International Journal of Management Science
Vol 13, No 1, May 2007

Simulation Optimization with Statistical Selection Method

Jumi Kim*
Korea Small Business Institute (KOSBI)
16-2 Yoido-dong, Yeongdeungpo-ku, Seoul, 150-742, Korea

(Received Dec. 2005; Accepted Jan. 2007)

ABSTRACT

I propose new combined randomized methods for global optimization problems. These methods are based on the Nested Partitions (NP) method, a useful method for simulation optimization which guarantees global optimal solution but has several shortcomings. To overcome these shortcomings I hired various statistical selection methods and combined with NP method. I first explain the NP method and statistical selection method. And after that I present a detail description of proposed new combined methods and show the results of an application. As well as, I show how these combined methods can be considered in case of computing budget limit problem.

Keywords: Simulation Optimization, Nested Partitions Method, Statistical Selection Method

1. Introduction

Complex and large systems cannot be solved by simple analytical or mathematical methods. For this reason, using simulation is often necessary. Simulation optimization is optimization method with uncertainty. It has been found to be useful in areas such as designing manufacturing systems, evaluating the requirements of computer systems, determining policies in inventory systems, designing and operating transportation facilities, evaluating designs for service organization and analyzing financial systems. These days, it has become one of the most widely used tools in operations research and management science, especially when large but finite feasible region is given.

Evaluating the performance of every feasible point using simulation optimiza-

* Email: jmkim@kosbi.re.kr

tion is very time consuming. Even though many simulation optimization algorithms have been developed, there are some difficulties when applying these algorithms to real world problems. One of the reasons is that there is no guarantee for convergence to the optimal; therefore, new algorithms are needed to overcome this problem. There are many methods for simulation optimization. Deciding which method to use depends on the problem structure. For example, gradient estimation, stochastic approximation, and sample path optimization are applicable when the feasible input variables are continuous. On the other hand, the random search method and statistical method are applicable for discrete input variables. In a recent paper, Shi and Ólafsson [22] introduced an optimization method, the nested partitions (NP) method, for global optimization when the objective function is deterministic or stochastic. In this context, the method has been found to be quite efficient for combinatorial optimization. Also, they show NP method is guaranteed convergence to an optimal solution [22].

Every algorithm that is discussed in this paper is based on the Nested Partitions (NP) method. Even if the Pure NP guarantees the optimal solution, it has two apparent shortcomings as observed in Shi and Ólafsson [22]. First, there are clearly two sources of error in the estimate of each region: the sampling error due to the use of a sample of the points in the region, and the estimation error due to the use of simulation. Secondly, in each iteration, there is no guarantee concerning whether the correct move is made. In Ólafsson [19], a two-stage NP is already proposed to address both of these concerns. By using statistical selection methods to determine a second-stage sample size, it is possible to assure that the correct move is made with a given probability while simultaneously controlling the total error, possibly by using different numbers of sample points in each region.

In this paper I advance two-stage NP method by hiring subset selection which filters inferior feasible region and several other statistical selection methods which improve computational effort. Also I suggest appropriate parameter level to control the computing budget limit.

The remainder of this paper is organized as follows. In Section 2, I define the problem and present NP Framework which is the basis of the paper. In Section 3, state of the art of statistical selection method is presented. In Section 4, I show how to combine NP method with statistical selection method. Experimental results from simulation are reported in Section 5, and Section 6 contains some concluding remarks.

2. NP Algorithm

In this section I present the nested partitions (NP) method for global optimization. This method is primarily motivated by solving problems that have a finite feasible region.

In mathematical notation, I want to solve the problem

$$\min_{\theta \in \Theta} J(\Theta),$$

where Θ is a finite feasible region, and $J: \Theta \to R$ is a performance function to be optimized. In other words, for any feasible point $\theta \in \Theta$, $J(\Theta)$ can not be evaluated analytically. Often $J(\theta)$ is an expectation of some random estimate of the performance of a complex stochastic system given a parameter θ, that is, $J(\theta) = E[L(\theta)]$. Here $L(\theta)$ is a random variable which depends on the parameter $\theta \in \Theta$. I assume that $L(\theta)$ is a discrete event simulation estimate of the true performance, and refer to it as the sample performance.

This problem can be solved by enumerating and comparing all the points to find the one with the best performance. But, the huge feasible region makes this kind of approach infeasible. Most of real problems have no structure that can be exploited to find the optimal solution without checking all the alternatives. A class of NP-complete problems falls within this framework.

The basic idea of the method is to systematically partition the feasible region into subsets and focus the computational effort in those subsets that are considered promising. The Nested Partitions (NP) method is mainly composed of 4 procedures: partitioning, sampling, estimating promising index, and backtracking. In each iteration of the algorithm, it is assumed there is a region, i.e., a subset of Θ, that is considered the *most promising region*. Then this most promising region is *partitioned* into M regions and the entire surrounding region is aggregated into one region. Therefore M+1 disjoint subsets of the feasible region Θ are looked for at each iteration. Each of these M+1 regions is *sampled* using some random sampling scheme, and for each region a *promising index* is calculated. These promising indices are then compared to determine which region is the most promising index in the next iteration. If one of the sub-regions is found to be best, this sub-region becomes the most promising region.

However, if the surrounding region is found to be the best, the algorithm *backtracks* and a larger region containing the current most promising region becomes the new most promising region. The new most promising region is then partitioned and sampled in a similar fashion. This process is repeated until the terminate criteria is satisfied. Generally, simulation is done when the maximum depth is reached. The singleton regions are called regions of maximum depth. Since the singleton regions cannot be partitioned further, they are considered regions of maximum depth.

NP method can be understood as an optimization framework that combines adaptive global sampling with local heuristic search. It uses a flexible partitioning method to divide the design space into regions that can be analyzed individually and then aggregates the results from each region to determine how to continue the search, that is, to concentrate the computational effort. Thus, the NP method adaptively samples from the entire design space and concentrates the sampling effort by systematic partitioning of the design space.

The key features in determining how to implement the method is developing a partitioning method, deciding how much sample effort to use in each region, and how much local search effort to use in each iteration. These factors are of course interconnected. A high quality partition will lessen the need for sampling and local search, and in general increased effort along one of these dimensions decreases the need for the other two. Implementing the NP method can therefore be quite problem dependent, in particular, partitioning schemes that have been devised in the past have drawn heavily on specific structure related to the application itself. This, however, requires substantial effort on part of the practitioner using the method, and in this paper I present a new framework for automating these decisions, namely an intelligent partitioning method, guided random sampling, and guided local search.

