Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

USING PARALLEL AND DISTRIBUTED COMPUTING TO INCREASE THE CAPABILITY OF SELECTION PROCEDURES

E. Jack Chen

BASF Corporation
333 Mount Hope Avenue
Rockaway, New Jersey 07866, U.S.A.

ABSTRACT

We present a framework for deploying selection procedures in a parallel and distributed environment to identify the stochastic systems with optimal expected response when the number of alternatives is large. The goal is to speed up the process of identifying a good design with a specified probability. We present a sequential selection procedure and discuss the validity of dividing the entire set of alternative designs into several groups that can be processed in a parallel and distributed fashion. The surviving designs from each group are then processed subsequently. An experimental evaluation demonstrates the validity and efficiency of the procedure.

1 INTRODUCTION

Parallel and distributed simulation (PADS) studies how a network of several interconnected models work together to support decision making by distributing the execution of a discrete event simulation (DES) program over multiple computers. *Parallel* DES programs are executed on multiprocessor computing platforms containing multiple central processing units that interact frequently. *Distributed* DES programs are executed on loosely coupled systems that may be geographically distributed and require longer interaction times. However, with new computing paradigms such as clusters of workstations and grid computing, the distinction has become less clear. In both cases the execution of a single simulation model, likely composed of several simulation programs, is distributed over multiple processors (computers) and can be executed concurrently. Hence, one can reduce the execution time by up to a factor equal to the number of processors that are used. Distributing the execution across multiple computers and utilizing the resources of many computer systems is also beneficial in allowing the execution of larger simulations where the capacity of one computer may not be enough to carry out the simulation. A more detailed discussion of distributed and parallel simulation can be found in Fujimoto (2000).

The objective of many simulation studies is to evaluate alternative system designs or control policies of a complex system, i.e., to find a system design that is the best, or near the best, with respect to some measure or measures of system performance that can not be obtained analytically. Suppose that there are $k \geq 2$ alternative system designs that we want to compare, where design i has unknown mean μ_i and unknown variance σ_i^2, where the variances may be different for each design. We would like to compare these alternatives and to control the probability that the selected design satisfies the specified requirements. This class of problems is known as ranking and selection (R&S) and has been studied extensively in both simulation and statistics literature. It is much easier to approximately rank relative order and select a design that is within the *indifference zone* (see Section 2.1) than to precisely estimate their performance measures.

Let CS denote the event of "correct selection." In a stochastic simulation, a CS can never be guaranteed with certainty. The probability of CS, denoted by P(CS), depends on sample sizes and becomes higher as sample sizes become larger. To obtain a pre-specified precision of the estimate for a design decision, a large number of samples are often required for each design alternative. Extensive literature exist in R&S to determine the required sample sizes. For a general overview of R&S, see Law and Kelton (2000), and Swisher et al. (2003). Various schemes have been proposed to enhance the effectiveness of R&S simulation experiments. On the analysis side, statistical efficiency of R&S can be improved by taking into account the difference of sample means and using common random numbers (CRN), see Chen and Kelton (2005). On the execution side, algorithms can be developed so that executions are carried out sequentially in a parallel and distributed environment. The objective of this article is to introduce a platform for combining the statistical efficiency of simulation optimization techniques with the effectiveness of parallel and distributed execution

algorithms. We propose to divide all design alternatives into several groups and process them concurrently. Nelson et al. (2001) also point out two contexts that processing R&S in several groups might be useful: 1) exploratory studies in which not all system designs of interest are initially known or available; 2) Heuristic procedures that work with groups of designs at each iteration.

A *logical process* (LP) is a distinct flow of control, containing a combination of computation and operation. The simulation of each design can be treated as a LP and selection involves simulating a collection of LPs. Since system designs are independent of each other, the simulation of each system, i.e, LP, can be performed independently in a parallel and distributed fashion. The main goal is to compute the results of the simulation as quickly as possible to improve the effectiveness of the simulation tool. Thus, our immediate concern is capability rather than run-time performance. Chen and Kelton (2005) show that if a better alternative is found early in the process, it can be used to eliminate inferior designs at an early stage during the simulation process. When all alternatives are divided into several non-overlapping groups, if the best sample mean of all alternatives at any given moment is used to eliminate only those inferior designs within the same group, the overall efficiency may suffer. However, we will be able to process R&S of several groups in parallel, i.e., the entire R&S can be performed by a set of concurrently executing processes. Thus, the duration of run time will be decreased. Furthermore, it is possible to compute the best sample mean of all alternative designs from all groups at any iteration and use the best sample mean to eliminate inferior designs in all groups. Even though the application of parallel simulation technology has been limited, there has been some work done in this area, for example, Luo et al. (2000) deploy OCBA (Optimal Computation Budget Allocation) to distribute simulation replications over the web for R&S problems.

The rest of this article is organized as follows: In Section 2, we provide the background of selection via all pairwise comparisons. In Section 3, we present the basis of performing selection in a parallel and distribution fashion. In Sections 4, we show our experimental results. In Section 5, we give concluding remarks.

2 BACKGROUND

In this section, we introduce the necessary notation and background:

- X_{ij}: the independent and normally distributed observations from the j^{th} replication or batch of the i^{th} design,
- r: the intermediate number of replications or batches at a particular iteration,
- N_i: the total number of replications or batches for design i,
- n_i: the intermediate number of replications or batches for design i,
- μ_i: the expected performance measure for design i, i.e., $\mu_i = E(X_{ij})$,
- $\bar{X}_i(n_i)$: the sample mean performance measure for design i with n_i samples, i.e., $\sum_{j=1}^{n_i} X_{ij}/n_i$,
- $\bar{X}_i$: the sample mean performance measure for design i shorthand for $\bar{X}_i(n_i)$,
- σ_i^2: the variance of the observed performance measure of design i from one replication or batch, i.e., $\sigma_i^2 = \mathrm{Var}(X_{ij})$,
- $S_i^2(n_i)$: the sample variance of design i with n_i replications or batches, i.e., $S_i^2(n_i) = \sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)^2/(n_i - 1)$.

