A GENERAL FRAMEWORK ON THE

COMPUTING BUDGET ALLOCATION PROBLEM

PUVANESWARI MANIKAM

(B.Sc.(Hons.), University Technology Malaysia)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTERS OF ENGINEERING

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

ACKNOWLEDGEMENT

The author would like to express her heartfelt gratitude to her supervisors, Associate Professor Chew Ek Peng and Dr Lee Loo Hay for their patient guidance and illuminating advice throughout her course of this research. This thesis would have been impossible without them.

The author would also like to extend her deepest gratitude to her employer, The Logistics Institute-Asia Pacific (TLI-AP) for their continuous encouragement throughout the research. Their unwavering support has made this academic exercise more meaningful and smooth going.

The author in also indebted to her husband, Sandra for his encouragement, support and understanding throughout this exercise of research, and to her son Puvannesan for his full cooperation. Sincere thanks is conveyed to all her family members for their limitless love and unfailing support. Last but not least, warm appreciation is extended to all those who helped her to make this thesis a success.

TABLE OF CONTENTS

SUMMARY

Because the design space is huge in many real world problems, estimation of performance measure has to rely on simulation which is time-consuming. Hence it is important to decide how to sample the design space, how many designs to sample and for how long to run each design alternative within a given computing budget. In our work, we propose an approach for making these allocation decisions. This approach is then applied to the problem of assemble-to-order (ATO) systems where the sampling average approximation (SAA) is used as a sampling method. The numerical results show that this approach provides a good basis for decisions.

LIST OF FIGURES

LIST OF TABLES

Chapter 1 INTRODUCTION

1.1 Background

In much of the industrial applications, it is often assumed that all information needed to formulate and solve a design and control problem is deterministic, which means all information is known. In this case, the solution is expected to be optimal and reliable. In reality however, randomness in problem data poses a serious challenge for solving many optimization problems. The fundamental reason for the randomness is due to the nature of the data which represents information about the future (for example, product demand and price over the next few months), and these data cannot be known with certainty. As a result, the randomness may be present as the error or noise in measurements in estimating the performance. As such, stochastic optimization problems arise from applications with inherent uncertainty. Some examples of the stochastic optimization problem in industrial applications can be seen in manufacturing production planning, machine scheduling, freight scheduling, portfolio selection, traffic management, automobile dealership inventory management and water reservoir management. A general problem of stochastic optimization can be defined by the mathematical expression that is represented by the minimization form in P(1).

P(1):

$$\min_{\theta \in \Theta} J(\theta) \quad \equiv \quad E[L(\theta, \xi)] \tag{1.1}$$

where Θ is a design space consisting of all potential candidates; θ is a design alternative; ξ is a random vector that represents uncertainties in the system; L is the sample performance which is a function of θ and ξ, and J is the performance measure which is the expectation of L.

P(1) poses two major challenges; the "stochastic" and the "optimization". The challenge in the "stochastic" aspect lies in the task of estimating $J(\theta)$. Often the corresponding expectation function is not possible to be computed exactly, and need to be estimated by simulation. Let ξ_i, $i = 1, 2, ..., N$ be a realization of the uncertainties in replication i and the expected performance value is estimated as follows,

$$E[L(\theta, \xi)] \approx \hat{J}(\theta) \equiv \frac{1}{N}\sum_{i=1}^{N} L(\theta, \xi_i) \tag{1.2}$$

The estimation of the expectation function in (1.2) may require a long computational time. To make matters worse, the notoriously slow convergence rate of the accuracy cannot be improved any further than $1/\sqrt{N}$.

The other limitation is the "optimization" part. When an optimization problem has the advantage of the design space structure and real-variable nature to work out effective algorithms for optimization, traditional analysis tools, such as infinitesimal perturbation analysis (IPA) can be used to estimate the gradient for determining the local search direction. However, when the problem becomes structureless and Θ becomes totally arbitrary, such advantage is no longer viable. As a result, combinatorial explosion of system designs occurs forcing us to consider a constrained set of possibilities to be the optimal design. In such cases, a random search becomes an alternative that may not be an effective approach for a simulation based optimization problem. Other alternatives to locate near-optimal designs include the use of some Artificial Intelligence optimization tools such as Neural Networks, Genetic Algorithm or Hybrid techniques.

Realizing the challenges posed by both the stochastic and optimization aspects in a stochastic optimization problem, the concept of ordinal optimization emerged. Unlike the concept of cardinal optimization that estimates the accurate values of design performance, the ordinal optimization is based on two advantageous ideas, (i) "order" converges exponentially fast while "value" converges at rate $1/\sqrt{n}$ (n:simulation length), that is, it is much easier to know whether "A>B" than to estimate the value of "A-B" (ii) Goal softening can make hard problem easier, that is, we settle for "good enough set with high probability" instead of "best for sure". Suppose G denotes the good enough subset of a search space Θ based on true performance value, and S denotes the selected subset of a search space Θ based on the observed sample performances. The quality of selection is then determined by the overlap of S with G which is quantified through the alignment probability, $P\{|G \cap S|\} \geq k$ where k is the number of minimum desired overlap between the two subsets. Alignment probability, also called the probability of correct selection in the context of simulation, is the measure of the goodness of the selection rules. In other words, the alignment probability in ordinal optimization tries to find what is the probability that among the set S that we have chosen, we have at least k members of G. Figure 1.1 illustrates the general concept of ordinal optimization.

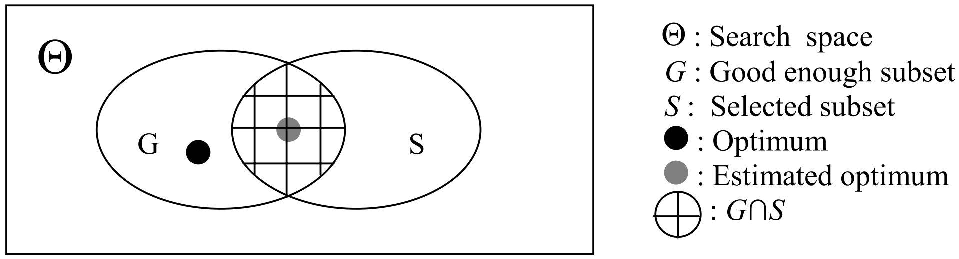

Figure 0.1.1: Softened definition of ordinal optimization

Note that the goal softening in ordinal optimization has advantage over the traditional optimization view where both the subsets *G* and *S* are no longer singletons. With this idea, the ordinal optimization has the ability to quickly separate the good designs from the bad one. We see that ordinal optimization has at least provided a means for narrowing down the search with higher probability of getting a good design, which otherwise is not possible. It has emerged as an efficient technique for simulation and optimization.

Ordinal optimization has provided a paradigm shift in optimization, and has also changed the way we should deal with stochastic optimization. Instead of running very long simulation for every design until we obtain its precise performance estimation, we should look at how to balance the effort spent in running the simulation and sampling the designs. Ranking and selection (R&S) procedure and the multiple comparison procedure (MCP) are among the methods that have been successfully used in spending the simulation effort of a set of design effectively. R&S is a statistical procedure developed in the simulation optimization to select the best design among a fixed set of designs. Generally, the design having the largest expected value is regarded as the "best" design. The R&S procedure usually guarantees a certain level of the probability of correct selection. There are two major approaches widely used in the R&S

procedures; the indifference zone (IZ) selection approach and the subset selection approach. The goal of the IZ selection approach is to select the design associated with the largest mean. In a stochastic simulation however, such a "correct selection" can never be guaranteed with certainty. Having such condition, a compromise solution offered by this approach is to guarantee to select the best design with high predefined probability whenever it is at least a user-specified amount better than the others. This practically-significant difference is called the indifference-zone. In contrast to the approach of IZ selection that attempts to select the single best design, the subset selection approach is a screening tool that aims to select a small subset of alternative design that includes the design associated with the largest mean.

Unlike the goal of R&S procedure which is to make a decision (i.e. select the best design) directly, the goal of MCP is primarily to identify the differences and the relationship between the designs' performance. MCP tackles the optimization problem by forming simultaneous confidence intervals (CIs) on the means. These CIs measure the magnitude and difference between the expected performance of each pair of the alternatives. One of the most widely used classes of MCP is the multiple comparisons with the best (MCB). In the MCB approach, the CIs are measured by the difference between the expected performance of each design and the best of the others. Other three classes of MCP developed includes the paired-t, Bonferroni, all-pairwise comparisons (MCA), the all-pairwise multiple comparisons (MCA) and the multiple comparisons with a control (MCC). In this thesis, the focus will be mainly on the R&S procedure.

Further with the idea of ordinal optimization, simulation efforts should now be spent wisely on the designs sampled by intelligently determining the number of simulation samples or replications among the different designs. Such effort called Optimal Computing Budget Allocation (OCBA) tries to optimally choose the simulation length for each design to maximize simulation efficiency within a given computing budget. Larger computing budget or simulation efforts should be invested on the potentially good designs to improve their performance, while limited computing resources should be allocated on the non-critical designs. The objective could be either to minimize the computational cost, subject to the constraint that the alignment probability is greater than a predefined satisfactory level, or to maximize the alignment probability, subject to a fixed computing budget.

While OCBA focus on allocating the simulation time for a fixed number of design alternatives, sampling effort further decide on the right number of designs to sample and how the sampling of designs should be performed. Blind picking or random sampling is one common method used for sampling designs. Although the time spent in sampling designs in such method is negligible, the design selection is not very good (we expect smaller overlap between subset S and G) in a random sampling method. However, if a sophisticated sampling method is used, some computational time will be required for sampling designs and the design selection is expected to improve. In such cases, besides allocating the computing time to estimate the performance measure of the designs, we also have to wisely allocate the time to spend to sample each design.

1.2 Objectives

In our work, we assume that a sampling method can be differentiated by the degree of information (sophistication) used. The degree of information will affect the time used for sampling and the resulting performance measure. Hence, given a fixed amount of computing time, we want to optimally decide on how to sample the designs, number of designs to sample and the simulation time allocated for each design so as to optimize the expected true performance of the finally selected design. We propose an approach on how to ideally decide these allocation decisions.

1.3 Scope

The remaining section of this thesis is organized as follows. In the following Chapter 2, the relevant literatures on ordinal optimization, R& S and OCBA are presented. In Chapter 3, we introduce the OCBA model and discuss how the distribution of performance measure and the distribution of estimation noise affect the results of our proposed approach. In Chapter 4, the proposed approach of our framework is demonstrated on an assemble-to-order (ATO) system where the sample average approximation (SAA) proposed by Shapiro (2001) is used as the sampling method. We present two different numerical examples of the ATO problem in Chapter 5. Finally in Chapter 6, important conclusions are drawn and some directions for future research are given.

Chapter 2 LITERATURE SURVEY

2.1 Introduction

In recent years, the need for stochastic optimization in Industrial and Systems Engineering has received increased recognition. The essential of the optimization under uncertainty is justified by the need of facing the real world problem in a more realistic ways. However, as discussed in Chapter 1, the solution to the stochastic optimization problem can be hardly obtained due to the "stochastic" and "optimization" challenges in the problem, and often the approximate solutions is obtained via simulation. Hence, much effort has been contributed by different authors over the years in coming up with various alternatives to tackle the challenges in the stochastic optimization and simulation.

We first review the literatures involved on the topic of ordinal optimization. As R&S method are related directly to ordinal optimization in performing the simulation of a set of designs effectively, we discuss in detail the progress of R&S methods over the years in the Section 2.3. Finally in Section 2.4, we present the evolving literature on the OCBA on determining the number of replication among different designs to optimize the simulation efficiency.

2.2 Ordinal Optimization

As an effort to soften the stochastic and optimization aspects in a stochastic optimization problem, Ho et al. (1992) proposed the concept of ordinal optimization. The idea of ordinal optimization is based on the fact that order converges very much faster than value. In this paper, the ordinal optimization concept was emphasized as a simple, general, practical and complementary approach as compared to the cardinal optimization which requires large computing efforts to be spent in obtaining the best estimates. Ordinal optimization can significantly reduce the simulation effort in estimating the performance measure by approximating the model and shortening the observations. More importantly, it was emphasized that with the parallel implementation of the ordinal optimization algorithm (one does not need to know the result of one experiment in order to perform another, i.e. the sequential approach) the repeated experiments in simulation can be performed easily to improve the system designs. In their work, the examples of buffer allocation problem and a cyclic server problem was used to illustrate the applicability of the approach.

Dai (1996), Xie (1997), Tang and Chen (1999) and Lee et al. (1999) provided theoretical evidence of the efficiency of ordinal optimization. Dai (1996) tackled the fundamental problem of characterizing the convergence of ordinal optimization. An indicator process was formulated and it was proved to converge exponentially, i.e. comparing the relative orders of performance measure, converges much faster than comparing the performance measure estimations. With this tenet of ordinal optimization, one will be able to identify the good designs very quickly.

An extension of the previous work was given in Xie (1997) in which the dynamic behaviours of ordinal comparison were investigated. Similarly he proved that for regenerative systems, the alignment probability converges at exponential rate. The classical large deviation result was used in the proof.

While Dai (1997) established the exponential convergence rate of the ordinal comparison algorithm for a classical regenerative process in the continuous-time and for the independent and identically distributed (i.i.d.) random sequence in the discrete-time, Tang and Chen (1999) proved the exponential convergence rate in the context of one-dependant regenerative processes instead. A systematic approach was developed using the stochastic Lyapunov function criterion to verify the exponential stability condition for Harris-recurrent Markov chains (HRMCs), a special case of one-dependant regenerative processes. Several examples in queuing theory were examined to illustrate the developed criterion.

Lee et al. (1999) further presented the detailed explanations and the theoretical proofs of goal softening in ordinal optimization. Using the order statistics formulation, it was established that the misalignment probability (a condition when there is no alignment in the selection) decreases by the exponential effect. Further, it was concluded that by softening (relaxing) the good enough subset and selected subset condition, one could achieve a significant improvement in the alignment probability.

While the previous works exploited the efficiency of ordinal optimization when the noise of the N designs is assumed to be i.i.d., Yang and Lee (2002) extended the existing methodology when the i.i.d. assumption of noise is relaxed. In order to generalize the ordinal optimization approach to problems where the noise term follows arbitrary distribution and design dependant, Yang and Lee (2002) proposed new selection scheme based on Bayesian model and distribution sensitive selection rule. This scheme used the selection index for every design, which is calculated from a proposed Bayesian model. It was also shown how this selection index could be used to maximize the alignment probability. Some application examples were illustrated to show how this selection scheme solved the non i.i.d. problem.

Ho et al. (2000) provided the efficiency of ordinal optimization in the context of simulation. It was emphasized that the ordinal optimization reduces the computational cost for design selection in a simulation effort. Further details and literatures on this computing budget allocation problem (OCBA) are discussed in Section 2.4. With the idea of ordinal optimization, simulation efforts should now be spent wisely on the designs sampled.

2.3 Ranking and Selection

Ranking and Selection (R&S) is a statistical method specifically developed to select the best design or the subset containing the best design from a fixed set of competing designs. In the examples of applications, ranking is also seen to be stabilizing very early in simulation (Ho et al. (1992)), and thus can be used efficiently to solve the ordinal optimization problem. There has been continuous development in research dealing with R&S issues in the field of simulation study.

As described in Chapter 1, there are generally two approaches that are widely used in the R&S works; the indifference-zone (IZ) selection and the subset selection approach. We first present the literature survey on the IZ selection approach, followed by the subset selection approach, and then the combined approach. Following this, the literatures on the R&S unified with the multiple comparison procedure (MCP) are discussed. Finally some recent developments in the R&S procedure are described.

The concept of R&S was first proposed by Bechhofer (1954). He suggested that the formulation of problem in terms of R&S approach is better than the classical test of homogeneity (analysis of variance) approach. The hypothesis that several essentially different systems have the same population mean yield is unrealistic one; different treatment must have produced some difference, though the difference may be small. Thus it is important to estimate the size of the differences in order to identify the best of the designs. This has emerged as the motivation for the R&S approach. Bachhofer (1954) first formulated the IZ approach for randomly sampled k normal populations with a common and known variance. In his approach, he was interested in selecting a single population such that there was at least the probability P^* of making the correct selection, provided the greatest population mean exceeds all other means by a user specified "indifference zone", δ^* where the differences of less than δ^* were considered practically insignificant. If the population means lie within the δ^*, the populations were viewed as the same and thus there exist no preference between the two alternatives. The N independent observation was picked from each of the k populations, and the decision was to choose the population with the largest observed sample mean. In his paper, he addressed the problem of determining the common sample size N that guarantees the predefined P^* under the indifference zone $\delta \geq \delta^*$.

As Bechhofer's approach (1954) described above is a single-stage procedure (i.e. the N required is determined by the choice of δ and P^*), Paulson (1964) formulated the same problem as a multi-stage (sequential) problem, which means they require two or more stages of simulation. In the first stage, a user-specified number of observations were fixed, and certain stopping criteria was checked. If the criterion was met, the user should stop the experiment and select the best design. Otherwise, he should proceed to the second stage and continue sampling until the stopping criterion is met at the rth stage. As the sequential sampling progresses, the inferior populations were eliminated from further consideration. Likewise in Bechhofer (1954), Paulson (1964) also assume a common and known variance of populations. Although a sequential procedure was proposed for the common but unknown variance in this paper, it was far from being the best solution. Bechoffer et al. (1954) also attempted to formulate the problem for the case of a common but unknown variance using a two-stage procedure.

All the literatures discussed above dealt with only the known or unknown common variance. In reality, often it is impossible to know about the performance variance of a design that does not exist physically. Even when the variance is known, ensuring the common variance for all the designs is another challenge. Realizing this bottleneck, modern IZ approaches were developed for the case that neither equal nor known variances were required.

Dudewicz and Dalal (1975) and Dudewicz (1976) are among the first articles that addressed the selection problem with IZ approach under normal means with unknown and unequal variances. They developed a two-stage procedure with user-specified δ and P^*. In the first stage, the experimenter chose N number of observations and the sample variance was estimated. Based on this value, the number of additional observations was determined in the second stage. Rinott (1978) developed a somewhat similar method with some modifications. This method however cannot tackle the large problem. Most IZ selection approaches used today are directly or indirectly developed based on Dudewicz and Dalal (1975) or Rinott (1978) selection procedure.

Koenig and Law (1985) generalized the two-stage procedure suggested in Dudewicz and Dalal (1975) for selecting the subset of size m containing the l best of k independent normal populations so that the selected subset will contain the best design with at least the probability P^*. This IZ approach was essentially a screening procedure developed to eliminate the inferior designs at the initial stage. This method required the selection of different table constant when computing the sample size in the second stage.

There are many real world applications of the R&S procedure (using the IZ approach) for selecting the best design among the competing designs. For example, the selection procedure in Koenig and Law (1985) was illustrated using a simulation study of an inventory system. Another application example involving the selection of the best airspace configuration to minimize the airspace route delays for a major European airport was presented in Gray and Goldsman (1988). Goldsman and Nelson (1991) applied the Rinott (1978) procedure to an airline reservation system problem. Besides being easy to use, the procedure also assured the selection of the good design with high probability. One disadvantage described was that this procedure at times requires more observation than necessary in order to configure a favorable design mean. In another work, Goldsman (1986) also provided a brief tutorial on the IZ approach for both the single-stage and multi-stage with common known variance.

In contrast to the IZ approach, there exist another large class of R&S procedure for the best design selection proposed by Gupta (1956) and (1965), i.e. the subset selection approach. The subset selection approach is a method for producing a subcollection of alternatives that has random size, and this subset contains the best population with the guaranteed probability P^*. The advantage of this approach was that it enabled the experimenter to screen a large set of alternatives, and allowed adequate resources to be allocated to the selected subset so that it can be examined more thoroughly with a follow up study. To better illustrate the subset selection approach, Gupta and Hsu (1977) presented an application example of motor facility data.

As the initial methodology on subset selection approach required common and known variances, Sullivan and Wilson (1989) worked a modern approach that allowed unknown and unequal variances for the normal population. Using the subset selection approach, they developed two different procedures of random sampling scheme to compare transient or steady-state simulation models; the exact procedure was designed based on the single independent replications for each design, while the heuristic procedure was based on single lengthy run for each of the design. As it is more rewarding to decide on the best design rather than to identify a subset that contains the best design, the IZ selection approach has emerged as a more favorable approach compared to the subset selection approach.

On the other hand, when the number of design alternatives was large, Nelson et al. (2000) suggested using the idea of sample-screen-sample-select to reduce the computational effort. This is a subset selection and IZ selection combined method. In the first stage, the subset selection approach was used to screen out the noncompetitive designs, and the IZ selection was then used to select the design among the survivors of the screening.

In Matejcik and Nelson (1993), it was shown that by combining the R&S procedure (i.e. the IZ selection approach) with the multiple comparison procedure (MCP) (i.e. the multiple comparisons with the best (MCB) approach), a better procedure could be designed for selecting the best design. The applicability of this simultaneous procedure was illustrated with an inventory example problem.

A review on modern approaches in the R&S and the MCP to compare designs via the computer simulations was presented in Goldsman and Nelson (1994). The various approaches (including the combined approaches) in the statistical procedures used in a simulation were given for four classes of subprobelms; screening a large number of system designs, selecting the best system, comparing all designs to a standard and comparing alternatives to a default. For example, the two-stage procedures (using the IZ selection approach and the MCB approach) for comparing a fixed set of designs with a single standard design in simulation experiments were presented in Nelson and Goldsman (2001). Given k alternative designs and a standard, the comparison was based on their expected performance. The goal of this procedure was to check if there is any other design with a better performance than the standard, and if so to identify them.

Another complete review on the existing literatures on the R&S and the MCP were given in Swisher and Jacobson (1999). The existing approaches in each of the procedures were presented along with the recent unified approaches. These works emphasized on the advantages of the unified approaches. Besides leading to better methods to make a correct selection, by unifying these procedures, one will be able to compare the best design to each of the other competitors. This information can provide inference about the relationships between designs which may facilitate decision-making based on secondary criteria that are not reflected in the output performance measure selected.

It is known that most IZ selection approaches guarantee MCB confidence intervals (CIs) with half-width corresponding to the indifference amount (Chen and Kelton (2003)). In this latest work, they presented the statistical analysis of MCB and multiple comparisons with a control (MCC) with CIs. For the MCC approach, the CIs bound the difference between the performance of each design and a specified design as the control, while for the MCB approach, the CIs bound the difference between the performance of each design and the best of the others. Chen and Kelton (2003) further established that the efficiency of the selection procedures could be improved by taking into consideration of the differences of sample means, using the variance reduction technique of common random numbers and also by using the sequential selection procedures.

Goldsman and Marshall (2000) recently extended the R&S procedures for use in steady-state simulation experiments. The Extended-Rinott Procedure (ERP) and the Extended-Fully Sequential Procedure (EFSP) were the two sequential procedures developed with the aim to select the design with the minimum (or maximum) steady-state mean performance. For the ERP, the first stage variance estimator was replaced with marginal asymptotic variance estimator, while for the FSP the estimator was replaced with an estimator of the asymptotic variance of the difference between pairs of systems.

The procedures discussed above assumed that the observations recorded are independent and identically normally distributed. In reality though, often it is not a valid assumption when dealing with simulation outputs. Realizing this challenge, Goldsman and Nelson (2001) presented three procedures for selecting the best design when the underlying (i.i.d) assumption of observations is relaxed. The first procedure was a single stage procedure for finding the most probable multinomial cell, the second was a sequential procedure and finally the third is a clever augmentation that makes more efficient use of the underlying observations.

The R&S procedure can also be used together with other methods to achieve better results. Butler et al. (2001) exploited the R&S procedure for making comparisons of different designs that have multiple performance measures. They developed and applied a procedure that combines multiple attribute utility (MAU) theory (an analytical tool associated with decision analysis) with R&S procedure to select the best configuration design from a fixed set of possible configuration designs. To achieve this goal, the famous IZ selection approach of the R&S procedure was utilized. In Ahmed and Alkhamis (2002), the simulated annealing method was combined with the R&S procedure for solving discrete stochastic optimization problems. The unified procedure converged almost to the global optimal solution.

2.4 Optimal Computing Budget Allocation (OCBA)

The performance of ordinal optimization is further improved by intelligently determining the number replication for the different designs sampled in the OCBA problem. Chen et al. (1997) presented an OCBA model to decide on how to allocate the computing budget to the designs so as the predefined probability of correct selection could be satisfied. First all designs were simulated with the same number of replications and the probability of correct selection was approximated. If the probability did not achieve the predefined level, an additional allocation of simulation replications would be given to the more promising designs and the marginal increase in the correct selection probability would be estimated. In their approach, the optimal allocation problem was solved using the gradient method. This effort was further extended in Chen et al. (1998) where they incorporated the impact of different system structures by considering different computation costs occurred in each design.

A new asymptotical allocation rule was developed by Chen et al. (2000) to give a higher efficiency when solving the optimal budget allocation problem where the simulation costs of all the designs were the same. This approach gave higher probability of correct selection even with a relatively small number of replications. Chen et al. (2003) recently extended this work. They developed an asymptotical approach in which the objection function was replaced with an approximation that could be solved analytically. A significant advantage of this method was that this approximated allocation problem could be solved with negligible computational cost. Moreover with the restriction of the equal cost of all designs being relaxed, it enabled a more general formulation of the allocation problem. The ultimate idea of all these efforts is to optimally allocate the available computing resources to all the potential designs so as to maximize the probability of correct selection.