In each iteration of the NP method it maintains what is called the most promising region, that is, a sub-region which is considered the most likely to contain the best solution. This most promising region is partitioned into a given number of sub-regions, these sub-regions and the surrounding region is sampled using random sampling, and the sampling information is used to determine which region should be the most promising region in the next iteration.

As opposed to purely heuristic optimization methods, the NP method guarantees that the optimum solution is eventually found [18]. To make clear understanding, I take an example from Shi and Ólafsson [22].

Example 1. Consider a feasible region that consists of eight points $\eta_0 = \Theta = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and that in each iteration we partition the current most promising region into $M = 2$ disjoint sets. In the first iteration, the current most promising region η_0 is partitioned into two sub-regions, $\eta_1 = \{1, 2, 3, 4\}$ and $\eta_2 = \{5, 6, 7, 8\}$. In our terminology, η_0 is then super-region of η_1 and η_2. Both η_1 and η_2 are sampled and the promising index of each region is estimated. Assume that the estimated promising index of η_1 is better than for η_2. We then select η_1 as the most promising region in the second iteration and further it into two sub-regions $\eta_3 = \{1, 2\}$ and $\eta_4 = \{3, 4\}$. In the second iteration, η_3 and η_4, and their surrounding region, η_2, are sampled. If the estimated promising index of η_3 is the best, we then select η_3 to be the most promising region in the third iteration and partition η_3 further into another two sub-regions $\eta_5 = \{1\}$ and $\eta_6 = \{2\}$. Hence in the third iteration, η_5, and η_6, and their surrounding region, $\eta_0 \setminus (\eta_5 \cup \eta_6)$, are sampled and their promising index of the surrounding region is the best, we backtrack to a larger region containing η_3. In this case that would be either η_1 or η_0.

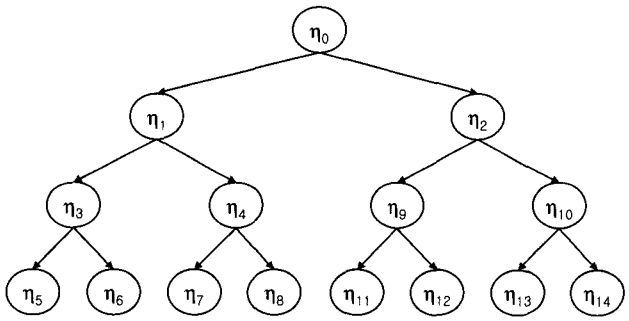

Figure 1. Example of partition generated by the NP method

I define major notations used in this paper and summarized below.

Σ	=	$\{\sigma \subseteq \Theta \mid \sigma$ is a valid region given a fixed partitioning $\}$
$\Sigma_0 \subset \Sigma$	=	$\{\sigma \subseteq \Theta \mid \sigma$ is of maximum depth $\}$
$\sigma(k)$	=	The most promising region in the k^{th} iteration
$\sigma_j(k)$	=	Subregions, $j = 1, 2, \cdots, M$
$\sigma_{M+1}(k)$	=	Surrounding region
$s(\sigma) \in \Sigma$	=	The superregion of $\sigma \in \Sigma$

3. Statistical Selection Method

Discrete-event stochastic simulation is often used to choose the best system among a set of proposed systems where the best is defined by the maximum or minimum expected simulation output. Thus, considerable interests exist for Ranking and Selection (R&S) procedures. The fundamentals of R&S were first proposed by Bechhofer [1]. The original indifference zone R&S procedure proposed by Bechhofer [1] is single-stage and assumes unknown means and known, common variances for all systems. But indifference zone R&S procedures need not be single-stage. By defining the user-specified number of observations, they can extend to multi-stage procedures (sequential procedures) assuming common, known variances. Paulson [20] and Bechhofer *et al.* [2] present such methodologies. Koeing and Law [17] extend the indifference zone approach for use as a screening procedure. Unlike the articles discussed, Dudewicz and Teneja [9] present a multivariate procedure which does not require reduction to a univariate model. If the indifference zone procedures use a least-favorable configuration (LFC) to allocate additional replications, the optimal computing budget allocation (OCBA) [4] and Bayesian decision-theoretic methods [3, 13, 6, 7] use an average case analysis to allocate additional replications [15]. All three procedures assume that simulation output is independent and normally distributed having unknown mean and variance and applicable to both two-stage and sequential procedures. Inoue *et al.* [15] show empirically that the two-stage procedure of Rinott [21] performs competitively with sequential OCBA and Bayesian decision-theoretic methods when the number of systems under consideration is small $(k < 5)$. For a large number of systems $(k \geq 5)$, or when the difference in the mean output of the best system and other systems varies significantly, the Rinott procedure is less effective at identifying the best system. Among two-stage procedures, the Bayesian decision-theoretic procedures have the best overall performance characteristics.

Recently, many articles have tried to unify the fields of R&S and Multiple Comparison Procedures (MCPs). Multiple Comparisons with the Best (MCB) is one of the most widely used MCPs. To apply MCB in a discrete-event simulation, the simulation runs must be independently seeded and the simulation output must be normally distributed, or averaged so that the estimators used are somewhat normally distributed. There are four R&S-MCB procedures having normally distributed data, but do not

Table 1. Summary of Statistical Methods (X_{ij} : The output of the jth replication of system i)

Method	CRN	Single/Two/Sequential Stage	Major Assumption
Procedure P_B Bechhofer [1]		Single (Indifference zone)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$ σ^2 : common, known
Paulson [19] Bechhofer *et al.* [2]		Sequential	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$ σ^2 : common, known
Procedure P_{DD} Dudewicz and Dalal [10]		Two	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Procedure P_{DD} Dudewicz and Zaino [11]		Single (MCP)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Rinott [21]		Two (Indifference zone)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Procedure *R* Nelson and Matejcik [18]		Two (MCB, Indifference zone)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Procedure *DD* Nelson and Matejcik [18]		Two (MCB, Indifference zone)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Procedure *CY* Nelson and Matejcik [18]	*	Two (MCB, Indifference zone)	$X_j \sim N(\mu,\Sigma)$ μ : unknown matrix Σ : unknown variance–covariance matrix
Procedure *NM* Nelson and Matejcik [18]	*	Two (MCB, Indifference zone)	$X_j \sim N(\mu,\Sigma)$ μ : unknown matrix Σ : unknown variance-covariance matrix
Procedure *OCBA* Chen *et al.* [4]		Two/Sequential (Optimal computing budget allocation (OCBA))	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$
Procedure 0-1(*B*) Chick and Inoue [6]		Two/Sequential (Bayesian decision-theoretic methods)	$X_{ij} \overset{i.i.d}{\sim} N(\mu,\sigma^2)$

