$.
3. *SAMPLE $\Delta$ POINTS:*
   Determine $N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$ from (10) with the variable $T_{d_k}$ set to $\Delta$. *SAMPLE* the new points based on $N^{(k)}_{\gamma^{(b,k-1)}(d'_k,\cdot)}$.
4. *CHECK STOPPING CONDITION:*
   Compute

$\hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}, \hat{s}^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$

and

$\hat{\alpha}^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d,\cdot)}, \hat{s}^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d,\cdot)}$ for each $d \in \mathbb{D}^{(k)}_r$

by *MLE* on points sampled from the corresponding boxes.
Based on (10) determine

$N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$ such that $T_{d_k} = \frac{T_r}{D_r(k-1)}$

and

$N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d,\cdot)}$ such that $T_d = \frac{T_r}{D_r(k-1)}$.

Compute $APCS_{stay} = APCS(d_k, T_{d_k} = \frac{T_r}{D_r(k-1)}, T_{d_{k+1}} = \frac{T_r}{D_r(k-1)}, ..., T_{d_D} = \frac{T_r}{D_r(k-1)})$ for each $d \in \mathbb{D}^{(k)}_r$.
Again, based on (10) determine

$N^{(k)}_{\gamma^{(b,k-1)}(d_k,\cdot)}$ such that $T_{d_k} = 0$

and

$N^{(k+1)}_{\gamma^{(b,k-1)}(d_k,m^b_{d_k})(d,\cdot)}$ such that $T_d = \frac{T_r}{D_r(k)}$ for each $d \in \mathbb{D}^{(k)}_r$ .

Compute $APCS_{leave} = APCS(d_k = d, T_{d_k} = 0, T_{d_{k+1}} = \frac{T_r}{D_r(k)}, ..., T_{d_D} = \frac{T_r}{D_r(k)})$. If $APCS_{leave} > APCS_{stay}$ go to *STEP 5* (*DETERMINE INTERVAL*) otherwise return to *STEP 3* (*SAMPLE $\Delta$ POINTS*) with $T_r = T_r - \Delta$.

5. *DETERMINE INTERVAL :*
Set $m^b_{d_k}$ to the $\arg\min_{m=1,...,M} \hat{\alpha}^{(k)}_{\gamma^{(b,k-1)}(d_k,m)}$. Add $(d_k, m^b_{d_k})$ to the vector $\gamma^{(b,k-1)}$ and set $k = k+1$. If there are no remaining dimensions to optimize or $T_r = 0$ then we *TERMINATE* the algorithm, otherwise return to *STEP 2* (*DIMENSION DETERMINATION*).

## 3 NUMERICAL RESULTS

In the following section we apply the lookahead algorithm described in Section 2 to a variety of numerical problems. We use the following test functions to evaluate the performance of the algorithm and compare with equally allocating budgets through dimensions from Chen et al. (2014).

- $H_1(x)$ is a function with two symmetric dimensions where $x_1, x_2 \in [1,7]$ with optimal solution $x^* = (2,2)$.

$$H_1(x) = \sum_{i=1}^{2} \left( -\frac{16}{\sqrt{2\pi}} e^{-(x_i-5)^2/2} - \frac{20}{\sqrt{2\pi}} e^{-200(x_i-2)^2} \right)$$

- $H_2(x)$ is a dimensional asymmetric function combining three negative normal density functions with different variances, where $x_1, x_2, x_3 \in [-3,3]$ with optimal solution $x^* = (0,0,0)$.

$$H_2(x) = -\sum_{i=1}^{3} \frac{1}{\sqrt{4i\pi}} e^{-\frac{x_i^2}{4i}}$$

- $H_3(x)$ is a dimensional asymmetric function combining one dimension of $H_1$ and a function $g(x)$ mixing a step function and a negative normal density function with different variances, where

$x_1, x_2, x_3 \in [-3,3]$ with optimal solution $x^* = (-2,0,0)$.

$$H_3(x) = -\frac{16}{\sqrt{2\pi}} e^{-(x_1-9)^2/2} - \frac{20}{\sqrt{2\pi}} e^{-200(x_1-6)^2} - \sum_{i=2}^{3} g(x_i),$$

where

$$g(x_i) = \begin{cases} \frac{\sqrt{i}}{\sqrt{2\pi}} e^{-\frac{ix_i^2}{2}}, \text{ if } x > 1 \text{ or } x < -1 \\ -2, \text{ otherwise.} \end{cases}$$

For test function $H_1$, $N_0$ and $\Delta$ are set to 200 and 50 respectively. Test functions $H_2$ and $H_3$ are conducted with $N_0 = 1000$ and $\Delta = 200$. All test functions are tested with $M = 3$. Therefore, the $\theta^*$ of $H_1$ is $\theta^{(2)}_{(1,1),(2,1)}$. For $H_2$ and $H_3$, the $\theta^*$ are $\theta^{(3)}_{(1,2),(2,2),(3,2)}$ and $\theta^{(3)}_{(1,1),(2,2),(3,2)}$ respectively.

