USING RANKING AND SELECTION TO "CLEAN UP" AFTER SIMULATION OPTIMIZATION

JUSTIN BOESEL

The MITRE Corporation, 1820 Dolley Madison Boulevard, McLean, Virginia 22102, boesel@mitre.org

BARRY L. NELSON

Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208-3119, nelsonb@northwestern.edu

SEONG-HEE KIM

School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, skim@isye.gatech.edu

In this paper we address the problem of finding the simulated system with the best (maximum or minimum) expected performance when the number of systems is large and initial samples from each system have already been taken. This problem may be encountered when a heuristic search procedure—perhaps one originally designed for use in a deterministic environment—has been applied in a simulation-optimization context. Because of stochastic variation, the system with the best sample mean at the end of the search procedure may not coincide with the true best system encountered during the search. This paper develops statistical procedures that return the best system encountered by the search (or one near the best) with a prespecified probability. We approach this problem using combinations of statistical subset selection and indifference-zone ranking procedures. The subset-selection procedures, which use only the data already collected, screen out the obviously inferior systems, while the indifference-zone procedures, which require additional simulation effort, distinguish the best from the less obviously inferior systems.

Received August 1999; revision received June 2002; accepted September 2002.

Subject classifications: Simulation, statistical analysis: selecting the best system. Simulation, efficiency: large-scale screening. Programming/stochastic: terminal inference.

Area of review: Simulation.

1. INTRODUCTION

This paper presents novel ranking-and-selection procedures that identify the system with the best expected performance when presented with a large number of systems, each of which has been simulated, but not necessarily for an equal number of observations. This situation is likely to be encountered at the end of a simulation-optimization run, where a heuristic search procedure may have uncovered very good systems, but cannot guarantee which system is the true best of those visited, due to sampling variability.

Interest in the topic of simulation optimization has been revived in the past several years. For recent overviews, see Andradóttir (1998), Fu (2002), Fu et al. (2000), and Banks et al. (2001). There are two reasons behind this renewed interest:

- Commercial add-on products that employ heuristic optimization techniques in conjunction with simulation models written in an existing simulation package are readily available.
- Increased computer processing speeds make the approaches used in the add-on products feasible by allowing faster evaluations of systems via simulation.

Typically, these add-on products, such as SimRunner® (PROMODEL Corporation) and OptQuest (OptTek Systems, Inc.), employ a combination of heuristic optimization methods (genetic algorithms, tabu search, etc.) originally designed for use in a deterministic setting. Rather than evaluating each alternative system with an objective function, each alternative is evaluated by a simulation model written in an existing simulation package (Glover et al. 1996, Benson 1997). In the past, the sample mean (based on several simulation replications) was often used in the optimization algorithm as though it were the output of a deterministic objective function; more recent versions employ some error control to keep the search from being badly misled by stochastic variation.

At the end of a simulation-optimization run, there is typically a database with a large number of different systems visited by the search, each with output data from one or more replications. The system with the best sample mean is returned as the best. There is no practical way to tell if this system is the best in the entire search space. In a stochastic setting, heuristic search algorithms usually do not guarantee convergence to a globally optimal solution, while provably convergent algorithms are only guaranteed to work as simulation effort goes to infinity. Suppose, however, that the search algorithm does actually *visit* the true best alternative in the solution space. Even with this supposition, there is no assurance that the algorithm will correctly *identify* the best system.

A key feature of simulation optimization that makes it a difficult problem is the need to address the *search versus*

selection trade-off. Given a limited computing budget, how should that budget be allocated between searching over the feasible space for (potentially) better solutions, and determining which of the solutions that have been examined are actually good? In deterministic optimization problems the selection issue does not exist because the solutions are evaluated without noise; in simulation comparison problems to which ranking and selection procedures are typically applied, the focus is entirely on selection because all solutions will be evaluated.

Consider trying to allocate the next unit of simulation effort—say a replication—effectively in a simulation-optimization search. To do so requires at least partial knowledge of the following:

- How much improvement is possible, or how likely is the search to find better solutions, relative to the solutions that have already been examined? More simply, is it worth continuing to search?
- Are the "good" solutions clustered together in the solution space, or are they isolated from each other? Will following the response surface in improving directions tend to lead to the best solutions, or should the search be kept diverse?
- How much variability is there in the performance estimates for solutions that have already been visited and those that have not yet been discovered? In other words, can we recognize good solutions (and perhaps good search directions) easily, or do we need to expend significant computational effort to do so?

Without knowing a great deal about the problem structure, these questions are difficult, if not impossible, to answer. See Yakowitz et al. (2000) for an illustration of how they can be addressed when there is structural information.

The present paper does not solve the search versus selection problem. Instead, we take a practical perspective: In a simulation-optimization problem that employs an effective search heuristic, we assume the search will be successful in uncovering some very good solutions (and perhaps a large number of inferior ones). The user would like to spend as much of the budget as possible on search—to maximize the chance of encountering the optimal solution—but would also like to be confident that the selected solution is the best or near-best of all solutions that the search actually did encounter. Further, the user is able to quantify what "near-best" means in units of the performance measure (dollars, minutes, etc.).

In this paper we present ranking-and-selection procedures that take over when the search ends. Therefore, we are faced with the following situation:

1. The number of solutions we have to compare (all those encountered by the search) is large, and they may not have been simulated equally. Thus, we need to be able to handle unequal sample sizes.

2. There are some (apparently) very good solutions. Thus, we need to be able to exploit their performance estimates to discard (in a statistically precise way) inferior solutions without generating much or any additional data.

3. We do not insist on finding the unique best of all the solutions visited by the search, but we do want one that is close enough. Thus, we need to be able to incorporate the user's measure of "close enough."

4. We want the comfort of a statistical guarantee, but do not want to spend any more simulation effort than necessary to get it.

It is perhaps worth asking if the premise in Item 2 above is reasonable. In other words, when we use a heuristic optimization technique, is it safe to assume that it will uncover better and better solutions without having estimates precise enough to determine the true best solution at every step? We believe that the answer is yes, and our position is supported by research in the area of *ordinal optimization* (see, for instance, Dai 1996). Ordinal optimization demonstrates that less simulation effort is required to approximately order solutions than is required to estimate their actual performance values. In particular, it is much easier to identify solutions that are highly ranked than it is to determine the unique or near-best. We believe that the *ultimate* goal in simulation optimization is to identify the best or near-best, but a search does not have to do so at each step in order to make progress.

This paper combines and extends two types of statistical ranking-and-selection (R&S) techniques. The first of these techniques, known as *screening*, or *subset selection*, requires only one set, or *stage*, of simulation replications to be performed on each system. In a simulation-optimization setting, this means that no additional simulation effort beyond what has taken place in the search is required. Unfortunately, this type of procedure is not guaranteed to identify the *single* best system; instead, it only eliminates systems that are clearly inferior. If many systems have closely spaced sample means and large sample variances, such a procedure may be ineffective, eliminating few inferior systems and leaving the analyst with a large subset of indistinguishable systems, one of which is the best.

