EFFICIENT COMPUTING BUDGET ALLOCATION BY USING REGRESSION WITH SEQUENTIAL SAMPLING CONSTRAINT

HU XIANG

NATIONAL UNIVERSITY OF SINGAPORE

2012

EFFICIENT COMPUTING BUDGET ALLOCATION BY USING REGRESSION WITH SEQUENTIAL SAMPLING CONSTRAINT

HU XIANG

(B.Eng. (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

HU XIANG

07 December 2012

ACKNOWLEDGEMENT

During this study, I have received tremendous help and support from many parties to whom I would like to extend my sincerest gratitude and appreciation for their efforts and assistances.

Firstly and most importantly, I would like to thank my supervisors Associate Professor Lee Loo Hay and co-supervisor Associate Professor Chew Ek Peng, who provided me with guidance and help along this study. Though it can be challenging discussing my work with them, every meeting and discussion was inspirational and thought-provoking. They enlightened me with their wisdom and vision, which guided me in the right direction. Without their patience and encouragement, completing this study is not possible.

I would also like to thank Professor Chen Chun-Hung and Professor Douglas J. Morrice, who overviewed my research progress and provided me with invaluable feedback and suggestions based on their rich experience and expertise in this domain.

Last but not least, I would like to extend my appreciation to my family and friends to whom I am deeply indebted for their continuous support. In particular, I would like to thank Mr. Nguyen Viet Anh and Ms. Zhang Si for spending time discussing with me and providing me with indispensable suggestions.

TABLE OF CONTENTS

ACKNOWLEDGEMENT .... **I**

TABLE OF CONTENTS .... **II**

SUMMARY .... **IV**

LIST OF TABLES .... **V**

LIST OF FIGURES .... **VI**

LIST OF SYMBOLS .... **VII**

1. INTRODUCTION **13**

2. LITERATURE REVIEW **15**

3. SINGLE DESIGN BUDGET ALLOCATION **19**

3.1. PROBLEM FORMULATION 19

3.1.1. Problem Setting 19

3.1.2. Sampling Distribution of Design Performance 21

3.2. SOLUTIONS TO LEAST SQUARES MODEL 27

3.2.1. Lower Bound of Objective Function 27

3.2.2. Linear Underlying Function 29

3.2.3. Full Quadratic Underlying Function 32

3.2.4. Full Cubic Underlying Function 34

3.2.5. General Underlying Function 35

3.3. SDBA PROCEDURE AND NUMERICAL IMPLEMENTATION 37

3.3.1. SDBA Procedure 37

3.3.2. Full Quadratic Underlying Function with Homogeneous Noise 39

3.3.3. M/M/1 Queue with Heterogeneous Simulation Noise *41*

4. MULTIPLE DESIGNS BUDGET ALLOCATION **46**

4.1. PROBLEM SETTING AND PROBLEM FORMULATION 46

4.1.1. Problem Setting *46*

4.1.2. Sampling distribution of Design Performance *48*

4.1.3. Rate Function and Model Formulation *49*

4.2. PROBLEM SOLUTION 51

4.2.1. Condition for Decomposition *51*

4.2.2. Problem Decomposition *52*

4.3. SDBA+OCBA PROCEDURE AND NUMERICAL IMPLEMENTATION. 55

4.3.1. SDAB+OCBA Procedure *55*

4.3.2. Application of SDBA+OCBA Procedure *57*

4.3.3. Ranking and Selection of the Best M/M/1 Queuing System *57*

4.3.4. Ranking and Selection of the Best Full Quadratic Design *59*

5. CONCLUSION AND FUTURE WORK **63**

5.1. SUMMARY AND CONTRIBUTIONS 63

5.2. LIMITATIONS AND FUTURE WORK 64

BIBLIOGRAPHY **65**

SUMMARY

In this thesis, we develop an efficient computing budget allocation rule to run simulation for a single design whose transient mean performance follows a certain underlying functional form, which enables us to obtain more accurate estimation of design performance by doing regression. A sequential sampling constraint is imposed so as to fully utilize the information along the simulation replication. We formulate this problem using the Bayesian regression framework and solve it for some simple underlying functions under a few common assumptions in the literature of regression analysis. In addition, we develop a Single Design Budget Allocation (SDBA) Procedure that determines the number of simulation replications and corresponding run lengths given a certain computing budget. Numerical experimentation confirms the efficiency of the procedure relative to extant approaches.

Moreover, the problem of selecting the best design among several alternative designs based on their transient mean performances has been studied. By applying the Large Deviations Theory, we formulate our problem as a global maximization problem, which can be decomposed under the condition that the optimal budget allocation for each single design is independent of the computing budget allocated to that design. As a result, the SDBA+OCBA Procedure has been developed, which has been proved to be an efficient computing budget allocation rule that enables us to correctly select the best design by consuming much less computing budget than the other existing computing budget allocation rules, based on the numerical experimentation results.

LIST OF TABLES

Table 3 - 1 Numerical Experiment for SDBA Rule for Linear Underlying Function .. 31

Table 3 - 2 Numerical Experiment for SDBA Rule for Full Quadratic Underlying Function 34

Table 3 - 3 Numerical Experiment for SDBA Rule for Full Cubic Underlying Function 35

Table 3 - 4 Numerical Solutions for Various Types of Underlying Function 36

Table 3 - 5 Assumptions and Budget Allocation Strategy for Various Procedures and Approaches 43

Table 3 - 6 Numerical Experimentation Results for M/M/1 Queue Using Various Procedures 44

Table 3 - 7 Simulation Bias and MSE for Different Procedures 44

Table 3 - 8 Ratio of MSE between Various Procedures 44

LIST OF FIGURES

Figure 3 - 1 Comparison of Estimated Variance Obtained by Using Different Procedures with Full Quadratic Underlying Function 40

Figure 3 - 2 Numerical Experimentation Results for Simplified SDBA Procedure for Full Quadratic Underlying Function 41

Figure 4 - 1 Comparisons of the performances of various computing budget allocation rule on the selection of the best M/M/1 queuing system 59

Figure 4 - 2 Comparisons of the performances of various computing budget allocation rule on the selection of the best design with full quadratic underlying function 61

LIST OF SYMBOLS

x_M	The point of interest
T	The total computing budget available
$y(x)$	The expected mean performance of design at observation point x
n	The total number of feature functions in the underlying function
β_i	The unknown parameter in underlying function
$\Phi_i(x)$	The component feature function comprising the underlying function
$\boldsymbol{\beta}$	The unknown parameter vector
$\boldsymbol{b}$	The mean vector of the prior distribution of $\boldsymbol{\beta}$
$\boldsymbol{\Sigma_\beta}$	The variance-covariance matrix of the prior distribution of $\boldsymbol{\beta}$
$\boldsymbol{F}$	The vector of simulation output
$\boldsymbol{Y}$	The vector of expected mean performance of design
$\boldsymbol{\varepsilon}$	The vector of simulation noise
$f(x_i)$	The simulation output at observation point x_i
$y(x_i)$	The expected mean performance of design at observation point x_i
$\varepsilon(x_i)$	The simulation noise at observation point x_i
Σ	The variance-covariance matrix of simulation noise
$\widetilde{\boldsymbol{\beta}}$	The sampling distribution of the parameter vector
$\tilde{y}(x_M)$	The sampling distribution of the expected mean design performance at the point of interest
$Var(\tilde{y}(x_M))$	The estimated variance of expected mean performance of design at observation point x_i
K	The otal number of simulation groups
G_i	The i^{th} simulation group

N_i The total number of simulation replications in the i^{th} simulation group

l_i The simulation run length for the i^{th} simulation group

$\boldsymbol{F_i^j}$ The vector of simulation output for the j^{th} simulation replication in i^{th} simulation group

$f_i^j(x_t)$ The simulation output at observation point x_t for the j^{th} simulation replication in i^{th} simulation group

$\boldsymbol{X_i}$ The matrix of feature functions for the i^{th} simulation group

X_i The vector of feature functions for the i^{th} simulation group

$\widehat{\boldsymbol{\beta}}_{GLS}$ The sampling distribution of the parameter vector derived by using the GLS formula

$\boldsymbol{\Sigma_i}$ The prior variance-covariance matrix of the unknown parameter vector

$\tilde{y}(x_M)_{GLS}$ The sampling distribution of the expected mean design performance at the point of interest derived by using the GLS formula

$\boldsymbol{W_i}$ The weight matrix in the Weighted Least Squares model

e_{kk} The k^{th} diagonal element in the variance-covariance matrix $\boldsymbol{\Sigma_i}$

$\sigma_{x_k}^2$ The noise variance at observation point x_k

$\widetilde{\boldsymbol{\beta}}_{WLS}$ The sampling distribution of the parameter vector derived by using the WLS formula

$\tilde{y}(x_M)_{WLS}$ The sampling distribution of the expected mean design performance at the point of interest derived by using the WLS formula

$\widetilde{\boldsymbol{\beta}}_{LS}$ The sampling distribution of the parameter vector derived by using the LS formula

$\tilde{y}(x_M)_{LS}$	The sampling distribution of the expected mean design performance at the point of interest derived by using the LS formula
$Var\left(\tilde{y}(x_M)\right)_{LS}$	The estimated variance of expected mean performance of design at observation point x_i calculated from the LS formula
α_i	The proportion of total computing budget allocated to the i^{th} simulation replication
c	The nonzero $n \times 1$ vector
R	The $n \times n$ positive definite matrix
ξ_0	The c-optimal design
PVF_{Linear}^{K}	The PVF derived from the linear underlying function with K different simulation groups
$PVF_{Quadratic}$	The PVF derived from the quadratic underlying function
C	The constant
D	The constant
N_0	The number of initial simulation replications
Θ	The design space
θ_i	The i^{th} alternative design
P	The total number of alternative designs
$y_{\theta_i}(x)$	The expected transient performance of design θ_i at observation point x
n_{θ_i}	The total number of feature functions comprising the underlying function of design θ_i
$\beta_{\theta_i,j}$	The j^{th} unknown parameter for design θ_i
$\Phi_{\theta_i,j}(x)$	The one dimensional one-to-one feature function of design θ_i

$\boldsymbol{\beta}_{\theta_i}$ The unknown parameter vector for design θ_i

π_{θ_i} The total number of simulation replications that need to run for design θ_i

Q_{θ_i} The number of different simulation groups for design θ_i

$G_{\theta_i,q}$ The q^{th} simulation group for design θ_i

$n_{\theta_i,q}$ The number of simulation replications in the q^{th} simulation group for design θ_i

$l_{\theta_i,q}$ The run length of the simulation replications in the q^{th} simulation group for design θ_i

$\boldsymbol{F}_{\theta_i,q,t}$ The simulation output vector for the t^{th} simulation replication in group $G_{\theta_i,q}$

$\boldsymbol{Y}_{\theta_i,q}$ The vector of the expected mean design performance for all simulation replications in group $G_{\theta_i,q}$

$\boldsymbol{\varepsilon}_{\theta_i,q}$ The simulation noise vector for all simulation replications in group $G_{\theta_i,q}$

$f_{\theta_i,t}(x_p)$ The simulation output collected from the t^{th}simulation replication in group $G_{\theta_i,q}$ at observation point x_p

$y_{\theta_i}(x_p)$ The expected mean performance of the design at observation point x_p for design θ_i

$\Sigma_{\theta_i,q}$ The variance-covariance matrix for all simulation replications in group $G_{\theta_i,q}$

$\tilde{y}_{\theta_i}(x_M)$ The sampling distribution of the mean performance of design θ_i at the point of interest x_M

$\tilde{y}_{\theta_b}(x_M)$ The sampling distribution of the mean performance of the selected

	best design at x_M
$\boldsymbol{X}_{\theta_i,q}$	The $l_{\theta_i,q} \times n_{\theta_i}$ matrix of the feature function matrix for the simulation replications in group $G_{\theta_i,q}$
$X_{\theta_i,i}$	The $1 \times n_{\theta_i}$ feature function vector at simulation run length x_i for design θ_i
δ_{θ_i}	The estimated mean performance of the design θ_i at x_M
$\sigma^2_{\theta_i}$	The estimated variance of the design θ_i at x_M
$\sigma^2_{\theta_i,E}$	The unbiased estimator of the performance variance of design θ_i
E_{θ_i}	The probabilistic event
$\alpha_{\theta_i,q}$	The proportion of total computing budget allocated to the group $G_{\theta_i,q}$
α_{θ_i}	The proportion of total computing budget allocated to design θ_i
T_0	The initial simulation budget allocated to each design
Δ	The total computing budget allocated during each round of budget allocation

OCBA	Optimal Computing Budget Allocation
DOE	Design of Experiment
GLS	Generalized Least Squares
WLS	Weighted Least Squares
LS	Least Squares
PVF	Prediction Variance Factor
LGO	Lipchitz Global Optimizer
SDBA	Single Design Budget Allocation
MSE	Mean Squared Error

P{CS}	Probability of Correct Selection
P{IS}	Probability of Incorrect Selection

1. INTRODUCTION

Many industrial applications have proved that simulation-based optimization is able to provide satisfactory solution under the condition that computing budget and time for running simulation be abundant. Nevertheless, in reality, the latter condition is hardly met due to the constraint of limited computing budget or due to the requirement that the decision-making process based on optimization result shall be completed in a restricted time period. The computing budget and time required to obtain a satisfactory result might be very significant, especially when the number of alternative designs is large, as each design would require certain simulation replications in order to achieve a reliable statistical estimation. Several researchers have dedicated themselves in searching for an effective and intelligent way of allocating limited computing budget so as to achieve a desired optimality level, and the idea of Optimal Computing Budget Allocation has emerged to be either maximizing the simulation and optimization accuracy, given a limited computing budget, or minimizing the computing budget while meeting certain optimality level (Chen and Lee, 2011).