require known or equal variance: Rinott's Procedure (Procedure *R*), Dudewicz and Dalal's Procedure (Procedure *DD*), Clark and Yang's Procedure (Procedure *CY*), Nelson and Matejcik's Procedure (Procedure *NM*) [18]. Procedure *R* and Procedure *DD* are performed in the same manner with the only difference being in the calculation of the sample means. Both algorithms require independence among all observations. The total sample size depends on the sample variance of the systems. So the larger the sample variance, the more replications are required. Unlike these algorithms, Procedure *CY* and Procedure *NM* requires fewer total observations by employing the CRN. Clark and Yang [8] use the Bonferroni inequality to account for the dependence induced by CRN. However, Nelson and Matejcik [18] observed that the benefit

gained from using Procedure *CY* is diminished when the number of systems to be compared is large. To overcome this problem, they present Procedure *NM*. Procedure *NM* assumes that the unknown variance–covariance matrix exhibits a structure known as sphericity that implies the variances of all paired differences across systems are equal, even though the marginal variances and covariances may be unequal. The difference between Procedure *CY* and *NM* is the calculation of sample variance. This sample variance affects the total number of sample size for second-stage sampling. Nelson and Matejcik [18] reported that Procedure *NM* is superior to Procedure *R*, *DD* and *CY* in terms of the total observations required to obtain the desired confidence level. The only potential drawback with Procedure *NM* is that the assumption of sphericity may not be satisfied. Table 1 summarizes these characteristics.

4. Statistical Selection with NP

One key idea of many statistical ranking and selection methods is that the number of sample points obtained for each system should be proportional to the variance of the performance of each system. When incorporated into the NP method, this intuitively suggests that since the sizes of the regions vary greatly, and, in particular, the surrounding region tends to be much larger than the sub-regions, some regions can be expected to have higher variance and will therefore need a larger sampling size.

To state the two-stage NP approach rigorously, let $D_{ij}(k)$ be the i^{th} set of random sample points selected from the region $\sigma_j(k)$ in the k^{th} iteration, where $i \geq 1,\ j = 1, 2, \cdots, M+1$. Let $N = |D_{ij}(k)|$ denote the initial number of sample points, which is assumed to be constant. In addition let $\theta \in D_{ij}(k)$ denote a point in this set and let $L(\theta)$ be a simulation estimate of the performance of this point. Then in the k^{th} iteration, for every i,

$$X_{ij}(k) = \min_{\theta \in D_{ij}(k)} L(\theta) \tag{1}$$

is an estimate of the performance of the region $\sigma_j(k)$, which is referred to as the i^{th} system performance for the j^{th} system, $i \geq 1,\ j = 1, 2, \cdots, M+1$. The two-stage ranking

and selection procedure first obtains n_0 such system estimates, and then uses that information to determine the total number of N_j of system estimates needed from the j^{th} system, which is, subregion $\sigma_j(k)$. This number is selected to be sufficiently large so that the correct subregion is selected with probability at least P^*, subject to an indifference zone of $\varepsilon > 0$.

Three different methods are used to identify the best systems in terms of sample characteristics. Two of these methods have an assumption of independence of between systems. Generally, independence requires many sampled points. As a result, Nelson and Matejcik [18] suggest using Common Random Numbers (CRNs) for a small number of alternatives.

4.1 Two-Stage Sampling with Subset Selection

When using statistical selection methods, computation can be made more efficient by filtering inferior systems. A subset selection technique is used for filtering systems. The subset selection technique has been studied by many researchers. In 1965, Gupta proposed a single-stage procedure with the assumption that alternatives are independent equal-sized and normally distributed with the common unknown variance. This procedure produces random size subsets having an optimal system with prespecified probability P^* [21] without an indifference zone. In 1989, Sullivan and Wilson [23] proposed a general restricted subset selection procedure that allows unknown and unequal variance with an indifference zone having an exact size to be included in a subset. Unlike Gupta's method, the number of systems in the subset can be controlled. In 1993, Gupta and Santer extended the above methods for prespecifying the maximum size of a subset and showed relationship between indifference zone approaches. It is efficient if the size of a subset is clearly upper-bounding than having the exact size of the subset method. Because exact size of subset method was used, then some inferior system which is already known could be included. The shortcoming of the subset selection approach is that the best system cannot be found. As an illustration of the two-stage NP approach, Ólafsson [19] uses the classic Rinott's ranking and selection procedure. An indifference zone ε is assumed to be given that describes our tolerance for selecting a system that has up to ε units worse performance than the optimal performance. By using ε, the number of systems being compared, and the desired probability P^* of correct selection, a constant h is calcu-

lated. Then n_0 initial samples are obtained from each system. After calculating sample variance S_j^2 for each system, the second stage sample size for each system is calculated according the following formula.

$$N_j(k) = \max\left\{n_0 + 1, \left\lceil \frac{h^2 S_j^2(k)}{\varepsilon^2} \right\rceil\right\} \tag{2}$$

After finishing second stage sampling, a system with performance within ε of the optimal performance is selected with probability P^* [21].

Another incorporation method with NP is Procedure P_{DD}. Procedure P_{DD} is originally proposed by Dudewicz and Dalal [10]. This method assumes normality and independence of observations. Procedure P_{DD} is almost the same as Rinott's Procedure but the difference is the selection of the best system is based on weighted averages. They use the weighted average of each stage to find the best system. Weights are calculated according the following formula.

$$W_{j1}(k) = \frac{n_0}{N_j(k)}\left[1 + \left\{\left(1 - \frac{N_j(k)}{n_0}\left(1 - \frac{(N_j(k) - n_0)\varepsilon^2}{h^2 S_j^2(k)}\right)\right)^{1/2}\right\}\right], \quad W_{j2}(k) = 1 - W_{j1}(k) \tag{3}$$

The shortcomings of the Rinott procedure are well documented. Most notably, the derivations of equation (2) assumes the least favorable configuration among the system, which typically leads to a very conservative value for the number of sample points which tends to require too much sampling effort. Also, equation (2) only uses the variance, such that there is no consideration for the mean performance in the first stage. Thus, it may be beneficial in terms of computation time to filter out such inferior systems, which can be accomplished by combining it with a subset selection procedure, resulting in the following algorithm:

Algorithm NP/Subset/Rinott

Step 1. *Initialization*

Set $k = 0$ and $\sigma(k) = \Theta$.