2.1 Indifference zone

Simulation enables the comparison of various design alternatives before implementing any of the required physical changes. Let μ_{i_l} be the l^{th} smallest of the μ_i's, so that $\mu_{i_1} \leq \mu_{i_2} \leq \ldots \leq \mu_{i_k}$. Our goal is to select a design with the smallest expected response μ_{i_1}. However, in practice, if the difference between μ_{i_1} and μ_{i_2} is very small, there may be a negligible difference in mistakenly selecting design i_2, whose expected response is μ_{i_2}. The smallest "practically significant" difference d^* (a positive real number) between the best and a satisfactory design is called the *indifference zone* in statistics literature, and it represents the smallest difference that we care about. Therefore, we want a procedure that avoids making a large number of replications or batches to resolve differences less than d^*. We are indifferent to the selection of either "good" design.

The indifference-zone approach selects a design i such that $\mu_i - \mu_{i_1} < d^*$. Some literature refers to this event as the probability of good selection (P(GS)) and use P(CS) to indicate the probability of selecting design i_1. In this article, we do not distinguish the difference between the two and use P(CS) to indicate the event in which we select a good design. If the goal is to select a design with the largest expected response, one can multiple the performance measures with -1.

2.2 Null Hypothesis Tests

The conventional statistic for determining the significance of a difference of means is by the null hypothesis test. When testing the null hypothesis $\mathrm{H}_0 : \mu_{i_l} \leq \mu_{i_1}$, the test statistic that will be used to decide whether or not to reject the null

hypothesis is

$$T_{i_l} = \frac{\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1})}{S_{\bar{X}_{i_l}-\bar{X}_{i_1}}},$$

where $S^2_{\bar{X}_{i_l}-\bar{X}_{i_1}} = S^2_{i_l}(N_{i_l})/N_{i_l} + S^2_{i_1}(N_{i_1})/N_{i_1}$. Let f_{i_l} be

$$\frac{(S^2_{i_l}(N_{i_l})/N_{i_l} + S^2_{i_1}(N_{i_1})/N_{i_1})^2}{(S^2_{i_l}(N_{i_l})/N_{i_l})^2/(N_{i_l}-1) + (S^2_{i_1}(N_{i_1})/N_{i_1})^2/(N_{i_1}-1)} \quad (1)$$

and let $t_{1-\alpha,f}$ denote the $1-\alpha$ quantile of the t distribution with f degrees of freedom (df). We reject the null hypothesis only if $T_{i_l} > t_{1-\alpha,f_{i_l}}$, or similarly

$$\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1}) > t_{1-\alpha,f_{i_l}} S_{\bar{X}_{i_l}-\bar{X}_{i_1}} = w_{i_l},$$

where w_{i_l} is the one-tailed $1-\alpha$ confidence interval (c.i.) half width. By the property of c.i. half-width, w_{i_l} ensures $\mathrm{P}[\mu_{i_l}-\mu_{i_1} \geq \bar{X}_{i_l}(N_{i_l})-\bar{X}_{i_1}(N_{i_1})-w_{i_l}] \geq 1-\alpha$. Moreover, for us not to reject the null hypothesis with $1-\alpha$ confidence that $\mu_{i_l} > \mu_{i_1}$, the lower endpoint of the one-tailed $1-\alpha$ c.i. must be positive, i.e., $\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1}) - w_{i_l} > 0$. For details on the duality of c.i. and hypothesis tests see Rice (1995).

By symmetry of the normal distribution,

$$\mathrm{P}[\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1}) + w_{i_l} \geq \mu_{i_l} - \mu_{i_1}] \geq 1-\alpha. \quad (2)$$

To achieve

$$\mathrm{P}[\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1}) > 0] \geq 1-\alpha,$$

the sample sizes N_i should be large enough so that $\mu_{i_l}-\mu_{i_1} > w_{i_l}$. Note that the half-width w_{i_l} depends on sample sizes, sample variances, and confidence level. To achieve the pre-specified precision, the half-width only needs to be smaller than the difference between the true means or the indifference amount, see Chen (2004). For example, if the number of designs under consideration is $k = 2$ and $\mu_{i_1} + d^* < \mu_{i_2}$, then sample sizes that achieve $d^* > w_{i_2}$ will guarantee that $\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) > \bar{X}_{i_1}(N_{i_1})] \geq P^*$. We know that the sample sizes determined by Rinott's procedure guarantee

$$\mathrm{P(CS)} = \mathrm{P}[\bar{X}_{i_2}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1}) > 0] \geq P^*,$$

and under the least favorable configuration, i.e., $\mu_{i_1} + d^* = \mu_{i_2}$,

$$\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) - \bar{X}_{i_1}(N_{i_1}) \geq (\mu_{i_2}-\mu_{i_1}) - d^* = 0] \geq P^*$$

and

$$\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) - \bar{X}_{i_1}(N_{i_1}) \geq (\mu_{i_2}-\mu_{i_1}) - w_{i_2} \geq 0] \geq P^*.$$

Furthermore, we can achieve the pre-specified precision by ensuring

$$\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) - \bar{X}_{i_1}(N_{i_1}) \geq (\mu_{i_2}-\mu_{i_1}) - d_{i_2}] \geq P^*,$$

where $d_{i_2} = \max(d^*, \mu_{i_2}-\mu_{i_1})$. If $\mu_{i_2}-\mu_{i_1} \geq d^*$, then $\xi = \mu_{i_2}-\mu_{i_1}-d_{i_2} = 0$, and

$$\mathrm{P}[\bar{X}_{i_2}(N_{i_l}) - \bar{X}_{i_1}(N_{i_l}) \geq 0] \geq P^*.$$

If $\mu_{i_2}-\mu_{i_1} < d^*$, then $\xi = \mu_{i_2}-\mu_{i_1}-d_{i_2} < 0$, and

$$\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) - \bar{X}_{i_1}(N_{i_1}) \geq \xi] \geq P^*.$$

Therefore, $\mathrm{P}[\bar{X}_{i_2}(N_{i_2}) - \bar{X}_{i_1}(N_{i_1}) > 0]$ may be less than P^*. However, if we select design i_2, it will be considered a correct selection by definition.