This thesis provides an OCBA formulation for estimating the transient mean performance at the point of interest for a single design. We derive theoretical and numerical results that characterize the form of the optimal solution for polynomial regression functions up to order three. Polynomial functions represent an important class of regression models since they are often used in practice to model non-linear behaviour. Additionally, we provide more limited results on the optimal solutions for sinusoidal and logarithmic regression functions. The results extend both the simulation and statistical DOE literatures. To apply the theory, we propose an algorithm and numerically assess its efficacy on an M/M/1 queuing example. The performance of our approach is compared against other extant procedures.

Moreover, we develop an efficient computing budget allocation algorithm that can be applied to select the best design among several alternative designs. By applying the Bayesian regression framework and the Large Deviations Theory, we formulate our Ranking and Selection problem as a maximization problem of the convergence rate of the probability of the correct selection. We decompose the problem into two sub-problems under certain conditions, and the SDBA+OCBA Procedure has been developed when the condition is met. Numerical experimentation has confirmed the efficiency of this newly developed SDBA+OCBA Procedure.

The remainder of this thesis will be structured in the follow manner. Chapter 2 presents some of the work that is related to our problem in the literature, based on which we define our problem setting and the goals we would like to achieve in this study. Chapter 3 shows how we could improve the prediction accuracy of the transient design performance by doing regression analysis based on certain assumptions. The SDBA Procedure would be presented at the end of the chapter. Chapter 4 presents how we could make use of the SDBA Procedure to develop an efficient Ranking and Selection Procedure by using Large Deviation Theory. Chapter 5 concludes the whole thesis with a summary of what we have achieved, the practical importance and usefulness of our study. Some limitations and future works are also discussed at the end of the thesis.

2. LITERATURE REVIEW

Since the very beginning of the idea conception of OCBA, the world has witnessed incredibly fast development of OCBA, thanks to many researchers who have been diligently working on this topic. With their continual and significant contribution, basic algorithms to effectively allocate computing budget have been developed (Chen, 1995) and further improved to enable people to select the best design among several alternative designs with a limited computing budget (Chen, Lin, Yücesan and Chick, 2000). The OCBA technique has also been extended to solve problems with different objectives but of similar nature, and these problems include the problem of selecting the optimal subset of top designs (Chen. , He, Fu and Lee, 2008), the problem of solving the multi-objective problem by selecting the correct Pareto set with high probability(Chen and Lee, 2009; Lee, Chew, Teng and Goldsman, 2010), the problem of selecting the best design when samples are correlated (Fu, Hu, Chen and Xiong, 2007), the problem of OCBA for constrained optimization (Pujowidianto, Lee, Chen and Yep, 2009), etc. The application of OCBA can be found in various domains, such as in product design (Chen, Donohue, Yücesan and Lin, 2003), air traffic management (Chen and He, 2005), etc. Furthermore, the OCBA technique has been extended to solve large-scale simulation optimization problem by integrating it with many optimization search algorithms (He, Lee, Chen, Fu and Wasserkrug, 2009; Chew, Lee, Teng and Koh, 2009). Last but not least, the OCBA framework has been expanded to solve problems beyond simulation and optimization, such as data envelopment analysis, design of experiment (Hsieh, Chen and Chang, 2007) and rare-event simulation (Chen and Lee, 2011).

Among the diverse extensions of OCBA technique proposed by various researchers, the Ranking and Selection Procedure for a linear transient mean performance measure developed by (Morrice, Brantley and Chen, 2008) is of particular interest as it incorporates the regression analysis in the computing budget allocation and addresses the problem in which the transient design performances are not constant but follow certain underlying function. Simulation outputs are collected at the supporting points, which are used to estimate design performances by doing regression. They further generalize the regression approach of estimating design performances to the problem in which the underlying function of design performance is a polynomial of up to order five (Morrice, Brantley and Chen, 2009). Each simulation replication is run up to the point where prediction of transient design performance is to be made, and the sequential sampling constraint is imposed and multiple simulation output collection is conducted to maximize the information we could use to make prediction. They also show that significant variance reduction can be achieved by estimating design performance using regression. A heuristic computing budget allocation procedure, which would be referred to as the Simple Regression+OCBA Procedure, has been proposed, hoping to make advantage of the variance reduction achieved by doing regression.

In this thesis, we aim at developing an efficient Ranking and Selection Procedure that enables us to quickly select the best design among several alternative designs. In order to do so, more accurate estimation of the design performances are desired, especially when the design performances are transient, thus are difficult to predict. Once we are able to develop a more efficient computing budget allocation procedure to estimate transient design performances, we could make use of the newly developed procedure to further improve the current Simple Regression+OCBA Procedure.

Analysis of transient behavior is an important simulation problem in, for example, the initial transient problem (Law and Kelton, 2000) and sensitivity analysis (Morrice and Schruben, 2001). Transient analysis is also important in so-called “terminating simulations” (Law and Kelton, 2000) that have finite terminating conditions and never achieve steady state. Examples of transient behavior 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 multiple different prototypes are being simulated simultaneously. In this application, the prototype that is able to achieve the best specifications (e.g., based on performance, quality, safety, etc.) after a certain amount of development time wins. 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 (Tanrisever, Morrice and Morton, 2012).

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 thesis, and calculate the sample mean and sample variance by using the simulation outputs collected at that point. Another more sophisticated way is to use a regression approach which incorporates all information along the simulation replication instead of only at the point of interest. The regression approach is expected to provide more accurate estimation since more information is used. For example, Kelton and Law (1983) develop a regression-based procedure for the initial transient problem and Morrice and Schruben (2001) use a regression approach for transient sensitivity analysis.

Morrice, Brantley and Chen (2008) derive formula to calculate the mean performance of design when its transient mean performance follows a linear function, with the simulation outputs collected at the 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 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 thesis.

As a matter of fact, our problem is related to the Design of Experiment (DOE) literature. In particular, it is related to the c-optimal design problem in which we seek to minimize the estimated variance of the mean design performance measure at the point of interest, which is a linear combination of the unknown parameters, assuming that the underlying function can be expressed as a sum of several feature functions (Atkinson, Donev and Tobias, 2007). El-Krunz and Studden (1991) give a Bayesian version of Elfving's theorem regarding the c-optimality criterion with emphasis on the inherent geometry. In the case of homogeneous simulation noise over the domain, several results on the local c-optimal designs for both linear and nonlinear models have been generated (Haines 1993; Pronzato 2009) based on the work done by Elfving (1952). However, the problem of c-optimal design under the sequential constraint has not been studied. In this thesis, we would present some analytical and numerical solutions to this problem when the undelrying function takes certain forms.

3. SINGLE DESIGN BUDGET ALLOCATION

3.1. PROBLEM FORMULATION

3.1.1. Problem Setting

In this thesis, we would like to improve the Simple Regression Procedure by using the notion of Optimal Computing Budget Allocation (OCBA) (Chen and Lee, 2011). We aim at improving the estimate accuracy of the transient mean performance of the design at the point of interest 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 100th customer's waiting time, we need to run simulation 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 the simulation outputs we collect. Therefore, based on the aforementioned constraints and assumptions, our problem becomes the problem of determining the optimal simulation run lengths for all simulation replications, in order to obtain the best (minimum variance) estimate of the design's mean performance at the point of interest by doing regression, subject to limited simulation computing budget.

To put the aforementioned assumptions and considerations into mathematical expressions, we would like to 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 design at observation point x. The function $\Phi_i(x)$ is a univariate one-to-one feature function, which can be any continuous function. Without loss of generality, 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 prior distribution follows a multivariate normal distribution with mean $\boldsymbol{b}$ and variance-covariance matrix $\Sigma_{\boldsymbol{\beta}}$. The sampling distribution of $\boldsymbol{\beta}$ 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 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. 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 sampling distribution of the unknown parameter vector as $\widetilde{\boldsymbol{\beta}}$ and the sampling 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 $Var\left(\tilde{y}(x_M)\right)$, 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 $Var\left(\tilde{y}(x_M)\right)$.

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 one 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.

$$\begin{aligned} &\boldsymbol{obj}. && \boldsymbol{\min_{K}} \ \boldsymbol{\min_{l_1,l_2,\dots,l_K,N_1,N_2,\dots,N_K}} \ Var(\tilde{y}(x_M)) && (3.1)\\ &\boldsymbol{s.t.} && \sum_{i=1}^{K} l_i N_i \leq T \\ & && 0 < l_i \leq T; \qquad l_i \in \mathbb{N}^0, \forall i = 1,2,\dots,K \\ & && N_i > 0; \qquad N_i \in \mathbb{N}^0, \forall i = 1,2,\dots,K \\ & && K > 0 \qquad K \in \mathbb{N}^0 \end{aligned}$$

3.1.2. Sampling Distribution of Design Performance

Let $\boldsymbol{F_i^j} = \left[f_i^j(x_1), f_i^j(x_2), \dots f_i^j\left(x_{l_i}\right)\right]^T; i = 1,2,\dots,K, j = 1,\dots,N_i$ 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 $n \times l_i$ 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 $\boldsymbol{\Sigma_i}$.. Based on this assumption, the unknown parameter vector $\boldsymbol{\beta}$ can be estimated by minimizing the squared Mahalanobis length of the residual vector $\widehat{\boldsymbol{\beta}}_{GLS} = \boldsymbol{argmin_\beta} \sum_{i=1}^{K} \sum_{j=1}^{N_i} \left(\boldsymbol{F_i^j} - \boldsymbol{X_i\beta}\right)^T \boldsymbol{\Sigma_i^{-1}} \left(\boldsymbol{F_i^j} - \boldsymbol{X_i\beta}\right)$. We obtain the generalized least squares estimate of $\boldsymbol{\beta}$ below:

$$\widehat{\boldsymbol{\beta}}_{GLS} = \left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T \Sigma_i^{-1} X_i}\right)^{-1} \left(\sum_{i=1}^{K} \sum_{j=1}^{N_i} \boldsymbol{X_i^T \Sigma_i^{-1} F_i^j}\right)$$

Furthermore, the sampling distribution of the generalized least squares estimate of $\boldsymbol{\beta}$ can be expressed as follows (DeGroot, 2004; Gill, 2008).

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

Since $\tilde{y}(x_M)$ is a linear combination of $\widetilde{\boldsymbol{\beta}}$, the sampling distribution of the expected mean performance, which is denoted as $\tilde{y}(x_M)_{GLS}$, is also a linear combination of $\widetilde{\boldsymbol{\beta}}_{GLS}$, thus it is also normally distributed:

$$\tilde{y}(x_M)_{GLS} \sim N\left[X_M^T \left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T \Sigma_i^{-1} X_i}\right)^{-1} \left(\sum_{i=1}^{K} \sum_{j=1}^{N_i} \boldsymbol{X_i^T \Sigma_i^{-1} F_i^j}\right), X_M^T \left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T \Sigma_i^{-1} X_i}\right)^{-1} X_M\right] \qquad (3.2)$$

In order to minimize the objective in (3.1), it is always better to exhaust the available computing budget (Brantley, Lee, Chen and Chen, 2011). Hence the inequality budget constraint in model (3.1) can be replaced by an equality constraint. Therefore the problem of minimizing the estimated variance can be modelled as the following generalized Least Squares (GLS) Model.

Generalized Least Squares (GLS) Model

$$\boldsymbol{obj}. \qquad \underset{K}{\boldsymbol{min}} \underset{l_1,l_2,\ldots,l_K,N_1,N_2,\ldots,N_K}{\boldsymbol{min}} X_M^T \left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T} \boldsymbol{\Sigma_i^{-1}} \boldsymbol{X_i} \right)^{-1} X_M \tag{3.3}$$

$$\boldsymbol{s}.\boldsymbol{t}. \qquad \sum_{i=1}^{K} l_i N_i = T$$

$$0 < l_i \leq T; \qquad l_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K$$

$$N_i > 0; \qquad N_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K$$

$$K > 0 \qquad K \in \mathbb{N}^0$$

We note that the estimated variance depends on the variance-covariance matrix of the simulation noise, as a result, the objective function in the GLS Model could be too complex to handle. In order to simplify the problem, we look at two special cases in which the simulation outputs are uncorrelated or homogeneous.

Under the special case that the simulation noise is uncorrelated, the variance-covariance matrix $\boldsymbol{\Sigma_i}$ is a diagonal matrix, whose inverse is also a diagonal matrix. We denote the inverse of $\boldsymbol{\Sigma_i}$ as $\boldsymbol{W_i}$, whose diagonal element e_{kk} is equal to $\frac{1}{\sigma_{x_k}^2}$, and $\sigma_{x_k}^2$ is the noise variance at the observation point x_k. Therefore, under this special case, the sampling distribution of the unknown parameter and the transient design performance at the observation point x_M can be expressed as

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

$$\tilde{y}(x_M)_{WLS} \sim N\left[X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i} \boldsymbol{X_i^T W_i F_i^j}\right), X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1} X_M\right] \tag{3.4}$$

In fact, the above expression can be derived by minimizing the weighted least squared error terms $\sum_{i=1}^{K}\sum_{j=1}^{N_i}\left(\boldsymbol{F_i^j} - \boldsymbol{X_i\beta}\right)^T \boldsymbol{W_{ii}}\left(\boldsymbol{F_i^j} - \boldsymbol{X_i\beta}\right)$, with $\boldsymbol{W_i}$ being the weight matrix. Hence when the simulation outputs are uncorrelated, the GLS Model, can be reformulated as the following Weighted Least Squares (WLS) Model.