The other R&S technique, known as *selection*, or two-stage *indifference-zone* (*IZ*) *ranking*, guarantees to identify the single best system, but requires us to take two stages of simulation replications on each system. In a simulation-optimization setting, this means that we must perform additional simulation replications on top of those done during the search. More formally, a two-stage IZ procedure guarantees the selection of the best system with probability at least $1-\alpha$ whenever the best is at least a user-specified amount, δ, better than the others (Bechhofer et al. 1995). Furthermore, these procedures often guarantee returning a system within δ of the best if the best system is not δ better than the next best (Nelson and Matejcik 1995). This user-specified quantity, δ, defines the indifference zone, and it represents the smallest difference worth detecting. Regrettably, an IZ procedure can require a large number of additional replications from each system. The number of additional, or *second-stage*, replications required per system increases as the probability guarantee, $1-\alpha$, increases; the

indifference level, δ, decreases; the number of replications taken in the first stage, n_0, decreases; the sample variance of each system increases; or the number of systems being compared increases.

Although both subset-selection and IZ ranking procedures have shortcomings, the two approaches can work together. A straightforward combined procedure uses subset selection to eliminate the clearly inferior systems, then applies an IZ procedure *only* on the survivors. Because no second-stage data are collected on the inferior systems, it can deliver the single best system with less simulation effort than would be required by the IZ procedure alone. (Nelson et al. 2001). The combined procedures presented in this paper, which are outlined below, require even less simulation effort than a straightforward combined procedure.

- **Screen and Restart.** Given the search data, we perform a subset-selection procedure to screen out inferior systems. Then, rather than going directly to the second stage of the IZ procedure, we perform an entirely new two-stage IZ procedure. (For technical reasons we will discuss later, this effectively reduces the number of systems being compared, which reduces the additional effort required.) Furthermore, restarting allows us to choose an adjusted first-stage sample size n_0, which reduces the *total* number of replications required.
- **Sort and Iterative Screen.** Given the search data, we sort the systems by sample mean and take additional second-stage replications on the most promising system *before* performing any screening. These additional replications reduce the sample variance of the most promising system, making it much more effective at screening out inferior systems. One by one, the next-most-promising system faces screening by those who have gone before. Only if a system passes screening do we perform additional replications on it.

The Sort-and-Iterative-Screen procedure is a direct descendant of a Group-Screening procedure presented in Nelson et al. (2001) which extended ideas from ranking and selection to contexts in which the number of alternatives is large. A brief description of the Group-Screening procedure, and the differences between it and the Sort-and-Iterative-Screen procedure, can be found in §6 of the journal's online companion. Boesel et al. (2003) implemented ideas from Nelson et al. (2001) and the present paper in software that combines heuristic optimization, via a genetic algorithm, with statistical ranking and selection. The practical experience obtained in creating and using that software provided much of the research direction for the present paper.

Chick and Inoue (2001) have developed procedures for selecting the best simulated system based on Bayesian expected value-of-information arguments, rather than on the frequentist R&S techniques used in this paper. Their procedures, which allocate simulation effort adaptively, tend to be less conservative than those presented here and have produced good results on a number of empirical problems (see, for instance, Inoue et al. 1999, and Chen 1996). Unlike the procedures presented in this paper, however, they do not provide a probability of correct selection guarantee, nor have they been designed for very large numbers of alternatives.

The remainder of this paper is organized into three sections. Section 2 provides a more thorough overview of screening and selection procedures, and presents extensions that allow them to work when the initial number of replications varies from one system to another. Section 3 presents two distinct strategies for combining screening and selection. Section 3.1 describes the Screen-and-Restart Procedure; an algorithm to find the best first-stage sample size when restarting is also given. Section 3.2 describes the Sort-and-Iterative-Screen Procedure. An empirical evaluation and comparison of both approaches, and some earlier variants, is provided in §4. For brevity's sake, proofs of the validity of the procedures developed in the paper, as well as other supporting procedures and results, are included in the journal's online companion.

2. BASIC METHODS AND EXTENSIONS

2.1. Setting and Notation

We assume that a preliminary or *first-stage* set of simulation output data (possibly generated by a search procedure) are "dropped into our laps." Let k be the number of different systems in the data set, and let n_{0i} be the number of replications already performed on system i. Notice that we do not require equal first-stage sample sizes, since a search procedure may revisit systems or take differing numbers of replications from them. Let X_{im} be the output from replication m of system i. Systems are to be compared based on their true means, $\mu_i = E[X_{im}]$, and we assume that larger μ_i is better throughout this paper.

The first-stage sample mean of system i is

$$\overline{X}_i(1) = \frac{1}{n_{0i}} \sum_{m=1}^{n_{0i}} X_{im},$$

while $S_i^2(1)$ is the first-stage sample variance of system i; that is

$$S_i^2(1) = \frac{1}{n_{0i} - 1} \sum_{m=1}^{n_{0i}} (X_{im} - \overline{X}_i(1))^2.$$

Let $\overline{X}_{[i]}(n)$ be the nth-stage sample mean of the system whose true mean is the ith smallest (thus, $[k]$ is the index of the best system among the k that are available). To use the procedures described in this paper, we need the sample mean, sample variance, and number of replications taken from each system visited by the search. Define correct selection (CS) as the event where a procedure returns system $[k]$, the one with the best true mean. The probability of making a correct selection is denoted by PCS.

The procedures presented in this paper are derived under the assumption that the simulation output data are normally

distributed, and our empirical evaluation retains this setting. The assumption is reasonable when simulation performance estimates are averages of large numbers of more basic observations, either within a replication or "batch." Nevertheless, the issue of robustness to nonnormal output data is relevant. Both Nelson et al. (2001) and Nelson and Goldsman (2001) report empirical studies that evaluate the robustness of R&S procedures—similar to the ones in this paper—to departures from normality. Their results showed that mild departures cause no significant degradation in performance, as measured by probability of correct selection. Severe departures, however, do degrade PCS, as one would expect.

Our procedures assume that the search heuristic delivers i.i.d. samples from each system that it visits, and further that the samples across systems are also independent. These assumptions raise two issues in our setting: the first involves the use of common random numbers (CRN), and the second involves the possibility of dependence among systems visited by a simulation-optimization search procedure.

It is well known that the use of CRN to induce dependence across systems can sharpen comparisons, where in our context "sharpening" means reducing the total number of observations required to achieve the desired PCS. There are IZ procedures that exploit CRN (e.g., Nelson and Matejcik 1995), but they either become extremely conservative, or invalid, as the number of systems becomes large. We suspect that the procedures introduced in this paper will still preserve the required PCS if CRN is employed, but that is not the same as exploiting CRN to improve efficiency. Development of procedures that do exploit CRN when the number of systems is large is an open research problem.