As much of the literature focused on allocating the simulation time for a fixed number of design alternatives, Lee and Chew (2003) widened the scope by considering how many designs to sample when the design space is huge. A simulation study was presented to show that the sampling distributions (distribution of performance measure and distribution of estimation noise) will affect the decision on how to perform sampling and run simulation efficiently. They assumed the designs were randomly sampled and the time spent in sampling designs was negligible. However, if a sophisticated sampling method is used, some computational time will be required for sampling designs. In such cases, besides allocating our computing time to estimate the performance measure of the designs and number of designs to sample, we also have to wisely allocate the time to spend to sample each design. Our work is an extension of this idea. Given a fixed computing budget, we want to decide on how to sample the design space, how many designs to sample and how long to run the simulation for each design so as to obtain a good performance measure.

Chapter 3 SAMPLING, RANKING AND SELECTION

3.1 Introduction

In order to design and compare the alternatives of large man-made system designs such as the inventory systems, communication network, manufacturing and traffic systems, it is often necessary to apply extensive simulation since no closed-form analytical solutions exist for such problems. Unfortunately, using simulation can be both expensive and time-consuming, and this may preclude the feasibility of simulation for sampling, ranking and selection problems. This challenge becomes even more critical when we are limited with a fixed computing budget. Thus it becomes crucial in the optimal computing budget allocation (OCBA) problem to wisely determine the computation costs allocation while obtaining a good decision in simulation.

In this chapter, we model the OCBA to determine on how much information to use to sample a design, how many designs to sample and how long to run the simulation in order to estimate the performance measure for our problem. Before presenting the OCBA model, we first define the notations to aid clarity. In our problem, given a fixed computing budget, it becomes crucial to find a balance between the allocation decisions. We therefore discuss about the trade-offs involved. Following this, we discuss and present the assumptions and models in the allocation problem when the distribution of true performance value of the designs sampled follows different types of distributions.

3.2 OCBA Model

In the OCBA model for our problem, we associate n_0 with a sampling scheme. Let n_0 represent the degree of information (sophistication) used to sample a design and $t_{(n_o)}$ as the time taken to sample a design when n_0 degree of information is used. The higher the value of n_0, the more information is used and better designs can be sampled. However with larger values of n_0, more time will also be needed in sampling the designs. Let n_1 denote the number of designs to sample and n_2 denote the number of replications of the simulation run for each design. For n_2, we assume horse race selection method is used, which means that n_2 is the same for all the designs. Our objective of this problem is to find the optimal allocation decision of n_0, n_1 and n_2 under a fixed computing budget that minimizes the expected true performance of the observed best design, $E[J_{[\tilde{1}]}]$. The OCBA problem is as follows,

P(1):

$$\min E\,[J_{[\tilde{1}]}] \tag{3.1}$$

subject to

$$(\, t_{(n_o)} + s\, n_2\,)\, n_1 \leq K \tag{3.2}$$

$$\tilde{J} = J + w \tag{3.3}$$

where J is the true performance, $\tilde{J}$ is the observed performance, the subscript $[i]$ is the design with true rank i, $[\tilde{i}]$ is the design which is observed as rank i and w is the noise. Note that $\tilde{J}$ is an estimation of J and the order of the observed performance of N designs can be written as $\tilde{J}_{[\tilde{1}]} \leq \tilde{J}_{[\tilde{2}]} \leq ... \leq \tilde{J}_{[\tilde{N}]}$. We model the OCBA in term of time unit, where s is the time to run one replication of simulation and K is the given computing budget in unit time. Equation (3.2) states that the total time spent for sampling and running the simulation has to be less than K. Equation (3.3) defines the relationship between the observed performance and the true performance. From this equation, it is shown that the observed performance is confounded by noise.

For an ideal case, we always hope that n_0, n_1 and n_2 are high. However, given a fixed computing budget, it is not realistic to set all three allocation decisions to be high. For example, when n_0 is large, (n_1 and n_2 is small), we can use more information to sample a design, but only few designs will be sampled with few replications to run for each design. Generally with large n_0, good designs are sampled, but they may be confounded with large noise. Hence, we may end up picking the worst designs within the sampled designs. For the case when n_1 is large, (n_0 and n_2 is small), we will have many designs with each design being sampled using less information and with fewer replications. As a result, there will be higher chance of getting good designs, but we may fail to locate the good designs due to the large noise. On the other hand, when n_2 is large, (n_0 and n_1 is small), a large portion of computing time for simulation is allocated for the few designs which has been sampled using less information. Although we will be able to select the design with low noise within the n_1 designs sampled, this design however may not be good as the good designs may not have been sampled. Therefore, it becomes important for us to decide on the best trade-offs between n_0, n_1 and n_2 under a given computing budget so as to minimize the expected true performance of the observed best.

Generally, P(1) is not an easy problem as there is no close form solution for $E[J_{[\tilde{1}]}]$. From the model, we know that $E[J_{[\tilde{1}]}]$ depends on n_0, n_1 and n_2. The n_0 will affect the probability distribution of performance measure J. For example, when n_0 is randomly sampled (n_0 = 0, i.e. no information is used to sample a design), we expect the performance J to be mediocre. However when n_0 takes a value, some information is used and better designs will be sampled. As a result, the performance J tends to follow a skewed distribution. In the following subsections, we propose a general framework to address the allocation problem when the distribution of true performance of the samples follows normal and Weibull distributions.

3.2.1 Model Derivation for Normal Distribution of True Performance

When the true performance and the noise follow normal distributions, P(1) can be solved numerically. Note that with the different degree of n_0, we will have different normal distributions for the true performance, where the mean and the standard deviation of the distribution are denoted by $\mu_{x(n_0)}$ and $\sigma_{x(n_0)}$ respectively. Following are the assumptions made.

1. The true performance is normally distributed with $J \sim N(\mu_{x(n_0)}, \sigma_{x(n_0)})$
2. The noise is normally distributed with $w \sim N(0, \sigma_N)$

3. The standard deviation of the noise for one replication of the simulation run is equal to σ_{N_0}. Thus the standard deviation of the noise for the average of n_2 replications of the simulation run is as given below,

$$\sigma_N = \frac{\sigma_{N_0}}{\sqrt{n_2}}. \tag{3.4}$$

From equation (3.3),

$$\tilde{J} = \mu_{x(n_0)} + \sigma_{x(n_0)}\left[J' + w'\right] \tag{3.5}$$

where $J' \sim N(0,1)$ and $w' \sim N\left(0, \frac{\sigma_N}{\sigma_{x(n_0)}}\right)$. From the derivation given in Lee and Chew (2004), the expected true performance of the observed best when the true performance and noise follow normal distributions is

$$E[J_{[\tilde{1}]}] = \mu_{x(n_0)} + \frac{\sigma^2{}_{x(n_0)}}{\sigma^2{}_{x(n_0)} + \sigma^2{}_N} E[\tilde{J}_{[\tilde{1}]}] \tag{3.6}$$

where

$$E[\tilde{J}_{[\tilde{1}]}] = \sqrt{\sigma^2{}_N + \sigma^2{}_{x(n_0)}}\, E[z_{[1]}]$$

and $E[z_{[1]}]$ is the 1st order statistics of n_1 standard normal variables. From (3.5) and (3.6), the expected true performance of the observed best is

$$E[J_{[\tilde{1}]}] = \mu_{x(n_0)} + \frac{\sigma_{x(n_0)}}{\sqrt{1 + \sigma^2{}_N / \sigma^2{}_{x(n_0)}}} E[z_{[1]}] \tag{3.7}$$

$$= \mu_{x(n_0)} + \frac{\sigma_{x(n_0)}}{\sqrt{1 + \sigma^2{}_{N_0} / (n_2 \sigma^2{}_{x(n_0)})}} E[z_{[1]}]. \tag{3.8}$$

Note that when n_0 is fixed ($E[J_{[\tilde{1}]}]$ is only influenced by n_1 and n_2), the mean $\mu_{x(n_0)}$ and the standard deviation $\sigma_{x(n_0)}$ of the true performance in equation (3.8) become constants.

In order to compute the expected true performance of the observed best, $E[J_{[\tilde{1}]}]$ in equation (3.8), we need to find the $\mu_{x(n_0)}$ and $\sigma_{x(n_0)}$ of the true performance, the noise to signal ratio, i.e. $\sigma_{N_0} / \sigma_{x(n_0)}$ and the $E[z_{[1]}]$. The values of $\mu_{x(n_0)}$, $\sigma_{x(n_0)}$ and $\sigma_{N_0} / \sigma_{x(n_0)}$ can be estimated from the screening experiment which will be discussed later. As for the $E[z_{[1]}]$, it has been tabulated in Lee and Chew (2004). We refer the $E[J_{[\tilde{1}]}]$ that we obtain from (3.8) as the "normal table" value.

3.2.2 Model Derivation for Weibull Distribution of True Performance

We expect that better designs are sampled when a more sophisticated sampling method is used and the distribution of the true performance will be skewed to the left. Hence the Weibull distribution will be used to approximate such distribution. The different degree of n_0 used in the sampling method will now affect the scale parameter $\alpha_{(n_o)}$ and shape parameter $\beta_{(n_o)}$ of the Weibull distribution. The same assumptions mentioned in

Section 2.1 are made for this case, except for assumption (1), where the true performance now follows Weibull distribution i.e. $J \sim W\left(\alpha_{(n_0)}, \beta_{(n_0)}\right)$. Note that equation (3.3) can be rewritten as,

$$\tilde{J} = \sigma_N \left[\frac{J}{\sigma_N} + \frac{w}{\sigma_N} \right] \tag{3.9}$$

or

$$\tilde{J} = \sigma_N \left[J' + w' \right]. \tag{3.10}$$

Denote $\tilde{J}' = J' + w'$ with

$$J' = \frac{J}{\sigma_N} \sim W(\alpha', \beta') \tag{3.11}$$

and

$$w' = \frac{w}{\sigma_N} \sim N(0,1). \tag{3.12}$$

The Weibull distribution in (3.11) has new parameter of α' and β' where,

$$\alpha' = \frac{\alpha_{(n_0)}}{\sigma_N} \quad \text{and} \quad \beta' = \beta_{(n_0)} \tag{3.13}$$

or

$$\alpha' = \frac{\alpha_{(n_0)}}{\sigma_{N_0} / \sqrt{n_2}} \quad \text{and} \quad \beta' = \beta_{(n_0)}. \tag{3.14}$$

The expected true performance of the observed best is

$$E[J_{[\tilde{1}]}] = \sigma_N E[J'_{[\tilde{1}]}] \tag{3.15}$$

or

$$E[J_{[\tilde{1}]}] = \frac{\sigma_{N_o}}{\sqrt{n_2}} E[J'_{[\tilde{1}]}]. \tag{3.16}$$

In order to compute the expected true performance of the observed best, $E[J_{[\tilde{1}]}]$ in equation (18), we need to find the σ_{N_0} and $E[J'_{[\tilde{1}]}]$. The value of σ_{N_0} can be estimated from the screening experiment, while the value of $E[J'_{[\tilde{1}]}]$ can be obtained from the "Weibull table" which we have developed for a general case. The detail steps on how to compile the Weibull table through the Monte Carlo simulation are summarized in the Appendix A.

Similar to the normal table, the Weibull table can be used to compare the performance of different computing allocations of n_0, n_1 and n_2 under a fixed computing budget. First, we estimate the σ_{N_o}, $\alpha_{(n_0)}$ and $\beta_{(n_0)}$ from the screening experiment. Given the n_0 and n_2, we then estimate the α' and β' using the equation (3.14). With the n_1, α' and β' values, we can now use the Weibull table to compute the expected true performance of the observed best, $E[J_{[\tilde{1}]}]$.

Chapter 4 ATO PROBLEM

4.1 Literature on ATO Problem

ATO is a policy widely applied in inventory policies among companies. Unlike the traditional way of Make-to-Stock which often results in high opportunity cost due to the mismatch between the demand and the supply, ATO, is an effective way which can help the companies to reduce the cost. In an ATO system, several different products will usually share the same components to make the end products. The components are typically stored as inventory until they are required for assembly when the demands arrive. Besides decreasing the total component inventory cost, such a policy will help reduce the safety stock levels owing to the risk pooling effects. Some of the available literatures on this research issue are as follows.

Baker (1985) showed the reduced number of safety stocks as a result of component commonality. However, it was highlighted that the link between safety factor and service level in commonality is more complicated than that of non-commonality. Gerchak and Henig (1986) formulated a profit maximization model for selecting optimal component stock levels for a single period in an ATO system. Under the commonality effect, it was shown that the stock level of the product-specific component is always higher compared to when one is operating under a non-commonality environment. The effect of commonality in two-product, two-component configuration with different component cost structure was also examined in a single period by Eynan and Rosenblatt (1996) (using the model of Baker (1985) and Baker et al. (1986)). If the common component was cheaper than the component that it replaced, it was always worthwhile to use the advantage of commonality. However, if the reverse was true, it was not always desirable to introduce commonality. Conditions were provided under which introducing commonality will reduce the inventory cost.

As the models described above all dealt with single period, Gerchak and Henig (1989) further extended to properties of ATO in a multi-period scenario, and proved that the solution is myopic. Hillier (1999) extended the two-product, two-level inventory model of Eynan and Rosenblatt (1996) in the multi-period environment to study the relative cost effectiveness of incorporating commonality. In contrast to the single-period model by Eynan and Rosenblatt (1996), Hillier (1999) and (2000) showed that the multi-period model almost never reflected any advantage in using common components when they were more expensive than the components it would replace.

The literatures discussed above are among the initial works that show the advantages and limitations in the application of component commonality. The following are some of the literatures on the various methods used to estimate the near optimal solution for the ATO problem.

Realizing that a single universal algorithm cannot be used to solve all stochastic models, Wets (1989) demonstrated that the major obstacle in solving the probabilistic constrained programming numerically, comes from the need to calculate gradients of the expectation function. An algorithm approach was taken by Kannan et al. (1995) where a randomized polynomial-time algorithm (based on random walk) was developed to achieve near optimal solution with high probability. The problem of minimizing total cost while satisfying the probability of meeting demands under a given stock level was modeled as a stochastic program with probabilistic constraints. Tayur (1995) modeled a cost minimization problem and solved the multi-period case by decomposing them into many single-period recourse problems. The derivatives of cost with respect to component stock levels were estimated using simulations and these estimates were later used to solve the optimal stocking levels for common components using a gradient search method.

The profit maximization problem for one and two common components in a single-period was solved analytically by Rudi (2000). An analytical characterization for the optimal inventory levels and some new insights in ATO systems were presented. Hillier (2000) developed a heuristic method which gives near optimal solution with the objective of minimizing production, holding and storage cost. Recently Mirchandani and Mishra (2002) considered three components (one common, and two product-specific components) to be assembled into two end products. Unlike Eynan and Rosenblatt (1996) who used aggregate service level in the model, they studied and compared the effect of product-specific service level constraints in both case of prioritized and non-prioritized products and solved a nonlinear program so as to obtain the optimal level of inventory.

From the literatures above, it is known that cost minimization in an ATO system is a hard problem to solve even for a single period model. The major obstacle in solving the stochastic optimization problem comes from the need to compute the expected cost. Hence in our work, we propose an approach using the sampling average approximation (SAA) method to tackle this problem.

4.2 ATO Model

In this thesis, we model a single period cost minimization ATO problem with stochastic demand. Holding and penalty costs are considered in the problem and we attempt to find the optimal inventory levels of components to be ordered which can minimize the total cost.

The assumptions made in the ATO problem are as follows. First, components are acquired to stock. Components are purchased only once to satisfy all future demands. When the demands of end products are known, the available components are allocated and assembled to satisfy the demand. In our problem, due to the uncertainties of demand, there will incur some holding cost for the excess component inventories that we hold. However, when there is a shortage of certain components required to assemble a specific product, the assembly of the end product cannot be completed and hence, it leads to unsatisfied demand and a penalty is associated with it. The following notations are used in the model:

h_i : holding cost of each excess inventory of component i

p_j : penalty imposed for the unsatisfied demand j

Q_i : inventory level of component i, $i = 1, 2, \ldots, n$

$\hat{Q}$: vector of component inventory levels

D_j : demand for product j, $j = 1, 2, \ldots, m$

$\hat{D}$: vector of product demands

S_j : number of product j to be assembled, $j = 1, 2, \ldots, m$

α_{ij} : number of component i needed to assemble one unit of product j, $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$

The α_{ij} S_j can be implied as the number of component i allocated for product j. In general, the problem can be modeled as follows,

P(1):

$$\min_{\hat{Q}} \quad E_{\hat{D}}\left[\sum_{i=1}^{n} h_i\left(Q_i - \sum_{j=1}^{m} \alpha_{ij} S_j(\hat{Q},\hat{D})\right) + \sum_{j=1}^{m} p_j\left(D_j - S_j(\hat{Q},\hat{D})\right)\right] \tag{4.1}$$

where for the given $\hat{Q}$ and $\hat{D}$, $S_j(\hat{Q},\hat{D})$ is the solution for

$$\min_{S_j} \quad \sum_{i=1}^{n} h_i\left(Q_i - \sum_{j=1}^{m} \alpha_{ij} S_j\right) + \sum_{j=1}^{m} p_j\left(D_j - S_j\right) \tag{4.2}$$

subject to

$$\sum_{j=1}^{m} \alpha_{ij} S_j \le Q_i \tag{4.3}$$

$$S_j \le D_j \tag{4.4}$$

$$Q_i \ge 0 \tag{4.5}$$

$$S_j \ge 0 \tag{4.6}$$

We want to find the minimum inventory of the components to stock over the time, $\hat{Q}$ which will be used to satisfy the future demand. The optimal allocation of the components for assemble depends on the inventory levels of components $\hat{Q}$ and demands of end products $\hat{D}$. Given these two variables, the optimum allocation S_j can be decided by solving the allocation problem (4.2) – (4.6). As our ability to fulfill the demand depend solely on the availability of components, our total allocations are no greater than the available inventories (4.3). Constraint (4.4) shows that we do not assemble the products more than its demand. Constraints (4.5) and (4.6) are the non-negativity constraints.

In theory, P(1) can be solved by finding the derivative of the cost function integral in (4.1) over all the component inventories to find the optimal level of inventories. However in this problem, the expectation function becomes complex and there is no close form solution for P(1). Hence in this section, we propose to use the SAA approach to sample the inventory levels, Q_i (the design) and then simulation is used to estimate the expected cost of every design, and the design with the lowest expected cost will be selected. In SAA, we approximate the objective function (4.1) in P(1) by using the sample average over n_0 demand realizations, and the problem P(1) can then be modified as P(2), which is given as follows,

P(2):

$$\min_{\hat{Q}} \ \frac{1}{n_0}\sum_{r=1}^{n_0}\left[\sum_{i=1}^{n} h_i\left(Q_i-\sum_{j=1}^{m}\alpha_{ij}S_j(\hat{Q},\hat{d}_r)\right)+\sum_{j=1}^{m}p_j\left(d_{rj}-S_j(\hat{Q},\hat{d}_r)\right)\right] \tag{4.7}$$

subject to the (4.2) - (4.6).

d_{rj} is the demand realization for product *j* at replication *r*, where $r = 1, 2, \ldots, n_0$ and $\hat{d}_r$ is a vector of demand realization at replication *r*. Since the demand d_{rj} is known, the problem now becomes deterministic and it can be solved by appropriate mathematical algorithm. Note that when n_0 approaches infinity, the minimum $\hat{Q}$ obtained will converge to the optimal solution $\hat{Q}^*$. However solving for very large n_0 is very time-consuming. Alternatively, we can reduce the n_0 and repeat solving the problem P(2) several times by using different sets of demand realizations. Every time when the $\hat{Q}$ is obtained, it can be treated as a design, and we hope that some good designs will be sampled. We then further spend more time to run simulation on these sampled designs to estimate the performance of each design. The following algorithm describes how SAA is used as the sampling method in this problem.

Algorithm:

Step 1: The demand vector $\hat{d}_r$ is generated at replication *r*, where $r = 1, 2, \ldots, n_0$.

Step 2: The SAA problem P(2) is solved as a linear programming problem with $\hat{Q}$ and $\hat{S}$ being the decision variables, and the $\hat{Q}$ obtained will be considered as the design.

Step 3: In order to randomly sample n_1 different designs, the SAA problem in Step 2 is solved repeatedly using n_1 randomly generated sets of demand vectors $\hat{D}$.

Step 4: For each design obtained in Step 3, we run simulation with n_2 replications to estimate its performance. To run one replication of simulation for each sampled design (the design $\hat{Q}$ is fixed), one set of demand vector $\hat{D}$ is generated and its observed performance is computed. This step is repeated n_2 times and the mean of the observed performance (average cost) is recorded.

Step 5: The n_1 designs are then ranked based on the observed performance values, and the design which has the minimum cost will be picked as the best design.

For comparison purpose, we also simulate the observed best design with a very large number of replications in order to estimate its true performance value.

4.3 A Review on SAA

SAA is a Monte Carlo simulation-based approach and it appears as an appealing method in solving the stochastic optimization problem. The SAA sampling method results in better designs compared to the random sampling, as it utilizes uncertain demand information. In many scenarios, the SAA method can be very efficient and easily implementable. Besides having good convergence properties, often one can use the existing software due to the ease of numerical implementation introduced by this method. SAA approach is also easily amendable to variance reduction techniques and at the same time is ideal for parallel computations. The various properties of SAA method was discussed in Shapiro (2001). The statistical inference on the convergence rate of SAA was further discussed by Kleywegt et al. (2001). They showed that with the increase of sample size, the probability approaches one at exponential fast rate.

This indicates that an optimal solution of the SAA problem provides an exact optimal solution of the true problem. The SAA approach is then applied to a stochastic knapsack problem. Note that the SAA method is not an algorithm since the user still has to choose a particular numerical procedure in order to solve the SAA problem in P(2).

The idea of SAA is simple and natural. The basic idea is that a random sample is generated and the expected value function is approximated by the corresponding sample average function. The obtained sample average optimization problem is solved, and the procedure is repeated several times. The method of SAA however is noticed to have some limitations that have to be addressed. We discuss on the existing approach and present how our approach can complement the existing technique.

It is explained in the literature that the SAA problem can be solved repeatedly M times (resulting in M designs) using N independent random samples for each SAA problem. The probability of finding the optimal design increases with larger M and N. Kleywegt et al. (2001) proposed to use a stopping criterion based on the optimality gap to decide on the optimal M and N. In their algorithm, the M and N are adjusted dynamically, depending on the results of preliminary computations until the optimality gap is satisfied. Note that this procedure may become critical particularly for our case when there is only a limited computing budget. In such case, we have to wisely select on the optimal M and N so as to increase the chances of obtaining the good enough design. Moreover, as discussed in Section 3.2, solving with large M and N alone do not necessarily guarantee the selection of good design. Furthermore, solving for large N can be very time-consuming. Hence, as discussed in the previous section, besides reducing the N and repeat solving the SAA problem for M times, we further spend some time to run simulation on the sampled designs to reduce the noise in the performance estimation of each design. We will finally select the design with the minimum performance value as the good enough design under the given computing budget.

Chapter 5 NUMERICAL RESULT OF ATO PROBLEM

5.1 Conducting the Numerical Experiment

In this chapter, we describe and present the numerical result for selecting the best allocation decisions for two different ATO problems. The first ATO problem (Problem I) has 2 common components and 3 end products while the second problem (Problem II) has 6 common components and 9 end products. All the simulation works are carried out using a Pentium(R) IV computer (CPU 2.40GHz and 512MB of RAM). The Solver in Microsoft Excel is used to solve the SAA problem.

The presentation of our numerical result is organized as follows. First in Section 5.2, we describe on how the screening experiment is carried out to estimate the relevant parameters for our problem. Section 5.3 and Section 5.4 present the detailed problem descriptions and the numerical results for Problem I and Problem II respectively. For comparison of different scenarios in Problem I, we split the numerical result presentation into three cases, and the numerical results are approximated by using the normal table estimation, the Weibull table estimation and the simulation result. In the first case, we do not use the SAA as the sampling method, but instead used random sampling to sample the designs. This implies that the term of n_0 does not exist in this case. The objective is to find the best allocation decision pair of (n_1, n_2) among the different combinations when we random sample the designs. The underlying distribution of the true performance and the noise is referred to check on the appropriateness of the normal and Weibull table estimation in selecting the best allocation decision. In the second case, we use SAA as the sampling method and the numerical result when n_0 is fixed is presented. Similar to the first case, the best allocation decision pair of (n_1, n_2) is selected and the appropriateness of the normal and Weibull table estimations is discussed when the SAA is used as the sampling method. As the second case is restricted for a fixed n_0, we generalize our problem in the third case, where the n_0 is varied as well. We developed the OCBA model for Problem I in order to find the optimal computing budget allocation decision of (n_0, n_1, n_2) that minimizes the expected true performance of the observed best when the computing budget is fixed at a certain level for this problem. As for Problem II, we directly generalized the problem and presented the third case to illustrate the applicability of our approach.

5.2 Screening Experiment

In order to know how to determine the optimum allocation decisions using the normal and Weibull table estimations, we first run a screening experiment to estimate the required parameters. For the normal table, we estimate the standard deviation of the performance ($\sigma_{x(n_0)}$), the standard deviation of the noise (σ_{N_0}) and the noise to signal ratio ($\sigma_{N_0} / \sigma_{x(n_0)}$). For the Weibull table, we estimate the parameters $\alpha_{(n_o)}$ and $\beta_{(n_o)}$. To estimate these values, we sample 25 designs and then run 50 replications for each design.