Weighted Least Squares (WLS) Model

$$\boldsymbol{obj.} \qquad \underset{K}{\boldsymbol{min}} \underset{l_1,l_2,\dots,l_K,N_1,N_2,\dots,N_K}{\boldsymbol{min}} X_M^T\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T W_i X_i}\right)^{-1} X_M \tag{3.5}$$

$$\boldsymbol{s.t.} \qquad \sum_{i=1}^{K} l_i N_i = T$$

$$0 < l_i \leq T; \qquad l_i \in \mathbb{N}^0, \forall i = 1,2,\dots,K$$

$$N_i > 0; \qquad N_i \in \mathbb{N}^0, \forall i = 1,2,\dots,K$$

$$K > 0 \qquad K \in \mathbb{N}^0$$

Under the even more special case that the simulation noise is uncorrelated and homogeneous, the simulation noises at all observation points follow the same normal distribution with mean zero and variance σ_E^2. In practice, σ_E^2 is calculated as the unbiased estimator of the performance variance of the design. Based on this uncorrelated homogeneous simulation noise assumption, the sampling distribution of the unknown parameter and the design performance can be written as

$$\widetilde{\boldsymbol{\beta}}_{LS} \sim N\left[\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i} \boldsymbol{X_i^T F_i^j}\right), \sigma_E^2\left(\sum_{i=1}^{K} N_i \boldsymbol{X_i^T X_i}\right)^{-1}\right]$$

$$\tilde{y}(x_M)_{LS} \sim N\left[X_M^T\left(\sum_{i=1}^{K} N_i\boldsymbol{X_i^T X_i}\right)^{-1}\left(\sum_{i=1}^{K}\sum_{j=1}^{N_i}\boldsymbol{X_i^T F_i^j}\right), \sigma_E^2 X_M^T\left(\sum_{i=1}^{K} N_i\boldsymbol{X_i^T X_i}\right)^{-1} X_M\right] \tag{3.6}$$

We could obtain the same expression as above by minimizing the least squared error terms $\sum_{i=1}^{K}\sum_{j=1}^{N_i}\frac{1}{\sigma_E^2}\left(\boldsymbol{F_i^j}-\boldsymbol{X_i\beta}\right)^T\left(\boldsymbol{F_i^j}-\boldsymbol{X_i\beta}\right)$. Because σ_E^2 is a constant, minimizing $Var\left(\tilde{y}(x_M)\right)_{LS}=\sigma_E^2 X_M^T\left(\sum_{i=1}^{K}N_i\boldsymbol{X_i^T X_i}\right)^{-1}X_M$ 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 we will refer to as the Prediction Variance Factor (PVF) (Morrice, Brantley and Chen 2009). It is noted that in our thesis, this PVF might be of different forms, depending on the types of the feature functions comprising the underlying function. Under this uncorrelated and homogeneous noise assumption, the WLS Model can be further simplified into a Least Squares (LS) Model below.

Least Squares (LS) Model

$$\boldsymbol{obj}. \qquad \underset{K}{\boldsymbol{min}}\ \underset{l_1,l_2,\ldots,l_K,N_1,N_2,\ldots,N_K}{\boldsymbol{min}}\ X_M^T\left(\sum_{i=1}^{K}N_i\boldsymbol{X_i^T X_i}\right)^{-1}X_M \tag{3.7}$$

$$\begin{array}{lll}\boldsymbol{s.t.} & \sum_{i=1}^{K} l_i N_i = T & \\ & 0 < l_i \le T; & l_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K \\ & N_i > 0; & N_i \in \mathbb{N}^0, \forall i = 1,2,\ldots,K \\ & K > 0 & K \in \mathbb{N}^0\end{array}$$

Analytical solutions to the GLS Model and the WLS Model might not be available as solving these two models require us to have information on the variance-covariance matrix of simulation noise, which is usually unavailable. Nevertheless, analytical solutions to the LS Model might exist as the objective function is independent of the noise variance. Hereafter, we would solve the LS Model analytically when the underlying function takes certain functional form.

One of the main challenges of solving the LS Model is the excessive complexity of the objective function since the objective function could be nonlinear and could be very complex depending on the feature functions comprising the underlying functions. 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. In order to solve the problem, the integer constraints in the initial LS Model has been relaxed and the LS Model is reformulated in the following way.

Relaxed Least Squares (LS) Model

$$\boldsymbol{obj.} \qquad \boldsymbol{\min_{K}} \; \boldsymbol{\min_{\alpha_1,\alpha_2,\ldots,\alpha_K,l_1,l_2,\ldots,l_K}} \frac{1}{T} X_M^T \left(\sum_{i=1}^{K} \frac{\alpha_i}{l_i} \boldsymbol{X_i^T X_i} \right)^{-1} X_M \qquad (3.8)$$

$$\boldsymbol{s.t.} \qquad \sum_{i=1}^{K} \alpha_i = 1$$

$$0 < \alpha_i < 1 \qquad \forall i = 1,2,\ldots,K$$

$$l_i > 0 \qquad \forall i = 1,2,\ldots,K$$

$$K > 0 \qquad K \in \mathbb{N}^0$$

In the above Relaxed LS model, α_i is the proportion of computing budget allocated to simulation group G_i in which all the simulation replications have the same run length l_i, thus $\alpha_i = \frac{l_i N_i}{T}$, $i = 1,2,\ldots,K$. Furthermore, we assume that the transient design performance follows certain simple underlying functions, such as some simple polynomials including linear, full quadratic or full cubic polynomials. The Relaxed LS Model is different from the traditional c-optimal design model as the sequential constraint is imposed, thus the complexity of the problem increases significantly. In the literature of DOE, the simple polynomial models are of particular importance and interest due to their relative ease of derivation and wide application. We also provide some optimization results for trigonometric and logarithmic feature functions. These problems are solved numerically either using the Lipchitz-continuous Global Optimizer (LGO) embedded in AIMMS (Pinter, 1996) or by using the computing software such as the *Mathematica* for a limited number of feature functions in order to avoid an excessively complex objective function which cannot be handled by the software.

3.2. SOLUTIONS TO LEAST SQUARES MODEL

3.2.1. Lower Bound of Objective Function

We present in Lemma 1 that regardless of the types of the underlying functions the transient design performances follow, the objective function in the Relaxed LS Model is always lower bounded by $\frac{1}{\mathrm{T}}$.

Lemma 1 *If the optimal solution to the Relaxed LS Model exists, the objective function is lower bounded by* $\frac{1}{T}$. *In other words, regardless of the types of the feature functions included in the underlying function, the PVF is lower bounded by* $\frac{1}{T}$.

<u>Proof</u>

According to El-Krunz and Studden (1991), given a nonzero $n \times 1$ vector c and a $n \times n$ positive definite matrix R, if ξ_0 is a c-optimal design, $inf_{\xi_0} c^T M_R^{-1}(\xi) c \geq sup_{d^* \in \mathfrak{D}^*} (\boldsymbol{d}^{*T} c)^2$, where n is the number of parameters we want to estimate, R is the prior variance-covariance matrix of the parameter vector $\boldsymbol{\beta}$, and $M_R^{-1}(\xi)$ is the unity posterior variance-covariance matrix of $\boldsymbol{\beta}$. $\boldsymbol{d}^* = \left(1 + \frac{1}{T}\boldsymbol{d}^T R\boldsymbol{d}\right)^{-\frac{1}{2}}\boldsymbol{d}$, where $\boldsymbol{d}$ is a $n \times 1$ vector such that $|\boldsymbol{d}^T\boldsymbol{f}(\boldsymbol{x})| \leq 1$ for all x, with $\boldsymbol{f}(\boldsymbol{x}) = [\Phi_1(x), \Phi_2(x), \dots, \Phi_n(x)]^T$.

In our problem, $c = [\Phi_1(x_M), \Phi_2(x_M), \dots, \Phi_n(x_M)]^T$. As the total computing budget goes to infinity, $\frac{1}{T}\boldsymbol{d}^T R\boldsymbol{d} \to 0$, thus $\boldsymbol{d}^* \to \boldsymbol{d}$. Consequently, when the total computing budget goes to infinity, $\frac{1}{T}c^T M_R^{-1}(\xi)c$ is just the objective function in the Relaxed LS Model, and we can conclude that $sup_{d^*\in\mathfrak{D}^*}\,(\boldsymbol{d}^{*T}c)^2 \xrightarrow{T\to\infty} sup_{d^*\in\mathfrak{D}^*}\,(\boldsymbol{d}^T c)^2 = sup_{d^*\in\mathfrak{D}^*}\,(\boldsymbol{d}^T\boldsymbol{f}(\boldsymbol{M}))^2 = 1$, or $inf_{\xi_0}c^T M_R^{-1}(\xi)c \geq 1$, leading to the result that $\frac{1}{T}c^T M_R^{-1}(\xi)c \geq \frac{1}{T}$. Therefore, if the optimal solutions to the Relaxed LS Model exist, the minimum value the objective function can take is $\frac{1}{T}$. ■

When the objective function in the Relaxed LS Model obtains its minimum value $\frac{1}{T}$, all the T simulation outputs collected along the simulation replication could be considered as T simulation outputs collected at the point of interest by doing regression analysis.

Part of our problem is to determine the optimal number of different simulation groups we need such that we can achieve the minimum PVF, and this optimal number of simulation groups might vary as the types of feature functions comprising the underlying function differ. There might also exist multiple optimal solutions, as the objective function could be non-convex. In the case of multiple optimal solutions, we will focus our study on the optimal solutions with the minimum number of different simulation groups K, since simplicity is always appreciated when we apply the budget allocation rule. In particular, if for an underlying function model, the optimal solution can be obtained with $K = 1$, meaning that all simulation replications have the same run length, the objective function in the Relaxed LS Model can be expressed as a univariate function due to the equality budget constraint, with the variable being either the number of simulation replications or the simulation run length of each simulation replication. Therefore, the global minimum of the objective function can be obtained numerically by using computing software, regardless of the types of the feature functions included in the underlying function. In the case that the optimal solution cannot be obtained with $K = 1$, when the underlying function takes a certain form, one would need to use the LGO Solver to solve the problem numerically. In the following sections, we would determine the optimal solutions to the LS Model when the underlying function takes certain form.

3.2.2. Linear Underlying Function

In the case of linear underlying function, the transient mean performance of the design follows a linear function $f(x) = \beta_1 + \beta_2 x$. Based on Lemma 1, we present Lemma 2 in which one analytical solution to the Relaxed LS Model when the underlying function is a linear function is obtained.

Lemma 2 *When the underlying function is a linear function, the objective function in the Relaxed LS Model obtains its minimum value* $\frac{1}{T}$*, when all the simulation replications have the same run length* $2x_M$ *−1.*

<u>Proof</u>

We define PVF_{Linear}^{K} as the PVF derived from the linear underlying function with K different simulation groups. Hence the objective function in the Relaxed LS Model can be rewritten as $min_K \, min_{\alpha_1,\alpha_2,\ldots,\alpha_K,l_1,l_2,\ldots,l_K} \, PVF_{Linear}^{K}$.

From Lemma 1, we know that $min_K\, min_{\alpha_1,\alpha_2,\dots,\alpha_K,l_1,l_2,\dots,l_K}\, PVF^K_{Linear} \geq \frac{1}{T}$, resulting in that $min_{\alpha_1,\alpha_2,\dots,\alpha_K,l_1,l_2,\dots,l_K}\, PVF^K_{Linear} \geq \frac{1}{T},\ K = 1,2,\dots,\frac{n(n+1)}{2}$. Part of our problem is to find the minimum K such that the equality holds, thus we would study the problem by first considering the simplest case in which all the simulation replications have the same run length. When $K = 1$, we have

$$PVF^1_{Linear} = [1, x_M]\left(\frac{T}{l_1}\boldsymbol{X_1^T X_1}\right)^{-1}[1, x_M]^T = [1, x_M]\left(\frac{T}{l_1}\boldsymbol{X_1^T X_1}\right)^{-1}[1, x_M]^T$$

$$= [1, x_M]\left(\frac{1}{T}\begin{bmatrix} 1 & \frac{l_1+1}{2} \\ \frac{l_1+1}{2} & \frac{(l_1+1)(2l_1+1)}{6} \end{bmatrix}^{-1}\right)[1, x_M]^T$$

$$= \frac{6}{Tl_1(l_1+1)}\left(x_M - \frac{l_1+1}{2}\right)^2 + \frac{1}{T}$$

Therefore, when all the simulation replications have the same run length, the minimum PVF^1_{Linear} we could obtain is $\frac{1}{T}$, when $x_M = \frac{l_1+1}{2}$, or $l_1 = 2x_M - 1$. According to Lemma 1, the PVF for all types of underlying functions is lower bounded by $\frac{1}{T}$. In other words, $K = 1, l_1 = 2x_M - 1$ is an optimal solution to the Relaxed LS Model when the underlying function is a linear function. ■

In practice, based on our problem setting, the simulation run length and the number of simulation replications in each simulation group should be integers. By referring to the optimal solution obtained when the integer constraint is relaxed, we come up with the following computing budget allocation rule to deal with the discrete budget allocation in a real life application.

SDBA - Linear Underlying Function *Based on Lemma 2, When the underlying function follows a linear polynomial, we would run as many simulation replications as possible at run length* $l_1 = 2x_M - 1$*, and we would use the remaining simulation budget to run a single simulation replication at run length* $l_2 = T - l_1 N_1$*, where* $N_1 = \left\lfloor \frac{T}{l_1} \right\rfloor$*, and* $\lfloor x \rfloor$ *is the floor function.*

We have tested the above budget allocation rule by doing a simple numerical experiment. Suppose that we would like to predict the mean performance of the design at the point of interest $x_M = 30$. The transient design performance has an underlying function of $y(x) = 1 + x$ and the total computing budget T that varies from 1000 to 4000, in increments of 1000. The values of the PVF obtained under various budget T are presented in Table 3-1.

Table 3 - 1 Numerical Experiment for SDBA Rule for Linear Underlying Function

T	x_M	*Lower Bound of* $PVF = \frac{1}{T}$	*PVF Obtained Using the SDBA Rule*	l_1	l_2	N_1	N_2
1000	30	0.00100000	0.00100002	59	56	16	1
2000	30	0.00050000	0.00050001	59	53	33	1
3000	30	0.00033333	0.00033334	59	50	50	1
4000	30	0.00025000	0.00025000	59	47	67	1

From the table we observe that as T increases, the PVF is very close to the lower bound. Thus in practice, it would be efficient and convenient to run as many simulation replications at run length $l_1 = 2x_M - 1$ as possible, and use the remaining budget to run a single simulation at run length $l_2 = T - l_1 N_1$, where $N_1 = \left\lfloor \frac{T}{2x_M - 1} \right\rfloor$.