2.3 Power of Hypothesis Tests

Calculations of power are an important part of planning experiments in determining correct sample sizes. The power of tests is the probability of rejecting the null hypothesis when it is false, Rice (1995). Let $\Delta_{i_l} = \mu_{i_l}$ - μ_{i_1}, the power of the null hypothesis test is

$$\mathrm{P}\left[\frac{(\bar{X}_{i_l}(N_{i_l}) - \bar{X}_{i_1}(N_{i_1})) - \Delta_{i_l}}{S_{\bar{X}_{i_l}-\bar{X}_{i_1}}} > t_{1-\alpha,f_{i_l}} - \frac{\Delta_{i_l}}{S_{\bar{X}_{i_l}-\bar{X}_{i_1}}}\right] = F(\frac{\Delta_{i_l}}{S_{\bar{X}_{i_l}-\bar{X}_{i_1}}} - t_{1-\alpha,f_{i_l}}),$$

where $F(x)$ is the cumulative distribution function of the t distribution with f_{i_l} df. The probability of having a correct observed order can be obtained by setting $\alpha = 0.5$ since $t_{0.5,f_{i_l}} = 0$. If the power of the test is P_t for each pairwise comparison with the unknown best, then

$$\begin{aligned}\mathrm{P}[\bar{X}_{i_1}(N_{i_1}) < \bar{X}_{i_l}(N_{i_l})] &= F(\frac{\Delta_{i_l}}{S_{\bar{X}_{i_l}-\bar{X}_{i_1}}}) \\ &= P_t = F(t_{P_t,f_{i_l}}).\end{aligned}$$

The power of the test depends on sample sizes, sample variances, and the difference between the true means. The allocated sample sizes should achieve

$$w_{i_l} = t_{P_t,f_{i_l}} S_{\bar{X}_{i_l}-\bar{X}_{i_1}} \leq \Delta_{i_l}.$$

Furthermore, if

$$S_{i_l}^2(N_{i_l})/N_{i_l} = S_{i_1}^2(N_{i_1})/N_{i_1} = (\frac{\Delta_{i_l}}{t_{P_t, f_{i_l}}})^2/2, \qquad (3)$$

then the power of the test is P_t. Note that if we set $P_t = 0.5$, then the precision of $\mathrm{P}[\bar{X}_{i_1}(N_{i_1}) < \bar{X}_{i_l}(N_{i_l})] = 1-\alpha$ can be achieved when

$$S_{i_l}^2(N_{i_l})/N_{i_l} = S_{i_1}^2(N_{i_1})/N_{i_1} = (\frac{\Delta_{i_l}}{t_{1-\alpha, f_{i_l}}})^2/2.$$

This is consistent with the fact that if $\mathrm{P}[\bar{X}_2 - \bar{X}_1 < a] = P_t$ and the one-tailed P_t c.i. half-width is w, then $\mathrm{P}[\bar{X}_2 - \bar{X}_1 < a - w] = 0.5$. Here $P_t > 0.5$, and a is a real number. See Chen (2004) for more detail.

It can be shown that the optimal sample sizes that achieve $\mathrm{P}[\bar{X}_{i_1}(N_{i_1}) < \bar{X}_i(N_{i_l})] = 1-\alpha$ is $N_{i_1} = (z_{1-\alpha}/\Delta_{i_l})^2(\sigma_{i_1} + \sigma_{i_l})\sigma_{i_1}$ and $N_{i_l} = (z_{1-\alpha}/\Delta_{i_l})^2(\sigma_{i_1} + \sigma_{i_l})\sigma_{i_l}$. However, it is not clear what the optimal sample sizes will be when there are $k-1$ ($k > 2$) pairwise comparisons between design i_1 and designs $i_l \neq i_1$. Nevertheless, Glynn and Juneja (2004) have developed a framework for determining the optimal sample sizes based on the large deviations theory.

2.4 A Sequentialized R&S Procedure

Most selection procedures require the input data to be i.i.d. (independent and identically distributed) normal. Many performance measures of interest are taken over some average of a sample path or a batch of samples. Thus, many applications tend to have a normally distributed simulation output. If the non-normality of the samples is a concern, users can use batch means (see Law and Kelton 2000) to obtain samples that are essentially i.i.d. normal.

Effective reduction of computation efforts while obtaining a good decision is crucial. Chen and Kelton (2005) and Chen (2005) propose to sequentialize selection procedures to eliminate the drawback of two-stage procedures and to improve its efficiency. Rinott (1978) procedure and its variants are derived based on P(CS)=$\mathrm{P}[\bar{X}_{i_1} < \bar{X}_{i_l}$, for $l = 2, 3, \ldots, k] \geq P^*$ and $\mathrm{P}[\bar{X}_{i_1} < \bar{X}_{i_l}] \geq (P^*)^{1/(k-1)}$, for $l = 2, 3, \ldots, k$. To further improve the efficiency of sequentialized selection procedure, Chen and Kelton (2005) incorporate all pairwise comparisons at each iteration. Let $P = 1-(1-P^*)/(k-1)$ and let the set I contain the competing designs. Inferior design j such that $\mathrm{P}[\bar{X}_j > \bar{X}_i] \geq P$ of some design $i \in I$ will be excluded from further simulation at each iteration.

