Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

EFFICIENT COMPUTING BUDGET ALLOCATION FOR A SINGLE DESIGN BY USING REGRESSION WITH SEQUENTIAL SAMPLING CONSTRAINT

Xiang Hu
Loo Hay Lee
Ek Peng Chew

Dept. of Industrial and Systems Engineering
National University of Singapore
10 Kent Ridge Crescent, 119260,
SINGAPORE

Douglas J. Morrice

Red McCombs School of Business
The University of Texas at Austin
1 University Station, B6500
Austin, TX 78712,USA

Chun-Hung Chen

Dept. of Systems Engineering and Operations Research
George Mason University
4400 University Drive, MS 4A6
Fairfax, Virginia 22030, USA

ABSTRACT

In this paper, we develop an efficient computing budget allocation rule to run simulation for a single design whose transient mean performance follows a certain underlying function, which enables us to obtain more accurate estimation of design performance by doing regression. The sequential sampling constraint is imposed so as to fully utilize the information along the simulation replication. We formulate this problem as a c-optimal design problem based on some common assumptions in the field of simulation. Solutions are generated for some simple polynomial, logarithmic, and sinusoidal functions. Based on the numerical solutions, we develop the Single Design Budget Allocation (SDBA) Procedure that determines the number of simulation replications we need to run, as well as their run lengths, given a certain computing budget. Numerical experimentation confirms the efficiency of the procedure.

1 INTRODUCTION

As is often the case, the mean performance of the design might not be constant, but a function of some variables that can be simulation time or simulation run length. A common practice to estimate the transient mean performance of the design and its variance is to run the simulation up to the point where we want to make a prediction which is called the point of interest in this paper, and calculate the sample mean and sample variance by using the simulation outputs collected at that point. Another more sophisticated way is to use the regression approach which would make use of all information along the simulation replication instead of only at the point of interest. The latter is expected to provide us with more accurate estimation since more information is used.

Analysis of transient behavior is an important simulation problem in, for example, the initial transient problem (Law and Kelton 2000, Section 9.5.1) and sensitivity analysis (Morrice and Schruben 2001). Kelton and Law (1983) develop a regression-based procedure for the initial transient problem. Transient analysis is also important in so-called "terminating simulations" (Law and Kelton 2000, Section 9.4) that have finite terminating conditions and never achieve steady state. Examples are found in many service systems like hospitals or retail stores that have closing times or clearly defined "rush hour" patterns. They are also found in new product development competitions where

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

multiple different prototypes are being simulated simultaneously and the one that is able to achieve the best specifications (e.g., based on performance, quality, safety, etc.) after a certain amount of development time is selected. The latter is an example of gap analysis which is found in many other applications such as recovery to regular operations after a supply chain disruption and optimality gap analysis of heuristics for stochastic optimization (see, for example, Tanrisever, Morrice, and Morton 2012).

Morrice, Brantley, and Chen (2008) derive the formula to calculate the mean performance of the design when its transient mean performance follows a linear function, with the simulation outputs collected at the two supporting points. They further generalize this result to the problem when the underlying function is a polynomial of up to order five and the sequential sampling constraint is imposed so that information is collected at all observation points along the simulation replication which is run up to the point of interest (Morrice, Brantley, and Chen 2009). They show that significant variance reduction can be achieved by using this regression approach, which we refer to as the Simple Regression Procedure in this paper.

In this study, we aim at further improving the Simple Regression Procedure by running simulation replications to certain run lengths instead of running all of them to the point of interest. We assume that the transient mean performance of the single design follows a certain underlying function which can be expressed as a sum of several univariate one-to-one feature functions. Sequential multiple simulation output collection is conducted at all observation points along the simulation replication. We assume that the starting points of all simulation replications are fixed at a common point due to practical constraints. For example, in an M/M/1 queuing system, in order to estimate the 100^{th} customer's waiting time, we need to simulate from the very first customer. We further assume that the simulation budget needed to run the simulation from one observation point to the next is constant over the simulation replication and is equal to one unit of simulation budget. As a result, the run length of the simulation replication is equivalent to the number of observation points along the simulation replication, and the total computing budget can be considered as the total number of simulation outputs we collect. Therefore, based on the aforementioned constraints and assumptions, our problem becomes the problem of determining the simulation run lengths for all simulation replications, in order to better estimate the mean performance of the design at the point of interest by doing regression, subject to limited simulation computing budget.

The remainder of this paper will be structured in the following manner. Section 2 presents how we model the problem as an optimization problem by using a Bayesian regression framework. Section 3 presents the numerical solutions and the SDBA Procedure developed based on these numerical solutions. Section 4 gives the numerical experimentation to test the efficiency of the SDBA Procedure and gives an example of how we could handle heteroscedasticity in the application of the SDBA Procedure on a more realistic problem. Section 5 concludes the paper.