It is also noted that in order to achieve smaller PVF, it is better to run the simulations at a longer run length than the point of interest. Data collected beyond the point of interest are believed to help better define the overall shape of the underlying function as more information would always be helpful due to regression, resulting in a more accurate prediction at the point of interest.

3.2.3. Full Quadratic Underlying Function

In this case, we assume that the underlying function follows a full quadratic polynomial, namely, $y(x) = \beta_1 + \beta_2 x + \beta_3 x^2$. From Lemma 1, the minimum PVF we can achieve when the underlying function is a full quadratic polynomial is $\frac{1}{T}$, i.e.: $PVF_{Quadratic} \geq \frac{1}{T}$. By doing some simple calculation, it can be shown that when $K = 1$, the minimum PVF we could achieve is not $\frac{1}{T}$, hence the optimal number of simulation groups is at least two. When $K = 2$, if we could find l_1^*, l_2^*, α_1^* and α_2^* that make PVF equal to $\frac{1}{T}$, we could conclude that $K^* = 2$, l_1^*, l_2^*, α_1^* and α_2^* is an optimal solution to the LS Model. Otherwise, we can conclude that $K^* \geq 3$. In Lemma 3, we present an optimal solution to the Relaxed LS Model when the underlying function is a full quadratic polynomial.

Lemma 3 *When the underlying function is a full quadratic polynomial, the objective function in the Relaxed LS Model obtains its minimum value* $\frac{1}{T}$*, when* $K^* = 2$*,* $l_1^* \to 2x_M - 1$*,* $l_2^* \to +\infty$*,* $\alpha_1^* \to 1, \alpha_2^* \to 0$*, and* $\alpha_2^* = O\left(\frac{1}{l_2^2}\right)$*, where O(x) is a function such that* $lim_{x \to +\infty} \left|\frac{O(x)}{x}\right| = C$*, where C is a finite number.*

Proof

When $K = 2$, $l_1 \to 2x_M - 1$, $\alpha_2 = O\left(\frac{1}{l_2^2}\right) = \frac{C}{l_2^2}$, where C is a constant, by using the big O notation, the objective function in the Relaxed LS Model can be expressed as follows:

$$\frac{1}{T} X_M^T \left(\frac{\alpha_1}{l_1} \boldsymbol{X_1^T X_1} + \frac{\alpha_2}{l_2} \boldsymbol{X_2^T X_2}\right)^{-1} X_M = \frac{1}{T} \frac{9 l_2^{\frac{13}{2}} C^2 + O(l_2^6)}{9 l_2^{\frac{13}{2}} C^2 + O(l_2^6)}.$$

By making $l_2 \to +\infty$, $lim_{l_2 \to +\infty} \frac{1}{T} X_M^T \left(\frac{\alpha_1}{l_1} \boldsymbol{X_1^T X_1} + \frac{\alpha_2}{l_2} \boldsymbol{X_2^T X_2} \right)^{-1} X_M = \frac{1}{T}$. Since when $K^* = 2$, $l_1^* \to 2x_M - 1$, $l_2^* \to +\infty$, $\alpha_1^* \to 1, \alpha_2^* \to 0$ and $\alpha_2^* = O\left(\frac{1}{l_2^2}\right)$, the objective function in the Relaxed LS Model is equal to $\frac{1}{T}$, which is the minimum value it could take according to Lemma 1, we can conclude that $K^* = 2$, $l_1^* \to 2x_M - 1$, $l_2^* \to +\infty$, $\alpha_1^* \to 1, \alpha_2^* \to 0$, and $\alpha_2^* = O\left(\frac{1}{l_2^2}\right)$ is an optimal solution to the relaxed LS Model when the underlying function is a full quadratic. ■

Based on the analytical solution we obtained in the continuous case in Lemma 3, we present the following rule that deals with discrete computing budget allocation.

SDBA Rule - Full Quadratic Underlying Function *Based on Lemma 3, when the underlying function follows a full quadratic 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 by using computing software.*

We test the efficiency of the above budget allocation rule by doing a numerical experiment. The transient design performance has an underlying function of $y(x) = 1 + x + x^2$ and we would like to predict the design performance at the observation point x_M, with the total computing budget ranging from 1000 to 4000, in increments of 1000. The PVF obtained by using the above allocation rule is presented in Table 3-2. These results suggest that our computing budget allocation rule is able to give us a satisfactory outcome that is very close to the optimal solution.

Table 3 - 2 Numerical Experiment for SDBA Rule for Full Quadratic Underlying Function

T	x_M	***Lower Bound of*** $PVF = \frac{1}{T}$	***PVF Obtained Using the SDBA Rule***	***l₁***	***l₂***	***N₁***	***N₂***
1000	30	0.00100000	0.00114295	59	174	14	1
2000	30	0.00050000	0.00054597	59	230	30	1
3000	30	0.00033333	0.00035545	59	286	46	1
4000	30	0.00025000	0.00026332	59	283	63	1

3.2.4. Full Cubic Underlying Function

In this case the underlying function is assumed to be $y(x) = \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3$. Similar analysis as the full quadratic case has been done for this full cubic case and we present in Lemma 4 an optimal solution to the Relaxed LS Model when the underlying function is a full cubic polynomial.

Lemma 4 *When the underlying function is a full cubic polynomial, the objective function in the Relaxed LS Model obtains its minimum value* $\frac{1}{T}$, *when* $K^* = 2$, $l_1^* \to 2x_M - 1$, $l_2^* \to +\infty$, $\alpha_1^* \to 1, \alpha_2^* \to 0$, *and* $\alpha_2^* = O\left(\frac{1}{l_2^3}\right)$.

Proof

When $K = 2$, $l_1 \to 2x_M - 1$, $\alpha_2 = O\left(\frac{1}{l_2^3}\right) = \frac{D}{l_2^3}$, where *D* is a constant, the objective function in the Relaxed LS Model becomes

$$\frac{1}{T} X_M^T \left(\frac{\alpha_1}{l_1} \boldsymbol{X_1^T X_1} + \frac{\alpha_2}{l_2} \boldsymbol{X_2^T X_2}\right)^{-1} X_M = \frac{1}{T} \frac{16 l_2^{\frac{25}{2}} D^2 + O(l_2^{\frac{23}{2}})}{16 l_2^{\frac{25}{2}} D^2 - O(l_2^{12})}.$$

By making $l_2 \to +\infty$, $lim_{l_2 \to +\infty} \frac{1}{T} X_M^T \left(\frac{\alpha_1}{l_1} \boldsymbol{X_1^T X_1} + \frac{\alpha_2}{l_2} \boldsymbol{X_2^T X_2}\right)^{-1} X_M = \frac{1}{T}$. Since the objective function in the Relaxed LS Model is lower bounded by $\frac{1}{T}$ according to Lemma 1, we can conclude that $K^* = 2$, $l_1^* \to 2x_M - 1$, $l_2^* \to +\infty$, $\alpha_1^* \to 1, \alpha_2^* \to 0$ and $\alpha_2^* = O\left(\frac{1}{l_2^3}\right)$ is an optimal solution when the underlying function is a full cubic polynomial. ■

We present below the budget allocation rule based on the analytical solution obtained in Lemma 4.

SDBA Rule - Full Cubic Underlying Function *Based on Lemma 3, when the underlying function follows a 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 by using computing software.*

The efficiency of the above budget allocation rule has been confirmed by doing a numerical experiment in which the transient design performance follows the underlying function $y(x) = 1 + x + x^2 + x^3$, and we would like to estimate the transient design performance at the observation point $x_M = 30$, with the total computing budget varying from 1000 to 4000, in increments of 1000. The experiment result given in Table 3-3 reveals that using the SDBA Procedure is able to give us a close to optimal PVF.

Table 3 - 3 Numerical Experiment for SDBA Rule for Full Cubic Underlying Function

T	x_M	***Lower Bound of*** $PVF = \frac{1}{T}$	***PVF Obtained Using the SDBA Rule***	l_1	l_2	N_1	N_2
1000	30	0.00100000	0.00133471	59	292	14	1
2000	30	0.00050000	0.00059843	59	348	30	1
3000	30	0.00033333	0.00038198	59	404	46	1
4000	30	0.00025000	0.00027963	59	460	63	1

3.2.5. General Underlying Function

In this section, we look at the numerical solutions to some other simple underlying function models, obtained by solving the Relaxed LS Model. Due to the complexity of the objective function, analytical solutions to some of the underlying function models cannot be obtained. However, from Lemma 1, we know the minimum PVF we can achieve for all types of underlying functions is lower bounded by $\frac{1}{T}$. We determine the optimal number of simulation groups by studying the minimum PVF we achieve as K increases. Starting with $K = 1$, we stop the search for optimal K once the minimum PVF equals $\frac{1}{T}$. By doing so, the minimum number of simulation groups required to achieve the global minimum PVF for various types of underlying function are 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*
$\boldsymbol{\beta_1 + \beta_2 x}$	2	1	2
$\boldsymbol{\beta_1 + \beta_2 x^2}$	2	1	2
$\boldsymbol{\beta_1 + \beta_2 x^3}$	2	1	2
$\boldsymbol{\beta_1 + \beta_2 x + \beta_3 x^2}$	3	2	4
$\boldsymbol{\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3}$	4	2	4
$\boldsymbol{\beta_1 + \beta_2 ln(x)}$	2	1	2
$\boldsymbol{\beta_1 + \beta_2 sin\,(\frac{x}{50})}$	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 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 that 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. Based on this observation, we introduce the following SDBA Procedure for general underlying function.

SDBA Rule - 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.*

3.3. SDBA PROCEDURE AND NUMERICAL IMPLEMENTATION

3.3.1. SDBA Procedure

In this section, we would develop an efficient computing budget allocation algorithm that allows us to estimate accurately the transient performance of the design by doing regression, based on analytical and numerical results presented in Section 3. In practice, the underlying function of the design might be unknown and certain measures need to be taken to determine the best underlying function that captures the transient design performances.

SDBA Procedure

1. Conduct N_0 initial simulation replications at the run length $l=x_M$ and collect simulation outputs at all observation points along the simulation replication.
2. Average the simulation outputs at each observation point across replications.

3. Fit a regression model to the replication averages using adjusted R^2. The model that yield the highest R^2 is selected.
4. Calculate the simulation noise variance using the data collected in Step 1 at each observation point across replication and check for normality of the residuals.
5. If the normality test fails run an additional simulation replication at run length l and go to Step 2. Else
6. Determine the budget allocation strategy by solving the LS Model using the optimization solver or by doing numerical search. In the special case that the underlying function is a simple polynomial (linear, full quadratic or full cubic), apply the SDBA Rules developed in Section 3.3.

Remarks:

1. In Step 1, the initial run length of the simulation replications for the pilot runs is set to be x_M in the procedure presented above, which can be considered as a good choice when no additional information about the transient design performance is available. Nevertheless, a more sophisticated method such as determining the run length by assuming a certain underlying function can be applied, which might enable us to identify the best underlying function with less computing budget consumed during these pilot runs.
2. The value of N_0 should be small enough so that most of the computing budget is conserved for the simulation runs using the budget allocations scheme determined in Step 6. However, N_0 needs to be big enough to determine the best underlying function that captures the transient design performance, as well as an accurate description of the noise variance pattern.

In the next two sections, we present two numerical experimentations to test the efficiency of introducing run length optimization to the computing budget allocation and how we could use the SDBA Procedure to address real life problem.

3.3.2. Full Quadratic Underlying Function with Homogeneous Noise

In this numerical experimentation, we would like to test the efficiency of incorporating the concept of run length optimization to the determination of the efficient computing budget allocation strategy. 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 the sample mean at the point of interest is also used as a comparison procedure due to its wide application. We assume uncorrelated, homogeneous normal simulation noise along the simulation replication, with mean zero and variance one. The least squares formula that is used in the original Simple Regression Procedure, is used to calculate the design mean and variance during the simulation runs for all procedures. The results from a *MATLAB* simulation are presented in Figure 3-1. 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.

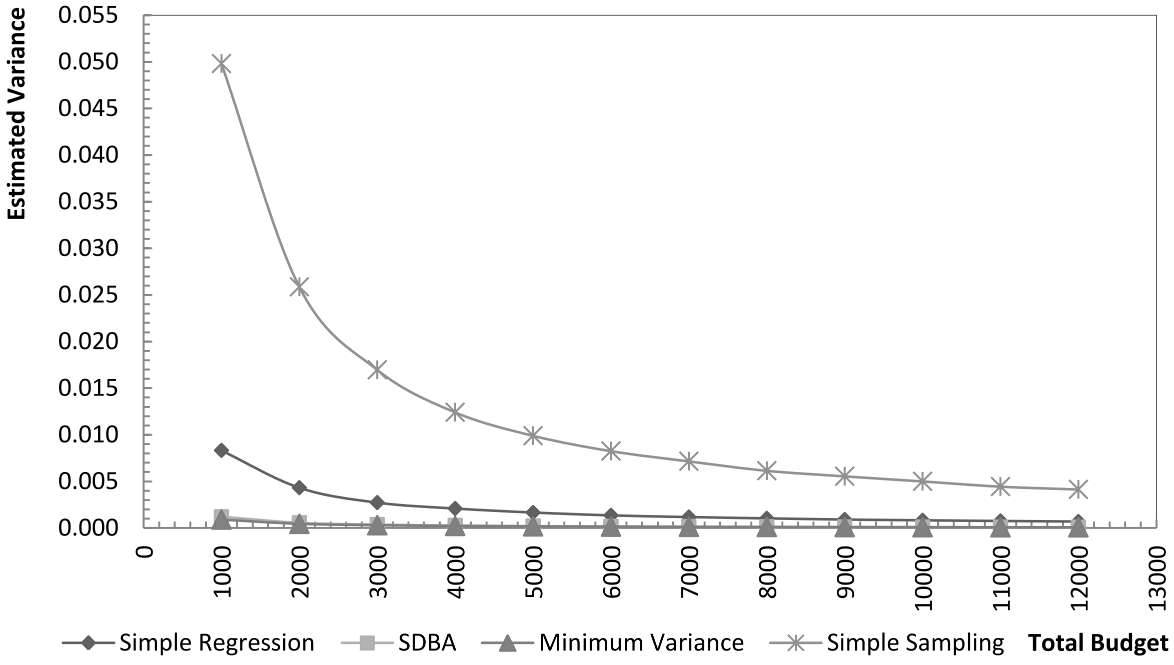

Figure 3 - 1 Comparison of Estimated Variance Obtained by Using Different Procedures with Full Quadratic Underlying Function

As illustrated in the diagram, given a certain amount of computing budget, using the regression procedures enables 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, Lemma 2 and Lemma 3 dictate that we run simulation replications at two different run lengths. In addition, one of these groups contains a single longer simulation replication. We now explore the impact of not using this single longer simulation run group for the SDBA Procedure. In Figure 3-2, we present the experiment results for the Simplified SDBA Procedure in which only a single simulation run length is used, under the same experiment setting as in Figure 3-1.