For completeness, the related theorem and proposition are listed here.

Theorem 1 (Chen 2004) *For k competing designs whose performance measure X_{ij} are independent and normally distributed with means $\mu_1, \mu_2, \ldots, \mu_k$ and unknown variances that need to be estimated by sample variances $S_1^2(r), S_2^2(r), \ldots, S_k^2(r)$, where r is the current sample size, P(CS) will be at least P^* when the sample size for design i is*

$$N_i = \max(r, \lceil (h_t S_i(r)/d_i)^2 \rceil), \ \textit{for } i = 1, 2, \ldots, k,$$

the critical value $h_t = \sqrt{2}t_{P,r-1}$, $P = 1-(1-p^*)/(k-1)$, and $d_i = \max(d^*, \mu_i - \mu_{i_1})$.

Let $\bar{X}_b = \min_{i \in I} \bar{X}_i$ at each iteration. Since the true means μ_i are unknown, d_i is conservatively estimated by $\hat{d}_i = \max(d^*, \bar{X}_i - U(\bar{X}_b))$, where $U(\bar{X}_b)$ is the upper one-tailed P^* confidence limit of μ_b, i.e., $\mathrm{P}[\mu_b \leq U(\bar{X}_b)] \geq P^*$. The value of $t_{P,r-1}$ can be approximated easily, see Hastings (1955).

Proposition 1 (Chen and Kelton 2005) *Let the set I contain k competing designs whose performance measure X_{ij} are independent and normally distributed with unknown means and unknown variances that need to be estimated by sample means $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_k$, and sample variances $S_1^2(n_1), S_2^2(n_2), \ldots, S_k^2(n_k)$. If $k-1$ designs are removed (eliminated) sequentially from I with each eliminated design j satisfies the equation that $\bar{X}_j(r) > \bar{X}_i(r) + w_{ij}$, $i, j \in I$, where r is the current sample size for each design, $w_{ij} = t_{P,r-1}\sqrt{S_i^2(r)/r + S_j^2(r)/r}$ and $P = 1-(1-p^*)/(k-1)$, then $P[i_1 \in I] \geq P^*$.*

The Sequential Selection Procedure via All Pairwise Comparisons (SAPC):

1. Initialize the set I to include all k designs. Let $N_{i,l}$ be the sample size allocated for design i and $\bar{X}_{i,l}$ be the sample mean of design i at the l^{th} iteration. Simulate n_0 replications or batches for each design $i \in I$. Set the iteration number $l = 0$, and $r = N_{1,l} = N_{2,l} = \cdots = N_{k,l} = n_0$, Set $P = 1-(1-P^*)/(k-1)$ and specify the value of the indifference amount d^*.
2. Perform all pairwise comparisons and delete inferior design j from I; i.e., $\bar{X}_j > \bar{X}_i + w_{ij}$, $i, j \in I$. Note that w_{ij} is the one-tailed P c.i. half-width.
3. If $w_{ij} < d^*$ and $\bar{X}_i > \bar{X}_j$, remove design i from I.
4. If there is more than one element (or the predetermined number of best designs) in I, go to step 8.
5. Compute the critical value $h_t = \sqrt{2}t_{P,r-1}$, where $t_{P,d}$ denotes the P quantile of the t distribution with d degrees of freedom.
6. Let $\bar{X}_{b,l} = \min_{i \in I} \bar{X}_{i,l}$. For all $i \in I$, compute $\hat{d}_{i,l} = \max(d^*, \bar{X}_{i,l} - U(\bar{X}_{b,l}))$, where $U(\bar{X}_{b,l})$ is the upper one-tailed P^* confidence limit of μ_b at

the l^{th} iteration, and compute

$$\delta_{i,l+1} = \lceil ((h_t S_i(r)/\hat{d}_{i,l})^2 - r)^+ \rceil.$$

7. Set $l = l + 1$. If $\delta_{i,l} = 0$, set $\delta_{i,l} = 1$. Set the incremental sample size at the l^{th} iteration $\delta_l = \min_{i \in I} \delta_{i,l}$. For $\forall i \in I$, simulate additional δ_l samples, set $r = r + \delta_l$. Go to step 2.
8. Return the values b and $\bar{X}_b(N_b)$, where $\bar{X}_b(N_b) = \min \bar{X}_i(N_i)$, $1 \le i \le k$ and i was not eliminated by all pairwise comparisons.

Instead of removing design i from further simulation when $\delta_{i,l} = 0$ we set $\delta_{i,l} = 1$. Theoretically, the incremental sample size for design i at iteration l $\delta_{i,l} = 0$ when the procedure has concluded that $\mathrm{P}[\bar{X}_{i_1} < \bar{X}_i] \ge P$ or $\mathrm{P}[\bar{X}_i \le \bar{X}_{i_1} + d^*] \ge P$. However, since sample means are used to estimate the required sample sizes, the sample sizes may not be large enough. On the other hand, if the sample sizes are in fact large enough, then design i will be eliminated by all pairwise comparison, namely between designs i_1 and i. Recall that, if $\mu_{i_l} - \mu_{i_1} > d^*$, the sample sizes that achieve $\mathrm{P}[\bar{X}_{i_1} < \bar{X}_{i_l}] \ge P$ also guarantees $\mathrm{P}[\bar{X}_{i_l} > \bar{X}_{i_1} + w_{i_l}] \ge 0.5$, where $w_{i_l i_1}$ is the one-tailed P c.i. half-width.