Specify the overall desired probability P^* of correct selection and indifference zone ε, the common initial sample size $n_0 \geq 2$, the number of sub-

regions M. Determine t from the t-distribution and h for Rinott's integral. t and h are constants which are determined by n_0, the minimum probability P^* of correct selection, and M (See the tables in Bechhofer *et al.* [2]).

$$t = t_{1-(1-\frac{\alpha}{2})^{\frac{1}{M-1}}, n_0-1}.$$

Step 2. *Partitioning*

Given the current most promising region $\sigma(k)$, partition $\sigma(k)$ into M sub-regions $\sigma_1(k), \cdots, \sigma_M(k)$, and aggregate the surrounding region $\Theta \setminus \sigma(k)$ into one region $\sigma_{M+1}(k)$.

Step 3. *First-Stage Sampling*

Step 3-1. Let $i = 1$.

Step 3-2. Use uniform sampling to obtain a set $D_{ij}(k)$ of N sampling points from region $j = 1, 2, \cdots, M+1$.

Step 3-3. Use discrete event simulation of the system to obtain a sample performance $L(\theta)$ for every $\theta \in D_{ij}(k)$ and estimate the performance of the region as

$$X_{ij}(k) = \min_{\theta \in D_{ij}(k)} L(\theta).$$

Step 3-4. If $i = n_0$ continue to Step 4. Otherwise, let $i = i+1$ $i = i+1$ and go back to Step 3-2.

Step 4. *Estimating Mean and Variance of First-Stage Sampling*

Calculate first-stage sample means and variances

$$\overline{X}_j^{(1)}(k) = \frac{1}{n_0}\sum_{i=1}^{n_0} X_{ij}(k), \quad S_j^2 = \frac{\sum_{i=1}^{n_0}[X_{ij}(k) - \overline{X}_j^{(1)}(k)]^2}{n_0 - 1}, \text{ for } j = 1, 2, \cdots, M+1.$$

Step 5. *Filtering*

Calculate the quantity

$$W_{ij}(k) = t\left(\frac{S_i^2(k) + S_j^2(k)}{n_0}\right)^{1/2} \text{ for all } i \neq j.$$

Include the i^{th} region in the selected subset I if $\overline{X}_i(k) \leq \overline{X}_j(k) + (W_{ij}(k) - \varepsilon)^+$ for all $i \neq j$.

Step 6. *Computing Total Sample Size*

If I contains only a single region, $I = \{\sigma_{\hat{j}}(k)\}$, then this has the best promising index so update $\sigma(k+1) = \sigma_{\hat{j}}(k)$ and go to Step 11. Otherwise, compute the total sample size for all $j \in I$

$$N_j(k) = \max\left\{n_0 + 1, \left\lceil \frac{h^2 S_j^2(k)}{\varepsilon^2} \right\rceil\right\}$$

where ε is the indifference zone and h is a constant determined by n_0 and the minimum probability P^* of correct selection.

Step 7. *Second-Stage Sampling*

Obtain $N_j(k) - n_0$ more simulation estimates of the system performance for all $j \in I$ as in Step 3-1 through Step 3-4 above.

Step 8. *Estimating Mean of Second-Stage Sampling*

Let the overall sample mean be the promising index for all $j \in I$,

$$\hat{I}(\sigma_j(k)) = \bar{X}_j(k) = \frac{\sum_{i=1}^{N_j(k)} X_{ij}(k)}{N_j(k)}.$$

Step 9. *Determining the Most Promising Index*

Select the index of the region with the best promising index,

$$\hat{j}_k \in \arg\min \hat{I}(\sigma_j(k)) \quad \text{for all } j \in I.$$

If more than one region is equally promising, the tie can be broken arbitrarily. If this index corresponds to a region that is a sub-region, $\sigma(k)$, then let this be the most promising region in the next iterations. Otherwise, if the index corresponds to the surrounding region, backtrack to a larger region containing the current most promising region. That is, let

$$\sigma(k+1) = \begin{cases} \sigma_{\hat{i}_k}(k), & \text{if } \hat{i}_k < M+1 \\ s(\sigma(k)), & \text{otherwise} \end{cases}$$

Step 10. *Checking Stopping Rule*

If $\sigma(k+1) \in \Sigma_0$, stop else $k = k+1$ and go back to Step 2.

Algorithm NP/Subset/DD

Step 1. *Initialization*

See Step 1 in Algorithm NP/Subset/Rinott.

Step 2. *Partitioning*

See Step 2 in Algorithm NP/Subset/Rinott

Step 3. *First-Stage Sampling*

See Step 3 in Algorithm NP/Subset/Rinott

Step 4. *Estimating Mean and Variance of First-Stage Sampling*

See Step 4 in Algorithm NP/Subset/Rinott

Step 5. *Filtering Subset*

See Step 5 in Algorithm NP/Subset/Rinott

Step 6. *Computing Total Sample Size for Second-Stage Sampling*

See Step 6 in Algorithm NP/Subset/Rinott

Step 7. *Second-Stage Sampling*

See Step 7 in Algorithm NP/Subset/Rinott

Step 8. *Estimating Mean of Second-Stage Sampling*

Calculate the second-stage sample means based on $N_j(k)-n_0$ replications

$$\overline{X}_j^{(2)}(k)=\frac{1}{N_j(k)-n_0}\sum_{i=n_0+1}^{N_j}X_{ij}(k).$$

Step 9. *Calculating Weights for each Stage Samples*

$$W_{j1}(k)=\frac{n_0}{N_j(k)}\left[1+\left\{\left(1-\frac{N_j(k)}{n_0}\left(1-\frac{(N_j(k)-n_0)\varepsilon^2}{h^2S_j^2(k)}\right)\right)^{1/2}\right\}\right],\quad W_{j2}(k)=1-W_{j1}(k).$$

Step 10. *Calculating Weighted Averages*

Calculate weighted averages for all $j\in I$

$$\overline{X}_j(k)=W_{j1}\overline{X}_j^{(1)}(k)+W_{j2}\overline{X}_j^{(2)}(k).$$

and let these weighted averages be the promising index for all $j\in I$,

$$\hat{I}(\sigma_j(k))=\bar{X}_j(k)$$

Step 11. *Determining the Most Promising Index*

See Step 9 in Algorithm NP/Subset/Rinott

Step 12 *Checking Stopping Rule*

See Step 10 in Algorithm NP/Subset/Rinott

4.2 Two-Stage Sampling with Nelson-Matejcik

One assumption in the statistical selection procedure used by the NP/Subset/Rinott Algorithm is that each system is independent, which implies the simulation samples for comparing the regions must also be independent. However, when comparing simulated systems, researchers prefer to use common random numbers (CRNs), thus making the systems independent. Hence it is important to consider statistical selection methods that allow for correlated systems. One such method is proposed by Nelson and Matejcik [18] which will now be incorporated into the NP framework. Given a fixed first-stage sample size n_0, first-stage samples are randomly obtained from each region by using the same stream of random numbers for each region. Using these samples, sample variance S of the difference of the sample means is determined, then use this to compute the final sample size given indifference zone ε

$$N = \max\left\{n_0, \left\lceil \left(\frac{gS}{\varepsilon}\right)^2 \right\rceil\right\} \tag{4}$$

Note that this requires computing the constant g which depends on the initial sample size n_0 and the number of regions M that are compared [18]. Furthermore, note that unlike Rinott's two-stage sampling, the sample size for each system is identical in the second stage.