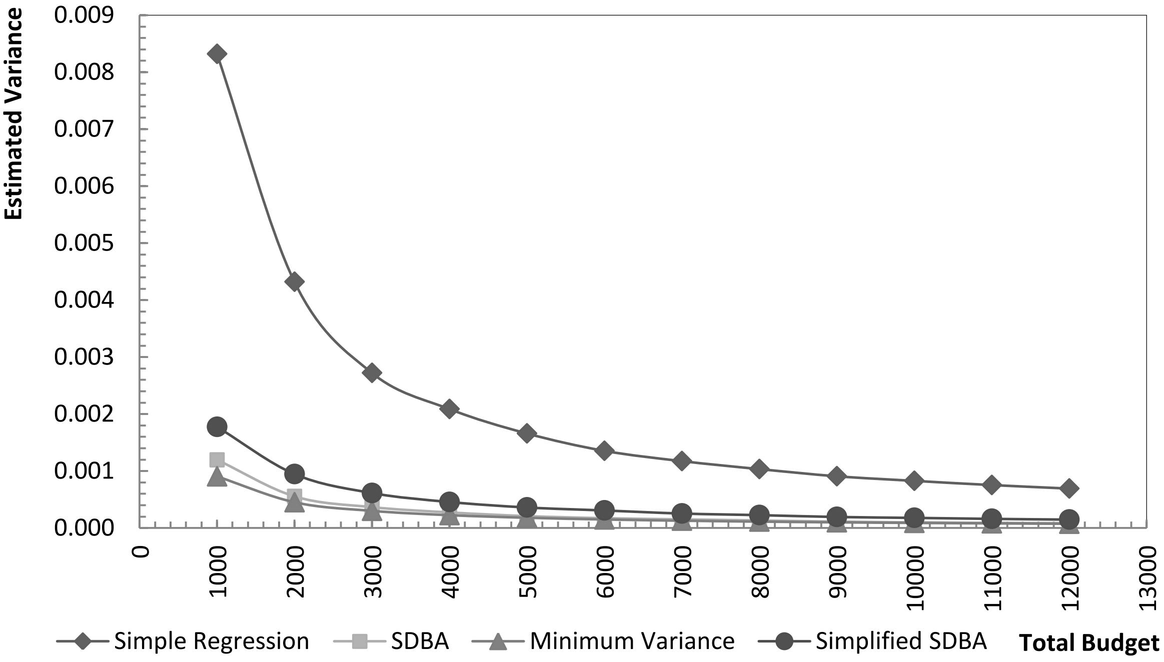

Figure 3 - 2 Numerical Experimentation Results for Simplified SDBA Procedure for Full Quadratic Underlying Function

Figure 3-2 illustrates 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 with a single run length is about twice the minimum PVF we get by using SDBA Procedure. Depending on whether or not this difference in performance is considered practically significant one might run all the simulation replications at the same run length due to the relative ease of implementation of the Simplified SDBA Procedure. Similar results can be obtained for the full cubic underlying function.

3.3.3. M/M/1 Queue with Heterogeneous Simulation Noise

It is noted that we assume uncorrelated and homogeneous simulation noise in the SDBA Procedure. However, in practice, these assumptions are often violated. In this section, we consider an implementation of the SDBA Procedure on a real life problem in which the simulation noise is correlated and heterogeneous.

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 set to be 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. The analytical value of the mean system waiting time of the 20th customer is known to be approximately 4.275 (Kelton and Law, 1985).

By running simulation and studying the average transient customer system waiting times during the pilot runs, 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. With a budget of 5000 for the pilot runs, this logarithm underlying function gives us $R^2 = 0.9623$ and the simulation noise follows approximately the normal distribution at all observation points. As we can see, the total computing budget consumed during the pilot runs is not very significant and yet is able to give us a pretty good estimation of the underlying function.

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. In fact, the simulation outputs are correlated and the simulation noise variance increases as the simulation run length increases. In this study, we present a Modified SDBA Procedure in which we approximate the noise variance using certain functional form. Different from the original SDBA Procedure in which the optimal run lengths are determined by solving the LS Model, in the Modified SDBA Procedure, the optimal run lengths are determined by solving the WLS Model by making use of the noise variance function. For example, in this M/M/1 queue problem, we approximate the noise variance by a linearly increasing function, namely,

$\varepsilon(x_i) \sim \mathcal{N}(0, ax_i + b)$, where a and b are real numbers, and $a \geq 0$. It is noted that this approximation may not be accurate. Nevertheless, it could 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.

A simpler way to get the budget allocation strategy is to apply the SDBA Procedure in which we assume uncorrelated and homogeneous noise, and numerically solve the LS. The SDBA Rules presented in the earlier part of the thesis might be applied when the underlying function follows certain forms.

In Table 3-5, we present the computing budget allocation strategies obtained by solving different models under different assumptions. The Simple Regression and the Simple Sampling Procedures are used as comparison procedures. It is noted that the run lengths obtained using the Modified SDBA Procedure and the SDBA Procedure are quite close to each other.

Table 3 - 5 Assumptions and Budget Allocation Strategy for Various Procedures and Approaches

Approach	**Modified SDBA Procedure**	**SDBA Procedure**	**Simple Regression Procedure**	**Simple Sampling Procedure**
Assumptions	Uncorrelated and linearly increasing noise variance.	Uncorrelated and homogeneous noise.	Uncorrelated and homogeneous noise.	N.A.
Budget Allocation Strategy	All the simulation replications would run up to the 50th customer entering the system.	All the simulation replications would run up to the 51th customer entering the system.	All the simulation replications would run up to the 20th customer entering the system.	All the simulation replications would run up to the 20th customer entering the system.

In Table 3-6, we present results on the prediction of the 20^{th} customer's system waiting time by running the simulation using different budget allocation strategies listed in Table 3-5.

Table 3 - 6 Numerical Experimentation Results for M/M/1 Queue Using Various Procedures

T	***Estimated Mean System Waiting Time of the 20th Customer***				***Estimated Variance of the System Waiting Time of the 20th Customer***			
	Modified SDBA	***SDBA***	***Simple Regression***	***Simple Sampling***	***Modified SDBA***	***SDBA***	***Simple Regression***	***Simple Sampling***
5000	4.31058	4.31615	3.94422	4.26865	0.00273	0.00274	0.00315	0.04996
10000	4.31587	4.32069	3.93716	4.27065	0.00138	0.00138	0.00158	0.02505
15000	4.32131	4.32848	3.94233	4.26853	0.00093	0.00093	0.00106	0.01668
20000	4.32429	4.33705	3.93910	4.27438	0.00070	0.00070	0.00079	0.01250
25000	4.32624	4.33238	3.94244	4.27471	0.00056	0.00056	0.00063	0.01000

We have also calculated the simulation bias and the Mean Squared Error (MSE) for the various procedures and they are illustrated in Table 3-7 and Table 3-8. As we can see, the Modified SDBA Procedure is able to achieve the best performance with the smallest MSE, and it also leads us to the conclusion that the approximation of linearly increasing noise variance helps enhance the estimation accuracy. The SDBA Procedure in which we assume homogeneous noise is slightly worse than the Modified SDBA Procedure, but it outperforms the Simple Regression Procedure and the Simple Sampling Procedure. Nevertheless, as the total computing budget consumed increases, the Simple Sampling Procedure would expect to achieve the smallest MSE as the procedure is unbiased.

Table 3 - 7 Simulation Bias and MSE for Different Procedures

T	***Bias***				***MSE***			
	Modified SDBA	***SDBA***	***Simple Regression***	***Simple Sampling***	***Modified SDBA***	***SDBA***	***Simple Regression***	***Simple Sampling***
5000	-0.03585	-0.04143	0.33050	0.00608	0.00401	0.00445	0.11238	0.04999
10000	-0.04114	-0.04596	0.33757	0.00407	0.00308	0.00349	0.11553	0.02507
15000	-0.04658	-0.05376	0.33240	0.00619	0.00310	0.00382	0.11154	0.01672
20000	-0.04956	-0.06232	0.33562	0.00034	0.00315	0.00458	0.11343	0.01250
25000	-0.05151	-0.05765	0.33229	0.00001	0.00321	0.00388	0.11105	0.01000

Table 3 - 8 Ratio of MSE between Various Procedures

T	***Percentage Improvement MSE***			
	Modified SDBA to Simple Sampling	***SDBA to Simple Sampling***	***Modified SDBA to Simple Regression***	***SDBA to Simple Regression***
5000	91.97%	91.09%	96.43%	96.04%
10000	87.73%	86.07%	97.34%	96.98%
15000	81.47%	77.17%	97.22%	96.58%
20000	74.78%	63.34%	97.22%	95.96%

25000	67.87%	61.19%	97.11%	96.50%

In the current SDBA Procedure and the Modified SDBA Procedure, we derive the sampling distribution of the design performance by applying the WLS formula. We have conducted similar study as the one presented in this section, in which the GLS formula or the LS formula have been used in lieu of the WLS formula. The experiment reveals that using the WLS formula would introduce the least bias and MSE, as compared to the GLS or the LS formula. Its advantage over the LS formula might be explained by the fact that the actual variance at various observation points have been used to predict design performance, thus the bias in prediction has been reduced. In theory the GLS formula should be favoured as no assumption has been made to model the simulation noise. However, applying the GLS formula requires estimation of the variance-covariance matrix, which can be often erroneous if the amount of data is not sufficient, thus its performance might not be guaranteed.

4. MULTIPLE DESIGNS BUDGET ALLOCATION

4.1. PROBLEM SETTING AND PROBLEM FORMULATION

4.1.1. Problem Setting

In this section, we would like to develop an efficient Ranking and Selection Procedure that select the best design among several alternative designs based on their transient mean performances at certain time or observation point, making use of the SDBA Procedure we developed in Chapter 3.

We assume that the number of designs is finite and we define the design space as $\Theta \equiv \{\theta_i, i = 1,2, \ldots, P\}$ where P is relatively small. The transient mean performance of each design is assumed to follow certain underlying function which is a sum of several one-to-one feature functions, and it can be expressed as $y_{\theta_i}(x) = \sum_{j=1}^{n_{\theta_i}} \beta_{\theta_i,j} \Phi_{\theta_i,j}(x)$, where $y_{\theta_i}(x)$ denotes the expected transient performance of design θ_i at observation point x. $\Phi_{\theta_i,j}(x)$ is a one dimensional one-to-one feature function of design θ_i, which can be either linear or non-linear. Without loss of generality, we assume that the first feature function in the underlying function of each design is constant, i.e. $\Phi_{\theta_i,j}(x) = 1$ $(\forall \theta_i \in \Theta)$. n_{θ_i} is the total number of feature functions comprising the underlying function of design θ_i. $\boldsymbol{\beta}_{\theta_i} = [\beta_{\theta_i,1}, \beta_{\theta_i,2} \ldots, \beta_{\theta_i,n}]^T$ is the unknown parameter vector for design θ_i which we want to estimate, whose sampling distribution can be determined by using the simulation outputs.

The total computing budget, which can be interpreted as the total number of simulation outputs we can collect, is distributed to each design to run several simulation replications which might not have the same run lengths. For example, suppose that for design θ_i , π_{θ_i} simulation replications would be conducted and by grouping those simulation replications with the same run length together, these π_{θ_i} simulation replications can be classified into Q_{θ_i} simulation groups that are denoted as $G_{\theta_i,q}, q = 1,2,\ldots,Q_{\theta_i}$. Each group $G_{\theta_i,q}$ would contain $n_{\theta_i,q}$ simulation replications of run length $l_{\theta_i,q}$.

The transient performance of design could be obtained by running simulation, and the relationship between the simulation output and the expected mean design performance is defined as $\boldsymbol{F}_{\theta_i,q,t} = \boldsymbol{Y}_{\theta_i,q} + \boldsymbol{\varepsilon}_{\theta_i,q}$, where $\boldsymbol{F}_{\theta_i,q,t} = \left[f_{\theta_i,t}(x_1), f_{\theta_i,t}(x_2), \ldots, f_{\theta_i,t}\left(x_{l_{\theta_i,q}}\right)\right]^T$ is the simulation output vector for the t^{th} $(t = 1,2,\ldots,n_{\theta_i,q})$ simulation replication in group $G_{\theta_i,q}$, and $f_{\theta_i,t}(x_p)$ is one simulation output collected from the t^{th}simulation replication in group $G_{\theta_i,q}$ at observation point x_p. The vector $\boldsymbol{Y}_{\theta_i,q} = \left[y_{\theta_i}(x_1), y_{\theta_i}(x_2), \ldots, y_{\theta_i}\left(x_{l_{\theta_i,q}}\right)\right]^T$ is the vector of the expected mean design performance for all simulation replications in group $G_{\theta_i,q}$ and $y_{\theta_i}(x_p)$ is the expected mean performance of the design at observation point x_p for design θ_i $(p = 1,2,\ldots,l_{\theta_i,q})$. Its value can be computed by using the parameter vector whose sampling distribution would be determined after running simulation. Finally, $\boldsymbol{\varepsilon}_{\theta_i,q} = \left[\varepsilon_{\theta_i}(x_1), \varepsilon_{\theta_i}(x_2), \ldots, \varepsilon_{\theta_i}\left(x_{l_{\theta_i,q}}\right)\right]^T$ is the simulation noise vector for all simulation replications in group $G_{\theta_i,q}$, which follows multivariate normal distribution $\mathcal{N}\left(0, \Sigma_{\theta_i,q}\right)$, where $\Sigma_{\theta_i,q}$ is the variance-covariance matrix.