In the sequential selection procedure, all the alternatives $1 \le i \le k$ are initially included in the set I for R&S. If all $k-1$ designs were eliminated from I through the two-sample-t test, then $\mathrm{P}[i_1 \in I] \ge P^*$. On the other hand, if some designs were eliminated from I because $w_{ij} < d^*$, then the procedure can only guarantee P(CS) $\ge P^*$. The basic idea is that the procedure sequentially removes $k-1$ designs from I. If $\mu_{i_2} - \mu_{i_1} \ge d^*$, then the probability of wrongly removing design i_1 is no more than $1 - P$ each time a design is removed. By the *Bonferroni* inequality $\mathrm{P}[i_1 \in I] \ge 1 - \sum_{i=1}^{k-1}(1-P) = P^*$. We use the equation $S_i^2(r) = (\sum_j^r X_{ij}^2/r - \bar{X}_i^2)r/(r-1)$ to compute the variance estimator so that we are only required to store the triple $(r, \sum_{j=1}^r X_{ij}, \sum_j^r X_{ij}^2)$, instead of the entire sequences $(X_{i1}, X_{i2}, \ldots, X_{ir})$.

In general, we can improve the efficiency of selection procedures with a *pre-selection*. The pre-selection approach is a screening device that attempts to select a (random-size) subset of the k alternative designs that contains the best one. Inferior designs will be excluded from further consideration, reducing the overall simulation effort. In this procedure, design i having the total required sample size $N_i = n_0$ can be viewed as being excluded from further consideration. Thus, the procedure has an intrinsic subset pre-selection built-in. Our tests indicate this sequential procedure is significantly faster than traditional two-stage procedures.

It is known that indifference-zone selection procedures also guarantee that the coverage of multiple comparisons with the best c.i.'s and multiple comparisons with a control, design i_1 being the control, c.i.'s with probability at least P^*. These properties are still applicable for SAPC, however, the c.i. half-width will be $\max(d_i, d_j)$ instead of the indifference amount d^*, see Chen and Kelton (2003).

3 METHODOLOGIES

In this section, we present the basis of selection in a parallel and distributed fashion. PADS attempts to decrease simulation analysis time by distributing the simulation workload among multiple computers (processors). A simulation program operates on a model's state variables by executing a time-ordered sequence of simulation events. Each event may change the state of the simulated system and schedule one or more future events. In most discrete-event simulation, the order in which events are executed is stored in the event list and is determined by a next-event time advance mechanism for the simulation clock (Law and Kelton 2000). Events are executed in nondecreasing time-stamp order so that the simulation clock always advances. A conventional PADS decomposes a simulation model into communicating LPs to perform different events. The PADS procedure maps each LP to a processor and uses interprocessor communication to allow LP on different processors to communicate with each other.

3.1 Inference From Indifference-Zone Selection

By design, indifference-zone selection procedures do not guarantee the order of competitive designs when the difference between them is less than d^*. On the other hand, indifference-zone selection procedures do guarantee the order of competitive designs when the difference between them is more than d^*. More specifically, if $\mu_b - \mu_{i_1} \ge d^*$, then indifference-zone selection procedures guarantee $\mathrm{P}[\bar{X}_{i_1} < \bar{X}_b] \ge (P^*)^{1/(k-1)}$. If $\mu_b > \mu_{i_1}$, the one-tailed $1-\alpha$ c.i. half-width w_{bi_1} guarantees that $\mathrm{P}[\bar{X}_b + w_{bi_1} \ge \bar{X}_{i_1}] = \mathrm{P}[\bar{X}_b - \bar{X}_{i_1} + w_{bi_1} \ge \mu_{i_1} - \mu_{i_1}] \ge \mathrm{P}[\bar{X}_b - \bar{X}_{i_1} + w_{bi_1} \ge \mu_b - \mu_{i_1}] \ge 1 - \alpha$. Moreover, since the sample sizes determined by the selection procedure SAPC ensures P(CS) $\ge P^*$, it ensures the one-tailed P c.i. half-width $w_{bi_1} \le \max(d^*, \mu_b - \mu_{i_1})$. If $\mu_b - \mu_{i_1} \ge d^*$, then $\mu_b - \mu_{i_1} - w_{bi_1} \ge 0$, so $\mathrm{P}[\bar{X}_b + d^* \ge \bar{X}_{i_1}] \ge \mathrm{P}[\bar{X}_b \ge \bar{X}_{i_1}] \ge \mathrm{P}[\bar{X}_b \ge \bar{X}_{i_1} + \mu_b - \mu_{i_1} - w_{bi_1}] \ge P$. If $\mu_b - \mu_{i_1} < d^*$, then $w_{bi_1} < d^*$, $\mathrm{P}[\bar{X}_b + d^* \ge \bar{X}_{i_1}] \ge \mathrm{P}[\bar{X}_b - \bar{X}_{i_1} \ge \mu_b - \mu_{i_1} - d^*] \ge \mathrm{P}[\bar{X}_b - \bar{X}_{i_1} \ge \mu_b - \mu_{i_1} - w_{bi_1}] \ge P$. We summarize the result as:

Theorem 2 *For k competing designs whose X_{ij} are independent and normally distributed with unknown means and unknown variances that need to be estimated by sample means $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_k$ and sample variances $S_1^2(r), S_2^2(r), \ldots, S_k^2(r)$, where r is the current sample size. The sample sizes determined by the indifference-zone selection procedure SAPC also guarantees $P[\bar{X}_{i_1} \le \bar{X}_b + d^*] \ge P$.*

3.2 Parallel and Distributed Selection

The application of parallel and distributed simulation has been limited. It is generally difficult to implement simulation in a parallel and distributed environment because of the sequential nature of most simulations. However, simulation-based ranking and selection is well suited for parallel and distributed simulation because the behavior of each system can be simulated independently. For example, a network of computers can be used to perform R&S problems, several computers can be assigned to simulate the performance of one or several systems, while a computer dedicated to perform the comparisons between systems orchestrates the overall simulation strategy.