2 PROBLEM FORMULATION

2.1 Problem Setting

In our problem, we would like to better estimate the expected mean performance of the design at the point of interest x_M, given a total computing budget T. The transient mean performance of the design is assumed to follow a certain underlying function which is defined as $y(x) = \sum_{i=1}^{n} \beta_i \Phi_i(x)$, where $y(x)$ denotes the expected performance of the design at observation point x. The function $\Phi_i(x)$ is a one dimensional one-to-one feature function, which can be any continuous function. As a common practice in the literature of regression analysis, we assume the first feature function to be a constant function, i.e. $\Phi_1(x) = 1$. Let n be the total number of feature functions comprising the underlying function and $\boldsymbol{\beta} = [\beta_1, \beta_2 \dots, \beta_n]^T$ represent the unknown parameter vector which we want to estimate, whose posterior distribution can be determined by running the simulation.

The transient mean performance of the design can be obtained by running the simulation, and the relationship between the simulation output and the expected mean performance is defined as

$$\boldsymbol{F} = \boldsymbol{Y} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{F} = [f(x_1), f(x_2), \dots, f(x_m)]^T$ is the vector of simulation outputs and $f(x_i)$ is the simulation output at observation point x_i. The vector $\boldsymbol{Y} = [y(x_1), y(x_2), \dots, y(x_m)]^T$ is the expected mean performance of the design and $y(x_i)$ is the expected mean performance of the design at observation point x_i. Finally, $\boldsymbol{\varepsilon} = [\varepsilon(x_1), \varepsilon(x_2), \dots, \varepsilon(x_m)]^T$ is the vector of simulation noise which follows a multivariate normal distribution $\mathcal{N}(0, \Sigma)$, where Σ is the variance-covariance matrix. In this paper, we assume that the simulation outputs are uncorrelated and therefore Σ is diagonal. If the data generated by the simulation do not follow a normal distribution, then one can always perform macro-replications as suggested by Goldsman, Nelson, and Schmeiser (1991).

We denote the posterior distribution of the unknown parameter vector as $\widetilde{\boldsymbol{\beta}}$ and the posterior distribution of the design performance at observation point x as $\tilde{y}(x)$. A good estimation of the mean performance of design at the point of interest x_M implies a small estimated variance at x_M. Therefore, the problem of efficiently allocating computing budget for a single design is equivalent to minimizing $Var(\tilde{y}(x_M))$, which is the estimated variance of the design performance at x_M. Hence our problem is actually to find out the optimal number of simulation replications we need, as well as to determine their run lengths, in order to minimize $Var(\tilde{y}(x_M))$. This also implies that we are dealing with a c-optimal design problem for which only limited results are available in the literature of DOE (Atkinson, Donev, and Tobias 2007).

We assume that the total computing budget T is allocated to K simulation groups $G_i, i = 1,2, \dots, K$, and each of the simulation groups contains N_i simulation replications that have the same simulation run length l_i. For a simulation replication of run length l_i, we have l_i observation points, namely from observation point 1 to observation point l_i, and the simulation outputs are collected at all these points. Based on the above problem setting, we can formulate our computing budget allocation problem in the following form.

$$
\begin{aligned}
\boldsymbol{obj.} \quad & \min_{K} \min_{l_1, l_2, \dots, l_K, N_1, N_2, \dots, N_K} Var(\tilde{y}(x_M)) \\
\boldsymbol{s.t.} \quad & \sum_{i=1}^{K} l_i N_i \le T \\
& 0 < l_i \le T; && l_i \in \mathbb{N}^0, \forall i = 1,2, \dots, K \\
& N_i > 0; && N_i \in \mathbb{N}^0, \forall i = 1,2, \dots, K \\
& K > 0 && K \in \mathbb{N}^0.
\end{aligned}
$$

2.2 Bayesian Regression Framework

Let $\boldsymbol{F_i^j} = \left[f_i^{\,j}(x_1), f_i^{\,j}(x_2), \dots, f_i^{\,j}(x_{l_i})\right]^T; i = 1,2, \dots, K, j = 1, \dots, N_i$ be the simulation output vector of the j^{th} simulation replication in group G_i. Let $\boldsymbol{X_i} = \left[X_1, X_2, \dots, X_{l_i}\right]^T$ denote the $l_i \times n$ matrix of feature functions for the simulation replications of run length l_i, where X_i is a $1 \times n$ vector of feature functions at observation point x_i, and is expressed as $X_i = [\Phi_1(x_i), \Phi_2(x_i), \dots, \Phi_n(x_i)]$.