Our target is to develop an efficient budget allocation rule to select the best design among all alternative designs. In other words, we would like to maximize the *Probability of Correct Selection* which would be denoted as P{CS}. Without loss of generality, we assume that the design with the minimum expected mean performance at the point of interest would be selected as the best design. Suppose that design θ_b is selected as the best design, the P{CS} is defined as $P\{CS\} = P\{design\ \theta_b\ is\ actually\ the\ best\ design\} = P\left\{\tilde{y}_{\theta_b}(x_M) < \tilde{y}_{\theta_i}(x_M), \forall i = 1,2,\ldots,K, i \neq b\right\}$, where $\tilde{y}_{\theta_i}(x_M)$ is the sampling distribution of the mean performance of design θ_i at the point of interest x_M, and $\tilde{y}_{\theta_b}(x_M)$ is the sampling distribution of the mean performance of the selected best design at x_M. Hence our problem can be formulated as a maximization problem in which we seek to determine the optimal run lengths of all simulation replications, which would maximize the probability of correct selection. The mathematical model is presented as follows.

$$\boldsymbol{obj}. \qquad \underset{\left\{l_{\theta_i,q}, n_{\theta_i,q} | q=1,2,\ldots,Q_{\theta_i}; i=1,2,\ldots,K\right\}}{max} P\{CS\} \tag{4.1}$$

$$\boldsymbol{s.t.} \qquad \sum_{i=1}^{K} \sum_{q=1}^{Q_{\theta_i}} l_{\theta_i,q} n_{\theta_i,q} = T$$

$$0 \le l_{\theta_i,q} \le T; \qquad l_{\theta_i,q} \in \mathbb{N}^0 \, ; \forall q = 1,2,\ldots,Q_{\theta_i}; \forall i = 1,2,\ldots,P$$

$$n_{\theta_i,q} \ge 0; \qquad n_{\theta_i,q} \in \mathbb{N}^0 \, ; \forall q = 1,2,\ldots,Q_{\theta_i}; \forall i = 1,2,\ldots,P$$

4.1.2. Sampling distribution of Design Performance

In order to obtain the expression for P{CS}, we need to derive the sampling distribution of the transient performances of all designs at x_M. Let $\boldsymbol{X}_{\theta_i,q}$ denote the $l_{\theta_i,q} \times n_{\theta_i}$ matrix of the feature function matrix for the simulation replications in group $G_{\theta_i,q}$, and it is expressed as $\boldsymbol{X}_{\theta_i,q} = \left[X_{\theta_i,1}, X_{\theta_i,2}, \ldots, X_{\theta_i,l_{\theta_i,q}}\right]^T$, where $X_{\theta_i,i}$ is a $1 \times n_{\theta_i}$ feature function vector at simulation run length x_i for design θ_i , and it is expressed as $X_{\theta_i,i} = [\Phi_{\theta_i,1}(x_i), \Phi_{\theta_i,2}(x_i), \ldots, \Phi_{\theta_i,n}(x_i)]$.

We assume that the vector $\boldsymbol{F}_{\theta_i,q,t}$ follows a multi-variant normal distribution with mean $\boldsymbol{X}_{\theta_i,q}\boldsymbol{\beta}_{\theta_i}$ and covariance matrix $\Sigma_{\theta_i,q}$. In order to simplify the problem, we assume that the simulation outputs are uncorrelated and homogeneous and e can derive the sampling distribution of the transient performance of design θ_i at x_M. Let δ_{θ_i}be the estimated mean performance of the design θ_i at x_M and let $\sigma_{\theta_i}^2$ be its estimated variance, we have the following expressions in which $\sigma^2_{\theta_i,E}$ is the unbiased estimator of the performance variance of design θ_i.

$$\delta_{\theta_i} = X^T_{\theta_i,M}\left(\sum_{q=1}^{Q_{\theta_i}} n_{\theta_i,q}\boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q}\boldsymbol{X}_{\theta_i,q}\right)^{-1}\left(\sum_{q=1}^{Q_{\theta_i}}\sum_{t=1}^{n_{\theta_i,q}}\boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q}\boldsymbol{F}_{\theta_i,t}\right) \tag{4.2}$$

$$\sigma^2_{\theta_i} = \sigma^2_{\theta_i,E}X^T_{\theta_i,M}\left(\sum_{q=1}^{Q_{\theta_i}} n_{\theta_i,q}\boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q}\boldsymbol{X}_{\theta_i,q}\right)^{-1} X_{\theta_i,M} \tag{4.3}$$

4.1.3. Rate Function and Model Formulation

Let's define P{IS} as the probability of incorrect selection, or $P\{IS\} = 1 - P\{CS\} = P\left\{\tilde{y}_{\theta_b}(x_M) \geq min_{i=1,2,\ldots,K;i\neq b}\left(\tilde{y}_{\theta_i}(x_M)\right)\right\}$. It is noted that P{IS} and P{CS} have the same rate of convergence.

Let's denote E_{θ_i} as the event that $\tilde{y}_{\theta_b}(x_M) \geq \tilde{y}_{\theta_i}(x_M)$, $i = 1,2,\ldots K$ and $i \neq b$. Hence $\tilde{y}_{\theta_b}(x_M) \geq min_{i=1,2,\ldots,K;i\neq b}\left(\tilde{y}_{\theta_i}(x_M)\right)$ is the union of all E_{θ_i}, i.e.: $P\left\{\tilde{y}_{\theta_b}(x_M) \geq min_{i=1,2,\ldots,K;i\neq b}\left(\tilde{y}_{\theta_i}(x_M)\right)\right\} = P\{\bigcup_{i=1;i\neq b}^{K} E_{\theta_i}\}$. Obviously, $E_{\theta_i} \in \bigcup_{i=1;i\neq b}^{K} E_{\theta_i}$, we have $P\{\bigcup_{i=1;i\neq b}^{K} E_{\theta_i}\} \geq P\{E_{\theta_i}\}$, $\forall i = 1,2,\ldots P; i \neq b$, thus $P\{\bigcup_{i=1;i\neq b}^{K} E_{\theta_i}\} \geq max_{i=1,i\neq b} P\{E_{\theta_i}\}$. By applying the Bonferroni inequality (Bratley, Fox and Schrage, 1987; Chick, 1997; Law, 2007), P{IS} is upper bounded by $\sum_{i=1,i\neq b}^{K} P\{E_{\theta_i}\} \leq (P-1)\, max_{i=1,i\neq b} P\{E_{\theta_i}\}$. Therefore, the following inequality holds.

$$P\{E_{\theta_i}\} \leq P\{IS\} \leq (P-1)\max_{i=1,i\neq b} P\{E_{\theta_i}\} \tag{4.4}$$

Inequality (4.4) implies that the P{IS} would have the same convergence rate as $max_{i=1,i\neq b} P\{E_{\theta_i}\} = max_{i=1,i\neq b} P\{\tilde{y}_{\theta_b}(x_M) \geq \tilde{y}_{\theta_i}(x_M)\}$.

As $\tilde{y}_{\theta_i}(x_{\theta_i,M}) \sim \mathcal{N}(\delta_{\theta_i}, \sigma^2_{\theta_i})$ follows normal distribution, $\tilde{y}_{\theta_b}(x_{\theta_i,M}) - \tilde{y}_{\theta_i}(x_{\theta_i,M})$ is also normally distributed, and $\tilde{y}_{\theta_b}(x_{\theta_i,M}) - \tilde{y}_{\theta_i}(x_{\theta_i,M}) \sim \mathcal{N}(\delta_{\theta_b,\theta_i}, \sigma^2_{\theta_b,\theta_i})$, where $\delta_{\theta_b,\theta_i} = \delta_{\theta_b} - \delta_{\theta_i}$ and $\sigma^2_{\theta_b,\theta_i} = \sigma^2_{\theta_b} + \sigma^2_{\theta_i}$. Moreover, according to Glynn and Juneja (2004), the rate function of P{IS} is given by $min_{\theta_i} \frac{\delta^2_{\theta_b,\theta_i}}{2(\sigma^2_{\theta_b,\theta_i})}$.

Let's define $\alpha_{\theta_i,q}$ as the proportion of total computing budget allocated to the group $G_{\theta_i,q}$, namely $\alpha_{\theta_i,q} = \frac{l_{\theta_i,q} n_{\theta_i,q}}{T}$, and $\sum_{i=1}^{K} \sum_{q=1}^{Q_{\theta_i}} \alpha_{\theta_i,q} = 1$. Let's define $\alpha_{\theta_i} = \sum_{q=1}^{Q_{\theta_i}} \alpha_{\theta_i,q}$, which is the proportion of total computing budget allocated to design θ_i, and we have $\sum_{i=1}^{K} \alpha_{\theta_i} = 1$. Let's further define $\rho_{\theta_i,q} = \frac{\alpha_{\theta_i,q}}{\alpha_{\theta_i}}$, which is the proportion of the computing budget allocated to θ_i that has been consumed by group $G_{\theta_i,q}$, and we have $\sum_{q=1}^{Q_{\theta_i}} \rho_{\theta_i,q} = 1$. Based on the new definition, $\sigma^2_{\theta_i}$ can be rewritten as

$$\sigma^2_{\theta_i} = \sigma^2_{\theta_i,E} X^T_{\theta_i,M} \left(\sum_{q=1}^{Q_{\theta_i}} \frac{T\alpha_{\theta_i,q}}{l_{\theta_i,q}} \boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q} \boldsymbol{X}_{\theta_i,q} \right)^{-1} X_{\theta_i,M} = \frac{\sigma^2_{\theta_i,E}}{T} X^T_{\theta_i,M} \left(\sum_{q=1}^{Q_{\theta_i}} \frac{\alpha_{\theta_i}\rho_{\theta_i,q}}{l_{\theta_i,q}} \boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q} \boldsymbol{X}_{\theta_i,q} \right)^{-1} X_{\theta_i,M}$$

$$= \frac{\sigma^2_{\theta_i,E}}{\alpha_{\theta_i}T} X^T_{\theta_i,M} \left(\sum_{q=1}^{Q_{\theta_i}} \frac{\rho_{\theta_i,q}}{l_{\theta_i,q}} \boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q} \boldsymbol{X}_{\theta_i,q} \right)^{-1} X_{\theta_i,M}$$

Let $V_{\theta_i} = \sigma^2_{\theta_i,E} X^T_{\theta_i,M} \left(\sum_{q=1}^{Q_{\theta_i}} \frac{\rho_{\theta_i,q}}{l_{\theta_i,q}} \boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q} \boldsymbol{X}_{\theta_i,q} \right)^{-1} X_{\theta_i,M}$, we have $\sigma^2_{\theta_i} = \frac{V_{\theta_i}}{\alpha_{\theta_i}T}$, and $\sigma^2_{\theta_b,\theta_i} = \frac{V_{\theta_b}}{\alpha_{\theta_b}T} + \frac{V_{\theta_i}}{\alpha_{\theta_i}T}$.

Our initial problem, which is the maximization of the P{CS} or the minimization of the P{IS}, can be solved equivalently by maximizing the convergence rate of P{CS} or P{IS}. As a result, our problem can be formulated into the following model.

$$\boldsymbol{obj}.\qquad \underset{\left\{\alpha_{\theta_i},\rho_{\theta_i,q},l_{\theta_i,q}|q=1,2,\dots,Q_{\theta_i};i=1,2,\dots,K\right\}}{max}\ \underset{\theta_i}{min}\frac{\delta^2_{\theta_b,\theta_i}}{2\left(\frac{V_{\theta_b}}{\alpha_{\theta_b}T}+\frac{V_{\theta_i}}{\alpha_{\theta_i}T}\right)} \tag{4.5}$$

$$\begin{aligned}
\boldsymbol{s.t.}\qquad & \sum_{i=1}^{K}\alpha_{\theta_i}=1 \\
& \sum_{q=1}^{Q_{\theta_i}}\rho_{\theta_i,q}=1 && \forall i=1,2,\dots,K \\
& 0\le\alpha_{\theta_i}\le 1; && \alpha_{\theta_i}\in R;\forall i=1,2,\dots,K \\
& 0<l_{\theta_i,q}\le\alpha_{\theta_i}T; && l_{\theta_i,q}\in\mathbb{N}^0\,;\forall q=1,2,\dots,Q_{\theta_i};\forall i=1,2,\dots,K \\
& 0<\rho_{\theta_i,q}\le 1; && \rho_{\theta_i,q}\in R\,;\forall q=1,2,\dots,Q_{\theta_i};\forall i=1,2,\dots,K
\end{aligned}$$

4.2. PROBLEM SOLUTION

The complexity of the objective function in model (4.5) could be very significant due to the fact that we estimated the design performances by using the regression approach. To simplify the problem, we would adopt the decomposition technique to find the optimal solution when certain condition is met.