To set up the procedure, let $G_1, G_2, \ldots, G_m$ be groups of designs such that $G_1 \cup G_2 \cup \ldots G_m = \{1, 2, \ldots, k\}$, $G_i \cap G_j = \phi$ for $i \neq j$, $k_i = |G_i| \geq 2$ for all i, where $|I|$ denotes the cardinality of the set I. When we perform selection in group e, the designs in G_e will be compared to others in the same group. Without loss of generality, assume $i_1 \in G_g$ for some $1 \leq g \leq m$. For each group e, let $\bar{X}_b^e = \min_{i \in G_e} \bar{X}_i$ denote the best sample mean in group e. In each group, designs $q \in G_e$ such that $\bar{X}_q \leq \bar{X}_b^e + d^*$ will be considered as surviving designs. If we perform the selection in each group e, then $P[i_1 \text{ surviving } G_g \text{ selection}] \geq P$, see Theorem 1. Furthermore, if $\bar{X}_i \leq \bar{X}_b^g + d^*$ for all $i \in G_g$ in selection of group g, then $P[i_1 \text{ surviving } G_g \text{ selection}] = 1$. Since the procedure has allocated substantial sample sizes to those surviving designs, the additional samples in subsequent selections will most likely be small.

The surviving designs are then grouped into a single group or multiple groups depending on the number of surviving designs. Note that the probability that design i_1 is among the set of surviving designs is at least P. To simplify our discussion, we group the surviving designs into a single group. Let k' denote the number of surviving designs and let the set I contain the k' surviving designs. We then select from all surviving designs in I. The subsequent selection is essentially the same as described in Section 2.4, except that the initial sample sizes of each design are not equal. Note that for each design $i \in I$, we have the triple $(n_i, \sum_{j=1}^{n_i} X_{ij}, \sum_j^{n_i} X_{ij}^2)$. We perform all pairwise comparisons among these k' designs to eliminate inferior designs, i.e., design j having $\bar{X}_j > \bar{X}_i + w_{ij}$ for some $i \in I$ will be excluded from further simulation. Because the sample sizes n_i and n_j are likely to be different, we use (1) to compute the df f of the critical constant $t_{P,f}$. In the case that $n_i = n_j$, we set df to $n_i - 1$. Furthermore, if $w_{ij} < d^*$ and $\bar{X}_i > \bar{X}_j$ for $i, j \in I$, remove i from I. Let $\hat{d}_i = \max(d^*, \bar{X}_i - U(\bar{X}_b))$ for $i \in I$, where $\bar{X}_b = \min_{i \in I} \bar{X}_i$. Compute $\delta_{i,l} = \lceil (h_t S_i(n_i)/\hat{d}_i)^2 \rceil - n_i$. Since the sample sizes n_i for $i \in k'$ may be different, we use $\min_{i \in I} n_i - 1$ as the df to compute h_t. If $\delta_{i,l} = 0$, set $\delta_{i,l} = 1$. Let $\delta_l = \min_{i \in I} \delta_{i,l}$. Simulate additional δ_l replications or batches for each design $i \in I$ at iteration l. If $\mu_{i_2} - \mu_{i_1} > d^*$, the probability of i_1 being eliminated by some design $i \neq i_1$ is less than $(1 - P^*)/(k - 1)$. Hence, by the Bonferroni inequality, the probability that design i_1 survives the selection is larger than $P^* = 1 - \sum_{i=1}^{k-1}(1 - P^*)/(k - 1)$.

We can increase the efficiency of deploying the selection procedure in a parallel and distributed environment by passing the best sample mean at each iteration to all groups. Let $\bar{X}_B$ be the best sample mean from all groups, i.e., $\bar{X}_B = \min_{1 \leq e \leq m} \bar{X}_b^e$, and pass the triple $(n_B, \sum_{j=1}^{n_B} X_{Bj}, \sum_{j=1}^{n_B} X_{Bj}^2)$ to all groups. Since $\bar{X}_B \leq \bar{X}_b^e$ for $1 \leq e \leq m$, the overall efficiency of selection may be improved. Note that $\bar{X}_b^e$ from different groups may be obtained at different iteration with different sample sizes.

Another benefit of distributed simulation is the ability to integrate several different simulators, i.e., different simulation software packages, into a single simulation environment. Since the underlying simulator used to generate the simulation results has no impact on the final selection, we can simulate alternative system designs with different simulation packages. Ryde and Taylor (2003) point out that due to the way many commercial off-the-shelf simulation packages are designed, adding inter-operability could be relatively straightforward. The ability to integrate different simulators allows the simulation models to be executed and developed concurrently.

SAPC offers a natural way of performing selection in a parallel and distributed fashion, providing further gains in simulation efficiency. Once the number of alternative designs and number of groups have been determined, several control computers, denoted Ω, can be used to orchestrate the execution of simulation. These Ω computers will initiate the execution of simulation and invoke a set of remote computers, for example through remote method invocation, to generate r samples of one or several system designs. That is, each system is evaluated through distributed runs of discrete simulation models. The simulation results can be written to a shared file server or ftp (file transfer protocol) to the computers Ω that are performing the R&S. Based on the analysis, computers Ω send the required additional sample size for each system design to the corresponding computer. The communication between these computers will repeat iteratively until simulation is complete. Deploying selection procedures in parallel and distributed environment requires little communication between processors and overhead, making it practical and attractive.