Algorithm NP/ NM

Step 1. *Initialization*

Set $k = 0$ and $\sigma(k) = \Theta$.

Specify the constants $\varepsilon, \alpha,$ and n_0 Let $g = T^{(\alpha)}_{k-1,(k-1)(n_0-1),0.5}$, an equicoordinate critical point of the equicorrelated multivariate central t-distribution; the constant can be found in Hochberg and Tamhane [14], Appendix 3, Table 4; Bechhofer *et al.* [2]; or by using the FORTRAN program AS251 of Dunnet [12].

Step 2. *Partitioning*

Given the current most promising region $\sigma(k)$ partition $\sigma(k)$ into M sub-regions

$\sigma_1(k), \cdots, \sigma_M(k)$, and aggregate the surrounding region $\Theta \setminus \sigma(k)$ into one region $\sigma_{M+1}(k)$

Step 3. *First-Stage Sampling*

Step 3-1. Let $i = 1$.

Step 3-2. Use uniform sampling to obtain a set $D_{ij}(k)$ of N sampling points from region $j = 1, 2, \cdots, M+1$ using CRN across regions.

Step 3-3. Use discrete event simulation of the system to obtain a sample performance $L(\theta)$ for every $\theta \in D_{ij}(k)$ and estimate the performance of the region as $X_{ij}(k) = \min_{\theta \in D_{ij}(k)} L(\theta)$.

Step 3-4. If $i = n_0$, continue to Step 4.

Otherwise, let $i = i+1$ and go back to Step 3-1.

Step 4. *Estimating the Variance of First-Stage Sampling*

Compute the approximate sample variance of the difference of the sample means

$$S^2(k) = \frac{2\sum_{j=1}^{k}\sum_{i=1}^{n_0}(X_{ij}(k) - \bar{X}_{i\cdot}(k) - \bar{X}_{\cdot j}(k) + \bar{X}_{\cdot\cdot}(k))^2}{(k-1)(n_0-1)}.$$

Where $\bar{X}_{i\cdot}(k) = \sum_{j=1}^{k} X_{ij}(k)/k$, $\bar{X}_{\cdot j}(k) = \sum_{i=1}^{n_0} X_{ij}(k)/n_0$ and

$$\bar{X}_{\cdot\cdot}(k) = \sum_{i=1}^{n_0}\sum_{j=1}^{k} X_{ij}(k)/kn_0$$

Step 5. *Computing Total Sample Size*

Compute the total sample size

$$N(k) = \max\left\{n_0, \left\lceil \left(\frac{gS(k)}{\varepsilon}\right)^2 \right\rceil\right\}.$$

Step 6. *Second-Stage Sampling*

See Step 7 in Algorithm NP/Subset/Rinott

Step 7. *Estimating Mean of Second-Stage Sampling*

See Step 8 in Algorithm NP/Subset/Rinott

Step 8. *Determining the Most Promising Index*

See Step 9 in Algorithm NP/Subset/Rinott

Step 9. *Checking Stopping Rule*
See Step 10 in Algorithm NP/Subset/Rinott

5. Numerical Results

To numerically evaluate the performance of suggested algorithms two different kinds of problems are considered.

The first problem is production system with a given number of M workstations configured in parallel and jobs that are to be processed by exactly one of the stations. The objective is to find the optimal resource allocation that minimizes the expected makespan having assumption that there are some R resources that can be assigned to perform the necessary work within each station, and those resources can be moved to other workstations upon completion of a job. This is a Monte Carlo simulation where the randomness derives from random processing times, subsequently referred in this paper as the Monte Carlo problem. The following parameters are used in all experiments. Let $M = 2, R = 5,$ and the first stage sample points in each region set to $n_0 = 20$. Twenty replications are used for each experiment which were performed with $P^* \in [0.55, 0.95]$.

Second problem is queuing problem. In this problem, each server represents a user and each buffer slot represents a resource that is to be allocated to a user. Jobs arrive at this system at a rate of λ. Each user is processing jobs at a rate $\mu_i, i = 1, 2, \cdots, N,$ and if a job is routed to a user with a full queue, the job is lost. Let $L_i(n_i)$ be the probability of the i^{th} server losing a job (n_i is the number of buffers allocated to the i^{th} server). The goal is to allocate all K available buffer slots to the users in order to minimize job loss. Let $\lambda = 10, \mu = 10, N = 6, K = 18$ and $n_0 = 10$. Twenty replications are used for each experiment with $P^* \in [0.55, 0.95]$.

5.1 Numerical Evaluation of Two-Stage Sampling with Subset Selection

One of the primary benefits of two-stage sampling is that more computational effort is allocated in regions where it is needed. To insure that the two-stage approach indeed makes a substantial difference, the total number of sample points is used at each

depth level. To show the results Monte Carlo problem which is mentioned above is used. Results are shown in Figure 2. These figures show that the computational effort decreases as the depth increases, although there is a peak at depth two because this is the first depth where a surrounding region is considered. Intuitively the reason for this may be that, as the depth increases, the sub-regions become more and more homogeneous leading to lower variance, and hence less effort is required to evaluate each region. The opposite is true for the surrounding region, but for Algorithms NP/Subset/Rinott and NP/Subset/DD, it may often be possible to filter this region out early, especially when substantial progress has been made and the quality of the subregion is high.