If a simulation-optimization search takes a fixed number of replications from each solution it examines, then our assumptions of i.i.d. sampling and independence across systems will stand up. However, if the search adjusts sample sizes based on observed performance or variability, then some dependence within and across systems' output data could be introduced. However, since the search heuristic is not trying to enforce overall statistical error control, and (at best) is exercising error control in local search steps, the induced dependence (if any) should be negligible.

2.2. Screening

In many cases there will be systems visited by the search that are clearly inferior to others visited by the search. We will use a *subset-selection* procedure to screen out these clearly inferior systems. A subset-selection procedure returns a subset (whose size can be random or predetermined) that contains the best of the k systems with probability $\geqslant 1-\alpha_0$ (Bechhofer et al. 1995). Nelson et al. (2001) developed a single-stage subset-selection procedure that permits unequal and unknown variances. Our Extended Screen-to-the-Best Procedure, presented below, extends their procedure to allow the unequal sample sizes that may be the result of a search. In the following procedure, the best system is the one with the largest true mean:

Extended Screen-to-the-Best Procedure.

1. Set $1-\alpha_0$ such that $1/k < 1-\alpha_0 < 1$.
2. Given X_{im}, $i = 1, 2, \ldots, k$, $m = 1, 2, \ldots, n_{0i}$, let

$$W_{ij} = \left(\frac{t_i^2 S_i^2(1)}{n_{0i}} + \frac{t_j^2 S_j^2(1)}{n_{0j}} \right)^{1/2}, \qquad \forall\, i \neq j$$

where $t_i = t_{(1-\alpha_0)^{1/(k-1)},\, n_{0i}-1}$ and $t_{\beta,\,\nu}$ is the β quantile of the t distribution with ν degrees of freedom.

3. Set $H = \{i\colon\ 1 \leqslant i \leqslant k \text{ and } \overline{X}_i(1) \geqslant \overline{X}_j(1) - W_{ij}, \forall\, j \neq i\}$.
4. Return H as the subset of retained systems.

A complete description of a slightly more general version of this procedure, and a proof of its validity ($\Pr\{[k] \in H\} \geqslant 1-\alpha_0$ for all configurations of the means), are included in §2 of the journal's online companion. The simulation package Arena (Rockwell Software) has incorporated the screening procedure described above into its output analysis package.

Although a single-stage subset-selection procedure, such as the one presented above, requires no additional sampling effort, the number of systems included in the subset is random. If one is fortunate, the subset includes only a single system, the best. If one is unfortunate, no systems are obviously inferior, so the subset includes all k systems and the procedure has not reduced the field.

Even if it does not identify the single best system, a screening procedure can provide useful information about the quality of the search. If one is left with just a few systems in the subset, this means that the ratio of between-system variation to within-system variation (stochastic noise) is relatively high, and that the search procedure found some systems that were clearly better than the others. If, on the other hand, the subset is very large, then the search procedure found no clear winners; perhaps additional effort could be spent allowing the search procedure to seek out some better systems, or perhaps the search did not take enough replications at each system.

2.3. Selection

To choose the single best system from among those systems that are *not* obviously inferior, we will employ a two-stage IZ *ranking* procedure, which requires one additional sampling stage from the competitive systems. Two-stage IZ procedures guarantee to select the best system with probability $\geqslant 1-\alpha_1$ whenever the best is at least a user-specified amount, δ, better than the others. If there are some near-best systems within δ of the best, our procedures will return the best or one of these near-best systems. The user-specified quantity δ defines indifference zone, and it represents the smallest difference worth detecting. In a typical IZ procedure, such as Rinott's (1978) procedure, the total sample size required of system i is:

$$N_i = \max\left\{ n_0, \left\lceil \left(\frac{h' S_i(1)}{\delta} \right)^2 \right\rceil \right\}, \tag{1}$$

where $\lceil \cdot \rceil$ means to round up, $h' = h(k, 1-\alpha_1, n_0)$ is a constant determined by k, the number of systems being compared; $1-\alpha_1$ is the desired confidence level; and n_0 is the number of first-stage observations. The constant h' increases in k, and decreases in α and n_0. Rinott's original paper assumed that the initial sample sizes were equal, but we have extended Rinott's procedure to allow unequal initial sample sizes.

Extended Rinott Procedure.

1. Set $1-\alpha_1$ such that $1/k < 1-\alpha_1 < 1$, and let $n_{\min} = \min_i\{n_{0i}\}$.
2. Set $h = h(2, (1-\alpha_1)^{1/(k-1)}, n_{\min})$.
3. Given $X_{im},\ i = 1, 2, \ldots, k,\ m = 1, 2, \ldots, n_{0i}$, determine the total required sample size for system i

$$N_i = \max\left\{n_{0i}, \left\lceil \left(\frac{hS_i(1)}{\delta}\right)^2 \right\rceil\right\}.$$

4. Take $N_i - n_{0i}$ additional replications from each system i.
5. Select as best the system i with the largest overall sample mean $\overline{X}_i(2) = \sum_{m=1}^{N_i} X_{im}/N_i$.

The validity of this extension (specifically, $\Pr\{\text{select } [k]\} \geqslant 1-\alpha_1$ whenever $\mu_{[k]} - \mu_{[k-1]} \geqslant \delta$) is proved in §3 of the journal's online companion. Furthermore, as a result of Theorem 1 in Nelson and Matejcik (1995), we guarantee that if $\mu_{[k]} - \mu_{[k-1]} < \delta$, then the procedure will select the best system or one within δ of the best.

REMARK. Notice that the constant

$$h = h(2, (1-\alpha_1)^{1/(k-1)}, n_{\min})$$

used to determine the second-stage sample size is based on the smallest of the first-stage sample sizes. Further, it is not $h' = h(k, 1-\alpha_1, n_{\min})$, the standard Rinott constant for comparing k systems. We conjecture that the procedure is still valid if h is replaced by h', but have been unable to prove this (unless all the n_{0i} are equal, in which case the proof is trivial). The constant h is in fact an upper bound on h'; in §3 of the journal's online companion, we provide some numerical examples showing that h is a tight upper bound, so we feel that little is lost in using h rather than h'.

IZ ranking satisfies our overall goal of finding the best or near-best system, but it may be statistically inefficient because it assumes that the systems' true means are arrayed in the so-called "Least-Favorable Configuration" (LFC). The LFC is the configuration of system means that would give a procedure the smallest probability of returning the statement "system i is the best" when system i is indeed the true best and is at least δ better than anything else. For IZ procedures, the LFC is typically the "slippage" configuration in which the true mean of the best system is exactly δ larger than the means of all the other systems. Of course, because nature rarely (if ever) places systems in the LFC, $1-\alpha_1$ is a lower bound on the probability that the IZ procedure returns a true statement. This makes IZ procedures inherently conservative.