Three algorithms are compared in this section, the algorithm by Chen et al. (2014), the proposed basic zero-level lookahead algorithm ($APCS_0$), and a modified lookahead algorithm one-level ($APCS_1$).

In order to evaluate the performance of algorithms, for each total budget $T$, each method is run for 100 replications to obtain an observed ratio of correctly selecting all intervals. Explicitly, this value is defined as number of correctly selected replications divided by 100.

Figure 2 (A) graphs the ratio of correctly selected replications versus total budget on test function $H_1$ for the three algorithms. The results graphed in Figure 2 (A) show that there is no significant difference between the three algorithms. The primary reason could be the dimensional symmetric nature of $H_1$. For a dimensional symmetric function, the order of dimension to be determined is trivial, which means the lookahead does not provide any advantage. Furthermore, as dimensions are identical, the lookahead's stopping condition will not provide more advantage than equally allocating budgets through dimensions.

Second, we test the three algorithms with the dimensionally asymmetric function $H_2$ in three dimensions. We test $H_2$ with fewer budgets than for $H_1$ because of the simple structure of $H_2$. The different variances lead to different difficulty for each dimension. Therefore, the zero-level lookahead algorithm is promising compared to Chen et al's algorithm in Figure 2 (B). However, one-level lookahead performs worse than the zero-level possibly because the stopping condition is too greedy.

Third, the test function $H_3$ is not only asymmetric in dimension but also has reduced variance along one dimension. In this case, the lookahead algorithms perform well as shown in Figure 2 (C). For this type of problem, the lookahead algorithm is able to identify the correct interval with low variance on the "easy" dimension first and reduce the variance of the difficult dimensions subsequently.

## 4 DISCUSSION

This paper has developed a lookahead approach to intelligently determine the order and the dimensional stopping condition for optimizing the computational budgets on a multi-dimensional black-box simulation problem. The paper examines a strategy for sequentially optimizing each dimensional interval relative to a budget. We derive a lower bound, *APCS*, for approximating the probability of correctly selecting intervals in each remaining dimension. Based on APCS, we employ a lookahead algorithm to determine the optimal order of dimensions and the optimal stopping time for maximizing the probability of correct interval selection.

Based on initial tests, the addition of a lookahead to determine order of dimensional optimization and stopping time provides some improvement on Chen et al.'s original algorithm. The numerical results show that the lookahead algorithm generates a larger number of correct selections over 100 replications for several asymmetric functions. However, for dimensionally symmetric functions, the lookahead algorithm performs similarly to the original algorithm featured in Chen et al. (2014).

We speculate that the lookahead provides more benefit when the function is asymmetrical in dimension and there are large differences in variability within dimensions. In these situations, correctly determining the interval on some dimensions first could reduce the variance on subsequent iterations.