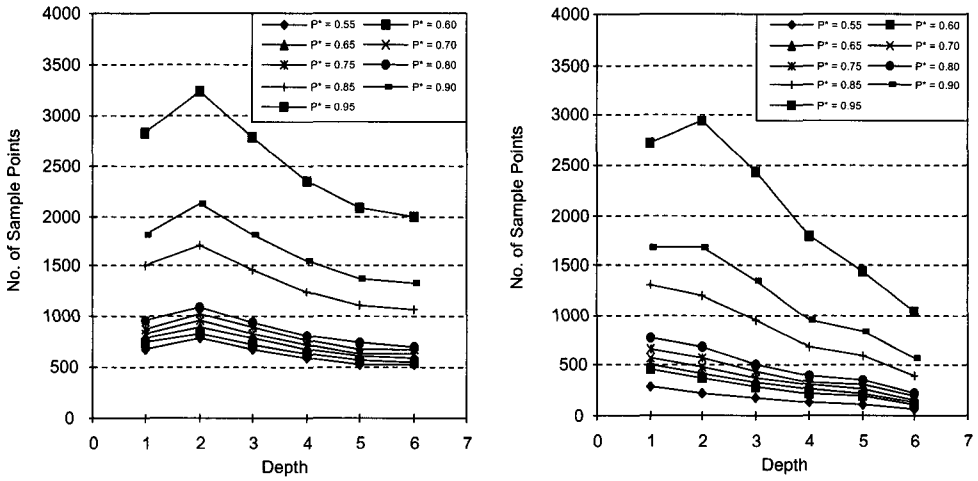

Figure 2. Total Number of Sample Points of Each Depth for NP/Rinott (Left) and NP/Subset/Rinott (Right)

The potential benefit of two-stage methods without subset selection is illustrated in both figures. When the pure NP method is used, the number of sample points is constant. The left plot of Figure 2 shows that over 3,000 sample points are needed to guarantee 95% success probability at depth two. Contrast with this, what is needed using the two-stage sampling at depth six is only 2,000 samples. Thus, variable sampling reduces total computational effort by one-third. In addition, the subset selection creates an even greater savings, and for many settings of P^* the effort that would be required for NP without two-stage sampling is three times that of which would be required for the NP/Subset/Rinott Algorithm.

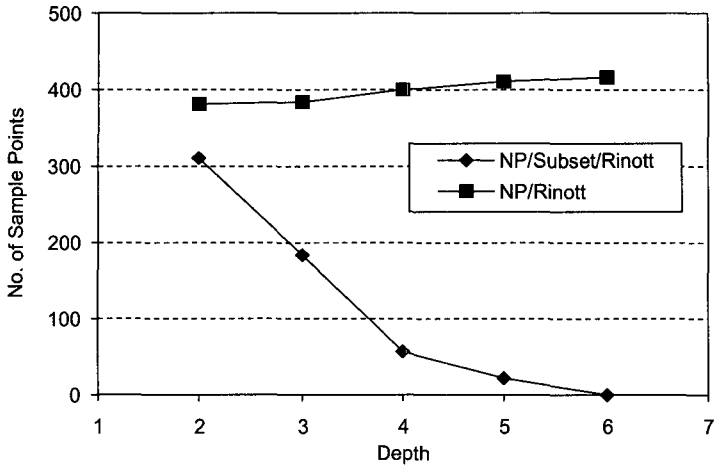

Figure 3. Computation effort in the Surrounding Region for NP/subset/Rinott and NP/Rinott

A comparison between two graphs in Figure 2 shows that the NP/Subset/Rinott Algorithm requires fewer sample points in every depth. In particular, if the P^* is low, the total number of sample points for the NP/Subset/Rinott Algorithm is less than half of that used by the NP/Rinott Algorithm; however, the relative difference between these algorithms is decreased by increasing P^*. I get the similar results from the NP/Subset/DD Algorithm. In conclusion, at least for this problem the NP/Subset/Rinott Algorithm is less computationally expensive than the NP/Rinott Algorithm and these benefits are higher when P^* is set to a low value.

The main benefit of subset selection is the improvement of the computational efficiency by eliminating inferior systems in the first stage. This to be particularly effective as the depth increases such that the surrounding region becomes larger, thus usually increasing the variance, which in turn dictates more computational effort. However, if the search is identifies a very good region, a thorough search of the surrounding region may become wasted effort and it would be beneficial to filter this region out early. Figure 3 shows the results for $P^* = 0.90$. As the depth increases, the NP/Subset/Rinott Algorithm filters out the surrounding region more and more frequently, resulting in lower average effort in the region. On the other hand, the NP/Rinott Algorithm uses more effort in the surrounding region, which is reasonable due to its high variance. Thus, the NP/Subset/Rinott Algorithm can realize substantial benefits over NP/Subset. I also similar results from NP/Subset/DD.

5.2 Numerical Evaluation of Two-Stage Sampling with Nelson-Matejcik

Since common random numbers are used in the NP/NM Algorithm, less sampling should be required. However, this algorithm will use the same amount of computational effort in each region; whereas, our numerical results from Section 5.1 indicate that substantial benefits could be obtained by using a variable sampling effort. Thus, since there are competing benefits to the two approaches, it is not clear which algorithm will perform better, the NP/NM Algorithm, the NP/Subset/DD Algorithm, or the NP/Subset/Rinott Algorithm; therefore it is necessary to evaluate this numerically.

For numerical evaluation, Monte Carlo problem and queuing problem which are mentioned in Section 5 are considered.

Table 2. Increase of Sample Points for Different P^* and Algorithms

P^* / Algorithm	Monte Carlo Problem		Queuing Problem	
	$P^*=0.55$	$P^*=0.95$	$P^*=0.55$	$P^*=0.95$
NP/NM	14000	29785	6783	153865
NP/Subset/Rinott	59585	249261	27623	201568
NP/Subset/DD	76638	302855	27360	200858

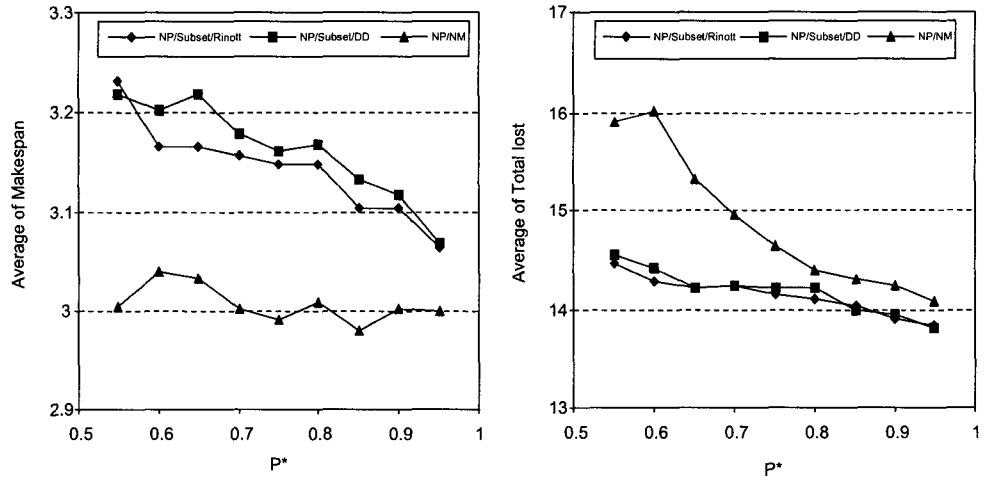

Figure 4. Average Performance of Final Solution for the Monte Carlo Problem (Left) and the Queuing problem (Right)