3. COMBINING SCREENING AND SELECTION

Both subset-selection and IZ ranking procedures have shortcomings that hamper their usefulness in a simulation-optimization setting. As mentioned above, a single-stage subset-selection procedure requires no additional simulation effort after the search has finished, but it may not eliminate many (or any) systems. On the other hand, an IZ procedure guarantees the return of a single system within δ of the best with a prespecified probability, but it may require an enormous amount of additional simulation effort to do so. As Equation (1) shows, N_i, the total number of replications required for system i, can become quite large, especially if the $S_i^2(1)$ are large and δ is small. For this reason, IZ procedures are best when k, the number of systems in contention, is small. In our environment, however, we may have hundreds or thousands of systems to consider. The simulation effort required to use an IZ procedure alone in such a setting quickly becomes prohibitive.

Fortunately, the two approaches (subset and IZ) can work together to deliver a single system that meets our requirements with less simulation effort than would be required by the IZ procedure alone (Nelson et al. 2001). As mentioned earlier, the IZ procedures assume that the means of the competing systems are arrayed in the LFC. Even after the first-stage data have been collected, the IZ procedures retain the LFC assumption, making no use of the information provided by the first-stage sample means. This strict adherence to the LFC assumption makes the IZ procedures wastefully conservative. Combining a subset-selection procedure with an IZ procedure can reduce this effort by using the first-stage sample data to "screen out" clearly inferior systems. This makes the IZ procedure more efficient by reducing the number of systems from which additional sampling is required.

In the next two subsections we describe procedures that employ various combinations of screening and selection, and discuss their application in a simulation-optimization setting. The first subsection describes a simple screen-and-restart procedure, then proposes a method that takes advantage of the variance estimates from the discarded first-stage samples to choose the size of the new first-stage samples. The second subsection describes a procedure that does not rerun any first-stage samples, but instead collects second-stage sample information from a few of the better systems, which decreases their sample variance and makes them tougher screeners. This increases the chance that they will eliminate inferior systems, reducing the total number of second-stage samples required.

3.1. Combination of Screen, Restart, and Select

Nelson et al. (2001) developed a provably valid screen-and-select procedure, which we will call the Screen-and-Continue procedure. Unfortunately, the validity guarantee for this procedure requires that the critical value h used in the IZ procedure be determined as though all k systems remain in contention, rather than just those that survive

screening. This is because the procedure uses the initial samples from the search in the IZ procedure (note that the conditional probability of selecting the best system, given it passed screening, depends upon whether or not the first-stage data are retained). Thus, h remains large, so N_i is also large. On top of this, because we must calculate h using $n_{\min}$ (the smallest of all of the initial sample sizes n_{0i} taken during the search) and because h is a decreasing function of n_{0i}, N_i will be further enlarged.

If, however, we rerun the first-stage samples of the systems that survive screening, then we can eliminate some of these problems. Let M be the number of systems that survive screening. Restarting allows us to use M, rather than the original k, in our determination of h. This could reduce h, perhaps dramatically. Furthermore, restarting gives us an opportunity to increase n_{0i}, which further reduces h. In many cases, the savings gained through these reductions in h more than offset the losses involved in rerunning the first-stage samples.

The combined procedure presented below is simple and statistically valid; it employs a subset-selection procedure to screen out inferior systems, then employs an independent IZ procedure on the survivors by taking a new first-stage sample from each.

Screen-Restart-and-Select Procedure.

1. Select the desired confidence level $1-\alpha$ such that $1/k < 1-\alpha < 1$, and the indifference level $\delta > 0$.

2. Given X_{im}, $i = 1, 2, \ldots, k$, $m = 1, 2, \ldots, n_{0i}$, run the subset procedure (described below). To obtain an overall confidence level of $1-\alpha$, we set $1-\alpha_0 = \sqrt{1-\alpha}$ for the subset procedure and $1-\alpha_1 = \sqrt{1-\alpha}$ for the selection procedure; however, any decomposition whose product is $1-\alpha$ could be used.

(a) Let

$$W_{ij} = \left(\frac{t_i^2 S_i^2(1)}{n_{0i}} + \frac{t_j^2 S_j^2(1)}{n_{0j}} \right)^{1/2} \tag{2}$$

where $t_i = t_{(1-\alpha_0)^{1/(k-1)}, n_{0i}-1}$.

(b) Set $H = \{i\colon\ 1 \leqslant i \leqslant k \text{ and } \overline{X}_i(1) \geqslant \overline{X}_j(1) - W_{ij}, \forall\, j \neq i\}$.

(c) Return H, the group of systems that survive the screen, and let $M = |H|$.

3. Take independent samples of size $n_{ri} \geqslant 2$ from each system $i \in H$ (discarding the initial first-stage sample), and calculate a new sample variance estimate, $S_i^2(r1)$. (The best setting of n_{ri} is discussed after the description of this procedure.)

4. Calculate the total required sample size from system $i \in H$, N_i, as

$$N_i = \max\left\{ n_{ri}, \left\lceil \left(\frac{hS_i(r1)}{\delta} \right)^2 \right\rceil \right\}, \tag{3}$$

where $h = h(2, (1-\alpha_1)^{1/(M-1)}, n_{\min})$ is Rinott's (1978) constant with $k = 2$, confidence level $(1-\alpha_1)^{1/(M-1)}$, and first-stage sample size $n_{\min} = \min_{i \in H}\{n_{ri}\}$.

5. Take $N_i - n_{ri}$ additional replications from each system $i \in H$.

6. Of the M surviving systems, select as best the system i with the largest overall sample mean $\overline{X}_i(2) = \sum_{m=1}^{N_i} X_{im}/N_i$, $i \in H$.

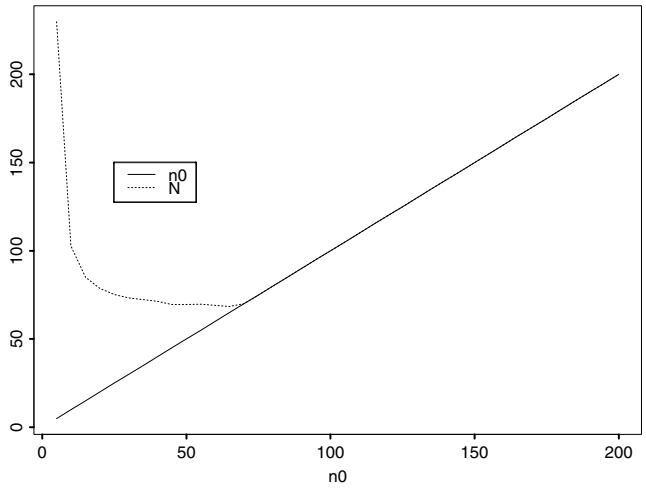
In §4 of the journal's online companion, we prove that $\Pr\{\text{select } [k]\} \geqslant 1-\alpha$ whenever $\mu_{[k]} - \mu_{[k-1]} \geqslant \delta$ under this procedure. Furthermore, as a result of Theorem 1 in Nelson and Matejcik (1995), we are guaranteed that if $\mu_{[k]} - \mu_{[k-1]} < \delta$, the probability of selecting a system *within* δ of the best is greater than or equal to $1-\alpha$.