We assume that the vector $\boldsymbol{F_i^j}$ follows a multivariate normal distribution with mean $\boldsymbol{X_i\beta}$ and variance-covariance matrix Σ_i. Based on these assumptions, the posterior distribution of $\boldsymbol{\beta}$ can be calculated and is given as follows (DeGroot 2004; Gill 2008):

$$
\widetilde{\boldsymbol{\beta}} \sim N\left[\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i} \boldsymbol{X_i^T W_i F_i^j}\right), \left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1}\right]
$$

In the above expression, W_i is a $l_i \times l_i$ diagonal weight matrix with its diagonal elements being the inverse of the simulation noise variance at various observation points. Since $\tilde{y}(x_M)$ is a linear combination of $\widetilde{\boldsymbol{\beta}}$, due to the property of Gaussian distribution, $\tilde{y}(x_M)$ is also normally distributed:

$$\tilde{y}(x_M) \sim N\left[X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i} \boldsymbol{X_i^T W_i F_i^j}\right), X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1} X_M\right] \quad (1)$$

Therefore the objective function can be expressed as

$$\min_{K} \min_{l_1,l_2,\ldots,l_K,N_1,N_2,\ldots,N_K} X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1} X_M \quad (2)$$

We note that the estimated variance depends on the variance of the simulation noise at each observation point. In fact, the above expression is derived by minimizing the weighted squared errors, with W_i being the weight matrix. In order to simplify the problem, we consider a special case in which the simulation noise is homogeneous, meaning that the simulation noise at all observation points follow the same normal distribution with mean zero and variance σ_E^2. In practice, this homogeneous noise variance is calculated as the unbiased estimator of the performance variance of the design. Based on this homogeneous simulation noise assumption, the posterior distribution of the design performance can be written as

$$\tilde{y}(x_M) \sim N\left[X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i} \boldsymbol{X_i^T F_i^j}\right), \sigma_E^2 X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1} X_M\right] \quad (3)$$

It is noted that expression (3) is actually the least squares formula for calculating the design mean performance and variance when the squared residuals are minimized. Hence, the objective function is simplified to $Var(\tilde{y}(x_M)) = \sigma_E^2 X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1} X_M$. Because σ_E^2 is a constant, minimizing $Var(f(x_M))$ is equivalent of minimizing $X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1} X_M$, which is called the Prediction Variance Factor (PVF), to be consistent with the notation and definition presented by Morrice, Brantley and Chen (2009). Therefore, the weighted least squares problem is simplified into a least squares problem, and it is reformulated as

$$\begin{aligned}
&\boldsymbol{obj.} && \min_{K} \min_{l_1,l_2,\ldots,l_K,N_1,N_2,\ldots,N_K} X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1} X_M \\
&\boldsymbol{s.t.} && \sum_{i=1}^{K} l_i N_i \le T \\
& && 0 < l_i \le T; \qquad l_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K \\
& && 0 < N_i; \qquad N_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K \\
& && K > 0 \qquad K \in \mathbb{N}^0.
\end{aligned}$$

The challenges of solving the above problem include the following. The objective function is nonlinear and can be very complex depending on the feature functions comprising the underlying functions. Thus an analytical closed-form solution might not be available. Moreover, there is no guarantee that the objective function is convex, which might result in multiple local optima. In general, when we are dealing with a multimodal objective function, finding the global optimum is not trivial. Secondly, based on our assumption, the simulation run lengths are discrete, and the number of simulation replications for each simulation group is also discrete. This integer constraint makes our problem even more difficult. Lastly, in our problem, the budget constraint is non-convex. Hence, our problem is a Mixed-Integer Nonlinear Programming problem, which can be very challenging to solve to optimality, except in special cases.

In order to get some insights without complicating the problem too much, we will solve the problem numerically by using the existing numerical optimization techniques for certain types of underlying functions. We observe that when the ranges of the decision variables are properly defined

and by relaxing the integer constraint, our objective function is Lipchitz continuous, and we can solve the problem numerically by using the Lipchitz-continuous Global Optimizer (LGO) embedded in AIMMS (Pinter 1996; Pinter 2005).

We will first study the case when the underlying function is a simple polynomial, including linear, full quadratic or full cubic polynomials. In the DOE literature, the simple polynomial models are of particular importance and interest due to their relative ease of derivation and wide application. We will also consider other simple underlying functions containing logarithmic and trigonometric feature functions. However, for these functions the number of feature functions is limited to two to avoid an excessively complex objective function which cannot be handled by the software.

3 NUMERICAL SOLUTIONS AND RESULT DISCUSSION

3.1 Linear Underlying Function

In the case of linear underlying function, the transient mean performance of the design follows a linear function $f(x) = \beta_1 + \beta_2 x$. Suppose that we would like to predict the mean performance of the design at the point of interest $x_M = 30$, with a total computing budget T that varies from 1000 to 4000, in increments of 1000. By using AIMMS, we can determine the minimum PVF given a certain computing budget, together with the corresponding optimal number of simulation replications and their run lengths. The numerical solutions indicate that we need one and only one simulation group to achieve the minimum PVF and the results are listed in Table 3-1 in which l_1 stands for the optimal simulation run length and N_1 stands for the optimal number of simulation replications we need.