The sampling in screening experiment is conducted differently for the designs that are randomly sampled and for the designs that are sampled using the SAA method. For the random sampling method, the 25 designs are randomly sampled from a Uniform distribution, between 0 and 4,000, i.e. $Q \sim U(0, 4000)$ (These parameter values of 0 and 4000 are obtained based on the initial trial run). As for the designs sampled using the SAA method, the 25 designs are sampled using SAA with n_0 degree of information. As we assume, the demands for all the end products are drawn from a normal distribution, $D \sim N(1000,100)$. From our initial trial run, it was also learned that the distribution of true performance of the designs sampled by SAA tends to favor good designs and the value of $\beta_{(n_o)}$ is close to one. This implies the distribution is an exponential distribution, which is a special case of Weibull distribution. Hence in this application problem, we fix $\beta_{(n_o)}$ at the value of one ($\beta_{(n_o)} = \beta' = 1$) for the cases when the designs are sampled by SAA.

5.3 Problem I: Problem Description

As mentioned in the beginning of the chapter, the first example of our ATO problem has 2 common components and 3 end products. For ease of reference, we refer to this problem as Problem I. Figure 5.1 shows the component allocation network of Problem I. In this problem, either one or none of component *i* is used to assemble one unit of product *j*. Thus, the number of component *i* needed to assemble one unit of product *j*, α_{ij} takes the value of either 1 (if allocated) or 0 (if not allocated), i.e. ($\alpha_{11}, \alpha_{12}, \alpha_{22}, \alpha_{23}$ = 1 and α_{13}, α_{21} = 0). Every component is used in assembling two end products. For product 1 and product 2, the component itself is made into the end product. The product 2 is however assembled with one unit of component 1 and one unit of component 2.

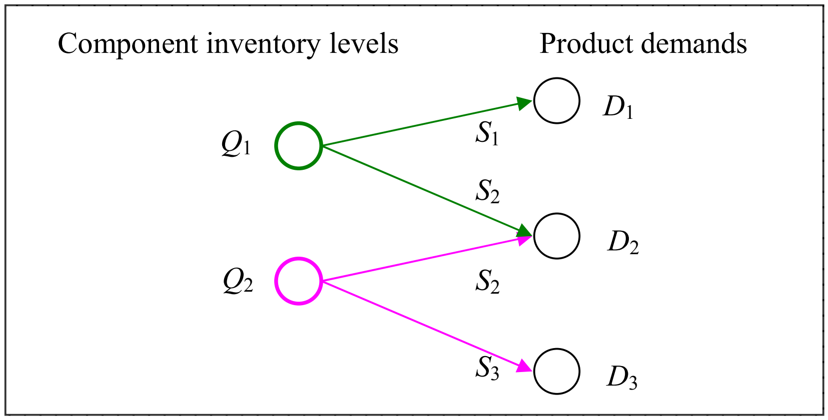

Figure 5.1: Problem I - 2 common components and 3 end products

Assume that all the components have the same holding cost h and all the products have the same penalty cost p, the SAA formulation for Problem I is given as P(1).

P(1):

$$\min_{Q_1,Q_2} \frac{1}{n_o} \sum_{r=1}^{n_o} \left[h\,(Q_1 + Q_2 - S_{r1} - 2S_{r2} - S_{r3}) + p\,(D_{r1} + D_{r2} + D_{r3} - S_{r1} - S_{r2} - S_{r3}) \right]$$

subject to

$$S_{r1} + S_{r2} \leq Q_1 \qquad \forall\, r$$

$$S_{r2} + S_{r3} \leq Q_2 \qquad \forall\, r$$

$$S_{r1} \leq D_{r1} \qquad \forall\, r$$

$$S_{r2} \leq D_{r2} \qquad \forall\, r$$

$$S_{r3} \leq D_{r3} \qquad \forall\, r$$

$$Q_1, Q_2, S_{r1}, S_{r2}, S_{r3} \geq 0 \qquad \forall\, r$$

In this special case, given the optimum component inventory level vector $\hat{Q}^*$ and the product demand vector $\hat{D}$ are known, the component allocation quantity S_j can be determined using the optimal allocation rule below:

$$S_1 = \min\{Q_1 * - S_2,\ D_1\}$$

$$S_2 = \min\{\max\{Q_1 * - D_1, Q_2 * - D_3, 0\}, D_2,\ Q_1 *, Q_2 *\}$$

$$S_3 = \min\{Q_2 * - S_2,\ D_3\}$$

The performance (minimum cost) based on the optimum allocation rule above can be computed as follows:

$$\text{min cost} = h(Q_1 * + Q_2 * - S_1 - 2S_2 - S_3) + p(D_1 + D_2 + D_3 - S_1 - S_2 - S_3)$$

The above optimal allocation rule and performance computation will be used when running simulation in Step 4 of Section 4.2. Note that the above optimum allocation rule is developed for a general case of Problem I. The rules for specific conditions of Problem I, i.e. when (D_1 + D_2) and (D_2 + D_3) is greater or equal and lesser or equal than Q_1* and Q_1* are given in Appendix B.

For our numerical experiment, the holding cost h is fixed at \$0.20 and the penalty cost p is fixed at \$0.50. In the following subsections, we present and discuss the numerical result for Problem I when the designs for this problem are random sampled and when the designs are sampled using the SAA method (for n_0 fixed and n_0 varied).

5.3.1 Numerical Result for Problem I for Case I : designs sampled by random sampling

For the first case, we want to select the best allocation decision of (n_1, n_2) when the designs are randomly sampled. In the randomly sampled designs, there is no information used in the sampling and thus we expect only mediocre designs to be sampled. As mentioned in the screening experiment, the designs are randomly sampled from $Q \sim U(0, 4000)$ and the demands of the end products are drawn from $D \sim N(1000,100)$. The computing budget is fixed at 25,000 runs (or 25,000 sets of demand vectors) and different pairs of (n_1, n_2) that satisfies (n_1 x n_2 = 25,000) are chosen. The screening experiment is first run and the information obtained from the screening experiment is as given below,

The estimated mean of the true performance distribution, $\mu_{x(n_0)}$ = \$ 707.40

The estimated standard deviation of the true performance distribution, $\sigma_{x(n_0)}$ = 329.51

The estimated standard deviation of the noise, σ_{N_0} = 64.78

The noise to signal ratio, $\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}}$ = 0.20

The estimated Weibull scale parameter, $\alpha_{(n_0)}$ = 638.47

The estimated Weibull shape parameter, $\beta_{(n_0)} = \beta' = 5.09$

The estimated Weibull location parameter, $\gamma_{(n_0)}$ = \$ 84.36

Based on the values obtained from the screening experiment, the normal and Weibull table estimations can be computed. Table 5.1 summarizes the numerical result for

Problem I when the designs are randomly sampled. We elaborate on how we obtain the normal and Weibull estimations and the simulation result in the following part.

Table 5.1: Numerical result for Problem I, designs randomly sampled

Computing Budget Allocation		**Normal table Estimation ($)**	**Weibull table Estimation ($)**	**Simulation Result ($)**
n_1	n_2			
5000	**5**	**703.74**	**200.54**	**67.02**
2500	10	703.91	216.38	68.86
1000	25	704.16	238.21	68.41
500	50	704.36	256.66	69.20
200	125	704.65	291.96	84.93
100	250	704.89	323.13	98.10

There are four main columns in Table 5.1; the computing budget allocation, the normal table estimation, the Weibull table estimation and the simulation result. The computing budget allocation column summarizes the different pairs of (n_1, n_2) run in the experiment. The normal and Weibull table estimation columns show the expected true performance of the observed best design $E[J_{[\tilde{1}]}]$ when the distribution of true performance is normal and Weibull respectively. For the normal table estimation, first the noise to signal ratio for each of the (n_1, n_2) pair is computed by $\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$. Based on this value, we subsequently refer to Lee and Chew (2004) for its normal table value. As this is a minimization problem, the normal table value obtained is multiplied with (-1). Finally the estimated mean of the true performance distribution, $\mu_{x(n_0)}$ (a constant) is added to each of the normal table value, and the end result is referred as the normal table estimation, i.e. $E[J_{[\tilde{1}]}]$. For the Weibull table estimation, the noise for each of the (n_1, n_2) pair is computed by $\frac{\sigma_{N_0}}{\sqrt{n_2}}$. At the same time, the Weibull table is referred to obtain the value of $E[J'_{[\tilde{1}]}]$. These two values are then multiplied together, i.e. $\frac{\sigma_{N_o}}{\sqrt{n_2}} E[J'_{[\tilde{1}]}]$. As we have initially removed the location parameter of $\gamma_{(n_0)}$ for ease of numerical computation, we now add the constant to $\frac{\sigma_{N_o}}{\sqrt{n_2}} E[J'_{[\tilde{1}]}]$ and the end result is the Weibull table estimation, i.e. $E[J_{[\tilde{1}]}]$ for Problem I. The detailed computation on how to compute the normal and Weibull table estimations based on the screening experiment values is presented in Appendix C. A long simulation (for 10,000 runs) is also run to estimate the expected true performance of the observed best design $E[J_{[\tilde{1}]}]$. Each pair of (n_1, n_2) in the experiment is repeated for 20 times and the average of the true performance is recorded in the simulation result column. From this column, we will be able to determine the optimal allocation decision, i.e. the (n_1, n_2) that gives the minimum expected true performance. This column will be used for comparison purposes.

The optimal allocation decision of (n_1, n_2) based on the simulation result is given in bold. For comparison purpose, the minimum expected cost suggested by the normal and Weibull table estimations are also given in bold, and the corresponding allocation decisions are the optimal selection suggested by them respectively. The allocation decisions suggested by the normal and Weibull table estimations are then compared with the optimal decision based on the simulation result. From the simulation result column, the optimal allocation decision for this case of Problem I is found to be (n_1, n_2) = (5000, 5), i.e. to sample 5000 designs and to run only 5 replications, and the expected minimum cost is \$67.02. The normal and Weibull table estimations select the same allocation decision with the expected minimum cost of \$703.74 and \$200.54 respectively.

We observe from the table that both the normal and Weibull table estimations are able to suggest the correct allocation decision. This is due to the fact that the true performance for the randomly sampled designs tends to have a mediocre distribution as shown in Figure 5.2. (This figure is obtained from an experiment run with very large number of designs and very long replications and such experiment is referred as "detailed experiment"). Therefore both the estimations are able to identify the correct selection. Moreover, the noise of the randomly sampled designs is also normally distributed (Figure 5.3) which is consistent with the underlying assumption made in the normal model development in Chapter 3.

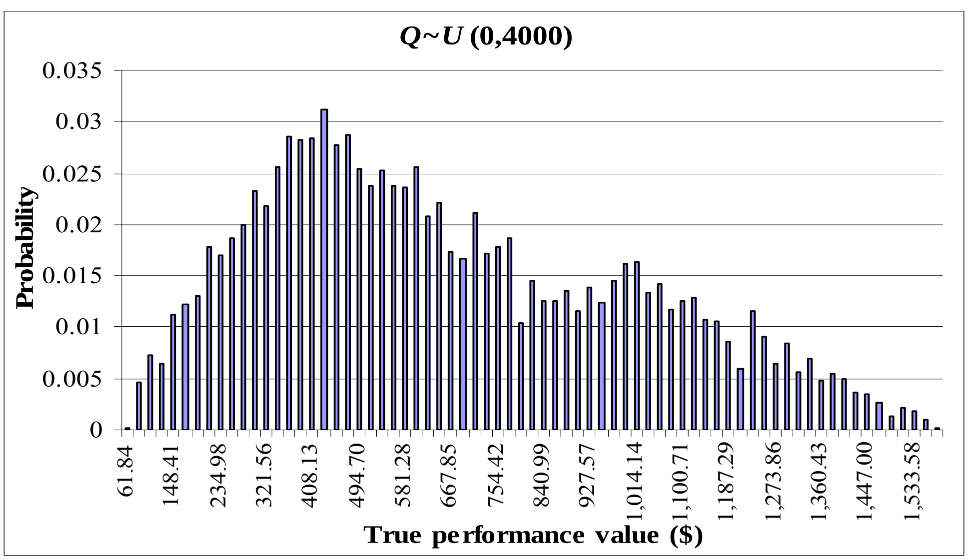

Figure 5.2: The distribution of the true performance for randomly sampled designs $Q \sim U(0,4000)$ for Problem I

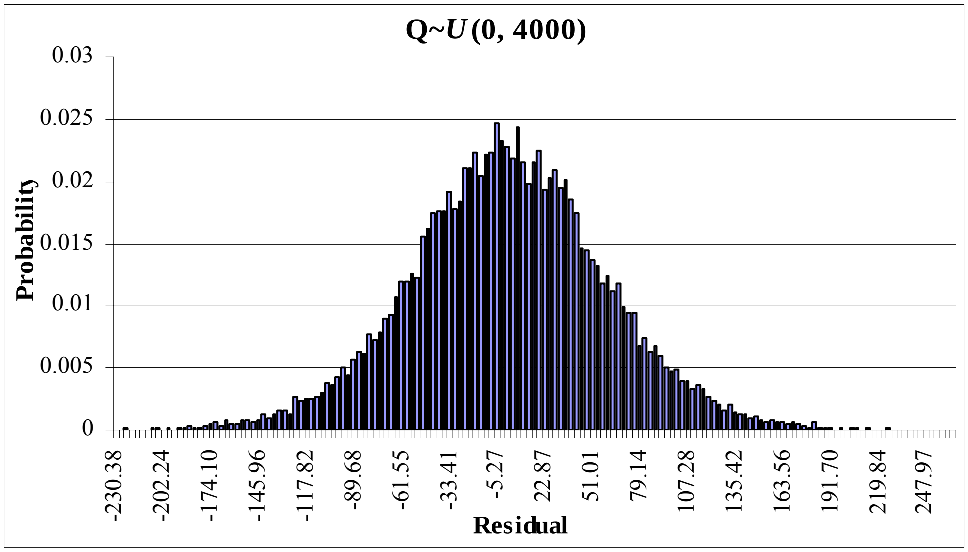

Figure 5.3: The distribution of the noise for randomly sampled designs Q~U(0,4000) for Problem I

5.3.2 Numerical Result for Problem I for Case II : designs sampled by SAA, n_0 fixed

In this section, we discuss the numerical result for Case II of Problem I when the SAA is used as the sampling method. In this case, the n_0 is fixed at certain level and the SAA is solved to sample the designs. For the first scenario of Case II, we fix n_0 at 5, and would like to select the optimal allocation decision for Problem I. We would also like to compare the optimal result suggested by both the normal and Weibull table estimations with the simulation result. Similar to Case I, the demands are also drawn from normal distribution, D~N(1000,100) and the computing budget is fixed at 25,000 runs. A few different pairs of (n_1, n_2) that satisfy the computing budget are chosen as the potential candidate for the best allocation decision.

In contrast with the random sampling, the designs sampled by SAA tend to sample better designs as some information is used in the sampling scheme. Thus, as mentioned earlier in Section 5.2 of this chapter, $\beta_{(n_0)}$ is fixed to 1 ($\beta' = \beta_{(n_0)} = 1$) when the SAA is used in sampling the designs. The screening experiment is first run and the information of the parameters is as given below,

$n_0 = 5$
$\mu_{x(n_0)} = 85.63$
$\sigma_{x(n_0)} = 8.43$
$\sigma_{N_0} = 40.91$
$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 4.83$
$\alpha_{(n_o)} = 13.74$
$\beta' = \beta_{(n_0)} = 1$
$\gamma_{(n_0)} = 74.46$

Table 5.2: Numerical result for Problem I, designs sampled using SAA ($n_0 = 5$)

Computing Budget Allocation		Normal table Estimation ($)	Weibull table Estimation ($)	Simulation Result ($)
n_1	n_2			
5,000	5	84.10	77.83	71.61
2,500	10	83.71	77.30	65.90
1,000	25	83.30	76.67	65.64
500	50	83.13	76.21	63.62
200	125	**83.10**	75.81	62.66
100	**250**	83.23	**75.52**	**61.85**

The numerical result for $n_0 = 5$ is presented in Table 5.2. The detailed computation on how to compute the normal and Weibull table estimations based on the screening experiment values is presented in Appendix D. Based the simulation result column in Table 5.2, the optimal allocation decision is found to be (n_1, n_2) = (100, 250), i.e. to sample 100 designs and to run 250 replications. This optimal allocation decision is expected to result in a minimum cost of $61.85. The Weibull table estimation suggests the same allocation decision option as the optimal decision, and the expected minimum cost is $75.52. The normal table estimation however picks a different allocation decision option, (200, 125) as the optimal selection with the expected cost of $83.10.

Unlike the example of Case I, in this case we observe from the table that only the Weibull table estimation is able to suggest the correct allocation decision. This is because the true performance for the designs sampled by SAA is exponentially distributed as shown in Figure 5.4. The noise is also normally distributed as shown in Figure 5.5. Both the distributions are consistent with the assumptions made in the Weibull model development in Chapter 3. (These distribution graphs are obtained from the detailed experiment).

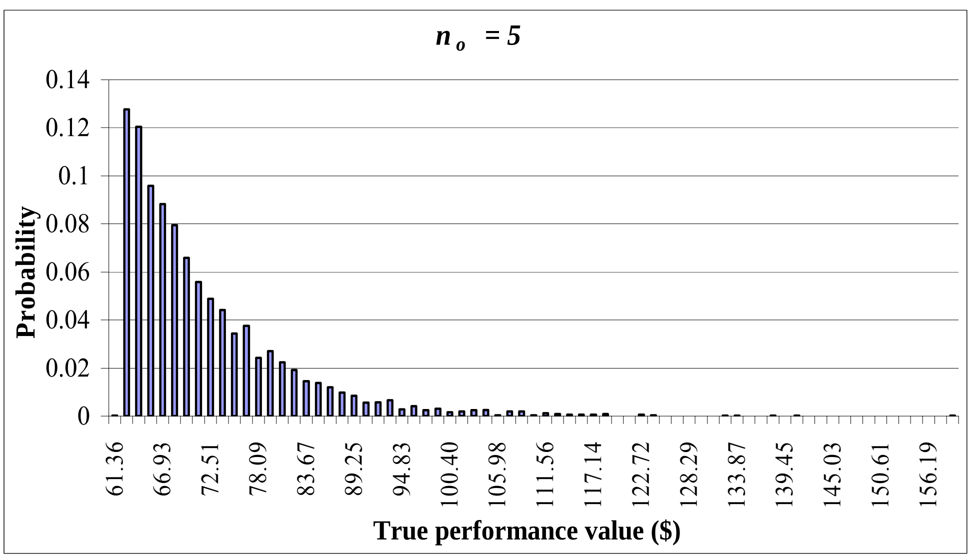

Figure 5.4: The distribution of the true performance value for SAA sampled designs ($n_0 = 5$) for Problem I

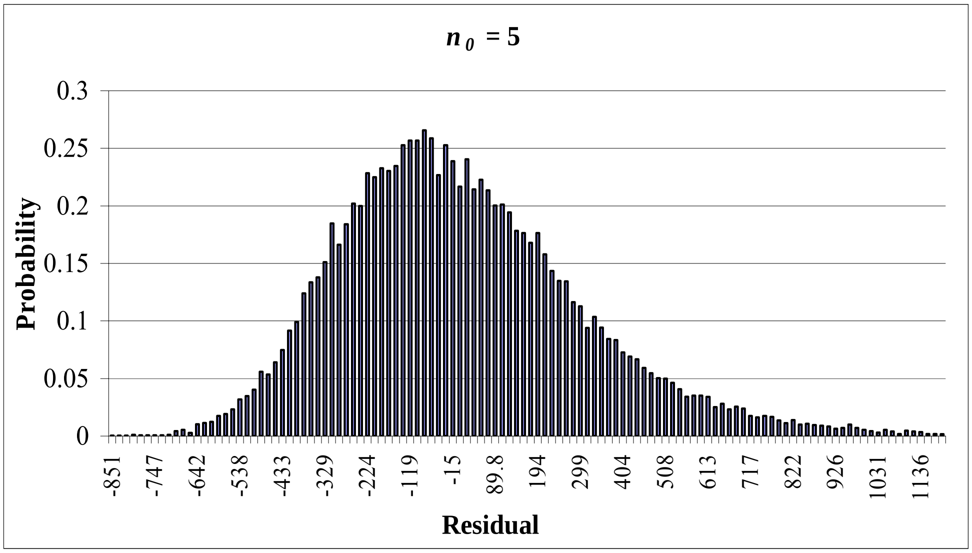

Figure 5.5: The distribution of the noise for SAA sampled designs ($n_0 = 5$) for Problem I

Therefore when the designs are sampled using SAA, the estimation by the Weibull is a better approximation for the correct selection as compared to the normal. Also, note that since we are only interested in determining the best allocation decision, we can do so by computing the Weibull table estimation alone, without having to run long simulation to estimate the simulation result.

The same experiment is repeated, with the n_0 fixed at 20 and 50. The values of screening experiment and numerical results are presented in Table 5.3 and Table 5.4 respectively. Also refer to Appendix D for the detailed computation.

$n_0 = 20$

$\mu_{x(n_0)} = 79.97$

$\sigma_{x(n_0)} = 5.67$

$\sigma_{N_0} = 34.85$

$\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 6.15$

$\alpha_{(n_o)} = 7.32$
$\beta' = 1$
$\gamma_{(n_0)} = 73.65$

Table 5.3: Numerical result for Problem I, designs sampled using SAA ($n_0 = 20$)

Computing Budget Allocation n_1	n_2	**Normal table Estimation ($)**	**Weibull table Estimation ($)**	**Simulation Result ($)**
5,000	5	78.73	76.00	63.22
2,500	10	78.37	75.75	62.90
1,000	25	77.92	75.29	62.75
500	50	77.68	74.98	62.35
200	125	**77.56**	74.70	62.26
100	**250**	77.63	**74.50**	**62.03**

$n_0 = 50$
$\mu_{x(n_0)} = 77.33$
$\sigma_{x(n_0)} = 5.35$
$\sigma_{N_0} = 34.74$
$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 6.49$
$\alpha_{(n_o)} = 5.09$
$\beta' = 1$
$\gamma_{(n_0)} = 73.58$

Table 5.4: Numerical result for Problem I, designs sampled using SAA ($n_0 = 50$)

Computing Budget Allocation n_1	n_2	**Normal table Estimation ($)**	**Weibull table Estimation ($)**	**Simulation Result ($)**
5,000	5	76.13	75.84	63.05
2,500	10	75.83	75.49	62.45
1,000	25	75.35	75.07	62.29
500	50	75.10	74.82	62.05
200	**125**	**74.95**	**74.51**	**61.91**
5,000	5	76.13	75.84	63.05

The best allocation decision for n_0 = 20 is (100, 250) with the expected minimum cost of $62.03, and for n_0 = 50 is (200, 125) with $61.91. Both the tables indicate that Weibull makes a correct selection, while normal only makes correct selection when n_0 = 50. This shows that the Weibull table estimation is quite promising in selecting the optimal allocation decision.

5.3.3 Numerical Result for Problem I for Case III : designs sampled by SAA, n_0 varied

In order to generalize Problem I in the OCBA model, we allow the n_0 to vary. In this case, the computing budget *K* is no longer in terms of number of runs, but we fix the *K* in terms of CPU time, i.e. 800 seconds and 3,600 seconds. The demands are still drawn from normal distribution, $D \sim N(1000,100)$. The Problem I is solved repeatedly with different combinations of (n_0, n_1, n_2) that satisfy the computing budget constraint. We attempt to find the optimal computing budget allocation decision of (n_0, n_1, n_2) that can minimize the expected cost of the problem. Note that in order to model the OCBA for Problem I, we have to first estimate the time to generate one design, $t_{(n_o)}$ in terms of n_0 and the simulation time to run one replication of simulation, *s*. Before developing the OCBA model for this problem, we look at how the different degree of n_0 used in the SAA sampling method affects the true performance. For the Weibull table computation, the empirical relationship of $\alpha_{(n_o)}$ when the n_0 is varied is also estimated.

Figure 5.6 shows the cumulative distribution function (CDF) of the true performance of the designs generated by SAA when n_0 is fixed at 1, 5, 25 and 50. The screening experiment is used to plot the CDF. It can be observed that when more information is supplied (the higher n_0), the better designs will be sampled. The standard deviation of true performance also decreases as the higher n_0 is used in the SAA sampling.