Restart provides an opportunity to reduce a system's *total* required sample size by increasing its *initial* sample size. Recall that n_{0i} is system i's initial sample size as a result of the search, and n_{ri} is i's initial sample size in the new sample. Under our procedure, the total required sample size for system i (after discarding the first n_{0i} replications) is given in Equation (3). The critical value, h, which helps to determine the total sample size, N_i, decreases as the initial sample size increases. The impact on N_i is illustrated in Figure 1, which plots N_i as a function of n_{ri} for $k = 25$ and $1-\alpha = 0.95$ for a fixed value of $(S_i(r1)/\delta)^2 = 1$. This figure suggests increasing n_{ri}, at least up to a point, to decrease N_i. Of course, if n_{ri} is increased too much, it will exceed $\lceil (hS_i(r1)/\delta)^2 \rceil$, which defeats the purpose.

Given that we have seen the results of the first-stage sample, how might we use that information to better set n_{ri}? Suppose we take a simplistic view, by assuming that the effects of rounding are negligible and that $S_i^2(r1) = S_i^2(1)$, where $S_i^2(1)$ is the sample variance of the initial first-stage sample and $S_i^2(r1)$ is the sample variance of the restarted first-stage sample. Let $h_{(n_{ri})}$ represent h as a function of n_{ri} for fixed values of k and α. In Figure 1, notice that N_i is quite low at the point where

$$n_{ri} = \left(\frac{h_{(n_{ri})} S_i(r1)}{\delta} \right)^2. \tag{4}$$

Figure 1. Sample size N as a function of the initial sample size n_r for $(S/\delta)^2 = 1$, $k = 25$, and $1-\alpha = 0.95$.



We will assume (without proof) that N_i is minimized at or near this point. The graphical analysis of Figure 1 applies only to setting n_{ri} for a single system; in our context the decision on how to set n_{ri} will be complicated by the fact that we have many systems with unequal variances. Although initial sample sizes need not be equal for an IZ procedure to be *valid*, h is set according to the smallest n_{0i}. Clearly, then, if one system is given a small initial sample size, then the benefit of giving other systems greater initial sample sizes is diminished. For this reason, we set a single n_r for use by all systems under restart. Notice that with n_r all equal, we can use $h(M, 1-\alpha_1, n_r)$ instead of $h(2, (1-\alpha_1)^{1/(M-1)}, n_{\min})$. To find the single best n_r, we will take advantage of the fact that the line $N_i = n_r$ is convex, and our conjecture that $h_{(n_r)}$ is convex.[1] If our conjecture is correct, then $(h_{(n_r)}S_i(r1)/\delta)^2$ is also convex. Therefore, the upper envelope

$$\max\left\{n_r, \left(\frac{h_{(n_r)}S_i(r1)}{\delta}\right)^2\right\}$$

is also convex. Because this is true for every system, the sum

$$\sum_{i=1}^{M}\left\{\max\left\{n_r, \left(\frac{h_{(n_r)}S_i(r1)}{\delta}\right)^2\right\}\right\}$$

is convex as well. Thus, we can use a search procedure such as the Golden-Section method (Bazaraa et al. 1993) to find the minimum. We defer discussion of a method for selecting $[n_{\text{low}}, n_{\text{high}}]$, the minimum-containing interval needed to start the Golden-Section method, to §5 of the journal's online companion.

Because we are restarting, the validity of carefully selecting n_r is not in question. The amount of simulation effort saved, however, is less clear. Although the restart procedure has easily provable statistical properties, it is unfortunate that it discards data. If the initial sample size is large or if the screen fails to eliminate many systems, rerunning the initial samples becomes wasteful. We conjecture that the biggest gains will occur when there are a large number of widely spaced systems and initial samples are small. In a search setting, we are very likely to encounter problems of this kind because a heuristic simulation-optimization search procedure (such as a genetic algorithm or a tabu search) will typically take just a few replications on a large number of systems with widely varying performance. In the empirical study in §4 we compare the restart procedure with the Screen-and-Continue procedure from Nelson et al. (2001), which does not discard data.

REMARK. One might wonder if the decision to screen and continue, or to restart, can be deferred until after examining the first-stage data to see how many systems survive screening. Boesel (2000) shows that this is indeed possible, but only if the confidence levels for screening and selection are adjusted at the cost of some efficiency.

3.2. Sorting and Iterative Screening

The procedure in the previous section employs screening and IZ selection in two separate, distinct phases: All systems are screened using first-stage data, then new first-stage data and second-stage data are collected to perform IZ selection. Nelson et al. (2001) describe and prove the validity of a procedure that completely processes a few systems at a time, rather than processing all systems in two distinct phases. Under this procedure, second-stage data are collected on a small group of systems; a second, distinct group of systems is formed, and the members of the second group are screened against members of the first group. Additional information is collected on a system in the second group only if it survives screening. This procedure then rolls from one group to the next, screening each subsequent system against all previously surviving systems. The use of second-stage information makes each screen much tighter, eliminating more systems and potentially reducing the total simulation effort required.

More formally, when screening system i against system j, screening is based on

$$\widetilde{W}_{ij} = \left(\frac{t_i^2 S_i^2(1)}{\widetilde{N}_i} + \frac{t_j^2 S_j^2(1)}{\widetilde{N}_j}\right)^{1/2},$$

where

$$\widetilde{N}_j = \begin{cases} n_{0j}, & \text{if system } j \text{ has only received first-stage sampling} \\ N_j, & \text{if system } j \text{ has received second-stage sampling.} \end{cases}$$

For systems j from previous steps that have already received second-stage sampling, we typically have $N_j \gg n_{0j}$, which shortens $\widetilde{W}_{ij}$, providing a tighter screening procedure. Notice that $S_i^2(1)$ and $S_j^2(1)$ are based on the first-stage data only.

In Nelson et al. (2001), the authors point out that if the procedure happens to encounter a system with a good sample mean early in the process, then that system will likely receive second-stage sampling and act as a very tough screen, eliminating many inferior systems, and reducing the total simulation effort required. Because we come in at the end of a simulation-optimization search procedure, our setting differs from that envisioned in Nelson et al. (2001), and we can assume that the first-stage samples from all of our systems have been given to us. Because all of this information is available at the same time, we can *sort* the systems from best to worst based on first-stage sample means. So, rather than *hope* that the procedure happens to encounter a good system early, we can *cause* the procedure to encounter a good system early by sorting. We call this the Sort-and-Iterative-Screen procedure, and present it below.