Table 3 - 1: Numerical Solution for Linear Underlying Function

T	x_M	*Minimum PVF*	l_1	N_1
1000	30	0.001000	59	16.9492
2000	30	0.000500	59	33.8983
3000	30	0.000333	59	50.8475
4000	30	0.000250	59	67.7966

It seems that $K = 1$, and the optimal value for this run length l_1 is a function of x_M, and $l_1 = 2x_M - 1$. In fact, when there is only one simulation group, the optimal simulation run length and the minimum PVF can be calculated easily by using the computing software. In order to minimize the PVF, it is always better to exhaust the available computing budget (Brantley et al. 2011). Hence the inequality budget constraint can be replaced by an equality constraint. When we have only one simulation group, we have two decision variables. Due to the equality budget constraint, we can express the objective function as a univariate function whose global minimum can be found numerically, regardless of the types of feature functions comprising the underlying function.

Moreover, in the case of linear underlying function, an analytical solution to the problem can be derived as the objective function is not too complex. After doing some derivation, the minimum PVF is found to be $\frac{1}{T}$ when $x_M = \frac{l_1+1}{2}$, or $l_1 = 2x_M - 1$. This result is consistent with our observation based on the numerical solutions generated by AIMMS.

3.2 Full Quadratic and Full Cubic Underlying Function

In this case, we assume that the underlying function follows a full quadratic polynomial, namely, $y(x) = \beta_1 + \beta_2 x + \beta_3 x^2$. Again we want to make prediction at $x_M = 30$, and the total computing budget T varies from 1000 to 4000. The numerical solutions tell us that we need two different simulation groups of different run lengths to achieve the minimum PVF, and we denote the number of simulation replications we need for run lengths l_1 and l_2 as N_1 and N_2, respectively. In Table 3-2, we summarize the numerical solutions obtained from AIMMS under various x_M and T.

Table 3 - 2: Numerical Solutions for Full Quadratic Underlying Function (Continuous Case)

T	x_M	*Minimum PVF*	l_1	l_2	N_1	N_2
1000	30	0.001003	59.5624	1000.00	16.7726	0.000980
2000	30	0.000501	59.3804	1496.11	33.6670	0.000563
3000	30	0.000333	59.2528	2266.91	50.6215	0.000236
4000	30	0.000250	59.1889	3038.92	67.5736	0.000129

The value of l_1 is observed to be approximately $2x_M - 1$. Similar to the linear underlying function, as shown in the continuous optimal solutions, the minimum PVF is lower bounded by $\frac{1}{T}$. We notice that the value of N_2 is almost zero. In practice, the number of simulation replications must be an integer. In order to determine whether it is necessary to run the simulation at run length l_2, we compute the optimal solutions when the integer constraint is imposed and when the number of different simulation groups is two, by comparing the values of PVF calculated from all possible combinations of l_1, l_2, N_1 and N_2. The results are presented in Table 3-3.

Table 3 - 3: Numerical Solutions for Full Quadratic Underlying Function (Discrete Case)

T	x_M	*Minimum PVF*	l_1	l_2	N_1	N_2
1000	30	0.001139	58	188	14	1
2000	30	0.000543	59	230	30	1
3000	30	0.000355	58	274	47	1
4000	30	0.000263	58	288	64	1

It is noted that we still have two different simulation groups in the optimal solutions for the discrete case, and we would run a single simulation replication at run length l_2, implying that it is better to run this longer simulation replication than discarding it.

We have done a similar study for the full cubic underlying function $y(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3$, and the same observations as in the full quadratic case can be made.

3.3 General Underlying Function

In this section, we look at the numerical solutions to some other simple underlying functions. The types of feature functions studied in this section include linear, quadratic, cubic, logarithmic and sine functions. Due to the complexity of the objective function, analytical solutions cannot be obtained, except in the linear polynomial case. However, there is still an interesting observation we can make based on the numerical solutions presented in Table 3-4.

Table 3 – 4: Numerical Solutions for Various Types of Underlying Function

Underlying Function	*Number of Feature Functions*	*Optimal Number of Simulation Groups*	*Optimal Number of Decision Variables*
$\beta_1 + \beta_2 x$	2	1	2
$\beta_1 + \beta_2 x^2$	2	1	2
$\beta_1 + \beta_2 x^3$	2	1	2
$\beta_1 + \beta_2 x + \beta_3 x^2$	3	2	4
$\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3$	4	2	4
$\beta_1 + \beta_2 ln\,(x)$	2	1	2
$\beta_1 + \beta_2 sin\left(\frac{x}{50}\right)$	2	1	2

$\boldsymbol{\beta_1 + \beta_2 x + \beta_3 sin\,(x)}$	3	2	4

We observe that the number of decision variables we need in order to achieve the minimum PVF is at least equal to the number of feature functions in the underlying function. The usefulness of this observation is that it enables us to determine the minimum number of simulation groups we need in order to achieve the minimum PVF, regardless of the types of the component feature functions in the underlying function.

An intuitive way to explain the results in Table 3-4 is that the number of component feature functions in the underlying function is the same as the number of parameters we want to estimate in order to predict the mean performance of the design at x_M. The parameter vector $\boldsymbol{\beta} = [\beta_1, \beta_2 \dots, \beta_n]^T$ contains n parameters and it has $n-1$ degrees of freedom. In order to estimate this parameter vector, we need at least n independent decision variables which give us $n-1$ degrees of freedom due to the equality budget constraint. Therefore, the number of decision variables should not be smaller than the number of parameters we want to estimate. Nevertheless, a more rigorous argument needs to be developed to justify this observation statement.