4.2.1. Condition for Decomposition

Assume that $\left\{\alpha^*_{\theta_i},\rho^*_{\theta_i,q},l^*_{\theta_i,q}|q=1,2,\dots,Q_{\theta_i};i=1,2,\dots,K\right\}$ is one of the optimal solutions to model (4.5). It is noted that $V^*_{\theta_i}=\sigma^2_{\theta_i,E}X^T_{\theta_i,M}\left(\sum_{q=1}^{Q_{\theta_i}}\frac{\rho^*_{\theta_i,q}}{l^*_{\theta_i,q}}\boldsymbol{X}^{\boldsymbol{T}}_{\theta_i,q}\boldsymbol{X}_{\theta_i,q}\right)^{-1}X_{\theta_i,M}$ depends on $\rho^*_{\theta_i,q}$ and $l^*_{\theta_i,q}$, and it might or might not depend on $\alpha^*_{\theta_i}$. If $V^*_{\theta_i}$ is independent of $\alpha^*_{\theta_i}$, $\alpha^*_{\theta_i}$ can be determined by solving the following optimization problem.

$$\boldsymbol{obj}.\qquad \underset{\left\{\alpha_{\theta_i}|i=1,2,\dots,K\right\}}{max}\ \underset{\theta_i}{min}\frac{\delta^2_{\theta_b,\theta_i}}{2\left(\frac{V^*_{\theta_b}}{\alpha_{\theta_b}T}+\frac{V^*_{\theta_i}}{\alpha_{\theta_i}T}\right)} \tag{4.6}$$

$$\boldsymbol{s.t.} \qquad \sum_{i=1}^{K} \alpha_{\theta_i} = 1$$

$$0 \le \alpha_{\theta_i} \le 1; \qquad \alpha_{\theta_i} \in R; \forall i = 1,2,\dots,K$$

Problem (4.6) is in fact a special case of problem (4.5), in which the values of $\{\rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\}$ have been pre-determined to be $\{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\}$. Indeed, this problem is almost the same as the problem of determining the optimal computing budget allocation rule among multiple designs whose performance variances are fixed, and the OCBA Procedure has been developed as a result (Chen, 1995). As a result, model (4.5) can be solved by first determining the values of $\{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\}$, followed by determining value of $\{\alpha^*_{\theta_i} | i = 1,2,\dots,K\}$ using the OCBA Rule.

When $V^*_{\theta_i}$ is not independent of $\alpha^*_{\theta_i}$, the above problem decomposition cannot be done. In the following section, we present in detail how we could decompose the problem when $V^*_{\theta_i}$ is independent of $\{\alpha^*_{\theta_i} | i = 1,2,\dots,K\}$ and how we determine the value of $\{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\}$.

4.2.2. Problem Decomposition

When V_{θ_i} is independent of α_{θ_i}, we would prove in Lemma 5 that $\{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\}$ can be determined by using the SDBA Procedure.

Lemma 5 *If the optimal computing budget allocation for any single design is independent of the computing budget allocated to that design, the Ranking and Selection Problem can be decomposed into two sub-problems, i.e.: the problem of optimal computing budget allocation among multiple designs and the optimal computing budget for a single design, which can be solved by applying the OCBA Rule and the SDBA Procedure respectively.*

Proof

If we use the SDBA Procedure to estimate the design performances, the values of $\left\{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\right\}$ can be determined by solving the mathematical model below.

$$\boldsymbol{obj}. \qquad \min_{\left\{\rho_{\theta_i,q}, l_{\theta_i,q} | q=1,2,\dots,Q_{\theta_i}\right\}} V_{\theta_i} \tag{4.7}$$

$$\boldsymbol{s.t.} \qquad \sum_{q=1}^{Q_{\theta_i}} \rho_{\theta_i,q} = 1$$

$$0 \le l_{\theta_i,q} \le \alpha_{\theta_i} T; \qquad l_{\theta_i,q} \in \mathbb{N}^0 \,; \forall q = 1,2,\dots,Q_{\theta_i}$$

$$0 \le \rho_{\theta_i,q} \le 1; \qquad \rho_{\theta_i,q} \in R \,; \forall q = 1,2,\dots,Q_{\theta_i}$$

In problem (4.7), we aim at minimizing V_{θ_i}, which is equivalent of minimizing $\frac{V_{\theta_i}}{\alpha_{\theta_i}T}$, the estimated variance of design θ_i, as $\alpha_{\theta_i}T$ is considered as a constant in the problem of optimal budget allocation for a single design. Let $\left\{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\right\}$ be the optimal solution to problem (4.7) when $\alpha_{\theta_i} = \alpha^*_{\theta_i}$. If $\left\{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\right\}$ is independent of $\alpha^*_{\theta_i}$, meaning that V'_{θ_i} is independent of $\alpha^*_{\theta_i}$, we would show that $\left\{\alpha^*_{\theta_i}, \rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}; i = 1,2,\dots,K\right\}$ is also the optimal solution to problem (4.5).

As the total amount of computing budget consumed increases, our estimation of the transient performances of the designs becomes more accurate and we could consider $\delta^2_{\theta_b,\theta_i}$ as a constant. Since $\left\{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2,\dots,Q_{\theta_i}\right\}$ minimizes $V_{\theta_i}(\forall i = 1,2,\dots,K)$ when $\alpha_{\theta_i} = \alpha^*_{\theta_i}$, we have $V^*_{\theta_i} \ge V'_{\theta_i}$, resulting in the inequality that

$$\frac{\delta^2_{\theta_b,\theta_i}}{2\left(\frac{V'_{\theta_b}}{\alpha^*_{\theta_b}T} + \frac{V'_{\theta_i}}{\alpha^*_{\theta_i}T}\right)} \ge \frac{\delta^2_{\theta_b,\theta_i}}{2\left(\frac{V^*_{\theta_b}}{\alpha^*_{\theta_b}T} + \frac{V^*_{\theta_i}}{\alpha^*_{\theta_i}T}\right)}; \ \forall i = 1,2,\dots,K; i \ne b$$

The above inequality implies that the convergence rate for each $P\{\tilde{y}_{\theta_b}(x_M) \geq \tilde{y}_{\theta_i}(x_M)\}$ when $\{\alpha_{\theta_i}, \rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\} = \{\alpha^*_{\theta_i}, \rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$ is at least as fast as that when $\{\alpha_{\theta_i}, \rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\} = \{\alpha^*_{\theta_i}, \rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$. Since $\{\alpha^*_{\theta_i}, \rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$ is the optimal solution to problem (4.5), $\{\alpha^*_{\theta_i}, \rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$ is also the optimal solution to problem (4.5).

$\alpha^*_{\theta_i}$ is obtained by applying the OCBA Rule with $\{\rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\} = \{\rho^*_{\theta_i,q}, l^*_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\}$. Since $\{\alpha^*_{\theta_i}, \rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$ is also an optimal solution to problem (4.5), by making $\{\rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\} = \{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\}$, we could obtain α'_{θ_i} by using the OCBA Rule, and $\{\alpha'_{\theta_i}, \rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}; i = 1,2, \dots, K\}$ would also be an optimal solution to problem (4.5).

Therefore, problem (4.5) can be solved by first determining the values of $\{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\}$ using the SDBA Procedure, followed by determining the value of $\{\alpha'_{\theta_i} | i = 1,2, \dots, K\}$ using the OCBA Rule with $\{\rho_{\theta_i,q}, l_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\} = \{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\}$, under the condition that the value of $\{\rho'_{\theta_i,q}, l'_{\theta_i,q} | q = 1,2, \dots, Q_{\theta_i}\}$ is independent of the amount of computing budget allocated to each single design. In other words, when the optimal computing budget allocation for each single design is independent of the computing budget allocated to them, the problem of maximization of the convergence rate can be decomposed and solved by first determining the optimal computing budget allocation strategy for each single design using the SDBA Procedure, followed by the optimal computing budget allocation among multiple designs using the OCBA Rule. ■

4.3. SDBA+OCBA PROCEDURE AND NUMERICAL IMPLEMENTATION

4.3.1. SDAB+OCBA Procedure

In this section we would develop the SDBA+OCBA Procedure to select the best design among all the K alternative designs by comparing their estimated mean performances at simulation run length x_M , when the optimal computing budget for a single design is independent of the total budget allocated to that design.

The SDBA Design Screening will be conducted before we apply the SDBA+OCBA Procedure to distribute the computing budget among the alternative designs. This is to ensure that SDBA Procedure can be applied to the alternative designs without violating the necessary assumptions for SDBA Procedure. T_0 simulation budget would be allocated to each design to run several simulation replications, and the simulation outputs at all observation points would be recorded. As we have seen in Chapter 3, the underlying function of the transient mean performance of design would be identified by doing curve fitting based on the recorded simulation outputs. Sometimes we might need to approximate the transient mean performance of design with certain underlying function. Moreover, the correlation test should be conducted on the simulation outputs to test whether the uncorrelated simulation output assumption still holds, and the assumption of homogeneous normal simulation noise at all observation points would be investigated. We would apply the SDBA Procedure to those designs which pass all the tests during the SDBA Design Screening. For the rest designs, we would use statistical sampling to estimate their means performances at the point of interest.

During each round of budget allocation, an incremental computing budget, Δ in total, would be distributed to each design based on OCBA Procedure, and the estimated mean and variance for each design would be updated accordingly based on the SDBA Design Screening results. The procedure would stop when we have exhausted all available computing budget T.

SDBA+OCBA Procedure

```
INPUT           $x_M, T, \Delta, T_0$
INITIALIZE      $l \leftarrow 0$;

                Perform the SDBA Design Screen for all designs;

                $T_1^l = T_2^l = \cdots = T_P^l = T_0$;

                $dT_i^l = T_i^l$.
LOOP            WHILE $\sum_{i=1}^{P} T_i^l < T$ DO
    UPDATE      Calculate estimated means and variances of design performances by using
                either the SDBA Procedure or Simple Sampling approach
    ALLOCATE    Increase the computing budget by $\Delta$ and calculate the new budget allocation,
                $T_1^{l+1}, T_2^{l+1}, \dots, T_P^{l+1}$ according to

                    1) $\frac{T_i^{l+1}}{T_j^{l+1}} = \frac{\delta_{\theta_b,\theta_j}^2 \sigma_{\theta_i}^2}{\delta_{\theta_b,\theta_i}^2 \sigma_{\theta_j}^2}$

                    2) $T_b^{l+1} = \sqrt{\sum_{i=1, i \neq b}^{K} \frac{\sigma_{\theta_b}^2}{\sigma_{\theta_i}^2} (T_i^{l+1})^2}$

    SIMULATE    $dT_i^{l+1} = T_i^{l+1} - T_i^l$;

                Run simulations by using the SDBA Procedure or Simple Sampling
                Procedure based on the Design Screening result, with computing budget
                $max\ (dT_i^{l+1}, 0)$ for design $\theta_i, i = 1,2,3 \dots P$;

                $l \leftarrow l + 1$.
END OF LOOP
```

4.3.2. Application of SDBA+OCBA Procedure

According to the SDBA Procedure, when the underlying function of the design transient performance consists of only one non-constant feature functions, we could achieve the minimum estimated variance by running all the simulation replications at the same run length. In this special case when all the simulation replications have the same run length, expression (4.3) can be simplified as

$$\sigma_{\theta_i}^2 = \frac{\sigma_{\theta_i,E}^2}{\alpha_{\theta_i} T} X_{\theta_i,M}^T \left(\frac{1}{l_{\theta_i,1}} \boldsymbol{X}_{\theta_i,1}^{\boldsymbol{T}} \boldsymbol{X}_{\theta_i,1} \right)^{-1} X_{\theta_i,M} \qquad (4.8)$$

The study on the optimal computing budget allocation suggests that the estimated variance in expression (4.8) can be minimized when $l_{\theta_i,1}$ takes some value at which we achieve a balance between the impact of increasing the number of simulation replications and the impact of running simulation at a longer run length. Both tactics would result in variance reduction but cannot be achieved at the same time due to the budget constraint. In other words, when T is sufficiently large, the optimal run length $l_{\theta_i,1}^*$ would take some finite value that is independent of T, which leads us to the conclusion that if the transient performances of all designs follow certain underlying functions that consist of only one non-constant feature functions, the SDBA+OCBA Procedure could be applied to optimally allocate computing budget among all these designs, in order to select the best design by using the least computing budget.

4.3.3. Ranking and Selection of the Best M/M/1 Queuing System

In this section, we present a numerical experimentation of the SDBA+OCBA Procedure in which the efficiency of the procedure has been examined in comparison with the other existing Ranking and Selection procedures. The original OCBA Procedure in which the mean and variance of the performance of design is calculated as sample mean and sample variance would be used as the comparison method. The heuristic procedure proposed by Morrice, Brantley and Chen (2009) would also be used as a comparison procedure, which would be referred to as the Simple Regression+OCBA Procedure, in which the computing budget would be allocated among all designs according to OCBA Procedure, and the budget allocated to each design would be used to run several simulations until to the point of interest and simulation outputs would be collected along the simulation replications and used to estimate the design performance by doing regression.

In this experiment, we have five M/M/1 queuing systems having the following traffic intensities: 0.9, 0.95, 1, 1,05 and 1.1. We would like to select the queuing system that has the shortest system waiting time (waiting time in the queue + service time) for the 20th customer joining the queue. All five queuing systems are initially empty with the servers being idle. The customer system waiting time is generated by running the simulation in MATLAB. The logarithm underlying function of the form $\beta_1 + \beta_2 ln\ (n)$ has been used to approximate the transient system waiting time of the n^{th} customers joining the queue. The weighted least squares formula has been used to compute the design performances due to heteroscedasticity. Moreover, the optimal run length for each single design is determined by assuming linearly increasing simulation noises along the simulation replication, since in practice, as the simulation run length increases, the uncertainty in prediction decreases, leading to a higher simulation noise at a longer run length. In Figure 4-1, we compare the efficiency of the aforementioned procedures on the selection of the best M/M/1 queuing system under the above experiment setting.