Sort-and-Iterative-Screen Procedure.

1. Select overall confidence level $1-\alpha$ such that $1/k < 1-\alpha < 1$, and the indifference level $\delta > 0$. Let $\alpha_0 = \alpha_1 = \alpha/2$. Set $t_i = t_{(1-\alpha_0)^{1/(k-1)}, n_{0i}-1}$ and $h = h(2, (1-\alpha_1)^{1/(k-1)}, n_{\min})$, where $n_{\min} = \min_i\{n_{0i}\}$ and h is Rinott's constant.

2. Given X_{im}, $i = 1, 2, \ldots, k$, $m = 1, 2, \ldots, n_{0i}$, compute $\overline{X}_i(1)$ and $S_i^2(1)$ and set $\widetilde{N}_i = n_{0i}$ for all i.

3. Sort by sample mean, reindexing systems such that $\overline{X}_1(1) \geqslant \overline{X}_2(1) \geqslant \cdots \geqslant \overline{X}_k(1)$.

4. Let $H_0 = \varnothing$ and $J_0 = \varnothing$, where H_i is the set of systems that have *passed* screening as of step i, and J_i is the set of systems that have *failed* screening as of step i.

5. Do the following for $i = 1, 2, \ldots, k$:

Compute $\widetilde{W}_{ij}$, $\forall\, j \in J_{i-1} \cup H_{i-1}$.
If $\overline{X}_i(1) \geqslant \overline{X}_j(1) - \widetilde{W}_{ij}$, $\forall\, j \in J_{i-1}$ and
$\overline{X}_i(1) \geqslant \overline{X}_j(2) - \widetilde{W}_{ij}$, $\forall\, j \in H_{i-1}$,
then
let $H_i = H_{i-1} \cup \{i\}$, and $J_i = J_{i-1}$.
Compute the second-stage sample size for system i

$$N_i = \max\left\{n_{0i}, \left\lceil \left(\frac{hS_i(1)}{\delta}\right)^2 \right\rceil\right\}.$$

Sample $N_i - n_{0i}$ additional replications, and compute the overall sample mean $\overline{X}_i(2)$. Set $\widetilde{N}_i = N_i$, and advance i.
Else
let $J_i = J_{i-1} \cup \{i\}$, $H_i = H_{i-1}$, and advance i.

6. Select as best the system $i \in H_k$ with the largest overall sample mean $\overline{X}_i(2)$.

Nelson et al. (2001) prove that whenever $\mu_{[k]} - \mu_{[k-1]} \geqslant \delta$, their procedure has $\Pr\{\text{select } [k]\} \geqslant 1 - 2\alpha_0 - \alpha_1$, where $1 - \alpha_0$ represents the confidence level used in the screening phase, $1 - \alpha_1$ represents the confidence level used in the selection phase, and $\alpha_0 + \alpha_1 = \alpha$. Furthermore, the authors provide substantial evidence that $\Pr\{\text{select } [k]\} \geqslant 1 - \alpha_0 - \alpha_1$ whenever $\mu_{[k]} - \mu_{[k-1]} \geqslant \delta$. Additionally, they guarantee with the same probability that the procedure will select a system within δ of the best if $\mu_{[k]} - \mu_{[k-1]} < \delta$.

The procedure described above differs from that presented by Nelson et al. (2001) in that:

- The first-stage sample sizes may be unequal,
- the first-stage sample means are sorted before any screening takes place, and
- each system faces screening by *all* previously screened systems, not just those that survived screening.

In §6 of the journal's online companion, we prove that the procedure presented above has the same guarantees and properties as the procedure presented in Nelson et al. (2001). Furthermore, in §4 we show that sorting can greatly reduce the simulation effort required to return these guarantees.

4. EMPIRICAL EVALUATION

We conducted an extensive empirical evaluation to compare the procedures introduced in this paper to existing procedures and to each other. The systems are represented as various configurations of k normal distributions. We evaluated the procedures on different variations of the systems, examining factors including the number of systems, k, the configuration of the means, μ_i, and the variances, σ_i^2, for $i = 1, 2, \ldots, k$.

In the first set of experiments, the following three procedures are compared: the Screen-and-Continue Procedure, as described in Nelson et al. (2001); the Screen-Restart-and-Select Procedure (described in §3.1), with the initial sample size under restart, n_r, not adjusted; and the Screen-Restart-and-Select Procedure with the initial sample size under restart, n_r, adjusted using the Golden Section method. In the second set of experiments, the Sort-and-Iterative-Screen Procedure (described in §3.2) is compared to an Iterative-Screen Procedure *without* sorting. The only difference between the two procedures is that the second does not sort the systems by sample mean. The third set of experiments compares the Screen-Restart-and-Select Procedure *with* n_r Adjustment to the Sort-and-Iterative-Screen Procedure.

4.1. Experiment Design

In all cases, the best system was system k and its true mean was set to 1. The indifference level δ also was set to 1. To examine a difficult scenario for the screening procedures, the slippage configuration (SC) of the means was used. In the SC, the mean of the best system was set exactly one indifference level, δ, above the other systems, and all of the inferior systems had the same mean. To investigate the effectiveness of the screening procedure in removing noncompetitive systems, monotone-decreasing means (MDM) were also used. In the MDM configuration, the spaces between the means of any two adjacent systems were set at δ/τ where τ was a constant within each experiment.

To gauge the effects of sorting by first-stage sample mean, we performed experiments under two different orderings of the means. In one set of experiments we ordered the systems from best to worst, to see how the nonsorting procedure (iterative-screening without sorting) would perform under the most fortunate circumstances. In another set of experiments we ordered the systems from worst to best, to gauge the performance of the nonsorting procedure under the most unfortunate circumstances.

In some cases the variances of all systems were equal ($\sigma_i^2 = 1$), while in others they differed. For each configuration, we examined the effects of equal and unequal variances on the procedures. In the unequal-variance case, the variance of the best system was set both higher and lower than the other systems. In the SC, $\sigma_1^2 = \rho\sigma^2$, with $\rho = 0.5, 2$ where σ^2 is the common variance of the inferior systems. In the MDM configurations, experiments were run with the variance directly proportional to the mean of each system, and inversely proportional to the mean of each system. Specifically, $\sigma_i^2 = |\mu_i - \delta| + 1$ to examine the effects of increasing variance as the mean decreases, and $\sigma_i^2 = 1/(|\mu_i - \delta| + 1)$ to examine the effect of decreasing variances as the mean decreases (since $\mu_{[k]} = 1$, the smallest means may be negative, but have large absolute values).