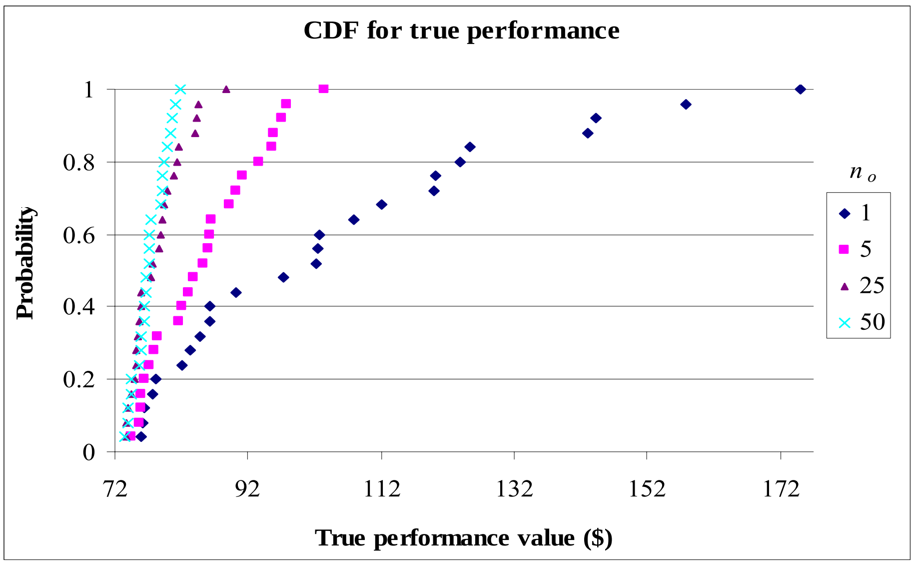

Figure 5.6: The improvement in the true performance value when n_0 is varied in Problem I

For simplicity in Figure 5.6, we present the distribution of true performance for only four different degrees of n_0. In our research work, we actually experimented for ten different degrees of n_0 (n_0 = 1, 3, 5, 10, 15, 20, 25, 30, 40, 50) using separate detailed experiments. The CDF of true performance, the distribution of true performance, the minimum true performance, the maximum true performance and the standard deviation of the true performance for each of the different n_0 experimented are recorded and presented in Appendix E. The observed best design, which gives the minimum true performance for each n_0 is also provided in the appendix. Similar findings with Figure 5.6, it can be observed from the Appendix E that better designs will be generated with higher degree of n_0. Also, the higher the degree of n_0, the smaller the standard deviation of true performance as more information is supplied in the sampling of the designs.

As discussed earlier, the $\alpha_{(n_o)}$ and $\beta_{(n_0)}$ are the two parameters estimated for the Weibull table. The $\beta_{(n_0)}$ is fixed to 1, and the remaining task is to estimate the $\alpha_{(n_o)}$ for different degrees of n_0. Figure 5.7 shows that the regression analysis for the estimated parameter of $\alpha_{(n_o)}$ when n_0 is varied. These values are computed from screening experiments. From this figure, observe that $\alpha_{(n_o)}$ decreases exponentially when n_0 increases. Using the least square method, the empirical relationship is given as $\alpha_{(n_o)} = 26.774\ n_0^{-0.4626}$. The coefficient of determination, R^2 is also measured to judge the adequacy of the model, and the exponential correlation appears as the best fit. This function can be used to estimate the value of $\alpha_{(n_o)}$ for any n_0 that lies between 1 and 50.

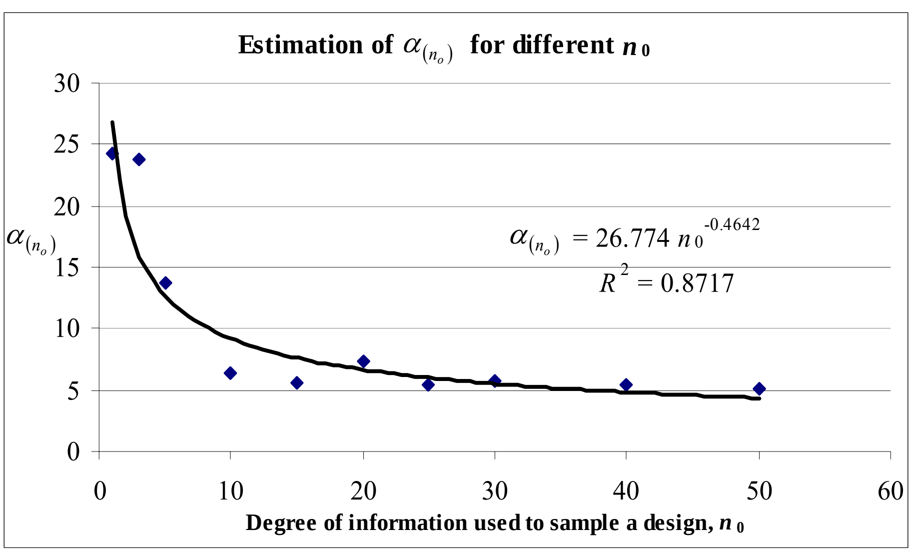

Figure 5.7: Estimation of $\alpha_{(n_o)}$ for Problem I

In order to develop the OCBA model for Problem I, we now estimate the function of the time to generate one design, $t_{(n_o)}$ in terms of n_0 and the simulation time to run one replication of simulation, s by using regression analysis. Figure 5.8 depicts the estimation of the $t_{(n_o)}$ in terms of the CPU time. It can be observed that the time we need to solve the SAA, $t_{(n_o)}$, increases when n_0 is increased, and the empirical relationship is $t_{(n_o)} = 0.0014\, n_0^{\,2} - 0.0062\, n_0 + 0.3496$. This empirical relationship fits the scatter diagram well with a high coefficient of determination ($R^2 = 0.998$). As for the estimation of s, the CPU time to perform the simulation, S is seen to increase linearly with the simulation length, n_2. This is represented in Figure 5.9. Note from this figure that the fitted linear regression line passes through many of the points ($R^2 = 0.9994$). The regression coefficient is the simulation time for one replication of simulation, s and it is estimated to be $1.5 \text{ X } 10^{-3}$ seconds. (For the numerical estimation of $\alpha_{(n_o)}$, $t_{(n_o)}$ and s for Problem I, please refer to Appendix E)

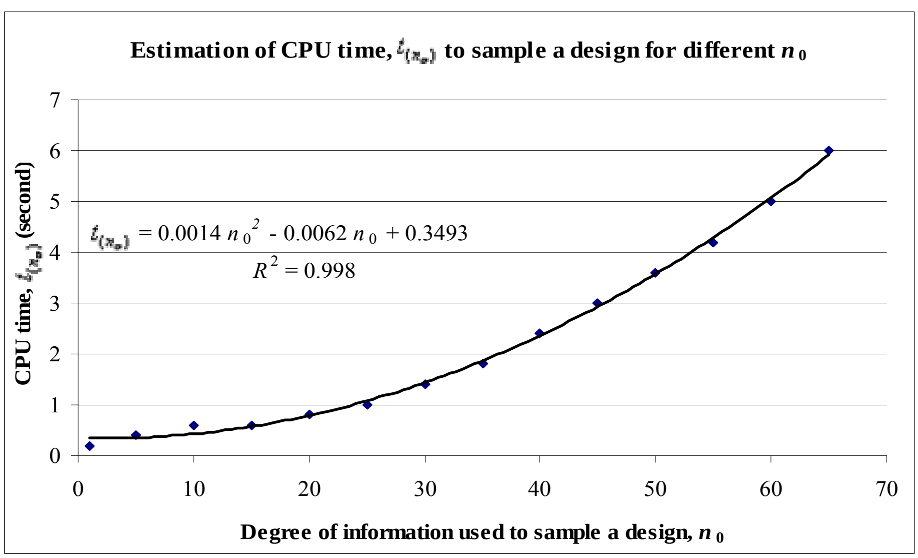

Figure 5.8: Estimation of $t_{(n_o)}$ for Problem I

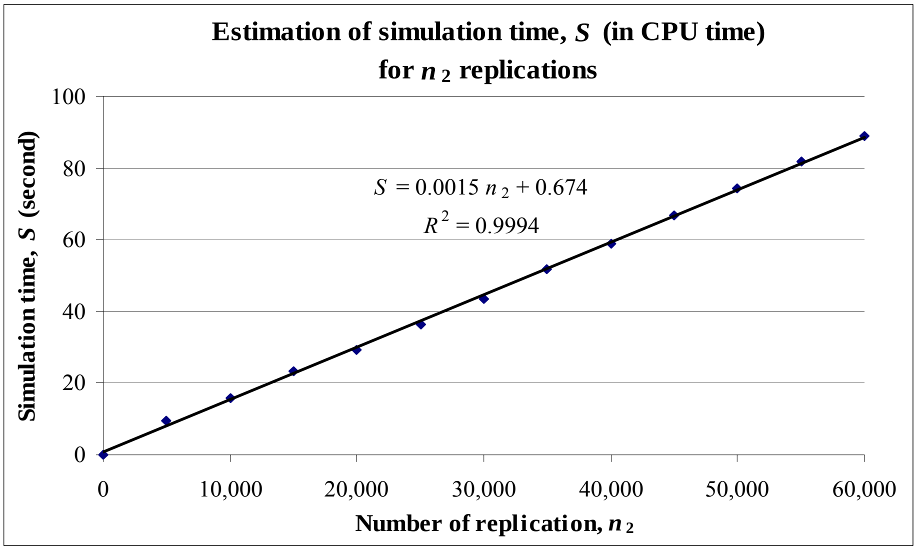

Figure 5.9: Estimation of *s* for Problem I

Based on the estimations, the OCBA model for Problem I is as follows,

P(6):

$$\min E\,[J_{[\tilde{1}]}] \tag{5.1}$$

subject to

$$(\,1.4\text{ X }10^{-3}\,n_0^{\;2} - 6.2\text{ X }10^{-3}\,n_0 + 3.493\text{ X }10^{-1} + 1.5\text{ X }10^{-3}\;n_2\;)\;n_1 \le\; K \tag{5.2}$$

$$E[J_{[\tilde{1}]}] \;=\; \frac{\sigma_{N_o}}{\sqrt{n_2}}\,E[J'_{[\tilde{1}]}] \tag{5.3}$$

Given K = 800 seconds, a few possible combinations of (n_0, n_1, n_2) that satisfy constraint (5.2) is selected. As discussed before, the screening experiment is first conducted with (n_0, 25, 50) and the information obtained is presented in Appendix F. The normal and Weibull table estimations are compared with the simulation result in Table 5.5 and the best allocation decision of (n_0, n_1, n_2) = (15, 200, 2290) which results in \$61.69 is given in bold. Based on this table, the Weibull again chooses the right allocation decision with the minimum expected cost of \$73.97. However, the normal table estimation fails to indicate the correct decision. It selects the higher n_0 and lower n_2, i.e. (50, 200, 300) as the best allocation.

Table 5.5: Numerical result for Problem I, designs sampled using SAA with n_0 varied (K =800 seconds)

Computing Budget Allocation			**Normal table Estimation (\$)**	**Weibull table Estimation (\$)**	**Simulation Result (\$)**
n_0	n_1	n_2			
5	1,500	120	82.55	75.60	63.78
15	**200**	**2,290**	75.73	**73.97**	**61.69**
20	800	145	77.14	74.48	62.39
50	200	300	**74.75**	74.21	62.06

The experiment is repeated when K = 3,600 seconds and the information on screening experiment and the computation are also presented in Appendix F. The numerical result is summarized in Table 5.6. The optimal allocation decision is (65, 500, 900) with \$61.67. Similarly, the Weibull is able to pick the correct decision with the expected minimum cost of \$73.89. In this case, normal estimation is also able to indicate the correct selection with the expected minimum cost of \$74.03. Also note that with higher computing budget time, better results can be achieved.

Table 5.6: Numerical result for Problem I, designs sampled using SAA with n_0 varied (K=3,600 seconds)

Computing Budget Allocation			**Normal table Estimation (\$)**	**Weibull table Estimation (\$)**	**Simulation Result (\$)**
n_0	n_1	n_2			
20	4,500	10	78.30	75.46	62.61
40	1,500	40	75.32	74.89	61.93
60	650	350	74.24	74.08	61.84
65	**500**	**900**	**74.03**	**73.89**	**61.67**

Based on all the experiments conducted in Case II and Case III for Problem I, the Weibull table estimation shows promise for making the correct allocation decision when the SAA is used as the sampling method. Hence, note that in future we can decide on the optimal allocation decision based on the Weibull table estimation alone.

5.4 Problem II : Problem Description

We now apply the same approach to another example of ATO problem. In this example, the ATO problem is made of 6 common components and 9 end products. Each common component is used in assembling three different end products, and each end product is assembled from two different common components. Hereafter, this problem will be referred as Problem II. The configuration of Problem II is illustrated in Figure 5.10.

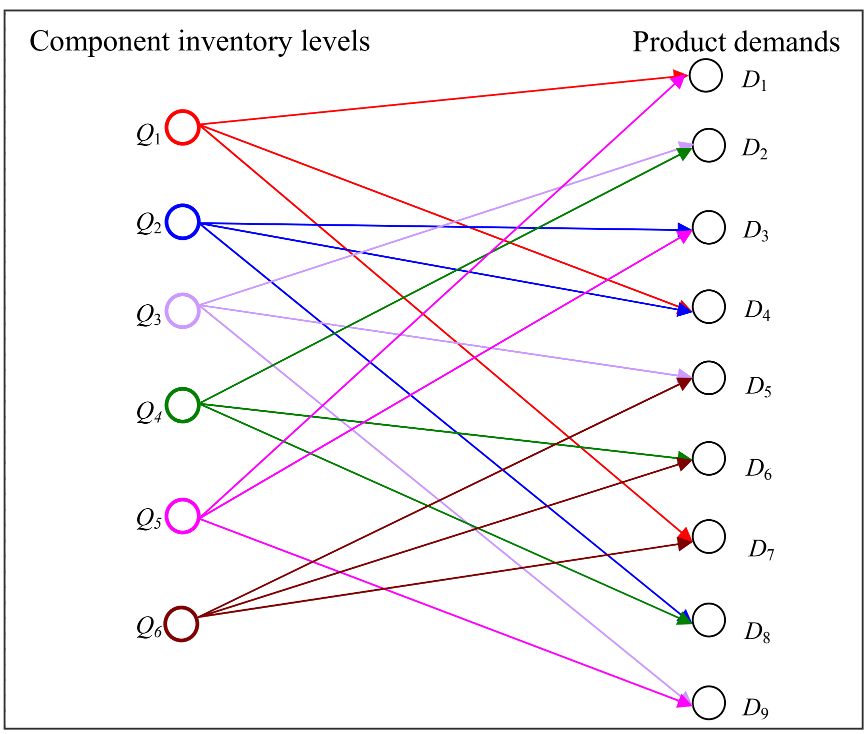

Figure 5.10: Problem II - 6 common components and 9 end products

In this problem, the number of component i needed to assemble one unit of product j, α_{ij} takes different values in each allocation. Different holding cost for component i, h_i and different penalty cost for the unsatisfied demand of product j, p_j are also imposed in Problem II. With these assumptions, the SAA formulation for Problem II is,

P(7):

$$\min_{Q_i} \frac{1}{n_o} \sum_{r=1}^{n_0} \begin{aligned} & h_1[Q_1 - (\alpha_{11}S_{r1} + \alpha_{14}S_{r4} + \alpha_{17}S_{r7})] \\ & + h_2[Q_2 - (\alpha_{23}S_{r3} + \alpha_{24}S_{r4} + \alpha_{28}S_{r8})] \\ & + h_3[Q_3 - (\alpha_{32}S_{r2} + \alpha_{35}S_{r5} + \alpha_{39}S_{r9})] \\ & + h_4[Q_4 - (\alpha_{42}S_{r2} + \alpha_{46}S_{r6} + \alpha_{48}S_{r8})] \\ & + h_5[Q_5 - (\alpha_{51}S_{r1} + \alpha_{53}S_{r3} + \alpha_{59}S_{r9})] \\ & + h_6[Q_6 - (\alpha_{65}S_{r5} + \alpha_{66}S_{r6} + \alpha_{67}S_{r7})] \\ & + p_1(D_{r1} - S_{r1}) + p_2(D_{r2} - S_{r2}) + p_3(D_{r3} - S_{r3}) \\ & + p_4(D_{r4} - S_{r4}) + p_5(D_{r5} - S_{r5}) + p_6(D_{r6} - S_{r6}) \\ & + p_7(D_{r7} - S_{r7}) + p_8(D_{r8} - S_{r8}) + p_9(D_{r9} - S_{r9}) \end{aligned}$$

subject to

$$\alpha_{11}S_{r1} + \alpha_{14}S_{r4} + \alpha_{17}S_{r7} \le Q_1 \qquad \forall\, r$$

$$\alpha_{23}S_{r3} + \alpha_{24}S_{r4} + \alpha_{28}S_{r8} \le Q_2 \qquad \forall\, r$$

$$\alpha_{32}S_{r2} + \alpha_{35}S_{r5} + \alpha_{39}S_{r9} \le Q_3 \qquad \forall\, r$$

$$\alpha_{42}S_{r2} + \alpha_{46}S_{r6} + \alpha_{48}S_{r8} \le Q_4 \qquad \forall\, r$$

$$\alpha_{51}S_{r1} + \alpha_{53}S_{r3} + \alpha_{59}S_{r9} \le Q_5 \qquad \forall\, r$$

$$\alpha_{65}S_{r5} + \alpha_{66}S_{r6} + \alpha_{67}S_{r7} \le Q_6 \qquad \forall\, r$$

$$S_{r1} \le D_{r1} \qquad \forall\, r$$

$$S_{r2} \le D_{r2} \qquad \forall\, r$$

$$S_{r3} \le D_{r3} \qquad \forall\, r$$

$$S_{r4} \le D_{r4} \qquad \forall\, r$$

$$S_{r5} \le D_{r5} \qquad \forall\, r$$

$$S_{r6} \le D_{r6} \qquad \forall\, r$$

$$S_{r7} \le D_{r7} \qquad \forall\, r$$

$$S_{r8} \le D_{r8} \qquad \forall\, r$$

$$S_{r9} \le D_{r9} \qquad \forall\, r$$

$$S_{rj} \ge 0 \qquad j = 1, 2, \ldots, 9 \qquad \forall\, r$$

$$Q_i \ge 0 \qquad i = 1, 2, \ldots, 6$$

The same approach as for the Problem I is used in this problem. Recall that in Problem I, we used an optimal allocation rule to run simulation on the sampled designs. In Problem II however, note that the number of variables increases and thus it is not feasible to develop an optimal allocation rule. Therefore, for Problem II, given $\hat{Q}^*$ and $\hat{D}$, we have to solve the component allocation $\alpha_{ij}S_j$ optimally by solving allocation problem in P(8) each time to run a simulation replication.

P(8):

$$\min_{Sj} \; h_1[Q_1 * -(\alpha_{11}S_1 + \alpha_{14}S_4 + \alpha_{17}S_7)] + h_2[Q_2 * -(\alpha_{23}S_3 + \alpha_{24}S_4 + \alpha_{28}S_8)]$$

$$+ h_3[Q_3 * -(\alpha_{32}S_2 + \alpha_{35}S_5 + \alpha_{39}S_9)] + h_4[Q_4 * -(\alpha_{42}S_2 + \alpha_{46}S_6 + \alpha_{48}S_8)]$$

$$+ h_5[Q_5 * -(\alpha_{51}S_1 + \alpha_{53}S_3 + \alpha_{59}S_9)] + h_6[Q_6 * -(\alpha_{65}S_5 + \alpha_{66}S_6 + \alpha_{67}S_7)]$$

$$+ p_1(D_1 - S_1) + p_2(D_2 - S_2) + p_3(D_3 - S_3) + p_4(D_4 - S_4) + p_5(D_5 - S_5)$$

$$+ p_6(D_6 - S_6) + p_7(D_7 - S_7) + p_8(D_8 - S_8) + p_9(D_9 - S_9)$$

subject to

$$\alpha_{11}S_1 + \alpha_{14}S_4 + \alpha_{17}S_7 \le Q_1 *$$

$$\alpha_{23}S_3 + \alpha_{24}S_4 + \alpha_{28}S_8 \le Q_2 *$$

$$\alpha_{32}S_2 + \alpha_{35}S_5 + \alpha_{39}S_9 \le Q_3 *$$

$$\alpha_{42}S_2 + \alpha_{46}S_6 + \alpha_{48}S_8 \le Q_4 *$$

$$\alpha_{51}S_1 + \alpha_{53}S_3 + \alpha_{59}S_9 \le Q_5 *$$

$$\alpha_{65}S_5 + \alpha_{66}S_6 + \alpha_{67}S_7 \le Q_6 *$$

$$S_1 \le D_1$$

$$S_2 \le D_2$$

$$S_3 \le D_3$$

$$S_4 \le D_4$$

$$S_5 \le D_5$$

$$S_6 \le D_6$$

$$S_7 \le D_7$$

$S_8 \leq D_8$

$S_9 \leq D_9$

$S_j \geq 0 \qquad j = 1, 2, \ldots, 9$

$Q_i \geq 0 \qquad i = 1, 2, \ldots, 6$

In the numerical experiment, the holding costs for the 6 components, h_i, the penalty costs for the 9 products, p_j and the number of component i needed to assemble one unit of product j, α_{ij} for Problem II are fixed as follows,

Parameter values of the holding cost for component i, h_i (\$):

h_1 = 0.30
h_2 = 0.80
h_3 = 1.30
h_4 = 0.25
h_5 = 0.95
h_6 = 0.70

Parameter values of the penalty cost for product j, p_j (\$):

p_1 = 0.90
p_2 = 0.85
p_3 = 1.35
p_4 = 1.43
p_5 = 0.58
p_6 = 2.30
p_7 = 1.50
p_8 = 0.50
p_9 = 1.10

Parameter values of the number of component *i* needed to assemble one unit of product j, α_{ij} :

α_{11} = 1

α_{14} = 2

α_{17} = 1

α_{23} = 4

α_{24} = 3

α_{28} = 2

α_{32} = 1

α_{35} = 1

α_{39} = 2

α_{42} = 2

α_{46} = 1

α_{48} = 1

α_{51} = 3

α_{53} = 1

α_{59} = 2

α_{65} = 1

α_{66} = 2

α_{67} = 1

For Problem II, we directly generalized the problem to test on the validity of our approach. Thus, the random sampling method as described in Case I for Problem I is not experimented. The designs for Problem II are sampled using the SAA sampling method with n_0 varied. The OCBA model for Problem II was developed and used to find the optimal computing budget allocation decision of (n_0, n_1, n_2) that minimizes the expected true performance of the observed best design.

5.4.1 Numerical Result for Problem II for Case III : n_0 varied

In this subsection, the numerical result for Problem II when the n_0 is varied is presented. The organization of this subsection is similar to the Subsection 5.3.3 for Problem I. We first discuss the effect of true performance when the n_0 is varied. We also estimate the Weibull parameter of $\alpha_{(n_o)}$ for the different degrees of n_0 in Problem II by using the screening experiment.

Following this, the function $t_{(n_o)}$ and the value of s is computed in terms of CPU time (in seconds) for the OCBA model for Problem II. The OCBA model is solved under two different fixed computing budgets K of 3,000 and 6,000 seconds. The demands for Problem II are also drawn from normal distribution, $D \sim N(1000,100)$. We select a few different combinations of allocation decision of (n_0, n_1, n_2) that satisfy the computing budget constraint and Problem II is solved for each of the allocation combinations. The optimal allocation decision that can minimize the expected true performance of the observed best design of Problem II is then identified. The normal and Weibull table estimations are also used to estimate the expected true performance of the observed best design and their performances are compared against the simulation result. In order to know how to determine the optimum allocation decisions using the normal and Weibull table estimations, we first run a screening experiment to estimate the required parameters.

The CDF of the true performance when SAA is used as the sampling method with a range of n_0 for Problem II is shown in Figure 5.11. For this case, the n_0 is fixed at 1, 5, 10 and 20. As expected, better designs are sampled when more information is supplied.

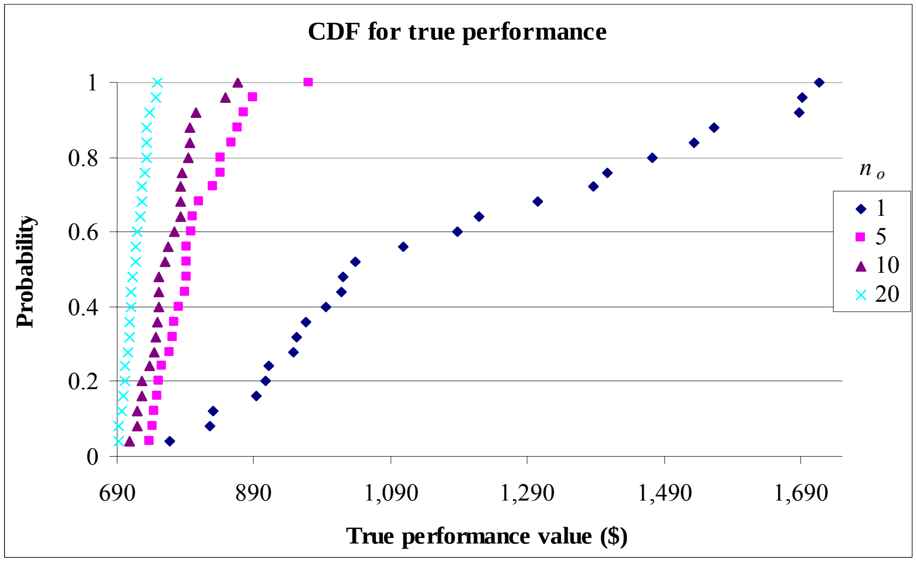

Figure 5.11: The improvement in the true performance value when n_0 is varied in Problem II

For the same reason of simplifying the graph, we present the distribution of true performance for only four different degrees of n_0. The true performance of eight different degrees of n_0 (n_0 = 1, 3, 5, 7, 10, 15, 18, 20) run with the detailed experiments are presented in Appendix G. Also please refer to Appendix G for more information on the CDF of true performance, the distribution of true performance, the minimum true performance and its observed best design, the maximum true performance and the standard deviation of the true performance for each of the different n_0 experimented. Based on the observations in the appendix, similar conclusions can be drawn; with higher level of n_0, better designs with lower standard deviation in the true performance are obtained. However note that unlike Problem I, the distribution of true performance of Problem II does not show the exponential distribution. This is because Problem II is more complex (more variables are involved) and we did not sample enough designs and run enough replications to represent the actual distribution due to the constraint of time.