3.4 SDBA Procedure

Based on the observations we made in the previous sections, we develop the following Single Design Budget Allocation (SDBA) Procedure, which helps us efficiently allocate computing budget to better estimate the transient design performance.

SDBA-Linear Underlying Function *When the underlying function follows a linear polynomial, we need one and only one simulation group in which all simulation replications would have the same run length* $l_1 = 2x_M - 1$*, where* x_M *is the point where the prediction is made.*

In practice, both the simulation run lengths and the number of simulation replications are discrete. When T is not divisible by x_m, we will run as many simulation replications of run length l_1 as possible, namely $N_1 = \left\lfloor \frac{T}{l_1} \right\rfloor$, where $\lfloor x \rfloor$ is a floor function. The remaining computing budget would be used to run a single simulation replication of run length $l_2 = T - N_1 \times l_1$.

SDBA-Full Quadratic/Cubic Underlying Function *When the underlying function follows a full quadratic or full cubic polynomial, we need two and only two simulation groups* G_A *and* G_B*. Group* G_A *contains several simulation replications of run length* $l_1 = 2x_M - 1$*. Group* G_B *contains a single simulation replication of run length* l_2*, whose value depends on the total computing budget and can be determined numerically.*

It is noted that the value of l_2 depends on T and its value can be calculated numerically since the objective function can be expressed as a univariate function using the pre-defined values for l_1 and N_2.

SDBA-General Underlying Function *When the transient mean performance of design follows a certain underlying function consisting of several feature functions, the minimum number of simulation groups (K) we need in order to achieve the minimum PVF and the number of component feature functions (n) comprising the underlying function are related by* $K = \left\lceil \frac{n}{2} \right\rceil$*, where* $\lceil x \rceil$ *is a ceiling function.*

4 IMPLEMENTATION OF SDBA PROCEDURE

4.1 Full Quadratic Underlying Function

In this numerical experimentation, we would like to test the efficiency of the SDBA Procedure. To do so, we consider the case when the transient mean performance of design follows a full quadratic underlying function $y(x) = 0.5227 + 0.1338x + 0.002x^2$. We would like to predict the mean performance of the design at point $x_M = 50$, which is expected to be 12.2127. The Simple Regression Procedure in which all simulation replications run up to the point of interest is used as the comparison procedure. The Simple Sampling Procedure in which the design performance is calculated as sample mean is also used as a comparison procedure due to its wide application. We assume homogeneous normal simulation noises along the simulation replication, with mean zero and variance one.

Expression (3) which is the least squares formula is used to estimate the design mean and variance. The simulation has been conducted in *MATLAB* and the result is presented in Figure 4-1, in which the *Minimum Variance* is the lower bound for the estimated variance calculated by using the formula $\frac{\sigma_E^2}{T}$, where σ_E^2 is the unbiased estimator of the variance of design performance, calculated from the simulation outputs.

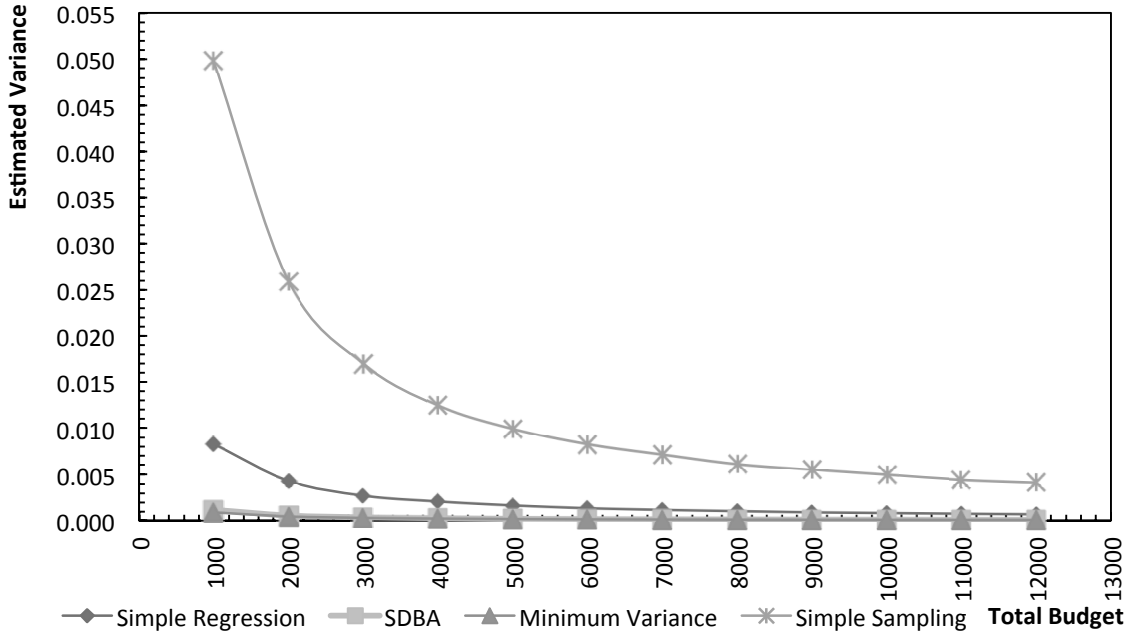

Figure 4 - 1: Comparison of Estimated Variance Obtained by Using Different Procedures with Full Quadratic Underlying Function