The performance of the three algorithms is compared along two dimensions:

speed as measured by the total number of simulation runs, and quality as measured by the average performance of the final solution obtained. Table 2 shows the total number of sample points of three algorithms for both problems. For both problems, notice that the NP/NM Algorithm requires substantially fewer sample points than the NP/Subset/Rinott and NP/Subset/DD. With respect to solution quality, Figure 4 shows the performance results for both problems. For the Monte Carlo problem, the NP/NM Algorithm performs better; however, in the queuing problem, the NP/Subset/Rinott and NP/Subset/DD Algorithms show better performance. Moreover, there was little difference between NP/Subset/Rinott and NP/Subset/DD Algorithms. Thus, which algorithm performs better depends on the problem structure, but these results also indicate that the NP/NM Algorithm is faster and thus better if the simulation budget is very limited.

5.3 Best Probability of Correct Selection having computing budget limit

One of the key parameters that must be carefully chosen for the two-stage NP method is the probability of correct selection (P^*) in each iteration. If computation time is not an issue, it can be set using some equation of Ólafsson [19] to set it according to the desired probability of terminating correctly. However, with limited computing budget it is of interest to empirically determine its best value.

This evaluation does that for both the Monte Carlo problem and queuing problem. Thus for this section, the algorithm is terminated only after a fixed number of simulation evaluations have been conducted. For the Monte Carlo problem, the simulation was run for six different sets of simulation estimates: 20,000, 30,000, 40,000, 50,000, 65,000, or unlimited for the NP/NM Algorithm nine different sets of simulation estimates: 75,000, 100,000, 125,000, 150,000, 175,000, 200,000, 225,000, 250,000, or unlimited for the NP/Subset/Rinott Algorithm, and 10 different sets of simulation estimates: 100,000, 125,000, 150,000, 175,000, 200,000, 225,000, 250,000, 275,000, 300,000, or unlimited for the NP/Subset/DD Algorithm. For the queuing problem, the simulation was run for 6 different sets of simulation estimates: 50,000, 75,000, 100,000, 125,000, 150,000, or unlimited for the NP/NM algorithm, nine different sets of simulation estimates: 75,000, 100,000, 125,000, 150,000, 175,000, 200,000, 225,000, 250,000, or unlimited for the NP/Subset/Rinott algorithm, and 10 different sets of simulation estimates: 100,000, 125,000, 150,000, 175,000, 200,000, 225,000, 250,000, 275,000, 300,000,

or unlimited for the NP/Subset/DD Algorithm. Figure 5 shows the result of the NP/NM Algorithm. For the range of $P^* \in [0.55, 0.70]$ there is no significant difference of average performance and sample points even if the simulation estimates are increased. Similar results are obtained for the NP/Subset/Rinott Algorithm and NP/Subset/DD Algorithm, as illustrated by Figure 6 and Figure 7. Figure 6 shows that it is possible to improve the performance found by increasing the P^* value to the range of $P^* \in [0.7, 0.8]$, but with limited computation budgets the performance degenerates very quickly for high P^* values. Furthermore, relatively low P^* values are recommended because the amount of computational effort increases exponentially with P^*.

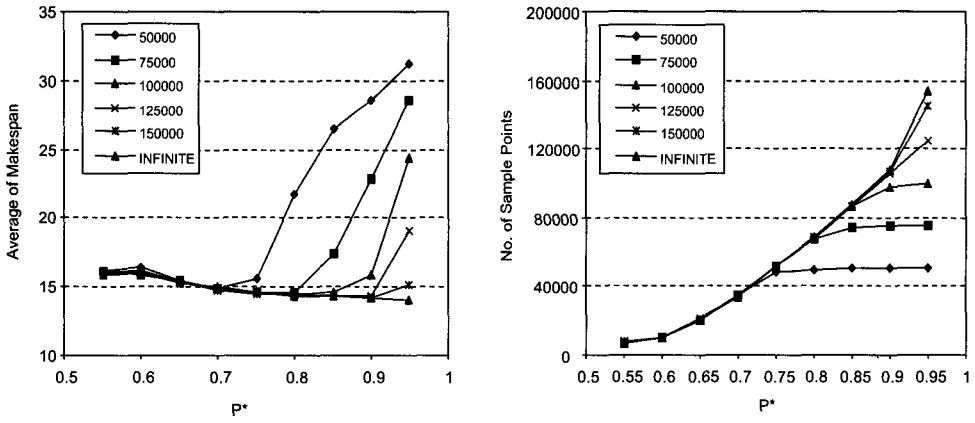

Figure 5. Performance for the Queuing Problem using NP/NM Algorithm

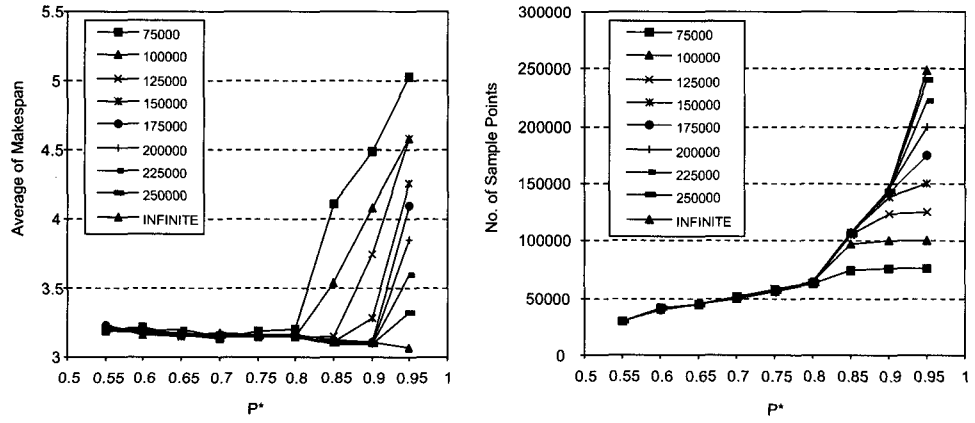

Figure 6. Performance for the Monte Carlo problem using NP/Subset/Rinott Algorithm