3.3 The Framework

The use of networks of workstations interconnected through LAN/WANs (Local Area Network/Wide Area Network) has evolved into an effective and popular platform for parallel and distributed computing. The advantages of these network computing environments include 1) ready availability,

2) low cost, and 3) incremental scalability. Furthermore, network computing environments retain their ability to serve as a general-purpose computing platform and to run commercially available software products, see Carothers et al. (1997).

One difficulty associated with PADS is time management when ensuring that the execution of the distributed simulation, i.e., LP, is properly synchronized. Communication must be sent between processors when corresponding parts of the model interact logically. As a result, issues concerning the sub model's ability to proceed at its own pace arise because generally PADS LP can schedule further events not only for itself but also for other LPs. Therefore, events cannot be executed in a straightforward time-stamp manner. As pointed out earlier, for the R&S procedure to perform correctly, it only needs the triple $(n_i, \sum_{j=1}^{n_i} X_{ij}, \sum_j^{n_i} X_{ij}^2)$ from each design. Thus, time management does not complicate processing R&S in parallel. In this case, each LP is viewed as an independent and autonomous discrete event simulator. This means that each LP maintains its local state information corresponding to the entities it is simulating and a list of events that have been scheduled for this LP. Furthermore, each LP only schedules additional events for itself but not for other LPs. Hence, it is straightforward to implement R&S in a parallel and distributed fashion.

For a generic framework of implementing distributed simulation, see Taylor et al. (1999). They discuss the use of a generic runtime infrastructure for distributed simulation, which is an execution environment capable of supporting a broad range of simulation types. The World Wide Web or the Internet is a loose coupling of thousands of networks and millions of computers around the globe. The Internet has become one of the most important information sources and communication platforms in industry. An inherent characteristic of the Internet is its distributed nature, hence, it provides an excellent basis for distributed simulation.

4 EMPIRICAL EXPERIMENTS

In this section, we list the empirical results of SAPC and running SAPC under a parallel and distributed environment, denoted NPDS. Furthermore, NPDS$_s$ denotes performing selection separately within each group, i.e., the best sample mean from other groups is not used to eliminate inferior designs in the current group. Instead of using stochastic systems to simulate examples, which offer less control over the factors that affect the performance of a procedure, we use various normally distributed random variables to represent the system performance measures.

In the experiment, there are 10 alternative designs under consideration, and each $X_{ij} \sim N(\mu_i, (\sqrt{10})^2)$, where $N(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. See Table 2 for the values of μ_i. We divide these designs into two non-overlapping groups: group 1 contains designs 1 through 5, group 2 contains designs 6 though 10. The indifference amount d^* is set to 1.0 in all cases. We set the initial replication $n_0 = 10$ and the minimal probability of CS $P^* = 0.90$ and $P^* = 0.95$. Furthermore, 10,000 independent experiments are performed to estimate the actual P(CS) by $\hat{P}$(CS): the proportion of the 10,000 experiments in which we obtained the correct selection: design 1.

Table 1: $\hat{P}$(CS) and Sample Sizes

	$P^* = 0.90$		$P^* = 0.95$	
Procedure	$\hat{P}$(CS)	$\overline{T}$	$\hat{P}$(CS)	$\overline{T}$
SAPC	0.9567	535	0.9801	746
NPDS	0.9638	653	0.9852	889
NPDS$_s$	0.9613	866	0.9827	1204

Table 2: Detailed Sample Sizes of NPDS

Design	μ	Sample	Within	Sample	Select
1	0	108	0.9944	116	0.9638
2	1	89	0.4306	90	0.0074
3	1	89	0.4231	89	0.0078
4	1	89	0.4260	89	0.0068
5	2	54	0.0094	54	0.0
6	1	55	0.4634	63	0.0141
7	2	41	0.0119	41	0.0000
8	2	41	0.0131	41	0.0000
9	2	40	0.0123	40	0.0001
10	3	25	0.0002	25	0.0

Table 1 lists the experimental results. The SAPC, NPDS, and NPDS$_s$ rows list the results of each procedure. The $\hat{P}$(CS) column lists the proportion of correct selection. The $\overline{T}$ column lists the average of the total simulation replications ($\overline{T} = \sum_{R=1}^{10000} \sum_{i=1}^{k} N_{R,i}/10000$, $N_{R,i}$ is the total number of replications or batches for design i in the R^{th} independent run) used in each procedure. The observed P(CS)'s are greater than the specified nominal levels of 0.90 and 0.95. The sample sizes allocated by NPDS are slightly greater than those allocated by SAPC since the best sample mean may come from a group that has completed the simulation run and is no longer improving its precision. However, NPDS can perform selection in parallel and may result in a shorter run time, especially when the best sample means in each group is about the same.

Table 2 lists the results when the best sample mean is available to all groups when $P^* = 0.90$. The μ column lists the true mean of design i. The Within column lists the proportion that $\bar{X}_i \leq \bar{X}_B + d^*$, where $\bar{X}_B = \min_{1 \leq i \leq k} \bar{X}_i$ is the best sample mean of all groups. The Select column lists the proportion that the particular design has the best sample mean. The first Sample column lists the resulting sample sizes from the subdivided groups. The second Sample column lists the final sample sizes. The procedure allocates less samples to inferior designs whose sample means are far

Table 3: Detailed Sample Sizes of $NPDS_s$

Design	μ	Sample	Within	Sample	Select
1	0	101	0.9968	109	0.9613
2	1	87	0.4182	87	0.0074
3	1	87	0.4199	87	0.0080
4	1	87	0.4147	87	0.0066
5	2	62	0.0073	62	0.0001
6	1	101	0.9960	107	0.0164
7	2	87	0.4149	87	0.0001
8	2	87	0.4180	87	0.0
9	2	86	0.4062	86	0.0001
10	3	62	0.0076	62	0.0

in excess of the best sample mean. The allocated sample size of design 6 (designs 7,8,9) in group 2 is less than those of designs 2, 3, 4 (design 5) in group 1 even though their true means are the same. This is because the number of alternatives in group 2 under consideration is reduced to one much earlier than in group 1.