As illustrated in the diagram, given a certain amount of computing budget, using the regression procedures would enable us to achieve smaller estimated variance than using the Simple Sampling Procedure. Moreover, the SDBA Procedure gives a much smaller estimated variance, compared to the Simple Regression Procedure. It is also noted that as the computing budget increases, we get closer to the minimum variance obtained in the continuous case, though our procedure uses a discrete computing budget. We have done similar numerical experimentation for the full cubic underlying function, and similar conclusions can be drawn.

When the underlying function is a full quadratic or full cubic polynomial, the SDBA Procedure suggests that we run simulation replications at two different run lengths. It is noted that we need to run a single simulation replication at the longer run length, whose value depends on the total computing budget. It seems that the impact of removing this run length from the SDBA Procedure might not be very significant. When there is a single simulation group, the optimal run length can be calculated numerically with a given x_M. The analytical solution to this optimal run length might not be available due to the excessive complexity of the objective function. In Figure 4-2, we present the experimental results for the Simplified SDBA Procedure in which only a single simulation run length is used, under the same experiment setting as in Figure 4-1.

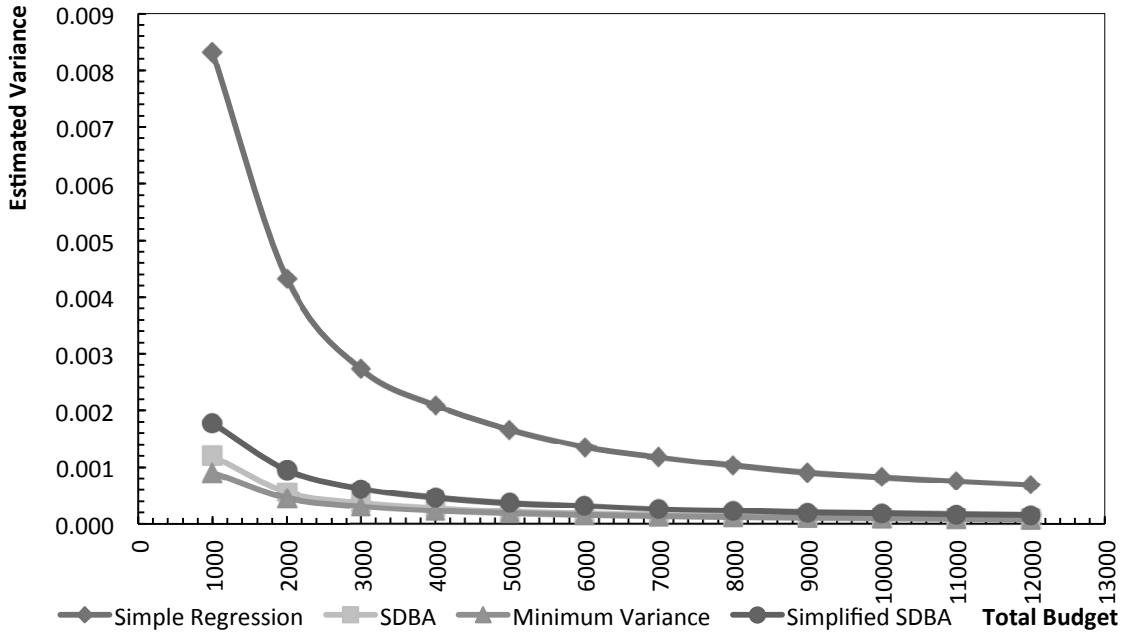

Figure 4 - 2: Numerical Experimentation Results for Simplified SDBA Procedure for Full Quadratic Underlying Function

As we can see from Figure 4-2, the Simplified SDBA Procedure is able to perform much better than the Simple Regression Procedure, though its performance is slightly worse than the SDBA Procedure with two run lengths, which is expected. In fact, the minimum PVF we get from the Simplified SDBA Procedure is about twice the minimum PVF we get by using SDBA Procedure. Due to this minor difference in performance, in practice, we might run all the simulation replications at the same run length due to its ease of implementation. Similar results can be obtained for the full cubic underlying function.