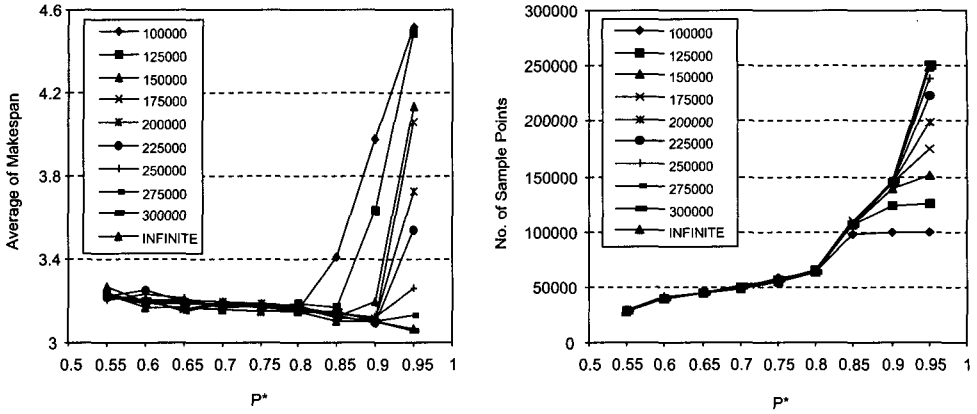

Figure 7. Performance for the Monte Carlo problem using NP/Subset/DD Algorithm

6. Conclusion

I have proposed new methodologies for simulation-based optimization that builds on the previously proposed nested partitions method of Shi and Ólafsson [22]. These new combined methods, namely Rinott's with subset procedure, *DD*'s with subset procedure, and Nelson Matejcik's procedure take advantage of the statistical selection procedure to guarantee that the search makes progress in every iteration with minimum probability.

In conclusion, the NP/Subset/Rinott Algorithm is less computationally expensive than the NP/Rinott Algorithm and these benefits are higher when P^* is set to a low value. And this result is also supported by watching surrounding region. I get the same results between NP/Subset/DD and NP/DD. When restricting computation time, that is budgeting computation time, the performance degenerates very quickly for high P^* values. Also, relatively low P^* values are advisable because the amount of computational effort increases exponentially with respect to P^*. Also I get empirical results that which algorithm is the best depends on the problem to solve. The NP/NM Algorithm is good for Monte Carlo problems. For the queuing problem, the NP/Subset/Rinott Algorithm and the NP/Subset/DD Algorithm give better results. And in both problems, the NP/Subset/Rinott Algorithm and the NP/Subset/DD Algorithm have little difference in the results. As expected the NP/NM Algorithm needs relatively less computational effort which is good for optimization with budget limits.

References

[1] Bechhofer, R. E., "A single-sample multiple decision procedure for ranking means of normal populations with known variances," *Annals of Mathematical Statistics* 25 (1954), 16-39.

[2] Bechhofer, R. E., J. Kiefer, and M. Soble, *Sequential Identification and Ranking Procedures*, The University of Chicago Press, Chicago, Illinois, 1968.

[3] Berger, J. O., "Bayesian approach to ranking and selection of related means with alternatives to analysis-of-variance methodology," *Journal of the American Statistical Association* 83, 402 (1988), 364-373.

[4] Chen, C.-H., H.-C. Chen, and L. Dai, "A gradient approach for smartly allocating computing budget for discrete event simulation," in *Proceedings of the Winter Simulation Conference* (1996), 398-405.

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

[6] Chick, S. E. and K. Inoue, "A Sequential allocation procedures that reduce risk for multiple comparisons," in *Proceedings of the Winter Simulation Conference* (1998), 669-676.

[7] Chick, S. E. and K. Inoue, "New two-stage and sequential procedures for selecting the best simulated systems," Technical report, University of Michigan, Ann Arbor, Dept. of Industrial & Operations Engineering, 1999.

[8] Clark, G. M. and W. N. Yang, "A Bonferroni selection procedure when using common random numbers with unknown variances," in *Proceedings of the Winter Simulation Conference* (1986), 372-375.

[9] Dudewicz, E. Z. and V. S. Taneja, "Multivariate ranking and selection without reduction to a univariate problem," in *Proceedings of the Winter Simulation Conference* (1978), 207-210.

[10] Dudewicz, E. J. and S. R. Dalal, "Allocation of Observations in Ranking and Selection with Unequal Variances," *Sankhya* 37B (1975), 28-78.

[11] Dudewicz, E. J. and N. A. Zaino, "Allowance for correlation in setting simulation run length via ranking and selection procedures," Technical Report. Department of Statistics, Stanford University, 1976.

[12] Dunnett, C. W., "Multivariate normal probability integrals with product correlation structure," *Applied Statistics* 38 (1989), 564-579. Correction: 42, 709.

[13] Gupta, S. S. and K. J. Miescke, "Bayesian look ahead one-stage sampling allocations for selecting the best population," *Journal of Statistical Planning and Inference* 54 (1996), 229–244.

[14] Hochberg, Y. and A. C. Tamhane, *Multipl Comparison Procedures*, NewYork: John Wiley, 1987.

[15] Inoue, K. and S. E. Chick, and C.-H. Chen, "An Empirical Evaluation of Several Methods to Select the Best System," *ACM Transitions on Modeling and Computer Simulation* 9 (1999), 381-407.

[16] Kim, J., "Optimization under uncertainty with application to data clustering," Ph.D. Dissertation, Iowa State University, 2002.

[17] Koeing, L. W. and A. M. Law, "A procedure for selecting a subset of size m containing the 1 best of l independent normal populations, with applications to simulation," *Communications in Statistics* B14, 3 (1985), 719-734.

[18] Nelson, B. L. and F. J. Matejcik, "Using common random numbers for indifference-zone selection and multiple comparisons in simulation," *Management Science* 41 (1995), 1935-1945.

[19] Ólafsson, S., "Iterative ranking-and-selection for large-scale optimization," in Proceedings of the Winter Simulation Conference (1999), 479-485.

[20] Paulson, E., "A sequential procedure for selecting the population with the largest mean from k normal populations," *Annals of Mathematical Statistics* 35 (1964), 174-180.

[21] Rinott, Y., "On two-stage selection procedures and related probability-inequalities," *Communications in Statistics* A1 (1978), 799-811.

[22] Shi, L. and S. Ólafsson, "Nested Partitions Method for Global Optimization," *Operations Research* 48 (2000), 390-407.

[23] Sullivan, D. W. and J. R. Wilson, "Restricted subset selection procedures for simulation," *Operations Research* 37 (1989), 52-71.