Figure 5.12 represents the correlation of the estimated parameter of $\alpha_{(n_o)}$ with the levels of n_0 varying from 1 to 20. The $\alpha_{(n_o)}$ decreases exponentially when n_0 increases with the empirical relationship given as $\alpha_{(n_o)} = 401.4\ n_0^{-0.7862}$. The coefficient of determination, R^2 is 0.9057. It is observed that the value of $\alpha_{(n_o)}$ is very much higher for the Problem II as compared to Problem I. For example, the $\alpha_{(n_o)}$ values for n_0 between 1 to 20 for Problem I are within 25, but for the Problem II, the values of the n_0 for the same range exceed 500. This function can be used for the Weibull table to estimate the value of $\alpha_{(n_o)}$ for any n_0 that lies between 1 and 20.

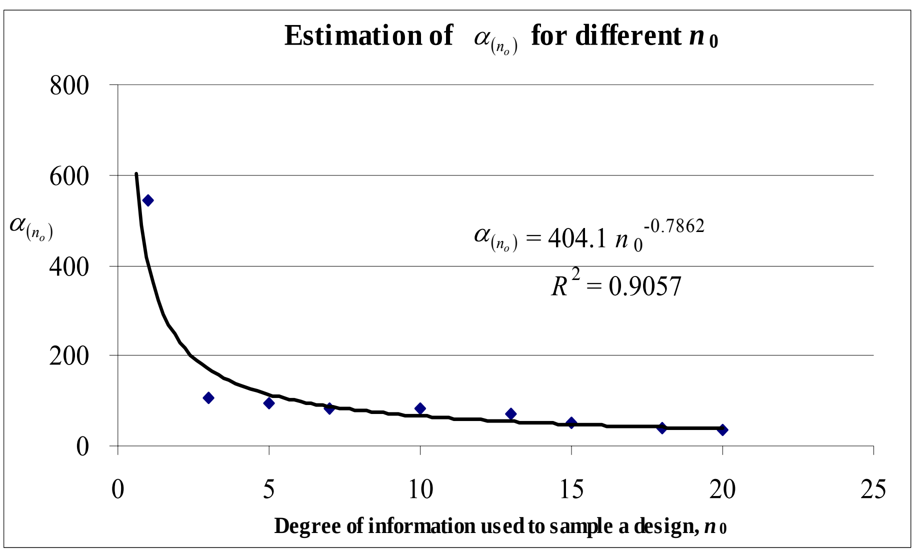

Figure 5.12: Estimation of $\alpha_{(n_o)}$ for Problem II

In the following work we use regression analysis to estimate the time to generate one design, $t_{(n_o)}$ in terms of n_0 and the simulation time to run one replication of simulation, s in terms of CPU time. Similarly, Figure 5.13 depicts that $t_{(n_o)}$ increases when n_0 is increased, and the empirical relationship is $t_{(n_o)}$ = 0.0224 $n_0{}^2$ - 0.1332 n_0 + 1.1496. The coefficient of determination is also high for this problem (R^2 = 0.9954). Observe that for the same degree of n_0, the $t_{(n_o)}$ is higher for Problem II as compared to Problem I. For example, the $t_{(n_o)}$ for n_0 = 1 for Problem I is 0.2 second, where as the $t_{(n_o)}$ for the same degree of n_0 for Problem II is 1 second. Another example can be seen for the higher degree of n_0 = 20. For Problem I, the $t_{(n_o)}$ = 0.8 second, and for Problem II, the $t_{(n_o)}$ = 7.4 seconds, which is almost ten times more of the time taken in Problem I. This is because more information of demand is supplied and more decision variables have to be solved in the SAA sampling in Problem II than in Problem I, and thus more time is taken to sample a design in Problem II.

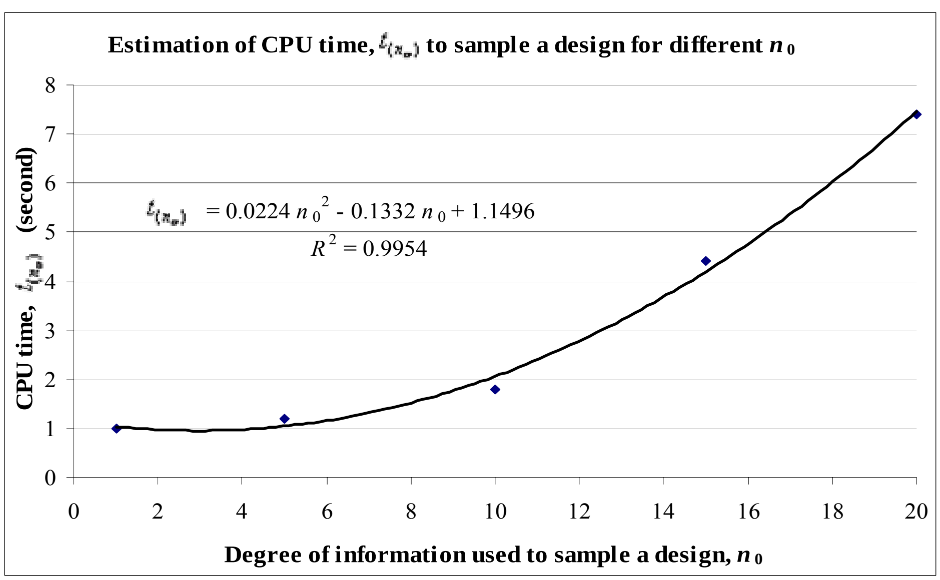

Figure 5.13: Estimation of $t_{(n_o)}$ for Problem II

From Figure 5.14, the CPU time to perform the simulation, S also is seen to increase linearly with the simulation length, n_2 and the linear regression fits the data well (R^2 is almost 1). The regression coefficient, which is also the estimation of simulation time for one replication of simulation, s is higher for Problem II ($s = 4.896 \text{ X } 10^{-1}$ seconds) as the LP in P(8) has to be solved for the component allocation in this problem. (The numerical estimation of $\alpha_{(n_o)}$, $t_{(n_o)}$ and s for Problem II is presented in Appendix G).

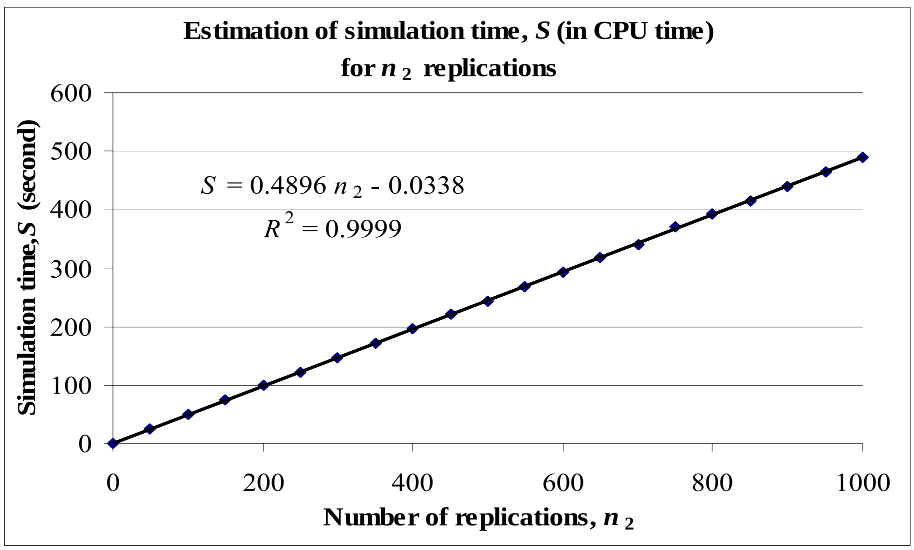

Figure 5.14: Estimation of s for Problem II

The OCBA model for Problem II is then as given in P(9).

P(9):

$$\min E[J_{[\tilde{1}]}] \tag{5.4}$$

subject to

$$(2.24 \text{ X } 10^{-2} n_0^{\,2} - 1.33 \text{ X } 10^{-1} n_0 + 1.1496 + 4.896 \text{ X } 10^{-1} n_2)\, n_1 \leq K \tag{5.5}$$

$$E[J_{[\tilde{1}]}] = \frac{\sigma_{N_o}}{\sqrt{n_2}} E[J'_{[\tilde{1}]}] \tag{5.6}$$

For the first numerical example of Problem II, the K is fixed at 3,000 seconds and some possible combinations of (n_0, n_1, n_2) that satisfy constraint (5.5) is selected. Table 5.7 presents the choices of (n_0, n_1, n_2) allocation and its numerical result based on the normal, Weibull and simulation estimations. Please refer to Appendix H for the complete computation based on the screening experiment. The optimal allocation based on the simulation result is given as (20, 100, 46) with \$719.43. The Weibull table estimation again suggests the correct allocation decision. The minimum expected true performance of the observed best design, $E[J_{[\tilde{1}]}]$ for the Weibull table estimation is recorded to be \$701.72. However, in this particular problem, the normal table estimation is also able to indicate the correct selection with the minimum expected cost of \$715.59. Note that the $E[J_{[\tilde{1}]}]$ for Problem II is generally higher than that of for Problem I.

Table 5.7: Numerical result for Problem II, designs sampled using SAA with n_0 varied (K=3,000 seconds)

Computing Budget Allocation			**Normal table Estimation (\$)**	**Weibull table Estimation (\$)**	**Simulation Result (\$)**
n_0	n_1	n_2			
7	50	120	769.81	725.38	730.63
20	**100**	**46**	**715.59**	**701.72**	**719.43**
5	1,000	4	802.61	766.05	754.29
10	4	1,530	765.33	727.48	745.04

The K is now fixed at 6,000 seconds. Appendix H also gives the parameter values and detailed computation for this problem. The summarized version of the numerical result is recorded in Table 5.8. Similarly, the Weibull is able to pick the best allocation decision of (20, 50, 230) as suggested by simulation result, with the expected minimum cost of \$698.46. Likewise the previous example, apparently the normal table estimation is also able to indicate the correct selection with the higher expected minimum cost of $715.22.

Table 5.8: Numerical result for Problem II, designs sampled using SAA with n_0 varied (K=6,000 seconds)

Computing Budget Allocation			**Normal table Estimation ($)**	**Weibull table Estimation ($)**	**Simulation Result ($)**
n_0	n_1	n_2			
1	50	243	1,171.37	789.35	862.36
20	102	105	715.25	699.88	709.53
7	2,615	2	770.73	747.59	721.08
20	**50**	**230**	**715.22**	**698.46**	**701.68**

In this chapter, we have conducted a number of experiments for Problem I and Problem II, when the SAA is used as the sampling. Based on the numerical results in these experiments, it is reasonable to conclude that the Weibull table estimation is quite reliable and promising in selecting the correct allocation decision when the SAA is used as the sampling method. Hence, in the future work, we can comfortably rely on the Weibull table estimation alone to decide on the optimal allocation decision for our problem.

Chapter 6 CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

In our work, we have described a general framework for solving the computing budget allocation problem. Given a fixed amount of computing budget, it is important to decide how to sample the design space, how many designs to sample and for how long to run each design alternative so as to optimize the expected true performance of the observed best design. In our work, we proposed an approach for selecting these allocation decisions. This approach was illustrated by two different ATO problems using the SAA as the sampling method.

From the experiment conducted, it is observed that the distribution of true performance and noise plays a vital role in deciding on how to perform the sampling efficiently. It is also found that different ways of sampling designs results in different distribution shapes of the true performance. For example when the sampling scheme is random, the distribution of true performance of the designs obtained is just mediocre. However, if the SAA is used as the sampling method, where some computational time is invested in sampling each design using the information provided, we have higher chance of getting good designs and thus the distribution of true performance have a skewed distribution which we approximate it with the Weibull (exponential) distribution.

In order to handle such cases when the true performance follows normal or Weibull distribution and the noise follows a normal distribution, we have developed normal and Weibull model to estimate the true performance of the observed best design. The normal and Weibull table estimations are used to decide on the correct allocation decision.

In order to know how to determine the correct allocation decisions using the normal and Weibull table estimations, we will have to first run a screening experiment to estimate the required parameters. Similarly, the time constraint in both the OCBA models has to be estimated by fitting the function to several sample points of the degree of information level and the simulation length. Of course much computing effort is required to get good fit and accurate estimations, especially for the degree of information level which requires more time to sample a design when the level is increased. However on the other hand, the computing expense for these screening experiment and time constraint function estimation has to be reasonable as compared to the entire budget. One way to limit the screening computing effort is by using smaller allocation decisions, but big enough to get good estimations to model the problem.

From the experiments conducted, it is observed that the approach is able to make correct selection on the allocation decision. It is shown that when the SAA is used as the sampling method, the Weibull table alone is able to indicate the right selection, and thus it can be used in future work for solving the computing budget allocation problem.

6.2 Future Works

There is scope for further work in this research area. Firstly, we assumed that the horse race selection method was used which meant that all the designs were run with equal replications; whereas, if we can allocate the simulation replication wisely to the different designs sampled, performance should be improved further. For example, less simulation replications should be assigned to the average designs, where as more simulation replications should be allocated for those critical designs sampled. By this way, the limited computation effort is allocated even more intelligently on the designs generated. Hence one direction of future work is to look at how to assign different number of replication for each of the design so as the simulation efficiency is further improved.

Secondly, in our work, we model the computing budget allocation problem when both the distributions of true performance and the noise are normal, and when the distribution of true performance is Weibull and the noise is normal. In future effort, we can investigate other cases where the distributions of true performance and the noise deviate from normal and Weibull. Appropriate models with well defined parameters should be derived to estimate the expected true performance of the observed best design for these cases. Some application examples can also be used to illustrate the applicability of the approach.

REFERENCES

1. Ahmed, A. A. and T. M. Alkhamis. Simulation-based Optimization using Simulated Annealing with Ranking and Selection, Computers and Operations Research, 29, pp.387-402. 2002.
2. Averill M. L., and W. David Kelton. Simulation Modeling and Analysis. pp. 566-575, Boston: McGraw-Hill. 2000.
3. Baker, K. R, M. J. Magazine, and H. L. W. Nuttle. The Effect of Commonality on Safety Stock in a Simple Inventory Model, Management Science, 32 (8), pp.982-988. 1986.
4. Baker, K. R. Safety Stocks and Component Commonality, Journal of Operations Management, 6 (1), pp.13-22. 1985.
5. Barry, L. N. Stochastic Modeling: Analysis and Simulation. New York; Singapore: McGraw-Hill. 1995.
6. Bechhofer, R. E., C. W. Dunnet and M. Sobel. A two-sample Multiple Decision Procedure for Ranking Means of Normal Populations with a Common Unknown Variance, Biometrika, 41, pp.170-176. 1954.
7. Butler, J., D. J. Morrice, and P. W. Mullarkey. A Multiple Attribute Utility Theory Approach to Ranking and Selection, Management Science, 47 (6), pp.800-816. 2001.
8. Chen, C.-H., J. Lin, E. Yücesan, and S. E. Chick. Simulation Budget Allocation for Further Enhancing the Efficiency of Ordinal Optimization, Journal of Discrete Event Dynamic Systems: Theory and Applications, 10, pp.251-270. 2000.

9. Chen, C.-H., Y. Yuan, C.-H. Chen, E. Yücesan, and L. Dai. Computing Budget Allocation for Simulation Experiments with Different Systems Structures, Proceedings of the 1998 Winter Simulation Conference, pp.735-741. 1998.

10. Chen, E. J, and W. D. Kelton. Inferences from Indifference-Zone Selection Procedures, to appear in the Proceedings of the 2003 Winter Simulation Conference.

11. Chen, H.-C., C.-H. Chen, L. Dai, and E. Yücesan. New Development of Optimal Computing Budget Allocation for Discrete Event Simulation, Proceedings of the 1997 Winter Simulation Conference, pp.334-341. 1997.

12. Dai, L.Y., Convergence Properties of Ordinal Comparison in the Simulation of Discrete Event Dynamic Systems. Journal of Optimization Theory and Application, 91(2), pp.363-388. 1996.

13. Dudewicz, E. J. and S. R. Dalal. Allocation of Observations in Ranking and Selection with Unequal Variances, The Indian Journal of Statistics 37B(1), pp.28-78. 1975.

14. Dudewicz, E. J. Statistics in simulation: How to Design for Selecting the Best Alternative, Proceedings of the 1976 Winter Simulation Conference, pp.67-71. 1976.

15. Eynan, A., and M. J. Rosenblatt. Component Commonality Effects on Inventory Costs, IIE Transactions, 28, pp.93-104. 1996.

16. Gerchak, Y. and M. Henig. An Inventory Model with Component Commonality, Operations Research Letters, 5 (3), pp.157-160. 1985.

17. Gerchak, Y. and M. Henig. Component Commonality in Assemble-to-Order Systems: Models and Properties, Operations Research Letters, Naval Research Logistics, 36, pp.61-68. 1989.

18. Goldsman, D. Tutorial on Indifference-zone Normal Means Ranking and Selection Procedures, Proceedings of the 1986 Winter Simulation Conference, pp.370-375. 1986.

19. Goldsman, D., and B. L. Nelson. Methods for Selecting the Best System, Proceedings of the 1991 Winter Simulation Conference, pp.177-186. 1991.

20. Goldsman, D., and B. L. Nelson. Ranking, Selection and Multiple Comparisons in Computer Simulation, Proceedings of the 1994 Winter Simulation Conference, pp.192-199. 1994.

21. Goldsman, D., and B. L. Nelson. Statistical Selection of the Best System, Proceedings of the 2001 Winter Simulation Conference, pp.139-146. 2001.

22. Goldsman, D., and W. S. Marshall. Ranking and Selection for Steady-State Simulation, Proceedings of the 2000 Winter Simulation Conference, pp.544-553. 2000.

23. Gray, D. and D. Goldsman. Indifference-zone Selection Procedures for Choosing the Best Airspace Configuration, Proceedings of the 1988 Winter Simulation Conference, pp.445-450. 1988.

24. Gupta, S. S. and J. C. Hsu. Subset Selection Procedures with Special Reference to the Analysis of Two-way Layout: Application to Motor-vehicle Fatality Data, Proceedings of the 1977 Winter Simulation Conference, pp.81-85. 1977.

25. Gupta, S. S. On a Decision Rule for a Problem in Ranking Means. Mimeograph Series No. 150, Institute of Statistics. University of North Carolina, Chapel Hill. 1956.

26. Gupta, S. S. On Some Multiple Decision (Selection and Ranking) Rules, Technometrics, 2 (2), pp.225-245. 1965.

27. Hillier, M. S. Component Commonality in Multi-Period Inventory Model with Service Level Constraints, International Journal of Production Research, 37 (12), pp.2665-2683. 1999.

28. Hillier, M. S. Component Commonality in Multiple-Period, Assemble-to-Order Systems, IIE Transactions, 32, pp.755-766. 2000.

29. Ho, Y. C., C. G. Cassandras, C.-H. Chen, and L. Dai. Ordinal Optimization and Simulation, Journal of Operations Research of Society, 51, pp.490-500. 2000.

30. Ho, Y. C., R. S. Sreenivas, and P. Vakili. Ordinal Optimization of Discrete Event Dynamic Systems, Journal of Discrete Event Dynamic Systems, 2, pp.61-68. 1992.

31. James, R. S., and S. H. Jacobson. A Survey of Ranking, Selection, and Multiple Comparison Procedures for Discrete-Event Simulation, Proceedings of the 1999 Winter Simulation Conference, pp.492-501. 1999.

32. Kannan, R., J. Mount, and S. Tayur. A Randomized Algorithm to Optimize Over Certain Convex Sets, Mathematics of Operations Research, 20 (3), pp.529-549. 1995.

33. Kleywegt, A. J., A. Shapiro, and T. Homem-De-Mello. The Sample Average Approximation Method for Stochastic Discrete Optimization, Siam Journal of Optimization, 12 (2), pp.479-502. 2001.

34. Koenig, L. W. and A. M. Law. A Procedure for Selecting a Subset of Size *m* Containing the *l* Best of *k* Independent Normal Populations, with Applications to Simulation, Communications in Statistics B14 (13): pp.719-734. 1985.

35. Lee, L. H. and E. P. Chew. A Simulation Study on Sampling and Selecting under Fixed Computing Budget, Proceedings of the 2003 Winter Simulation Conference, pp.535-542. 2003.

36. Lee, L. H. and E. P. Chew. A Simulation Study on Sampling and Selecting under Fixed Computing Budget, submitted to Journal of Optimization Theory and Application. 2004.

37. Lee, L. H., T. W. E., Lau and Y. C. Ho. Explanation of Goal Softening in Ordinal Optimization, IEEE Transaction on Automatic Control 44 (1), pp.94-99. 1999.

38. Matejcik, J. M. and B. L. Nelson. Simultaneous Ranking, Selection and Multiple Comparisons for Simulation, Proceedings of the 1993 Winter Simulation Conference, 1993.

39. Mirchandani, P. and A. K. Mishra. Component Commonality: Models with Product-Specific Service Constraints, Production and Operations Management, 11 (2), pp.99-215. 2002.

40. Nelson, B. L., and D. Goldsman. Comparisons with a Standard in Simulation Experiments, Management Science, 47, pp.449-463. 2001.

41. Nelson, B. L., J. Swann, D. Goldsman, and W. Song. Simple Procedures for Selecting the Best Simulated System when the Number of Alternatives is Large, Operations Research, 49, pp.50-963. 2000.

42. Paulson, E. A Sequential Procedure for Selecting the Population with the Largest Mean from *k* Normal Populations, Annals of Mathematical Statistics 35: pp. 174-180. 1964.

43. Rinott, Y. On Two-stage Selection Procedures and Related Probability Inequalities, Communications in Statistics-Theory and Methods, A7, pp.799-811. 1978.

44. Robert, E. B., Thomas, J. S., and M. G. David. Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons, pp.7-12, New York: Wiley. 1995.

45. Rudi, N. Dual Sourcing: Combining Make-to-Stock and Assemble-to-Order. University of Rochester. 1998.

46. Rudi, N. Optimal Inventory Levels in Systems with Common Components. University of Rochester. 2000.

47. Shanti, S. G., and S. Panchapakesan. Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations, pp.1-13, New York: Wiley. 1979.

48. Shapiro, A. Monte Carlo Simulation Approach to Stochastic Programming, Proceedings of the 2001 Winter Simulation Conference, pp.428-431. 2001. Simulation, Communications in Statistics B14(3): pp.719-734. 1985.

49. Srinivasan, R., R. Jayamaran, J. A. Rappold, R. O. Roundy, S. Tayur. Procurement of Common Components in a Stochastic Environment. IBM Research Report, RC-18580, 12/1992. 1998.

50. Sullivan, D. W. and J. R. Wilson. Restricted Subset Selection Procedures for Simulation, Operations Research, 37, pp. 52-71. 1989.

51. Swisher, J. R. and S. H. Jacobson. A Survey of Ranking, Selection and Multiple Comparison Procedures for Discrete-Event Simulation, Proceedings of the 1999 Winter Simulation Conference, pp.492-501. 1999.

52. Tang, Q. Y. and H. F. Chen. Convergence Rate of Ordinal Optimization for Stochastic Discrete Event Systems, Discrete Event Dynamic Systems: Theory and Applications, 9, pp.23-43. 1999.

53. Tayur, S. Computing Optimal Stock Levels for Common Components in an Assembly System. Working paper, 1995.

54. Xie, X. L. Dynamics and Convergence Rate of Ordinal Comparison of Stochastic Discrete Event Systems, IEEE Transaction on Automatic Control, 42 (4), pp.586-590. 1997.

55. Yang, M. S., and Lee, L. H. Ordinal Optimization with Subset Selection Rule, Journal of Optimization Theory and Applications, 113 (3), pp.597-620. 2002.

APPENDICES

APPENDIX A: EXPECTED TRUE VALUE FOR THE OBSERVED BEST

Following are the steps of running the Monte-Carlo simulation to develop the Weibull table for a fixed β':

Step 1: Fix the n_1.

Step 2: Fix the α'.

Step 3: Sample n_1 designs from a Weibull distribution, J' with the fixed parameters α' and β'.

Step 4: For each design, generate a noise, w' from standard normal distribution, $N(0,1)$.

Step 5: The true performance measure J' is then added to the noise w' to make up the observed performance value $\tilde{J}'$.

Step 6: The n_1 designs are then ranked based on the observed performance value $\tilde{J}'$ and the design which is ranked the best, its true performance value J' will be recorded.

Step 7: This experiment is repeated 1000 times to estimate the mean of the true performance, $E[J'_{[\tilde{1}]}]$. This is the Weibull table value recorded for the particular n_1 and α'.

Step 8: For the fixed n_1, vary the α' and repeat step 3 to step 7 and the Weibull table values are tabulated accordingly. This step is repeated until we exhaust all the different α'.

Step 9: Vary the n_1 and repeat step 2 to step 8.

Above are the steps to obtain the Weibull table values for a fixed β'. In order to estimate the expected true performance of the observed best, $E[J_{[\tilde{1}]}]$, the Weibull table value is multiplied with $\frac{\sigma_{N_o}}{\sqrt{n_2}}$. Given below are the Weibull table for β' between 1 to 10.