4.2 Heterogeneous Simulation Noise

In practice, the assumption of homogeneous simulation noise is often violated. Without knowing the exact noise distribution, it is very difficult to determine the optimal computing budget strategy. This problem can be addressed by assuming that the variance of the simulation noise follows a certain functional form and then determining numerically the optimal run length when a single simulation run length is used by minimizing the objective in expression (2). For example, if the simulation noise increases as the simulation run length increases, we might approximate the noise variance by a linearly increasing function. While this may be a crude approximation, it might provide us with a better budget allocation scheme than assuming homogeneous simulation noise, as it takes into account the fact that the simulation noise increases along the replication.

In this section, we consider an implementation of the SDBA Procedure assuming a certain simulation noise pattern along the simulation replication. The example we use is the M/M/1 queue which is of practical importance in many service systems like hospitals, in which the customer waiting time can be considered as a good indicator of system performance. The traffic intensity is 0.9 (mean service rate of 1 and mean arrival rate of 0.9), with the system being initialized empty and idle at time zero. Suppose we wish to estimate the system waiting time (i.e., waiting time in the queue plus service time) of the 20^{th} customer joining the queue using simulation. It is expected that as the simulation run length increases, the uncertainty in predicting the n^{th} customer's system waiting time increases, resulting in a higher simulation noise. Therefore, we will approximate the noise variance by a linearly increasing function as the number of customers joining the queue increases, namely, $\varepsilon(x_i) \sim \mathcal{N}\left(0, \frac{x_i}{a}\right)$, where a is a positive number. By simulating and studying the average transient

customer system waiting times, we find that the logarithm underlying function $y(x) = \beta_0 + \beta_1 ln\ (x)$ is a good approximation to the transient customer system waiting time. Using this underlying function model and the linear simulation noise variance pattern approximation, we can obtain the optimal simulation run length by minimizing expression (2). Once this optimal run length is determined, we can use it in the SDBA Procedure to run the simulation. Expression (1), which is the weighted least squares formula, is used to compute the design mean performance and variance.

In Table 4-1, we present results on the prediction of the 20^{th} customer's system waiting time by running the simulation using different budget allocation rules. The analytical value of the mean system waiting time of the 20th customer is known to be approximately 4.275 (Kelton and Law 1985). Of the three procedures, Simple Sampling has the lowest bias (see Table 4.2). This is to be expected since it is an unbiased estimator. The SDBA Procedure has lower bias than the Simple Regression Procedure in all cases. Additionally, the estimated variances for the regression procedures are much smaller than the estimated variances for the Simple Sampling Procedure. In general, the SDBA Procedure enables us to achieve about a 10% variance reduction as compared to the Simple Regression Procedure and a 90% variance reduction compared to Simple Sampling approach. Table 4-2, also presents the mean squared errors (MSE) for all three procedures calculated from the numerical results. As illustrated in Table 4-3, using the SDBA Procedure instead of the other two procedures benefits us with significant improvement in MSE.

Table 4 - 1: Numerical Experimentation Results for M/M/1 Queue Using Logarithm Underlying Function

T	***Estimated Mean System Waiting Time***			***Estimated Variance of System Waiting Time***		
	SDBA	***Simple Regression***	***Simple Sampling***	***SDBA***	***Simple Regression***	***Simple Sampling***
1000	4.088984	3.912666	4.283880	0.013467	0.015034	0.248412
2000	4.100969	3.932829	4.263112	0.006966	0.007727	0.124803
3000	4.102279	3.934631	4.264272	0.004696	0.005178	0.083473
4000	4.121475	3.935219	4.273659	0.003555	0.003918	0.062574
5000	4.126383	3.934695	4.277299	0.002859	0.003144	0.050014
6000	4.127300	3.937391	4.287525	0.002386	0.002610	0.041706

Table 4 - 2: Simulation Bias and MSE for Different Procedures

T	***Bias***			***MSE***		
	SDBA	***Simple Regression***	***Simple Sampling***	***SDBA***	***Simple Regression***	***Simple Sampling***
1000	0.185741	0.362059	-0.009155	0.047967	0.146121	0.250713
2000	0.173756	0.341896	0.011613	0.037157	0.124620	0.126184
3000	0.172446	0.340094	0.010453	0.034434	0.120842	0.084659
4000	0.153250	0.339506	0.001066	0.027041	0.119182	0.063305
5000	0.148342	0.340030	-0.002574	0.024864	0.118764	0.050632
6000	0.147425	0.337334	-0.012800	0.024120	0.116404	0.042288

Table 4 - 3: Ratio of MSE between Various Procedures

T	***Percentage Improvement MSE***	
	SDBA to Simple Regression	***SDBA to Simple Sampling***
1000	67.17%	80.87%
2000	70.18%	70.55%
3000	71.51%	59.33%
4000	77.31%	57.29%
5000	79.06%	50.89%
6000	79.28%	42.96%

5 CONCLUSION AND FUTURE WORK

In this study, we have looked into the problem of optimal computing budget allocation for a single design whose transient mean performance follows a certain underlying function. Numerical solutions to the problem have been obtained by using optimization solvers, and several observations have been made, based on which, the Single Design Budget Allocation (SDBA) Procedure has been developed. The numerical experimentation confirms the high efficiency of the SDBA Procedure, in comparison with the other existing budget allocation rules.