Table 3 lists the results when the best sample mean is available only to the group containing this particular design when $P^* = 0.90$. The Within column under NPDS without sharing lists the proportion that particular design has sample mean within the best sample mean in the same group (instead of the best sample mean of all groups) plus the indifference amount. All other fields are as defined in Table 2. The detailed results for $P^* = 0.95$ are similar. The allocated sample sizes for designs in the second group, i.e., designs 6 through 10, are significantly increased when the best sample mean of group 1 is not used to eliminate inferior designs at current group. When the selection in each group is performed separately, the resulting sample sizes and Within from each group are similar because the means within each group have similar configuration.

5 CONCLUSIONS

Parallel and distributed simulation reduces execution time for time-consuming applications, such as ranking and selection of stochastic systems. We have presented a framework for deploying selection procedures in a parallel and distributed environment. All the alternative designs are subdivided into several groups and the entire selection is performed by a set of concurrently executing processes. Thus, we may be able to shorten the run time. The procedure incorporates all pairwise comparisons to eliminate inferior designs at each iteration, which may reduce the overall computational effort. The proposed procedure takes into account the difference of sample means when determining the sample sizes and is suitable even when the number of designs under consideration is large. Furthermore, the proposed procedure is derived based on the Bonferroni inequality, so one can use CRN with the proposed procedure directly without any further assumptions.

While ordinal optimization (Ho 1992) is developed to evaluate thousands of alternative designs in its design of experiment, traditional R&S procedures only have the capability to evaluate a small number (less than 20) of alternative designs. When we take into account the different of samples means in determining the required sample sizes, we have increased the capability of R&S procedures to evaluate more designs. We can further increase the capability of R&S procedures by deploying them in a parallel and distributed environment. Since we have provided ways for the R&S statistics to deal with a much larger number of alternative designs, the proposed R&S procedure may be used to complement other existing techniques for optimization and search problems as well.

We have shown that the proposed procedure is versatile and easy to apply. The procedure preserves the simple structure of indifference-zone selection while increasing efficiency in both statistical and execution aspects, i.e., taking into account sample means and executed in a PADS environment. The simplicity of these methods should make them attractive to simulation practitioners and software developers.

ACKNOWLEDGMENTS

The author thanks Pamela Ting for her useful comments.

REFERENCES

Carothers, C. D., B. Topol, R. M. Fujimoto, J. T. Stasko, V. Sunderam. 1997. Visualizing Parallel Simulations in Network Computing Environments: A Case Study. *Proceedings of the 1997 Winter Simulation Conference*, ed. S.A. Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson, 110-117. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Chen, E. J. 2004. Using Ordinal Optimization Approach to Improve Efficency of Selection Procedures. *Journal of Discrete Event Dynamic Systems* 14(2): 153-170.

Chen, E. J. 2005. Some Insights of Indifference-Zone Selection via All-Pairwise Comparisons. *Proceedings of the 2005 Summer Computer Simulation Conference*, ed. A. G. Bruzzone, E. Williams, 118-124. San Diego, California: The Society for Modeling and Simulation International.

Chen, E. J., and W. D. Kelton. 2003. Inferences From Indifference-Zone Selection Procedures. *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 456-464. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Chen, E. J., and W. D. Kelton. 2005. Sequential Selection Procedures: Using Sample Means to Improve

Efficiency. *European Journal of Operational Research* 166(1): 133-153.

Fujimoto, R. M. 2000. *Parallel and Distributed Simulation Systems*. Wiley Interscience.

Glynn, P. W. and S. Juneja. 2004. A Large Deviations Perspective on Ordinal Optimization. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 577-585. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hastings, C. Jr. 1955. Approximations for Digital Computers. Princeton Univ. Press, Princeton, New Jersey.

Ho, Y. C., R. S. Sreenivas, and P. Vakili. 1992. Ordinal Optimization of DEDS. *Journal of Discrete Event Dynamic Systems* 2: 61–68.

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

Luo, Y-C, C. H. Chen, E. Yücesan and I. Lee. 2000. Distributed Web-Based Simulation Optimization. *Proceedings of the 2000 Winter Simulation Conference*, ed. J.A. Joines, R. Barton, P. Fishwick, and K. Kang, 1785–1793. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Nelson, B. L, J. Swann, D. Goldsman, and W. Song. 2001. Simple Procedures for Selecting the Best Simulated System when the Number of Alternative is Large. *Operations Research* 49: 950-963.

Rice, J. A., 1995. *Mathematical Statistics and Data Analysis.* Second ed. Belmont, California: Duxbury Press.

Rinott, Y., 1978. On Two-stage Selection Procedures and Related Probability Inequalities. *Communications in Statistics* A7: 799-811.

Ryde, M. D., and S. J. E. Taylor. 2003. Issues Using COTS Simulation Software Packages for the Interoperation of Models. *Proceedings of the 2003 Winter Simulation Conference*, ed. S.A. Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson, 772-777. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Swisher, J. R., S. H. Jacobson, and E. Yücesan. 2003. Discrete-Event Simulation Optimization using Ranking, Selection, and Multiple Comparison Procedures: A Survey. *ACM Transactions on Modeling and Computer Simulation* 13(2): 134-154.

Taylor, S. J. E., J. Saville, and R. Sudra. 1999. Developing interest management techniques in distributed interactive simulation using Java. *Proceedings of the 1999 Winter Simulation Conference*, 518-523. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHY

E. JACK CHEN is a Senior Staff Specialist with BASF Corporation. He received a Ph.D. from the University of Cincinnati. His research interests are in the area of computer simulation. His email address is <chenej@basf.com>.