Table E.1: The expected true performance of the observed best for Weibull table for β' between 1 to 10

β' = 1.0

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.2761	0.2657	0.2459	0.2400	0.2364	0.2342
2	0.3522	0.3117	0.3053	0.2988	0.2754	0.2706
4	0.4012	0.3714	0.3453	0.3270	0.3100	0.3064
8	0.4680	0.4392	0.4141	0.3715	0.3573	0.3536
16	0.5622	0.4837	0.4528	0.4129	0.3946	0.3943
32	0.6940	0.5758	0.5437	0.4708	0.4284	0.4034
64	1.0062	0.7130	0.6239	0.5229	0.5067	0.4692

β' = 1.1

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.3257	0.2889	0.2871	0.2721	0.2590	0.2535
2	0.3969	0.3557	0.3349	0.3314	0.3172	0.3092
4	0.4488	0.4095	0.3974	0.3781	0.3399	0.3376
8	0.5541	0.4634	0.4505	0.4329	0.4124	0.3995
16	0.6698	0.5583	0.5285	0.4888	0.4543	0.4475
32	0.8384	0.6501	0.6031	0.5368	0.5019	0.4913
64	1.2065	0.8955	0.7596	0.6741	0.5773	0.5483

β' = 1.2

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.3456	0.3425	0.2945	0.2831	0.2816	0.2637
2	0.3973	0.3965	0.3758	0.3636	0.3428	0.3354
4	0.5021	0.4584	0.4345	0.4093	0.4076	0.4051
8	0.5916	0.5192	0.5129	0.4796	0.4373	0.4261
16	0.7579	0.6347	0.5738	0.5022	0.5020	0.4715
32	1.0419	0.8164	0.6925	0.6477	0.5780	0.5546
64	1.5654	1.1102	0.9303	0.7733	0.7109	0.6470

β' = 1.3

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.3553	0.3450	0.3171	0.3138	0.3006	0.2990
2	0.4767	0.4288	0.4026	0.3877	0.3789	0.3645
4	0.5496	0.5103	0.5020	0.4608	0.4119	0.4117
8	0.6861	0.6103	0.5611	0.4968	0.4888	0.4624
16	0.8892	0.7302	0.6597	0.6079	0.5585	0.5353
32	1.2411	0.9438	0.8436	0.7251	0.6669	0.6411
64	1.9653	1.3874	1.1369	0.9647	0.7978	0.7417

β' = 1.4

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.3915	0.3760	0.3654	0.3367	0.3301	0.3262
2	0.5110	0.4438	0.4318	0.4287	0.4163	0.4071
4	0.6111	0.5688	0.5397	0.4967	0.4815	0.4710
8	0.7750	0.6702	0.5974	0.5886	0.5440	0.5098
16	0.9835	0.7910	0.7630	0.6707	0.6177	0.5954
32	1.3787	1.0998	0.9582	0.8422	0.7450	0.7251
64	2.4022	1.6394	1.3511	1.1734	0.9499	0.8715

β' = 1.5

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.3999	0.3955	0.3723	0.3636	0.3627	0.3508
2	0.5228	0.4915	0.4880	0.4538	0.4193	0.3983
4	0.6158	0.5872	0.5783	0.5267	0.5090	0.5075
8	0.7927	0.7376	0.6731	0.6551	0.6119	0.5712
16	1.0964	0.9635	0.8212	0.7527	0.6863	0.6749
32	1.6664	1.2590	1.0989	0.9688	0.8619	0.8134
64	2.9483	1.9452	1.6055	1.3013	1.0574	1.0544

$\beta' = 1.6$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.4413	0.4145	0.3945	0.3734	0.3701	0.3554
2	0.5647	0.5328	0.5158	0.4918	0.4756	0.4577
4	0.7348	0.6564	0.6113	0.5659	0.5443	0.5434
8	0.9254	0.7995	0.7355	0.7231	0.6435	0.6267
16	1.2467	1.0276	0.9262	0.8319	0.7418	0.7252
32	1.9489	1.4389	1.2456	1.0761	0.9573	0.9435
64	3.5042	2.4100	1.9153	1.5090	1.3587	1.1831

$\beta' = 1.7$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.4666	0.4376	0.4182	0.3994	0.3894	0.3891
2	0.6189	0.5540	0.5325	0.5314	0.4825	0.4745
4	0.7765	0.6839	0.6668	0.6153	0.5970	0.5702
8	0.9996	0.8671	0.8224	0.7521	0.7090	0.6619
16	1.3512	1.1135	0.9887	0.9247	0.8667	0.7812
32	2.1837	1.5774	1.4158	1.2677	1.0520	1.0248
64	3.9783	2.8035	2.2540	1.8649	1.5789	1.4738

$\beta' = 1.8$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.4857	0.4667	0.4488	0.4358	0.4103	0.4024
2	0.6357	0.5979	0.5701	0.5477	0.5125	0.5089
4	0.8624	0.7447	0.7080	0.6588	0.6530	0.6012
8	1.1037	0.9530	0.8599	0.7936	0.7391	0.7264
16	1.5578	1.2405	1.1166	1.0079	0.8966	0.8735
32	2.5181	1.8249	1.6484	1.3536	1.1836	1.1535
64	4.5623	3.1560	2.6758	2.1318	1.8209	1.5992

$\beta' = 1.9$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.4978	0.4611	0.4378	0.4474	0.4302	0.4183
2	0.6594	0.6372	0.6053	0.5875	0.5747	0.5530
4	0.8802	0.8094	0.7402	0.7538	0.6927	0.6743
8	1.1184	1.0160	0.9547	0.8725	0.7745	0.7670
16	1.6399	1.3889	1.2611	1.0802	1.0203	0.9583
32	2.7881	2.0989	1.8370	1.5264	1.3852	1.2593
64	5.0862	3.7237	2.9894	2.4607	2.0569	1.9723

$\beta' = 2.0$

	n₁					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.5183	0.4866	0.4704	0.4692	0.4534	0.4451
2	0.7040	0.6721	0.6516	0.6256	0.6021	0.5704
4	0.9264	0.8465	0.7889	0.7306	0.7259	0.7043
8	1.2446	1.0904	0.9930	0.9356	0.8902	0.8094
16	1.7816	1.4722	1.3397	1.1917	1.0891	1.0130
32	3.0671	2.3689	2.0550	1.7237	1.4495	1.4271
64	5.7909	4.2694	3.3921	2.8538	2.3599	2.2094

$\beta' = 5.0$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.8014	0.7961	0.7888	0.7658	0.7588	0.7570
2	1.4174	1.3482	1.3182	1.2956	1.2650	1.2624
4	2.1755	2.0598	1.9790	1.9136	1.8262	1.8222
8	3.4388	3.1885	2.9864	2.7927	2.6504	2.6366
16	6.2328	5.4852	5.1441	4.6817	4.3427	4.2520
32	11.9507	10.4009	9.6089	8.7090	8.0437	7.6728
64	23.8231	20.3559	18.7906	17.1846	15.7268	14.8393

$\beta' = 10.0$

	n_1					
α'	**100**	**200**	**300**	**500**	**800**	**1000**
1	0.9234	0.9189	0.9120	0.9119	0.9051	0.9063
2	1.7429	1.7336	1.7248	1.7174	1.7163	1.6952
4	3.1929	3.1131	3.0684	3.0161	2.9477	2.9448
8	5.5227	5.2707	5.1214	4.9341	4.7678	4.6967
16	10.0409	9.3793	9.1773	8.6983	8.3268	8.1994
32	19.4874	18.1911	17.4040	16.5683	15.9479	15.4621
64	38.5192	35.9764	34.5369	32.8570	31.2335	30.8217

APPENDIX B: OPTIMUM ALLOCATION RULE FOR SPECIAL CONDITIONS OF PROBLEM I

In Problem I, given the optimum component inventory level vector $\hat{Q}^*$ and the product demand vector $\hat{D}$ are known, the component allocation quantity S_j can be determined based on the conditions of $\hat{Q}^*$ and $\hat{D}$ using the optimal allocation rule given below. The problem can be categorized into 4 possible conditions:

Condition I:

$$D_1 + D_2 \leq Q_1^*$$
$$D_2 + D_3 \leq Q_2^*$$

Optimum allocation rule:

$$S_1 = D_1$$
$$S_2 = D_2$$
$$S_3 = D_3$$

Condition II:

$$D_1 + D_2 \leq Q_1^*$$
$$D_2 + D_3 \geq Q_2^*$$

Optimum allocation rule:

$$S_1 = D_1$$
$$S_2 = \min\{D_2, Q_1^* - D_1, Q_2^*\}$$
$$S_3 = Q_2 - \min\{D_2, Q_1^* - D_1, Q_2^*\}$$

Condition III: $D_1 + D_2 \geq Q_1{*}$

$D_2 + D_3 \leq Q_2{*}$

Optimum allocation rule: $S_1 = Q_1{*} - \min\{D_2,\ Q_2{*} - D_3,\ Q_1{*}\}$

$S_2 = \min\{D_2,\ Q_2{*} - D_3,\ Q_1{*}\}$

$S_3 = D_3$

Condition IV: $D_1 + D_2 \geq Q_1{*}$

$D_2 + D_3 \geq Q_2{*}$

Optimum allocation rule: $S_1 = \min\{Q_1{*} - S_2,\ D_1\}$

$S_2 = \min\{\max\{Q_1{*} - D_1,\ Q_2{*} - D_3,\ 0\},\ D_2,\ Q_1{*},\ Q_2{*}\}$

$S_3 = \min\{Q_2{*} - S_2,\ D_3\}$

The performance (minimum cost) based on the optimum allocation rule above can be computed as follows:

min cost $= h(Q_1{*} + Q_2{*} - S_1 - 2S_2 - S_3) + p\,(D_1 + D_2 + D_3 - S_1 - S_2 - S_3)$

The optimum allocation rule for Case IV above also applies for any general case of the Problem I.

APPENDIX C: ESTIMATION OF NUMERICAL RESULTS BASED ON THE SCREENING EXPERIMENT FOR PROBLEM I: CASE I

The detail information and computation on how to compute the normal and Weibull table estimations based on the screening experiment values for the randomly sampled designs (Case I) for Problem I are presented in Table C.1. The screening experiment is carried out with (n_1, n_2) = (25, 50), i.e. we sample 25 designs (random sampling) and then run 50 replications for each design.

$\mu_{x(n_0)}$ = \$ 707.40

$\sigma_{x(n_0)}$ = 329.51

σ_{N_0} = 64.78

$\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}}$ = 0.20

$\alpha_{(n_0)}$ = 638.47

$\beta_{(n_0)} = \beta'$ = 5.09

$\gamma_{(n_0)}$ = \$ 84.36

Table C.1: Computation of normal and Weibull table estimation for randomly sampled designs

Computing Budget Allocation		Computation for Normal Table			Computation for Weibull Table				
n_1	n_2	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	**Normal table estimation (\$)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	α'	**Weibull table value**	**Weibull table value** $*\frac{\sigma_{N_0}}{\sqrt{n_2}}$	**Weibull table estimation (\$)**
5000	**5**	0.088	**-3.6621**	**703.74**	28.97	22.04	4.010	116.18	**200.54**
2500	10	0.062	-3.4891	703.91	20.49	31.17	6.444	132.02	216.38
1000	25	0.039	-3.2387	704.16	12.96	49.28	11.875	153.86	238.21
500	50	0.028	-3.0353	704.36	9.16	69.69	18.807	172.30	256.66
200	125	0.018	-2.7456	704.65	5.79	110.19	35.827	207.60	291.96
100	250	0.012	-2.5074	704.89	4.10	155.83	58.277	238.78	323.13

APPENDIX D: ESTIMATION OF NUMERICAL RESULTS BASED ON THE SCREENING EXPERIMENT FOR PROBLEM I: CASE II

The detail information and table showing on how to compute the normal and Weibull table estimations based on the screening experiment values when the SAA is used as the sampling method (Case II) for Problem I for n_0 = 5, n_0 = 20 and n_0 = 50 are presented in Table D.1, Table D.2 and Table D.3 respectively. The screening experiment is carried out with (n_1, n_2) = (25, 50), i.e. we sample 25 designs by SAA and then run 50 replications for each design.

$n_0 = 5$

$\mu_{x(n_0)} = 85.63$

$\sigma_{x(n_0)} = 8.43$

$\sigma_{N_0} = 40.91$

$\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 4.83$

$\alpha_{(n_o)} = 13.74$

$\beta' = \beta_{(n_0)} = 1$

$\gamma_{(n_0)} = 74.46$

Table D.1: Computation of normal and Weibull table estimation for n_0 = 5

Computing Budget Allocation		**Computation for Normal Table**			**Computation for Weibull Table**				
n_1	n_1	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	**Normal table estimation ($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	α'	**Weibull table value**	**Weibull table value** $*\frac{\sigma_{N_0}}{\sqrt{n_2}}$	**Weibull table estimation ($)**
5,000	5	2.17	-1.5215	84.10	18.29	0.75	0.1842	3.3701	77.83
2,500	10	1.53	-1.9123	83.71	12.94	1.06	0.2195	2.8400	77.30
1,000	25	0.97	-2.3265	83.30	8.18	1.68	0.2700	2.2091	76.67
500	50	0.69	-2.4994	83.13	5.79	2.37	0.3010	1.7411	76.21
200	125	0.43	-2.5227	**83.10**	3.66	3.75	0.3672	1.3436	75.81
100	**250**	0.31	-2.3951	83.23	2.59	5.31	0.4066	1.0519	**75.52**

$n_0 = 20$
$\mu_{x(n_0)} = 79.97$
$\sigma_{x(n_0)} = 5.67$
$\sigma_{N_0} = 34.85$
$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 6.15$
$\alpha_{(n_o)} = 7.32$
$\beta' = 1$
$\gamma_{(n_0)} = 73.65$

Table D.2: Computation of normal and Weibull table estimation for $n_0 = 20$

Computing Budget Allocation		**Computation for Normal Table**			**Computation for Weibull Table**				
n_1	n_1	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	**Normal table estimation ($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	α'	**Weibull table value**	**Weibull table value** $*\frac{\sigma_{N_0}}{\sqrt{n_2}}$	**Weibull table estimation ($)**
5,000	5	2.75	1.2367	78.73	15.59	0.47	0.1510	2.3527	76.00
2,500	10	1.94	1.5950	78.37	11.02	0.66	0.1902	2.0963	75.75
1,000	25	1.23	2.0447	77.92	6.97	1.05	0.2359	1.6440	75.29
500	50	0.87	2.2910	77.68	4.93	1.48	0.2707	1.3344	74.98
200	125	0.55	2.4061	**77.56**	3.12	2.35	0.3373	1.0512	74.70
100	**250**	0.39	2.3362	77.63	2.20	3.32	0.3840	0.8465	**74.50**

$n_0 = 50$

$\mu_{x(n_0)} = 77.33$

$\sigma_{x(n_0)} = 5.35$

$\sigma_{N_0} = 34.74$

$\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}} = 6.49$

$\alpha_{(n_o)} = 5.09$

$\beta' = 1$

$\gamma_{(n_0)} = 73.58$

Table D.3: Computation of normal and Weibull table estimation for $n_0 = 50$

Computing Budget Allocation		**Computation for Normal Table**			**Computation for Weibull Table**				
$\boldsymbol{n_1}$	$\boldsymbol{n_1}$	$\dfrac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	**Normal table estimation ($)**	$\dfrac{\sigma_{N_0}}{\sqrt{n_2}}$	α'	**Weibull table value**	**Weibull table value** $* \dfrac{\sigma_{N_0}}{\sqrt{n_2}}$	**Weibull table estimation ($)**
5,000	5	2.90	1.1986	76.13	15.54	0.33	0.1454	2.2586	75.84
2,500	10	2.05	1.5028	75.83	10.99	0.46	0.1737	1.9081	75.49
1,000	25	1.30	1.9762	75.35	6.95	0.73	0.2142	1.4884	75.07
500	50	0.92	2.2347	75.10	4.91	1.04	0.2522	1.2392	74.82
200	**125**	0.58	2.3754	**74.95**	3.11	1.64	0.2987	0.9280	**74.51**
5,000	5	0.41	2.3202	76.13	2.20	2.32	0.3451	0.7582	75.84

Table 5.4: Numerical result for Problem I, designs sampled using SAA ($n_0 = 50$)

APPENDIX E: THE NUMERICAL VALUES AND PARAMETER ESTIMATIONS FOR PROBLEM I

Appendix E.1: CDF of True Performance

In our work, we also experiment for 10 different degrees of n_0 (n_0 = 1, 3, 5, 10, 15, 20, 25, 30, 40, 50). In order to get a clearer picture of the performance of the ATO system in Problem I, these experiments are conducted using separate experiments run with very large number of designs and long replications, i.e. (n_0, n_1, n_2) = (n_0, 5000, 10000). We refer to this experiment as the "detailed experiment". Figure E.1 depicts the CDF of true performance based on the detailed experiment for Problem I.

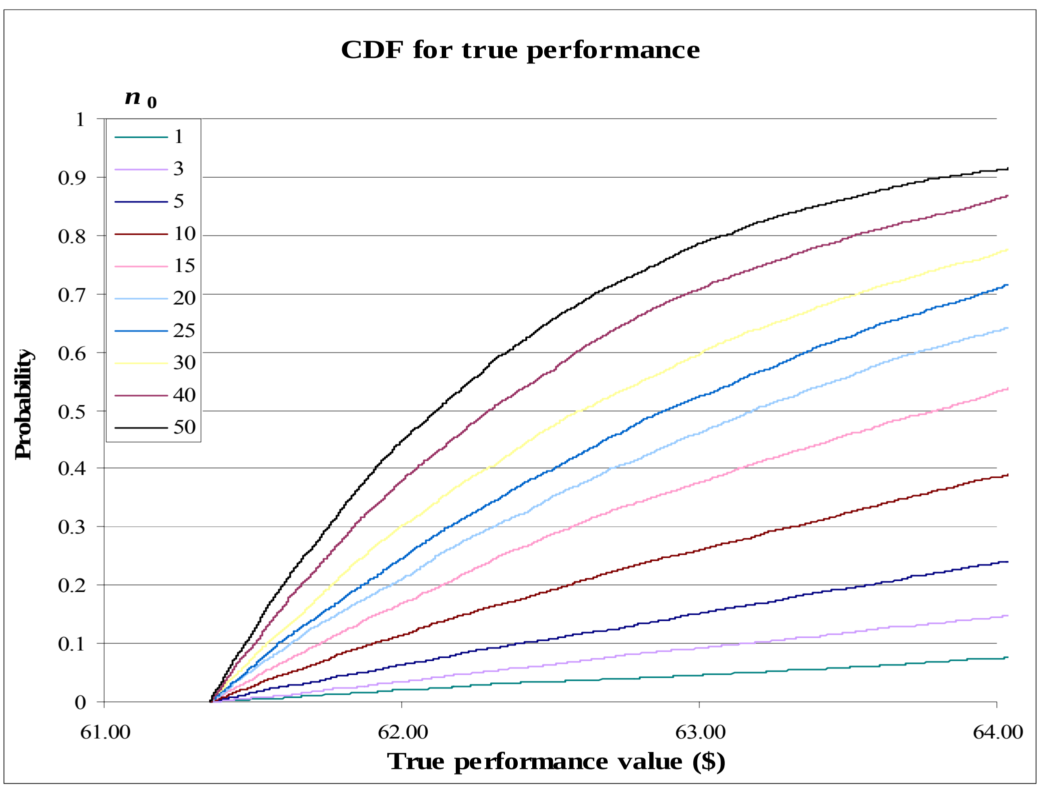

Figure E.1: The improvement in the true performance value when n_0 is varied in Problem I

Appendix E.2: pdf of True Performance

The probability density function (pdf) of the distribution of true performance for the different degrees of n_0 (n_0 = 1, 3, 5, 10, 15, 20, 25, 30, 40, 50) are presented in Figure E.2 to Figure E.11. These true performance values are also obtained via the detailed experiment.