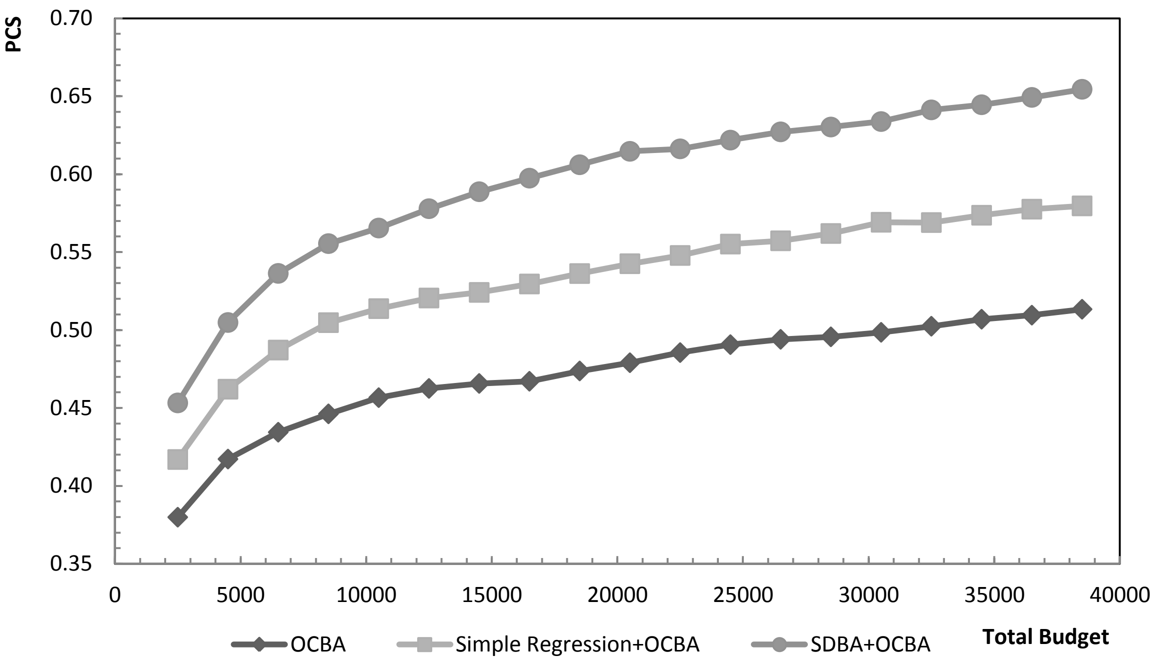

Figure 4 - 1 Comparisons of the performances of various computing budget allocation rule on the selection of the best M/M/1 queuing system

The experiment results reveal that the performance of the OCBA Procedure can be improved by incorporating the regression approach which is able to provide more accurate estimation of design performances. Moreover, the SDBA+OCBA Procedure outperforms the other two procedures and enables us to achieve the same probability of correction by using far less computing budget.

4.3.4. Ranking and Selection of the Best Full Quadratic Design

The SDBA Procedure suggests that when the underlying function of the design performance follows a full quadratic polynomial, in order to obtain the minimum variance, we need to run simulations at two different run lengths. Nevertheless, we would run most simulation replications at the first run length and we would run a single simulation replication at the second run length. Numerical experimentation has shown that the Simplified SDBA Procedure in which all simulation replications have the same run length, is able to provide us with very good estimation of design performances, though slightly worse than the SDBA

Procedure. In practice, the Simplified SDBA Procedure is much easier to implement. Moreover, since all the simulation replications have the same run length, the SDBA+OCBA Procedure could be used to select the best design among several alternative designs whose transient performances follow full quadratic polynomials.

In this section, we apply the SDBA+OCBA Procedure to select the best design among five alternative designs whose transient performances follow full quadratic polynomials. Since we would use the Simplified SDBA Procedure to allocate computing budget and estimate design performances, we would refer to this procedure as Simplified SDBA+OCBA Procedure. Again the original OCBA Procedure and the Simple Regression+OCBA Procedure are used as the comparison procedures. Moreover, we would also investigate the efficiency of the Heuristic SDBA+OCBA Procedure in which the computing budget allocation among multiple designs is done by using the OCBA rule, while the computing budget allocation for a single design is done by applying SDBA Procedure without simplification.

In Figure 4-2, we present the results we obtained by running the simulation in MATLAB using the four different procedures. The probabilities of correct selection after each round of budget allocation have been calculated for all the four procedures.

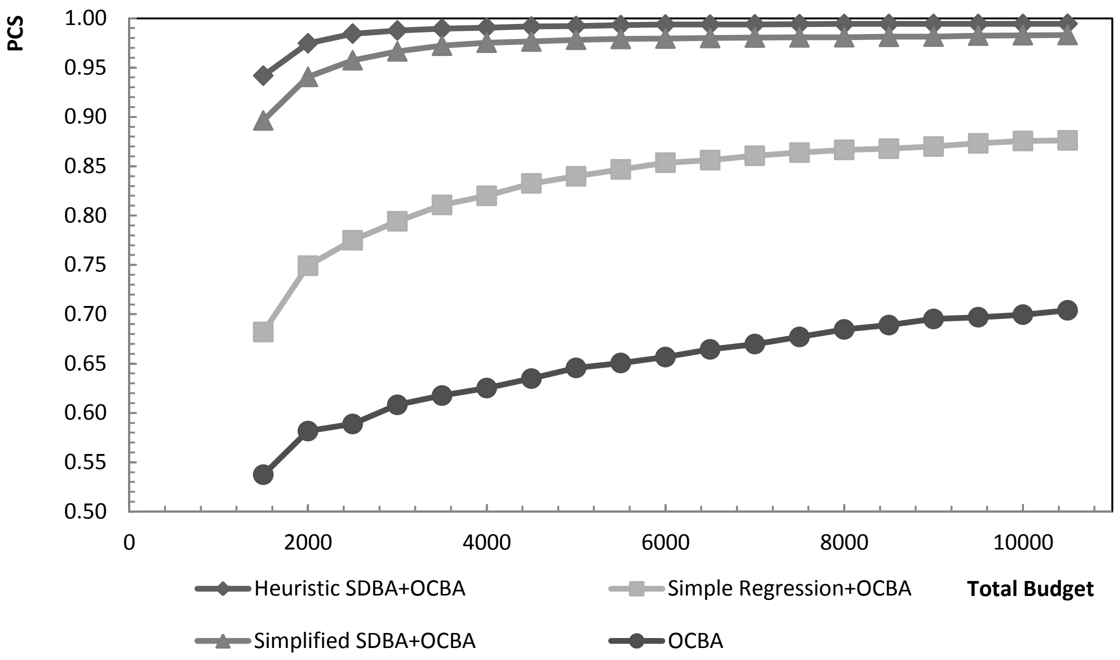

Figure 4 - 2 Comparisons of the performances of various computing budget allocation rule on the selection of the best design with full quadratic underlying function

Similar to the result we obtained in the first experiment, incorporating the regression approach to estimate the design performances would result in a higher probability of correct selection, and introducing the procedure of minimizing the estimated variance by doing regression could further increase the probability of correct selection.

Moreover, the Heuristic SDBA+OCBA Procedure is able to performance even better than the Simplified SDBA+OCBA Procedure. This is because the estimation of design performances by using the Heuristic SDBA+OCBA Procedure is better than the estimation by using the Simplified SDBA+OCBA Procedure, resulting in a higher convergence rate of the objective function. Moreover, as presented in the Single Design study, as the total computing budget allocated to design θ_i increases, $V_{\theta_i}^*$ would be very close to its lower bound, which is the constant one. Though $V_{\theta_i}^*$ would never be equal to one, its value is so close to one that it is almost a constant in problem (4.5) regardless of the value of α_{θ_i}, resulting in that the fact that even though the optimal budget allocation strategy for design θ_i is not independent of α_{θ_i}, model (4.6) could be a good approximation of model (4.5), hence the Heuristic SDBA+OCBA Procedure could give us a result that is close to optimal.

Nevertheless, in practice, the extra effort required to compute the second run length in the Heuristic SDBA+OCBA Procedure might be quite significant, thus the Simplified SDBA+OCBA Procedure might still be the first choice due to its ease of implementation and high efficiency.

5. CONCLUSION AND FUTURE WORK

5.1. Summary and Contributions

In this thesis, we have studied the problem of efficient computing budget allocation by using regression. In the first part of this study, we have looked into the problem of optimal computing budget allocation for a single design whose transient mean performance follows certain underlying function. The problem has been formulated as a global optimization problem based on the Bayesian Regression Framework. 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 budget allocation rules. In the second part of the thesis, we have looked into the problem of optimal computing budget allocation among several alternative designs by using regression, when the transient mean performances of designs follow certain underlying function. When the optimal computing budget allocation for a single design is independent of the computing budget allocated to that design, by approximating the probability of correct selection and by using the Large Deviation Theory, we have proved that the problem of maximizing the probability of correct selection can be decomposed into two sub-problems that could be solved by using the OCBA Procedure and the SDBA Procedure. As a result, the SDBA+OCBA procedure has been developed and based on the numerical experimentation, it has been proved to be an efficient ranking and Selection Procedure which enables us to select the best design among several alternative designs by using very little computing budget as compared to the other existing procedures.

5.2. Limitations and Future Work

We formulate and solve our problems based on certain assumptions which might not hold in real life application. Though a certain 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. Additionally, the problem of correlated simulation noise might be addressed by using certain regression model such as AR Model and more work is needed to investigate how we could incorporate the AR model into the SDBA Procedure.

Moreover, based on our study during the development of the SDBA+OCBA Procedure, the problem of maximizing the probability of correct selection can be done only when the budget allocation strategy for a single design is independent of the total budget allocated to that design. In practice, this condition is often violated thus the efficiency of the SDBA+OCBA Procedure might not be guaranteed. One can try to solve the original maximization problem numerically and observe if certain patterns exist in the solutions when the underlying functions of the designs follow certain functional forms.

BIBLIOGRAPHY

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

Bratley, P., Fox, B. and Schrage, L. (1987). *A guide to simulation* (2 ed.). Springer-Verlag .

Brantley, M., Lee, L., Chen, C. and Chen, A. (2011). Efficient Simulation Budget Allocation with Regression. Submitted to *IIE Transcations*.

Chen, C.-H. (1995). An Effective Approach to Smartly Allocate Computing Budget for Discrete Event Simulation. *The 34th IEEE Conference on Decision and Control*, (pp. 2598-2605).

Chen, C.-H. and He, D. (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. and Lee, L. (2011). Optimal Comupting Budget Allocation for variance Reduction in rare-event Simulation. In C.-H. Chen and L. Lee, *Stochastic simulation optimization : an optimal computing budget allocation* (pp. 169-171). World Scientific .

Chen, C.-H. and Lee, L. H. (2011). *Stochastic simulation optimization : an optimal computing budget allocation.* World Scientific.

Chen, C.-H., Donohue, K., Yücesan, E. and Lin, J. (2003). *Optimal Computing Budget Allocation for Monte Carlo Simulation with Application to Product Design. Journal of Simulation Practice and Theory*, 11, 57-74.

Chen, C.-H., He, D., Fu, M. and Lee, L. H. (2008). Efficient Simulation Budget Allocation for Selecting an Optimal Subset. *Informs Journal on Computing*, 20, 579-595.

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

Chen, E. and Lee, L. (2009). A multi-objective selection procedure of determining a Pareto set. *Computers and Operations Research*, 1872-1879.

Chew, E. P., Lee, L., Teng, S. and Koh, C. (2009). Differentiated Service Inventory Optimization using Nested Partitions and MOCBA. *Computers and Operations Research*, 36, 1703-1710.

Chick, S. (1997). Selecting the best system: A decision-theoretic approach. *Proceeding of Winter Simulation Conference*, (pp. 326-333).

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

Elfving, G. (1952). Optimal allocation in linear regression theory. *Ann. Math. Statist.* , 255-262.

El-Krunz, S. ad Studden, W. J. (1991). Bayesian Optimal Designs for Linear Regression Models. *The Annals of Statistics , 19,* 2183-2208.

Fu, M. C., Hu, J. Q., Chen, C.-H. and Xiong, X. (2007). Simulation Allocation for Determining the Best Design in the Presence of Correlated Sampling. *Informs Journal on Computing*, 19, 101–111.

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

Glynn, P. and Juneja, S. (2004). A large deviations perspective on ordinal optimization. *Proceeding of Winter Simulation Conference*, (pp. 577-585).

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

Haines, L. M. (1993). Optimal design for nonlinear regression models. . *Comm. Statist.*

He, D., Lee, L. H., Chen, C.-H., Fu, M. and Wasserkrug, S. (2009). Simulation Optimization Using the Cross-Entropy Method with Optimal Computing Budget Allocation. *ACM Transactions on Modeling and Computer Simulation*.

Hsieh, B. W., Chen, C.-H. and Chang, S. C. (2007). Efficient Simulation-based Composition of Dispatching Policies by Integrating Ordinal Optimization with Design of Experiment. *IEEE Transactions on Automation Science and Engineering*, 4, 553-568.

Kelton, W. and Law, A. (1983). A New Approach for Dealing with the Startup Problem in Discrete Event Simulation. *Naval Research Logistics , 30* (4), 641-658.

Law, A. (2007). *Simulation modeling and analysis*. McGraw-Hill.

Law, A. and Kelton, W. (2000). *Simulation Modeling and Analysis* (3 ed.). Boston: McGraw-Hill.

Lee, L. H., Chew, E. P., Teng, S. Y. and Goldsman, D. (2010). Finding the Pareto set for multi-objective simulation models. *IIE Transactions*.

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

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

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

Ng, T. S. and Goodwin, G. C. (1976). On optimal choice of sampling strategies for linear system identification. *International Journal of Control , 23* (4), 459-475.

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

Pronzato, L. (2009). Asymptotic properties of nonlinear least squares estimates in stochastic regression models over a finite design space. Application to self-tuning optimisation. *IFAC Proceedings*, *15*, pp. 156-161.

Pujowidianto, N. A., Lee, L. H., Chen, C.-H. and Yep, C. M. (2009). Optimal Computing Budget Allocation For Constrained Optimization. *Proceedings of 2009 Winter Simulation Conference*, (pp. 584-589)

Tanrisever, F., Morrice, D. and Morton, D. (2012). Managing Capacity Flexibility in Make-to-Order Production Environments. *European Journal of Operational Research , 216* (2), 334-345.