Though an approach has been proposed to handle heteroscedasticity when we apply the SDBA Procedure, more work is needed on this issue. Additionally, the assumption of uncorrelated simulation output might not hold in real life applications. Further study is required to justify the performance of the SDBA Procedure when the simulation outputs are correlated.

REFERENCES

Atkinson, A., A. Donev, and R. Tobias. 2007. *Optimum experimental designs, with SAS*. Oxford University Press.

Brantley, M., L. H. Lee, C. -H. Chen, and A. Chen. 2011. "Efficient Simulation Budget Allocation with Regression". Submitted to *IIE Transactions*.

DeGroot, M. 2004. *Optimal statistical decisions*. Wiley-Interscience.

Gill, J. 2008. *Bayesian methods: a social and behavioral sciences approach*. Chapman & Hall/CRC.

Goldsman, D., B. L. Nelson, and B. W. Schmeiser. 1991. Methods for Selecting the Best System. In B. L. Nelson, W. D. Kelton, & G. M. Clark (Ed.), *Winter Simulation Conference* (pp. 177-186). New Jersey: The Institute of Electrical and Electronic Engineers.

Kelton, W. D. and A. M. Law. 1983. A New Approach for Dealing with the Startup Problem in Discrete Event Simulation. *Naval Research Logistics 30:641-658.*

Kelton, W. D. and A. M. Law. 1985. Transient Behavior of the M/M/s Queue. *Operations Research 33(2): 378-396.*

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd ed. New York: McGraw-Hill, Inc.

Morrice, D., M. Brantley, and C. -H. Chen. 2008. "An efficient ranking and selection procedure for a linear transient mean performance measure". In *Proceedings of Winter Simulation Conference*, (pp. 290-296).

Morrice, D., M. Brantley, and C.-H. Chen. 2009. "A transient means ranking and selection procedure with sequential sampling constraints". In *Proceedings of Winter Simulation Conference*, (pp. 590-600).

Morrice, D., and L. W. Schruben. 2001. "A Frequency Domain Metamodeling Approach to Transient Sensitivity Analysis". *IIE Transactions* 33(3): 229-244.

Pinter, J. 1996. *Global optimization in action: continuous and Lipchitz optimization--algorithms, implementations, and applications.* Dordrecht; Boston: Kluwer Academic Publishers.

Pinter, J. 2005. *AIMMS/LGO Solver Engine. A Brief Introduction and User's Guide.* Pinter Consulting Services, Inc.

Tanrisever, F., D. J. Morrice, and D. Morton. 2012. "Managing Capacity Flexibility in Make-to-Order Production Environments". To appear in *European Journal of Operational Research*.

AUTHOR BIOGRAPHIES

XIANG HU is an M. Eng student in the Department of Industrial and Systems Engineering, National University of Singapore. He is also a final year student in École Centrale de Paris, where he is doing a concurrent double master degree. His research interest is on simulation optimization. He is currently working on the optimal computing budget allocation by using regression. His email address is a0040238@nus.edu.sg.

LOO HAY LEE is an Associate Professor and Deputy Head (Graduate Studies and Research) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his Ph.D. degree in 1997 from Harvard University. He is currently a senior member of IEEE, a member of INFORMS and ORSS. His research interests include simulation-based optimization, maritime logistics and supply chain systems. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is an Associate Professor and Deputy Head (Administration) in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B.Eng (Civil Engineering) degree from the National University of Singapore in 1987 and his Ph.D. degree from the Georgia Institute of Technology in 1992. His research interests include supply chain modeling, system simulation and optimization. His email address is isecep@nus.edu.sg.

DOUGLAS J. MORRICE is a Professor in Operations Management at The University of Texas at Austin. He is also Director of the University of Texas Supply Chain Management Center of Excellence. He has an ORIE Ph.D. from Cornell University. His research interests include simulation design, modeling, and analysis, and supply chain risk management. Dr. Morrice was Co-Editor of the Proceedings of the 1996 Winter Simulation Conference, and 2003 Winter Simulation Conference Program Chair. He is currently serving as a representative for the INFORMS Simulation Society on the Winter Simulation Conference Board of Directors. His email address is morrice@mail.utexas.edu.

CHUN-HUNG CHEN is a Professor of Systems Engineering and Operations Research at George Mason University. He received his Ph.D. from Harvard University in 1994. His research interests are mainly in development of very efficient methodology for simulation and optimization and its applications. Dr. Chen has served as Co-Editor of the Proceedings of the 2002 Winter Simulation Conference and Program Co-Chair for 2007 Informs Simulation Society Workshop. He is currently an associate editor of IEEE Transactions on Automatic Control, area editor of Journal of Simulation Modeling Practice and Theory, associate editor of International Journal of Simulation and Process Modeling, and simulation department editor for IIE Transactions. His email address is cchen9@gmu.edu.