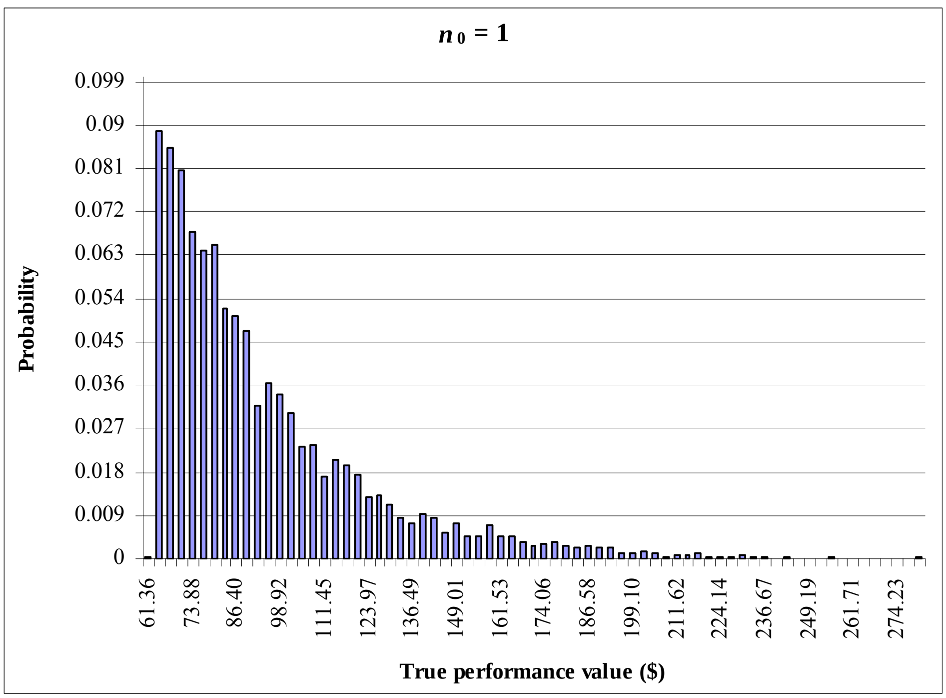

Figure E.2: The pdf of true performance for $n_0 = 1$

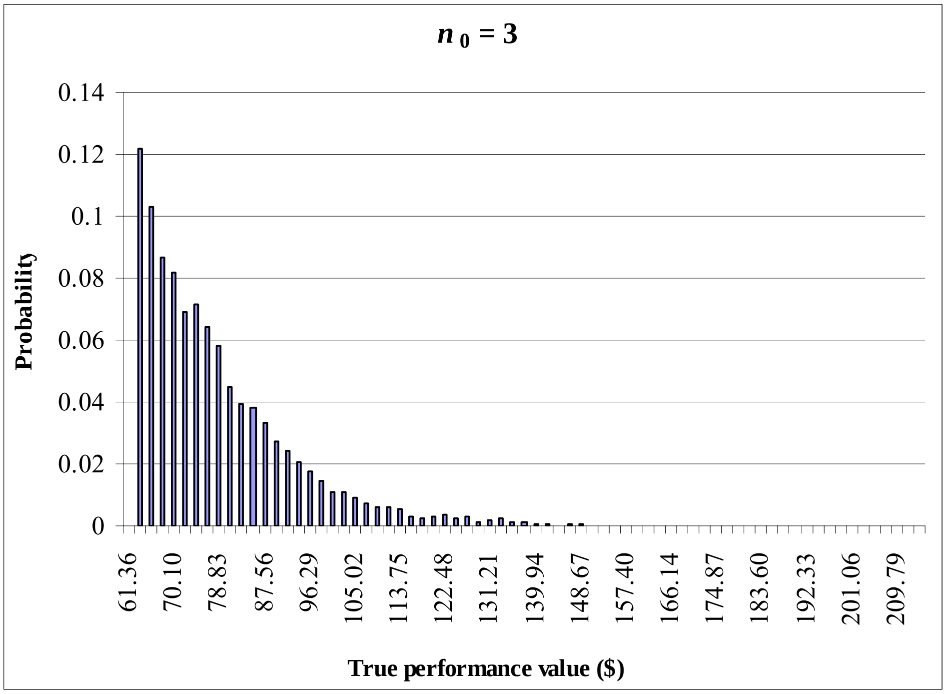

Figure E.3: The pdf of true performance for $n_0 = 3$

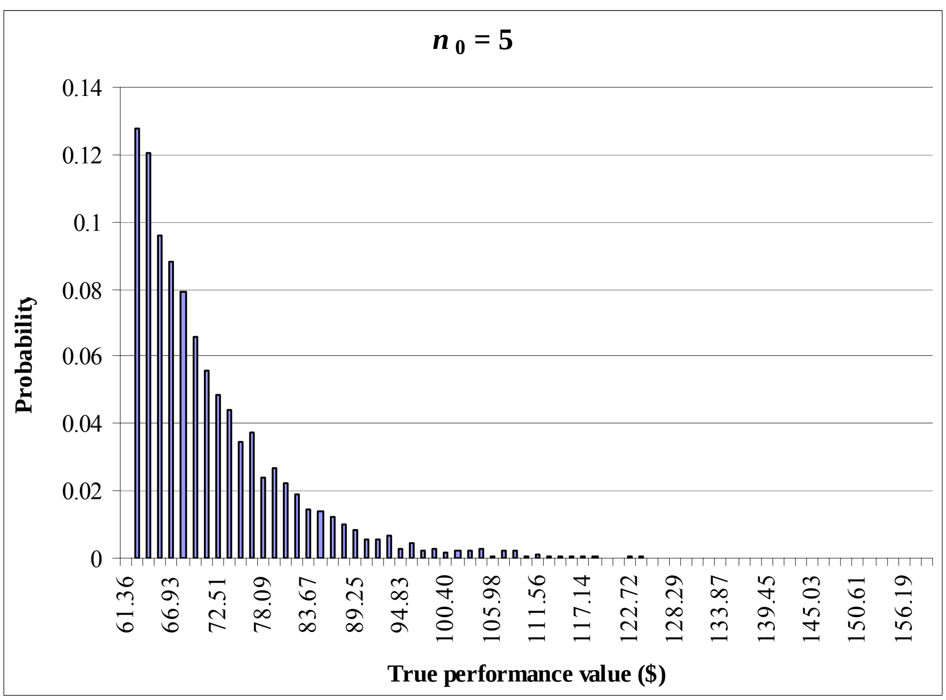

Figure E.4: The pdf of true performance for $n_0 = 5$

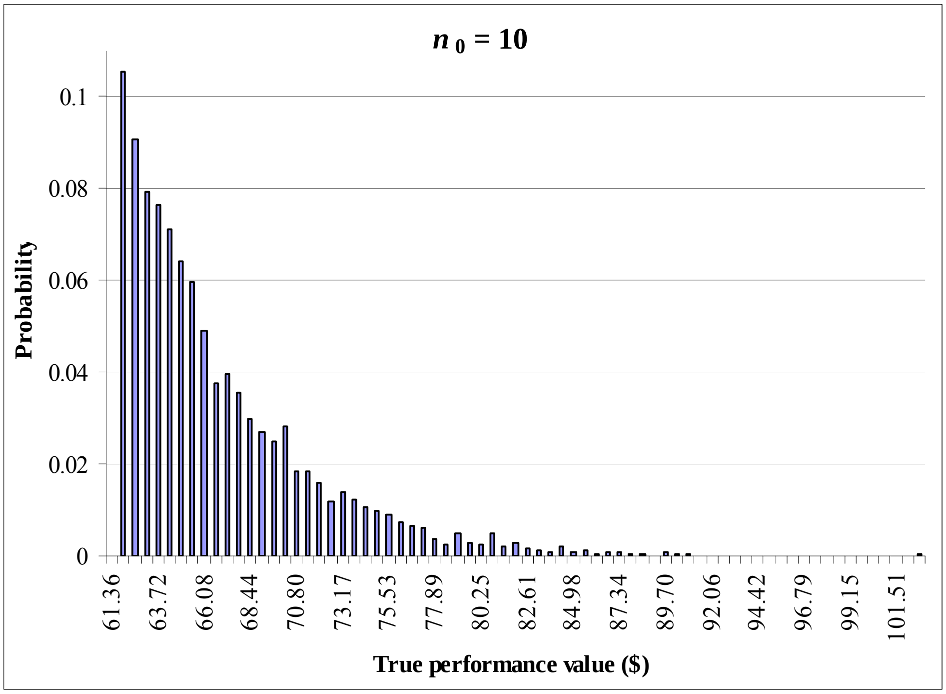

Figure E.5: The pdf of true performance for $n_0 = 10$

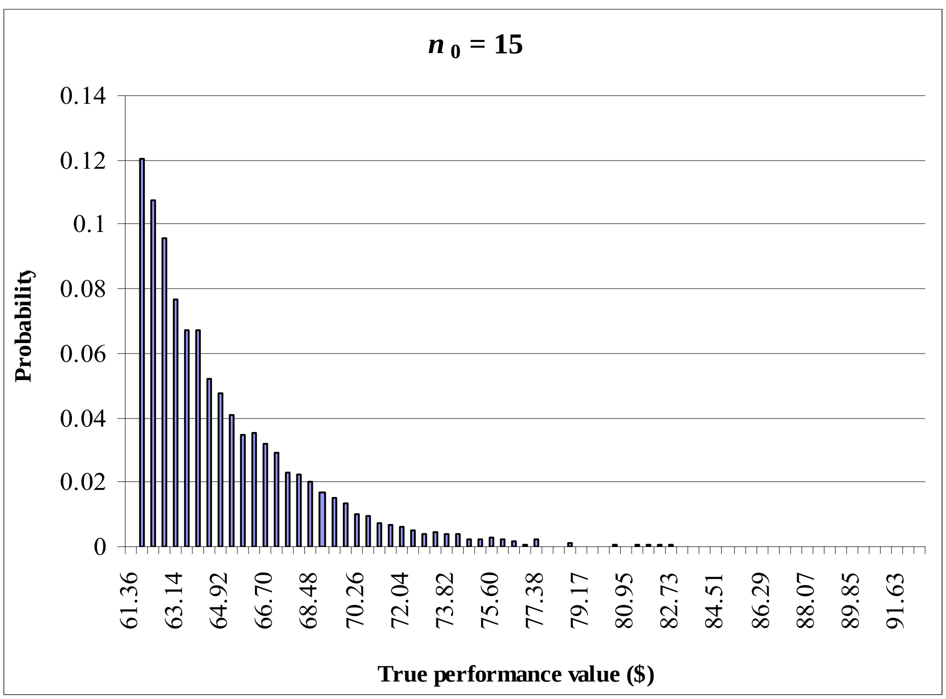

Figure E.6: The pdf of true performance for $n_0 = 15$

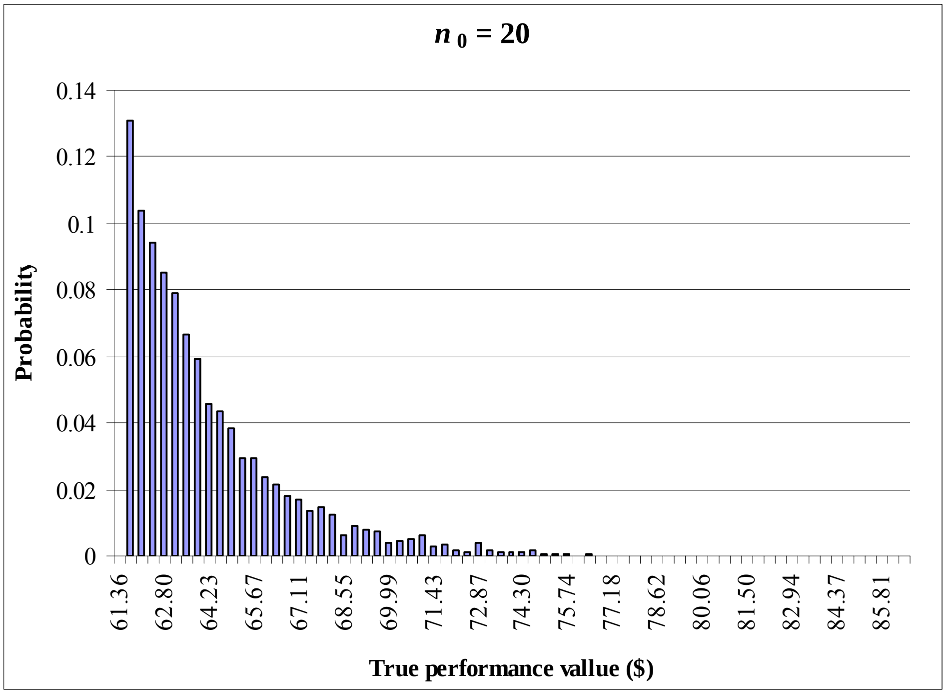

Figure E.7: The pdf of true performance for $n_0 = 20$

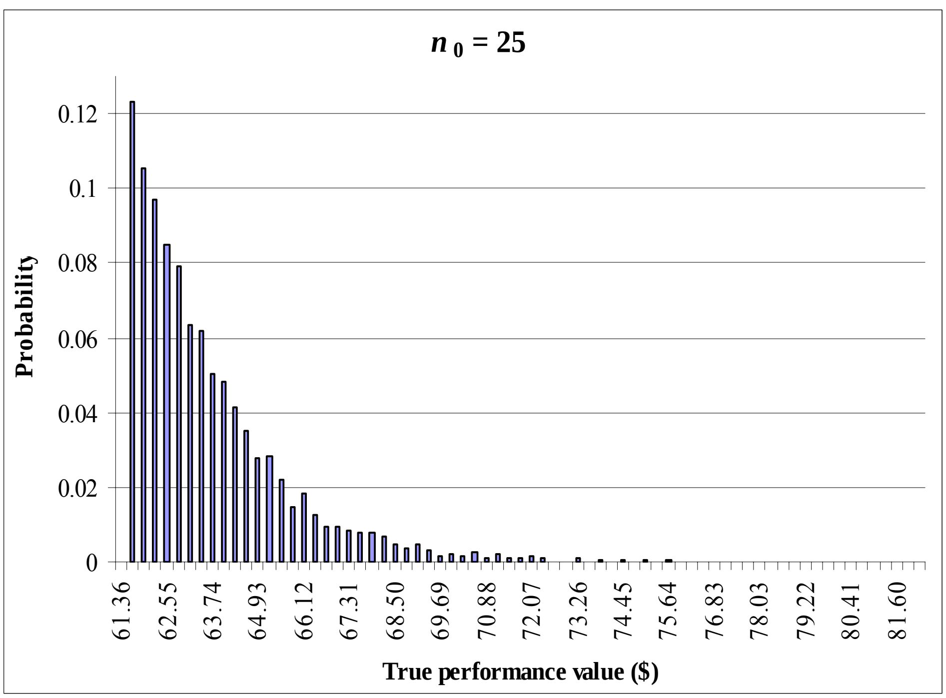

Figure E.8: The pdf of true performance for $n_0 = 25$

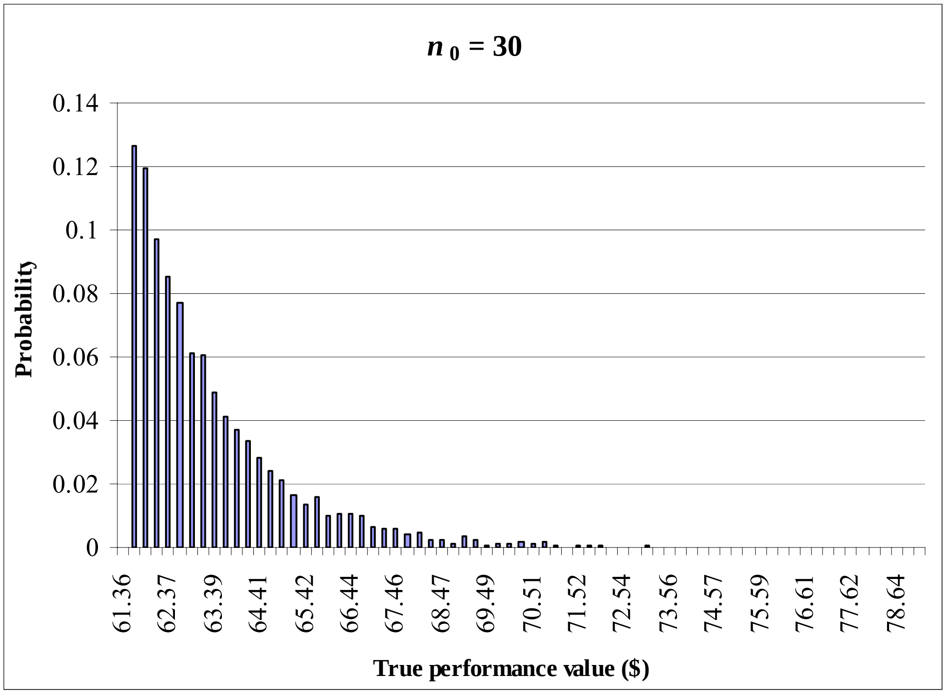

Figure E.9: The pdf of true performance for $n_0 = 30$

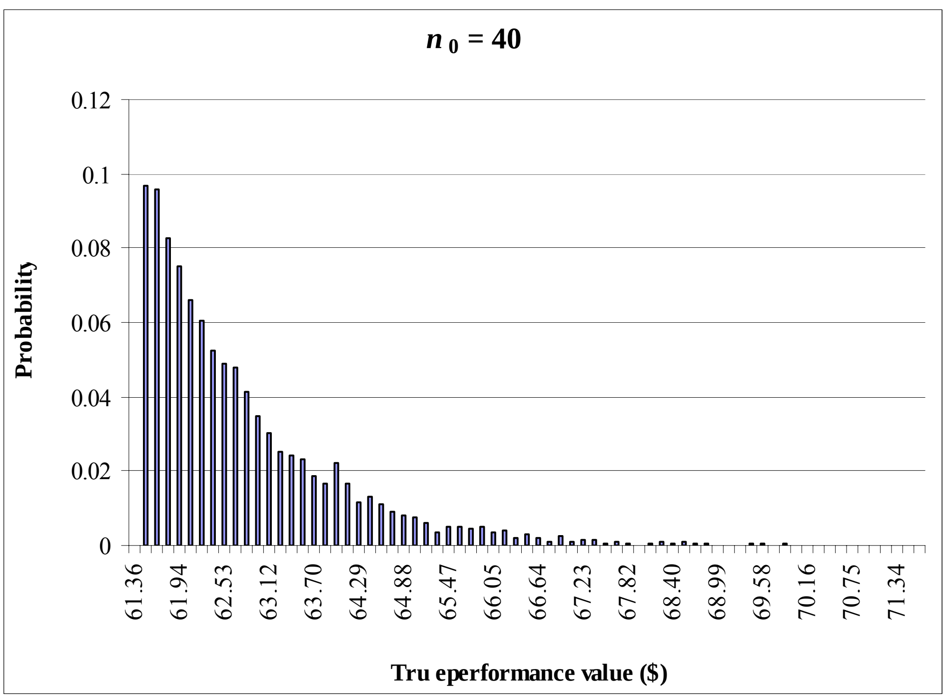

Figure E.10: The pdf of true performance for $n_0 = 40$

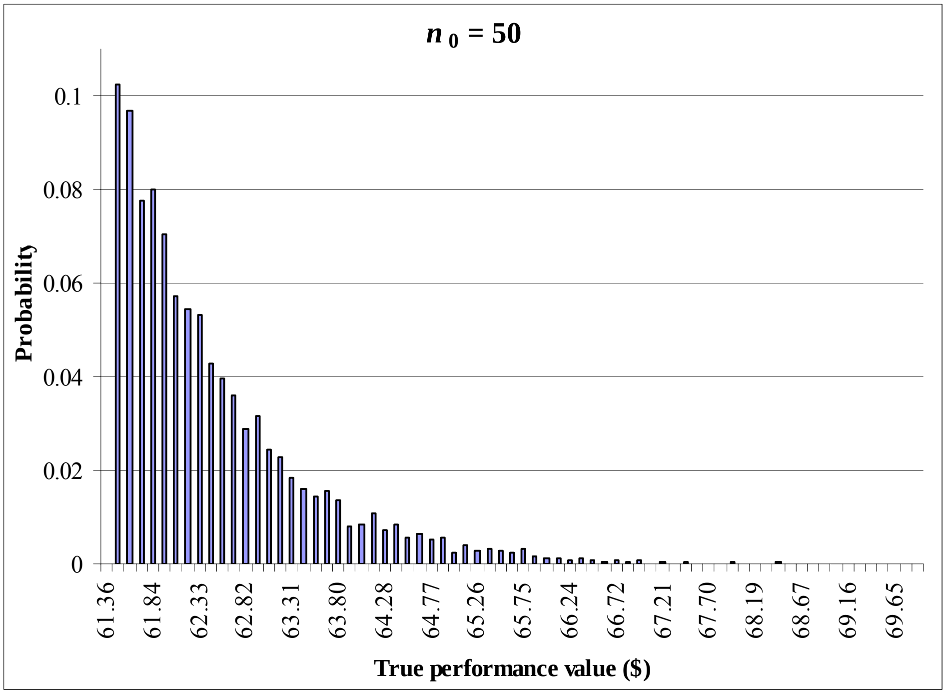

Figure E.11: The pdf of true performance for n_0 = 50

Appendix E.3: Numerical values and parameter estimations for n_0 varied (based on detailed experiment)

The numerical values for the detailed experiment as presented in Appendix E.1 and E.2 are recorded in Table E.1. The values of the minimum true performance (which is the $\gamma_{(n_0)}$), the maximum true performance and the standard deviation of the true performance for each of the different n_0 experimented are recorded in the table. The observed best design, which gives the minimum true performance for each n_0 is also provided in the table. Also presented are the estimated parameter values of $\alpha_{(n_o)}$ and $\mu_{x(n_0)}$ based on the detailed experiment.

Table E.1: Numerical values and parameter estimations based on the detailed experiment for the varied n_0 in Problem I

n_0	Q_1 (unit)	Q_2 (unit)	Expected true performance ($)		Standard deviation	$\alpha_{(n_o)}$	$\mu_{x(n_0)}$
			Min ($\gamma_{(n_0)}$)	Max			
1	2,049	2,052	61.3671	280.4942	29.80	31.31	92.30
3	2051	2049	61.3645	214.1559	14.61	16.32	77.21
5	2,049	2,051	61.3572	158.9745	9.54	9.72	70.99
10	2,055	2,055	61.3565	102.6921	5.01	5.27	66.56
15	2,050	2,052	61.3565	92.5231	3.41	3.49	64.82
20	2,050	2,050	61.3563	86.5326	2.60	2.64	63.99
25	2,049	2,052	61.3563	82.1925	2.05	2.17	63.48
30	2,050	2,050	61.3562	79.1497	1.80	1.82	63.16
40	2,049	2,051	61.3559	71.6325	1.30	1.35	62.69
50	2,049	2,051	61.3557	69.8945	1.06	1.09	62.43

Appendix E.4: Numerical values and parameter estimations for n_0 varied (based on screening experiment)

In Appendix E.3, we captured the information of the numerical experiments and the parameter estimations which is based on the detailed experiment. In this section, we present these values and estimations based on the screening experiment that we actually used for Problem I. As mentioned earlier in the main text, the screening experiment is a much simpler experiment conducted with only (n_0, n_1, n_2) = (n_0, 25, 50). The similar information based on the screening experiment is presented in Table E.2.

Table E.2: Numerical values and parameter estimations based on the screening experiment for the varied n_0 in Problem I

n_0	Expected true performance ($)		Standard deviation	$\alpha_{(n_o)}$	$\mu_{x(n_0)}$
	Min ($\gamma_{(n_0)}$)	**Max**			
1	76.00	175.06	27.65	24.24	105.27
3	74.86	131.68	16.36	23.70	97.03
5	74.46	103.58	8.33	13.74	85.63
10	73.67	102.72	7.53	6.45	80.35
15	73.65	91.23	4.87	5.55	78.46
20	73.65	89.61	4.35	7.32	79.97
25	73.64	88.71	3.97	5.50	78.42
30	73.64	87.87	3.74	5.70	78.26
40	73.60	85.00	2.62	5.39	77.41
50	73.58	81.95	2.39	5.09	77.33

Appendix E.5: Estimation of $t_{(n_o)}$ for Problem I

The numerical estimation of $t_{(n_o)}$ for Problem I is recorded in Table E.3.

Table E.3: Estimation of $t_{(n_o)}$ for Problem I

n_0	$t_{(n_o)}$ (seconds)					
	Trial 1	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**	**Average**
1	0	1	0	0	0	0.2
5	1	0	0	1	0	0.4
10	1	0	1	0	1	0.6
15	1	1	0	1	0	0.6
20	1	1	0	1	1	0.8
25	1	1	1	1	1	1.0
30	1	1	1	2	2	1.4
35	2	2	1	2	2	1.8
40	2	3	3	2	2	2.4
45	3	3	3	3	3	3.0
50	4	4	3	4	3	3.6
55	4	4	4	5	4	4.2
60	5	5	5	5	5	5.0
65	6	6	6	6	6	6.0

Appendix E.6: Estimation of *S* for Problem I

The numerical estimation of *S* for Problem I is recorded in Table E.3.

Table E.4: Estimation of *S* for Problem I

n_2	***S* (seconds)**			
	Trial 1	**Trail 2**	**Trail 3**	**Average**
5,000	10	9	9	9.33
10,000	16	16	16	16.00
15,000	24	23	23	23.33
20,000	30	29	29	29.33
25,000	36	37	36	36.33
30,000	44	44	43	43.67
35,000	52	51	52	51.67
40,000	59	59	59	59.00
45,000	67	67	66	66.67
50,000	75	74	74	74.33
55,000	80	83	82	81.67
60,000	88	90	89	89.00

APPENDIX F: ESTIMATION OF NUMERICAL RESULTS BASED ON THE SCREENING EXPERIMENT FOR PROBLEM I: CASE III

The detail information and table showing on how to compute the normal and Weibull table estimations based on the screening experiment values when the SAA is used as the sampling method with n_0 varied (Case III) for Problem I for K = 800 seconds and K = 3,600 seconds are presented in Table F.1 and Table F.2 respectively. The screening experiment for each allocation decision option is carried out with (n_0, n_1, n_2) = (n_0, 25, 50), i.e. we sample 25 designs by SAA using n_0 degree of information and then run 50 replications for each design.

Table F.1: Computation of normal and Weibull table estimation for K = 800 seconds

Computing Budget Allocation			**Computation for Normal Table**						**Computation for Weibull Table**						
n_0	n_1	n_2	σ_{N_0}	$\sigma_{x(n_0)}$	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	$\mu_{x(n_0)}$	**Normal table estimation ($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\alpha_{(n_o)}$	α'	**Weibull table value**	**Weibull table value*** $\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\gamma_{(n_0)}$	**Weibull table estimation ($)**
5	1,500	120	40.91	8.43	0.44	-3.0716	85.63	82.55	3.73	13.74	3.68	0.3032	1.1321	74.46	75.60
15	**200**	**2,290**	36.31	5.68	0.13	-2.7231	78.46	75.73	0.76	5.55	7.31	0.4130	0.3134	73.65	**73.97**
20	800	145	34.85	5.67	0.51	-2.8299	79.97	77.14	2.89	7.32	2.53	0.2861	0.8281	73.65	74.48
50	200	300	34.74	5.35	0.37	-2.5754	77.33	**74.75**	2.01	5.09	2.54	0.3157	0.6333	73.58	74.21

Table F.2: Computation of normal and Weibull table estimation for K = 3,600 seconds

Computing Budget Allocation			**Computation for Normal Table**						**Computation for Weibull Table**						
n_0	n_1	n_2	σ_{N_0}	$\sigma_{x(n_0)}$	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	$\mu_{x(n_0)}$	**Normal table estimation ($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\alpha_{(n_o)}$	α'	**Weibull table value**	**Weibull table value*** $\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\gamma_{(n_0)}$	**Weibull table estimation ($)**
20	4500	10	34.85	5.67	1.94	-1.6723	79.97	78.30	11.02	7.32	0.66	0.1645	1.8131	73.65	75.46
40	1500	40	37.13	4.65	1.26	-2.0862	77.41	75.32	5.87	5.39	0.92	0.2195	1.2887	73.60	74.89
60	650	350	34.80	5.27	0.35	-2.9391	77.18	74.24	1.86	4.92	2.65	0.3052	0.5675	73.51	74.08
65	**500**	**900**	35.84	5.89	0.20	-2.9776	77.01	**74.03**	1.19	5.01	4.19	0.3332	0.3981	73.49	**73.89**

APPENDIX G: THE NUMERICAL VALUES AND PARAMETER ESTIMATIONS FOR PROBLEM II

Appendix G.1: CDF of True Performance

Similar to Problem I, we also experiment for 8 different degrees of n_0 (n_0 = 1, 3, 5, 7, 10, 15, 18, 20) for Problem II. In order to get a clearer picture of the performance of the ATO system in Problem II, these experiments are conducted by experiments run with very large number of designs and long replications, i.e. (n_0, n_1, n_2) = (n_0, 500, 500). We refer to this experiment as the "detailed experiment". Figure G.1 depicts the CDF of true performance based on the detailed experiment for Problem II.

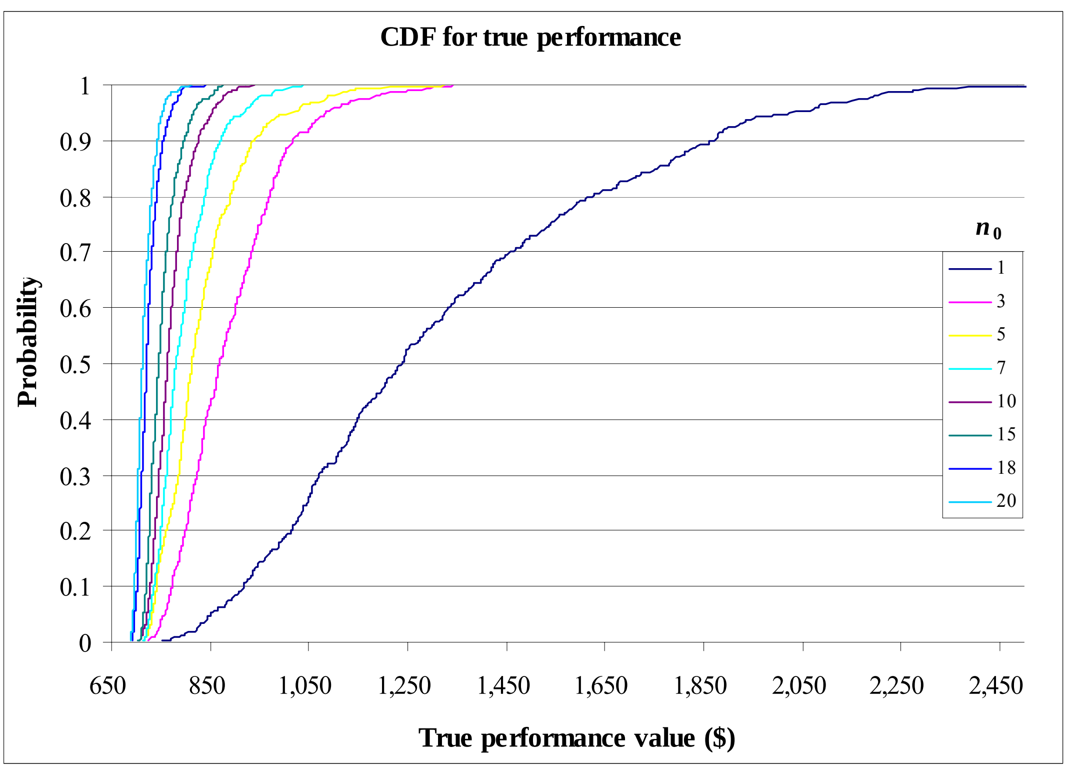

Figure G.1: The improvement in the true performance value when n_0 is varied in Problem II

Appendix G.2: pdf of True Performance

The probability density function (pdf) of the distribution of true performance for the different degrees of n_0 (n_0 = 1, 3, 5, 7, 10, 15, 18, 20) for Problem II are presented in Figure G.2 to Figure G.9. These true performance values are also obtained via the detailed experiment.