In the Restart experiments, 1,000 macroreplications (complete repetitions of the entire experiment) were performed for each configuration. In the sorting experiments, 500 macroreplications were performed.

In all experiments, the nominal probability of correct selection (PCS) was $1-\alpha = 0.95$. If the procedure's true PCS is close to the nominal level, then the standard error of the estimated PCS, based on 1,000 macroreplications, is near $\sqrt{0.95(0.05)/1000}$, which is approximately 0.0069. The standard error of the estimated PCS based on 500 macroreplications is near $\sqrt{0.95(0.05)/500}$, which is approximately 0.0097. Since the nominal PCS $= 1-\alpha$, we want to examine how close to $1-\alpha$ we get. If the actual PCS $\gg 1-\alpha$ for all configurations of the means, then the procedure is overly conservative.

The number of systems in each experiment varied over $k = 2, 5, 10, 25, 100, 500$. The first-stage sample size was set at $n_0 = 10$. Although we prove that unequal initial sample sizes (n_{0i}) can be used with ranking-and-selection procedures, to do so, one must calculate the constant h based on the minimum of the unequal sample sizes. In other words, one *can* use unequal initial sample sizes, if presented with them, but it does not help one reduce the additional effort required. Furthermore, use of unequal initial sample sizes may make the already-conservative procedures we are comparing even more conservative. This could only serve to blur, rather than to sharpen, the differences among our procedures. For these reasons, we used equal initial sample sizes in our experiments.

4.2. Summary of Results

We will not present comprehensive results from such a large simulation study. Instead, we present details of some typical examples after summarizing the overall results. The performance measures that we estimated in each experiment include the probability of correct selection (PCS), the average number of samples per system (ANS) over all k systems, and the percentage of systems that received second-stage sampling (PSS). Notice that ANS is a measure of a procedure's overall efficiency, while PSS measures the effectiveness of the screening component in eliminating inferior systems.

In the Restart experiments, we compared three procedures: Screen-and-Continue with no restart (shortened to "No Restart"), Screen-Restart-and-Select with n_r adjustments using the Golden Section method (shortened to "Restart *with* n_r Adjustment"), and Screen-Restart-and-Select without n_r adjustments (shortened to "Restart *without* n_r Adjustment"). In all but two instances, Restart *with* n_r Adjustment was more efficient than Restart *without* n_r Adjustment, often substantially so. When Restart *without* n_r Adjustment was better, it was only slightly better than Restart *with* n_r Adjustment. This suggests that our procedure for finding a good n_r under restart is useful, but could be improved somewhat.

The No Restart procedure was often more efficient than the Restart *without* n_r Adjustment procedure when the number of systems, k, was 2, 5, or 10. The Restart *without* n_r Adjustment procedure was clearly better than the No Restart procedure when $k \geqslant 25$. The Restart *with* n_r Adjustment procedure was almost always more efficient than No Restart for $k \geqslant 5$; even when $k = 2$, Restart *with* n_r Adjustment was more efficient than No Restart in about half of the trials.

Despite the improvements gained by adjusting n_r under restart, this procedure is still conservative; while the nominal PCS was 0.95, the actual PCS was rarely less than 0.99 unless the systems were in the slippage configuration. Under the SC, the actual PCS was usually between 0.97 and 0.99. The only exception was the Restart *with* n_r Adjustment procedure for $k = 500$ in the common variance case (PCS $= 0.963$) and the case where the inferior systems had smaller variance than the best (PCS $= 0.951$).

In the Sorting experiments, we compared three procedures: the Sort-and-Iterative-Screen Procedure (shortened to "Sort and Screen"); the Iterative-Screen Procedure with no sorting (shortened to "Screen No Sort"); and Rinott's Procedure with no screening. The differences were dramatic; Sort and Screen was vastly superior to Screen No Sort when the means were encountered in an unfavorable sequence. When the means were encountered in a favorable sequence (true best system first), the efficiencies of Sort and Screen and Screen No Sort were about the same, although sometimes (for $k = 500$) Screen No Sort was somewhat more efficient than Sort and Screen. We conjecture that the reason is that Sort and Screen sorts by first-stage sample mean, and this may sometimes place poorer screeners first, while Screen No Sort always used the true best as the first screener in this scenario. Rinott's procedure was more efficient than Sort and Screen only when the number of systems was small, $k = 2, 5$; that is, when the number of systems eliminated by screening could not make up for splitting α into α_0 for screening and α_1 for selection.

Comparing Restart *with* n_r Adjustment to Sort and Screen and Screen No Sort yields interesting results. In our experiments, Restart *with* n_r Adjustment was typically better than both Sort and Screen and Screen No Sort. One could imagine that if the initial sample size, n_0, were much larger than 10 that this would not be the case, because Sort and Screen would need fewer additional replications, but Restart *with* n_r Adjustment would still be discarding information. On the other hand, if n_0 were smaller, then Restart *with* n_r Adjustment may look stronger yet.

4.3. Restart Experiments—Detailed Results

In these experiments we compared No Restart to Restart *with* n_r Adjustment and Restart *without* n_r Adjustment. Our findings can be divided into four categories:

1. SC, many systems: In this case, Restart *without* n_r Adjustment was about as efficient as No Restart, but Restart *with* n_r Adjustment was about twice as efficient.

2. SC, few systems: The effects of restart and n_r adjustment were mixed.

Table 1. No Restart vs. Restart, with and without n_r adjustment.

	Number of Systems, $k = 2$			Number of Systems, $k = 500$		
	No Restart	Restart w/	Restart w/o	No Restart	Restart w/	Restart w/o
ANS	98	91	106	649	273	642
PSS	96%	96%	97%	100%	100%	100%
PCS	0.98	0.97	0.97	0.99	0.96	0.99

Note. The systems are in the slippage configuration, and all systems have equal variance.

3. MDM, many systems: Both Restart *with* n_r Adjustment and Restart *without* n_r Adjustment dominated No Restart, and Restart *with* n_r Adjustment tended to be better than Restart *without* n_r Adjustment.

4. MDM, few systems: Restart *with* n_r Adjustment was somewhat better than No Restart, but No Restart was better than Restart *without* n_r Adjustment.

Tables 1 and 2 show some of these results. When considering these results, keep in mind that the procedure for adjusting n_r under restart takes time, as the constant h must be obtained for a number of different values of n_r to solve the optimization problem. The comparisons in Tables 1 and 2 do not account for this effort.

4.4. Sorting Experiments—Detailed Results

Tables 3 and 4 show that Screen No Sort can be disastrous if variances are high and the procedure happens upon a poorly sequenced group of systems. As Table 3 shows, sorting caused most of the inferior systems to be screened out even in the unfavorable sequence (PSS = 2%). This was doubly helpful because under this configuration the systems with inferior means also had high variance. In Table 4, the inferior systems had much smaller variance, so their elimination had less of an impact.