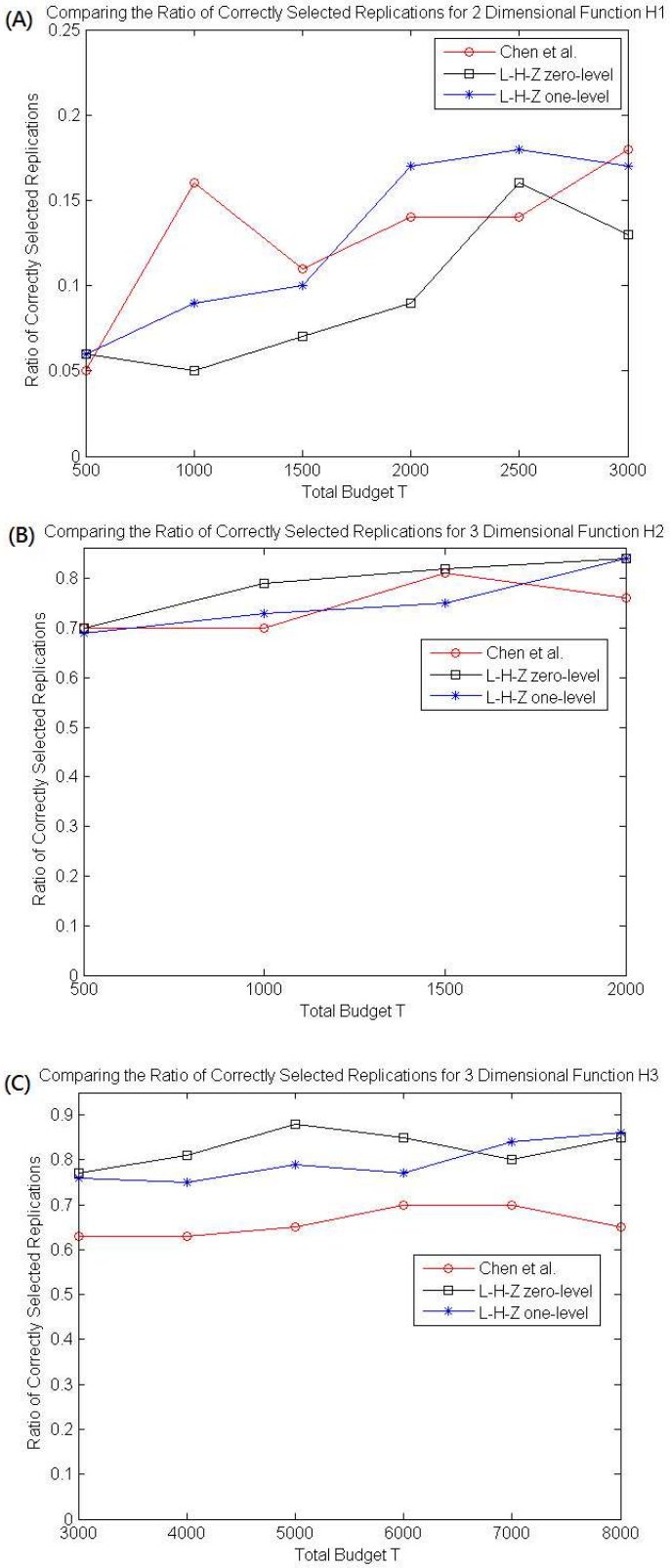


Figure 2: Comparing the fraction of correctly selected replications across three algorithms.

The performance of the algorithm needs to be tested on higher dimensional problems to determine the extent of the benefit from the lookahead approach. Moreover, the algorithm should be tested against a larger range of interval sizes $M$, it is very possible that the lookahead could provide better results under these circumstances.

## ACKNOWLEDGMENTS

This work has been funded in part by NSF grant CMMI-1235484.

## References

Brantley, M. W., L. H. Lee, C. H. Chen, and J. Xu. 2014. "An Efficient Simulation Budget Allocation Method Incorporating Regression for Partitioned Domains". *Automatica* 50 (5): 1391–1400.

Chen, C. H. 2011. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*.

Chen, C. H., and D. He. 2005. "Intelligent Simulation for Alternatives Comparison and Application to Air Traffic Management". *Journal of Systems Science and Systems Engineering* 14:37–51.

Chen, C. H., J. Lin, E. Yücesan, and S. E. Chick. 2000. "Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization". *Discrete Event Dynamic Systems: Theory and Applications* 10:251–270.

Chen, H. C., L. Dai, C. H. Chen, and E. Yucesan. 1997. "New Development Of Optimal Computing Budget Allocation For Discrete Event Simulation". *Winter Simulation Conference Proceedings,*.

Chen, W., S. Gao, C. H. Chen, and L. Shi. 2014. "An Optimal Sample Allocation Strategy for Partition-based Random Search". *IEEE Transactions on Automation Science and Engineering* 11 (1): 177–186.

Chew, E. P., H. L. Loo, S. Teng, and H. K. Choon. 2009. "Differentiated Service Inventory Optimization Using Nested Partitions and MOCBA". *Computers and Operations Research* 36:1703–1710.

Shi, L., and S. Ólafsson. 1998. "Nested Partitions Method For Global Optimization". *Operations Research* (December 2013).

Tang, Z. B. 1994. "Adaptive Partitioned Random Search to Global Optimization". *IEEE Transactions on Automatic Control* 39 (11): 2235–2244.

## AUTHOR BIOGRAPHIES

**DAVID D. LINZ** is a PhD student in the Department of Industrial and Systens Engineering at the University of Washington. His research interests include stochastic optimization and simulation optimization with applications to healthcare strategy. ddlinz@uw.edu.

**HAO H. HUANG** is a PhD candidate in the Department of Industrial and Systems Engineering at the University of Washington. His research interests include simulation optimization and healthcare applications. His e-mail is haoh7493@uw.edu.

**ZELDA B. ZABINSKY** is a Professor in the Department of Industrial and Systems Engineering at the University of Washington, with adjunct appointments in the departments of Electrical Engineering, Mechanical Engineering, and Civil and Environmental Engineering. She is an IIE Fellow. She has published numerous papers in the areas of global optimization, algorithm complexity, and optimal design of composite structures, and a book, *Stochastic Adaptive Search in Global Optimization*. She received an Erskine Fellowship from the University of Canterbury, Christchurch, New Zealand to collaborate internationally on global optimization methods. Professor Zabinsky's research interests are in global optimization under uncertainty for complex systems. Her email address is zelda@u.washington.edu.