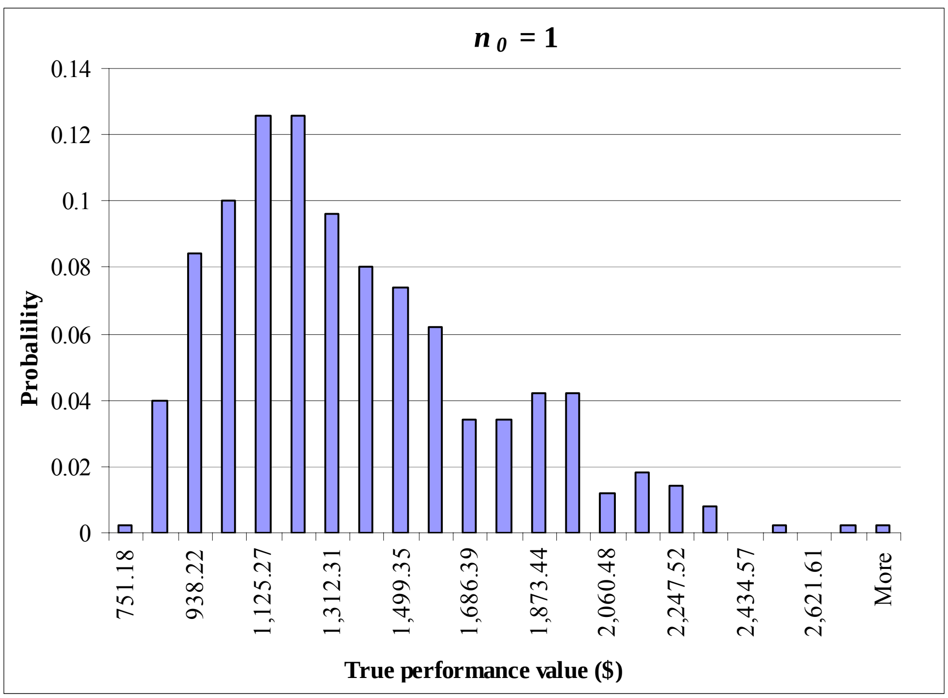

Figure G.2: The pdf of true performance for n_0 = 1

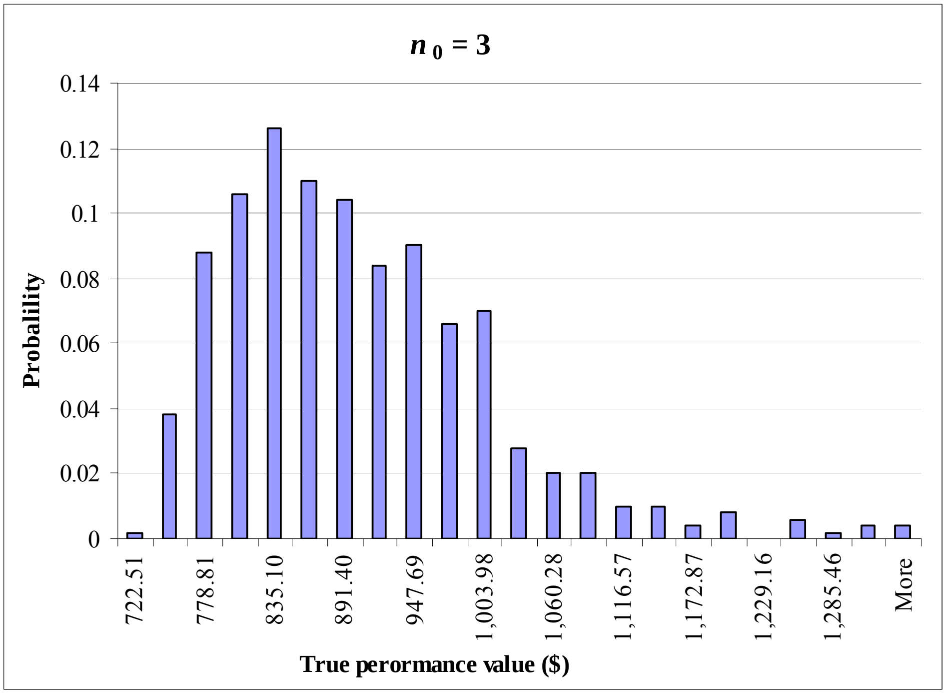

Figure G.3: The pdf of true performance for $n_0 = 3$

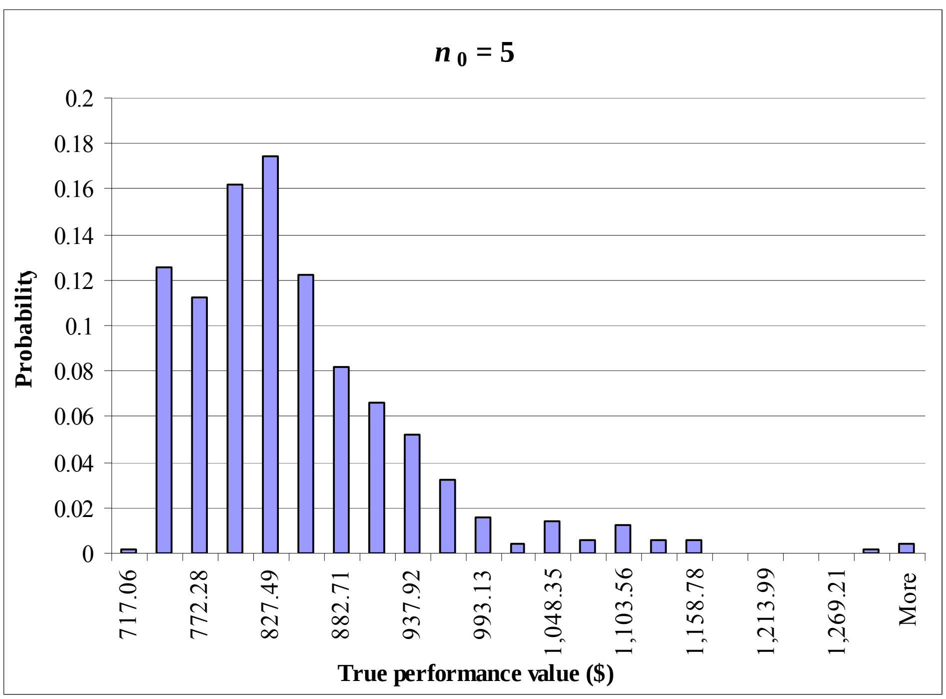

Figure G.4: The pdf of true performance for $n_0 = 5$

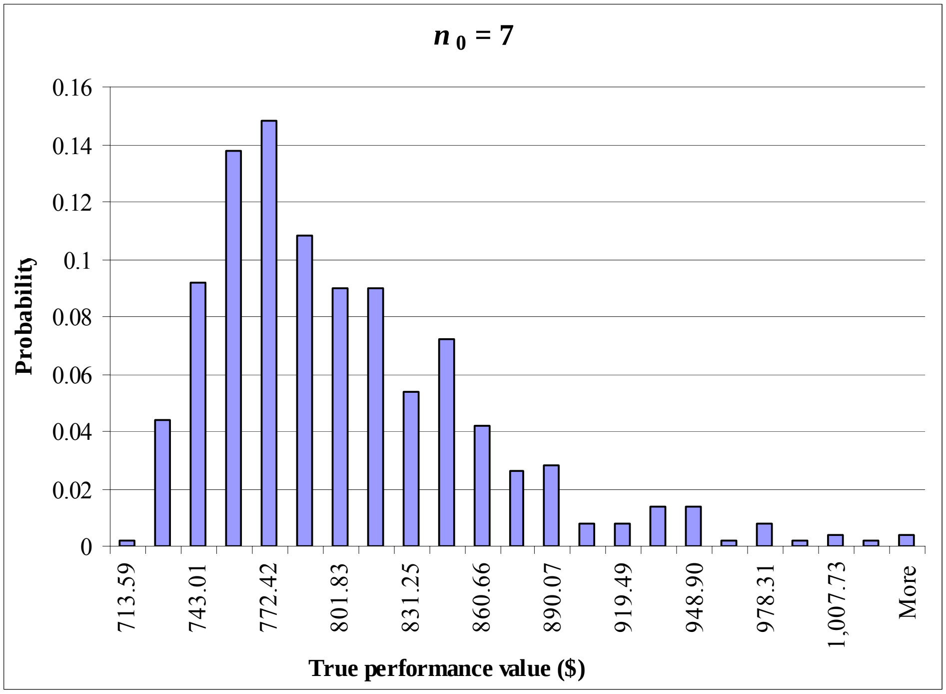

Figure G.5: The pdf of true performance for $n_0 = 7$

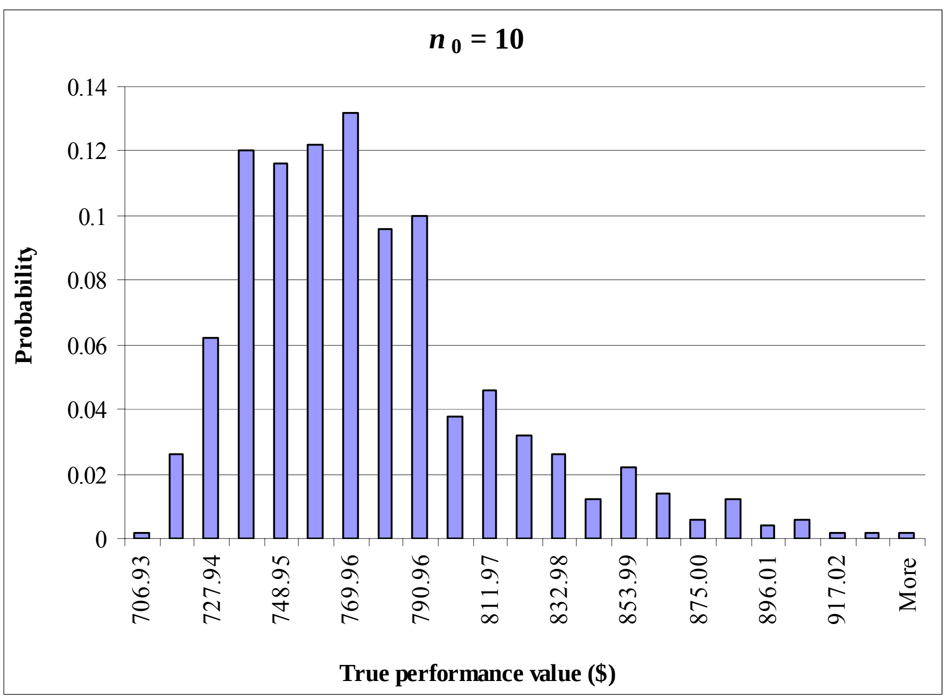

Figure G.6: The pdf of true performance for $n_0 = 10$

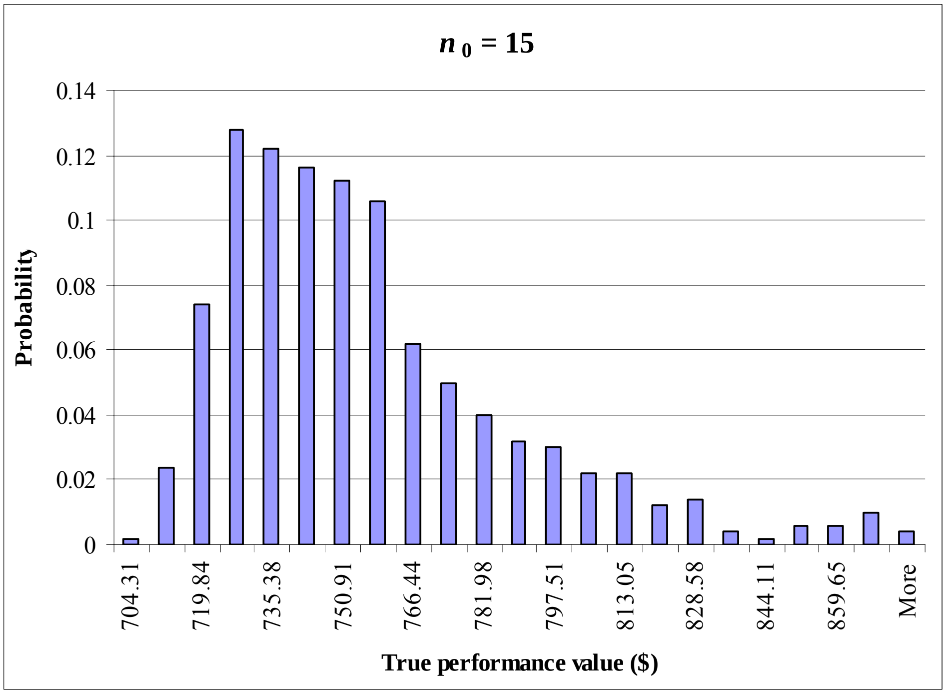

Figure G.7: The pdf of true performance for $n_0 = 15$

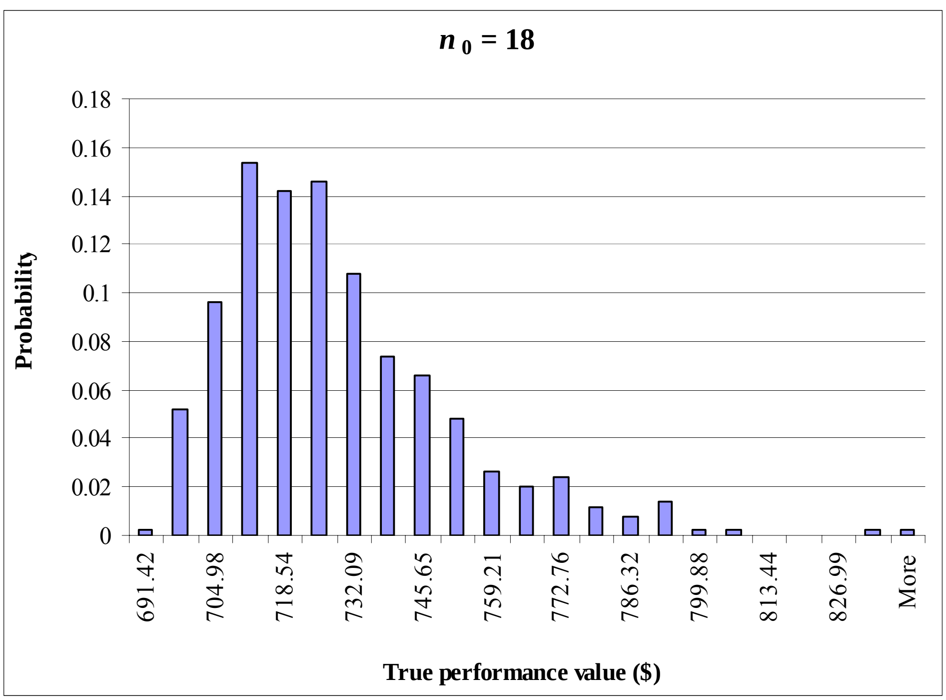

Figure G.8: The pdf of true performance for $n_0 = 18$

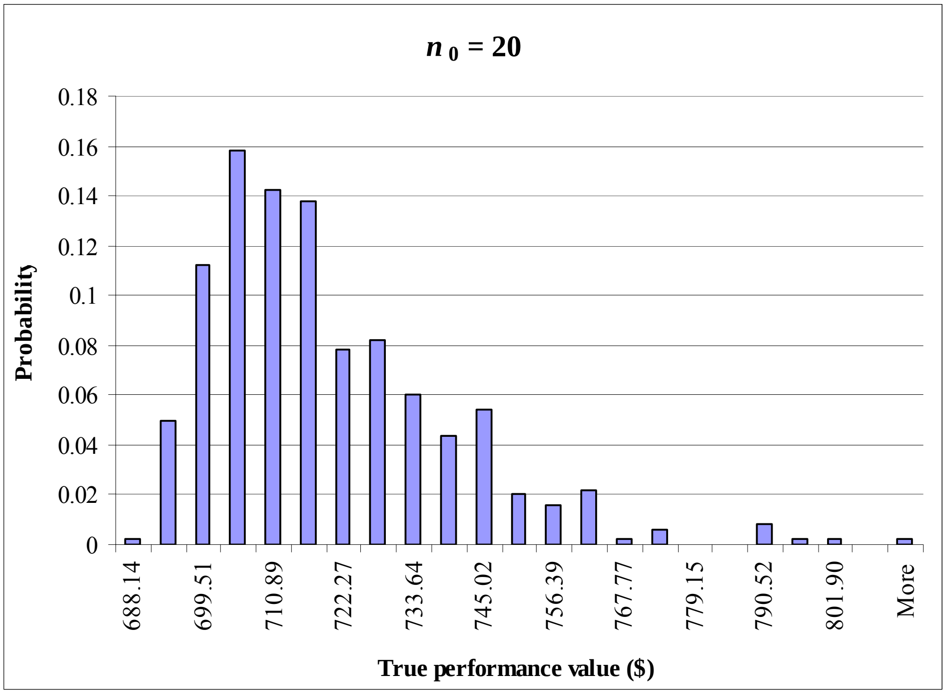

Figure G.9: The pdf of true performance for $n_0 = 20$

Appendix G.3: Numerical values and parameter estimations for n_0 varied (based on detailed experiment)

The numerical values for the detailed experiment as presented in Appendix G.1 and G.2 are recorded in Table G.1. The values of the minimum true performance (which is the $\gamma_{(n_0)}$), the maximum true performance and the standard deviation of the true performance for each of the different n_0 experimented are recorded in the table. The observed best design, which gives the minimum true performance for each n_0 is also provided in the table. Also presented are the estimated parameter values of $\alpha_{(n_o)}$ and $\mu_{x(n_0)}$ based on the detailed experiment.

Table G.1: Numerical values and parameter estimations based on the detailed experiment for the varied n_0 in Problem II

n_0	Q_1 (unit)	Q_2 (unit)	Q_3 (unit)	Q_4 (unit)	Q_5 (unit)	Q_6 (unit)	Expected true performance ($) Min ($\gamma_{(n_0)}$)	Expected true performance ($) Max	Standard deviation	$\alpha_{(n_o)}$	$\mu_{x(n_0)}$
1	4,097	8,651	3,857	3,775	5,637	4,001	751.18	2,808.65	365.69	785.10	1,321.20
3	3,882	8,611	3,838	3,920	5,496	3,996	722.51	1,341.75	107.46	234.26	890.67
5	3,956	8,712	3,745	3,922	5,535	3,945	717.06	1,324.42	90.31	148.13	834.00
7	4,026	8,471	3,801	3,993	5,558	3,957	713.59	1,037.14	57.71	112.03	796.55
10	3,958	8,470	3,800	3,932	5,584	3,937	706.93	938.03	39.83	91.43	771.13
15	4,148	8,874	4,040	3,991	6,075	4,086	704.31	875.18	32.35	68.09	752.84
18	3,988	8,574	3,768	3,837	5,563	3,981	691.42	840.55	22.55	46.15	725.39
20	3,969	8,574	3,743	3,937	5,576	3,959	688.14	813.27	19.50	38.76	716.79

Appendix G.4: Numerical values and parameter estimations for n_0 varied (based on screening experiment)

In Appendix G.3, we captured the information of the numerical experiments and the parameter estimations which is based on the detailed experiment. In this section, we present these values and estimations based on the screening experiment that we actually used for Problem II. As mentioned earlier in the main text, the screening experiment is a much simpler experiment conducted with only (n_0, n_1, n_2) = (n_0, 25, 50). The similar information based on the screening experiment is presented in Table G.2.

Table G.2: Numerical values and parameter estimations based on the screening experiment for the varied n_0 in Problem II

n_0	Expected true performance ($)		Standard deviation	$\alpha_{(n_o)}$	$\mu_{x(n_0)}$
	Min ($\gamma_{(n_0)}$)	Max			
1	767.27	1,715.87	301.44	545.48	1,173.67
3	759.94	1,015.11	70.72	107.59	852.45
5	736.76	970.03	55.34	95.53	804.00
7	714.28	885.55	41.70	81.30	771.99
10	708.80	866.95	38.54	83.05	766.81
13	699.36	814.05	31.61	71.16	748.99
15	696.71	796.22	20.81	52.38	730.57
18	697.34	782.92	24.34	38.64	728.43
20	691.85	749.20	16.13	34.37	717.29

Appendix G.5: Estimation of $t_{(n_o)}$ for Problem II

The numerical estimation of $t_{(n_o)}$ for Problem II is recorded in Table G.3.

Table G.3: Estimation of $t_{(n_o)}$ for Problem II

n_0	$t_{(n_o)}$ (seconds)					
	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Average
1	1	1	1	1	1	1.0
5	2	1	1	1	1	1.2
10	2	2	2	2	1	1.8
15	5	4	4	4	5	4.4
20	8	7	8	7	7	7.4

Appendix G.6: Estimation of *S* for Problem II

The numerical estimation of *S* for Problem II is recorded in Table G.3.

Table G.4: Estimation of *S* for Problem II

n_2	***S*** **(seconds)**			
	Trial 1	**Trail 2**	**Trail 3**	**Average**
50	25	24	24	24.33
100	49	49	49	49.00
150	74	73	74	73.67
200	98	99	98	98.33
250	123	123	123	123.00
300	147	147	148	147.33
350	170	171	171	170.67
400	195	195	195	195.00
450	220	220	220	220.00
500	244	244	244	244.00
550	269	269	269	269.00
600	292	293	293	292.67
650	318	318	318	318.00
700	341	341	341	341.00
750	372	372	371	371.67
800	392	392	392	392.00
850	416	415	415	415.33
900	441	441	441	441.00
950	464	464	463	463.67
1,000	490	490	490	490.00

APPENDIX H: ESTIMATION OF NUMERICAL RESULTS BASED ON THE SCREENING EXPERIMENT FOR PROBLEM II: CASE III

The detail information and table showing on how to compute the normal and Weibull table estimations based on the screening experiment values when the SAA is used as the sampling method with n_0 varied (Case III) for Problem II for K = 3,000 seconds and K = 6,000 seconds are presented in Table H.1 and Table H.2 respectively. The screening experiment for each allocation decision option is carried out with (n_0, n_1, n_2) = (n_0, 25, 50), i.e. we sample 25 designs by SAA using n_0 degree of information and then run 50 replications for each design.

Table H.1: Computation of normal and Weibull table estimation for K = 3,000 seconds

Computing Budget Allocation			**Computation for Normal Table**						**Computation for Weibull Table**						
n_0	n_1	n_2	σ_{N_0}	$\sigma_{x(n_0)}$	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	$\mu_{x(n_0)}$	**Normal table estimation ($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\alpha_{(n_o)}$	α'	**Weibull table value**	**Weibull table value*** $\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\gamma_{(n_0)}$	**Weibull table estimation ($)**
7	50	120	292.72	78.54	0.34	-2.1790	771.99	769.81	26.72	81.30	3.04	0.4155	11.1027	714.28	725.38
20	**100**	**46**	254.96	34.49	1.09	-1.6952	717.29	**715.59**	37.59	34.37	0.91	0.2627	9.8743	691.85	**701.72**
5	1,000	4	280.53	67.84	2.07	-1.3935	804.00	802.61	140.26	95.53	0.68	0.2088	29.2917	736.76	766.05
10	4	1530	285.41	50.91	0.14	-1.4872	766.81	765.33	7.30	83.05	11.38	2.5605	18.6829	708.80	727.48

Table H.2: Computation of normal and Weibull table estimation for K = 6,000 seconds

Computing Budget Allocation			**Computation for Normal Table**						**Computation for Weibull Table**						
n_0	n_1	n_2	σ_{N_0}	$\sigma_{x(n_0)}$	$\frac{\sigma_{N_0}}{\sigma_{x(n_0)}\sqrt{n_2}}$	**Normal table value**	$\mu_{x(n_0)}$	**Normal table estimati on($)**	$\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\alpha_{(n_o)}$	α'	**Weibull table value**	**Weibull table value*** $\frac{\sigma_{N_0}}{\sqrt{n_2}}$	$\gamma_{(n_0)}$	**Weibull table estimation ($)**
1	50	243	498.38	456.70	0.07	-2.2960	1,173.67	1,171.37	31.97	545.48	17.06	0.6904	22.0733	767.27	789.35
20	102	105	254.96	34.49	0.72	-2.0350	717.29	715.25	24.88	34.37	1.38	0.3225	8.0250	691.85	699.88
7	2,615	2	292.72	78.54	2.64	-1.2636	771.99	770.73	206.99	81.30	0.39	0.1609	33.3038	714.28	747.59
20	**50**	**230**	254.96	34.49	0.49	-2.0670	717.29	**715.22**	16.81	34.37	2.04	0.3933	6.6125	691.85	**698.46**