4.5. A Comparison of Sort-and-Iterative-Screen to Restart

In the previous sections we observed that Sort and Screen typically was better than Screen No Sort. We also observed that Restart *with* n_r Adjustment was better than No Restart and Restart *without* n_r Adjustment. To conclude, we compare Sort and Screen to Restart *with* n_r Adjustment. Some results are given in Tables 5–6.

Table 2. No Restart vs. Restart, with and without n_r adjustment.

	Number of Systems, $k = 2$			Number of Systems, $k = 500$		
	No Restart	Restart w/	Restart w/o	No Restart	Restart w/	Restart w/o
ANS	98	91	106	22	13	14
PSS	96%	96%	97%	2%	2%	2%
PCS	0.98	0.97	0.97	1.00	1.00	1.00

Note. The systems are in the MDM configuration, $\tau = 1$, and all systems have equal variance.

Table 3. The effect of screening *with* sorting relative to screening *without* sorting in the MDM configuration with $k = 500$ and $\tau = 1$.

	Favorable Sequencing		Unfavorable Sequencing	
	Screen No Sort	Sort and Screen	Screen No Sort	Sort and Screen
ANS	41	54	52,455	55
PSS	2%	2%	100%	2%
PCS	1.00	1.00	1.00	1.00

Note. In all cases, variance *increases* as sample mean decreases, that is, $\sigma_i^2 = |\mu_i - \delta| + 1$ for all i.

As one would expect, Sort and Screen, which uses second-stage sample information in screening, has a lower PSS than Restart *with* n_r Adjustment (see Table 5). This extra screening power, however, did not make up for the savings the restart procedure gained by lowering h, so Restart *with* n_r Adjustment had the lower ANS.

Of course, for the MDM configuration used in Table 5, screening out inferior systems is not such a tough job, so the extra screening power of Sort and Screen is not as critical. If we look at a similar, but more tightly spaced ($\tau = 3$ rather than $\tau = 1$) and highly variable configuration, the situation is less clear cut (see Table 6).

There are several factors at play here, and we could devise a configuration in which one or the other procedure would be better. For instance, if first-stage screening is easy, then the additional screening power of Sort and Screen is not as useful. On the other hand, if regular first-stage screening is much less effective than second-stage screening, then restarting will be a waste. Boesel (2000) shows that you can choose between Sort and Screen and Restart *with* n_r Adjustment *after* examining the first-stage data, but that you must pay a penalty for this choice by adjusting the confidence level downward. Also note that Restart *with* n_r Adjustment is a bit slower to execute than Sort and Screen, so if the two procedures have equal ANS, Sort and Screen will be faster. Of course, Restart *with* n_r

Table 4. The effect of screening *with* sorting relative to screening *without* sorting in the MDM configuration with $k = 500$ and $\tau = 1$.

	Favorable Sequencing		Unfavorable Sequencing	
	Screen No Sort	Sort and Screen	Screen No Sort	Sort and Screen
ANS	14	14	25	14
PSS	1%	1%	100%	1%
PCS	1.00	1.00	1.00	1.00

Note. In all cases, variance *decreases* as sample mean decreases, that is, $\sigma_i^2 = 1/(|\mu_i - \delta| + 1)$ for all i.

Table 5. Sort and Screen vs. Restart with n_r Adjustment.

	$k = 2$		$k = 25$		$k = 500$	
	Sort and Screen	Restart w/	Sort and Screen	Restart w/	Sort and Screen	Restart w/
ANS	95	91	66	44	19	13
PSS	92%	96%	18%	24%	1%	2%
PCS	0.97	0.97	1.00	0.99	1.00	1.00

Note. The systems are in the MDM configuration, $\tau = 1$, and all systems have equal variance, as in Table 2.

Adjustment could be improved with a better, and, way to determine the initial sample size under restart.

In a simulation-optimization setting, a heuristic search is likely to visit a large number of systems and will only expend a few simulation replications on each one. These circumstances should heavily favor the Restart *with* n_r Adjustment procedure, which performs better than Sort and Screen when the number of initial replications per system is low.

5. CONCLUSIONS

Our work is motivated by the fact that the demand for, and subsequent creation of, commercial simulation-optimization software is racing ahead of the supporting theory. In an ideal world, this software would be based on provably convergent algorithms that nevertheless provide good performance and precise statistical guarantees in finite time. However, since such algorithms do not yet exist for general classes of problems—and practitioners rarely have the time to figure out what "class" of problem they have—the commercial software is typically based on heuristics that have good empirical performance in difficult deterministic optimization problems. Such algorithms aggressively search the solution space and may uncover a number of good systems. Our goal is to provide some statistical support for the system that is ultimately selected, while reducing the additional simulation effort required beyond what has already been expended by the search. Specifically, we guarantee that the system selected is the best (or is within some user-specified amount δ from the best) of all those visited by the search. This is less than the global guarantee that we desire, but much more than provided by commercial products. And our procedures are independent of the search algorithm that was employed. See Boesel et al. (2003) for an example of how a search and selection procedure can be combined.

Table 6. Sort and Screen vs. Restart with n_r Adjustment.

	$k = 2$		$k = 25$		$k = 500$	
	Sort and Screen	Restart w/	Sort and Screen	Restart w/	Sort and Screen	Restart w/
ANS	104	120	407	254	102	128
PSS	97%	98%	66%	80%	5%	10%
PCS	1.00	1.00	1.00	1.00	1.00	1.00

Note. The systems are in the MDM configuration, $\tau = 3$, and variance increases as system mean decreases ($\sigma_i^2 = |\mu_i - \delta| + 1$ for all i).

The results of our empirical study show that the Restart Procedure *with* n_r Adjustment typically outperformed the Restart procedure *without* n_r Adjustment and the Iterative-Screening Procedures (with and without sorting). In our studies the first-stage sample size was $n_0 = 10$, but in a simulation-optimization setting, when a search procedure visits a large number of different systems, n_0 may be much smaller. This could strengthen Restart's advantage, although if n_0 is too small a poor variance estimate could mislead the sample-size optimization. Restart still has the unappealing feature of discarding the first-stage data, but using the first-stage variance information (and the Golden-Section method) to adaptively set the restarted first-stage sample size, n_r, ameliorates this drawback. Furthermore, the slowness of Restart's current method for finding a good n_r could be improved through use of approximations for h.

While Restart *with* n_r Adjustment is typically better, if one faces high-variance, closely spaced systems, and a large number of initial replications, then the Sort-and-Iterative-Screen procedure should prove superior.

ENDNOTE

1. Although we were unable to prove the convexity of $h_{(n_r)}$, we believe that it is true based on extensive numerical analysis.

ACKNOWLEDGMENTS

This research was partially supported by National Science Foundation Grant DMI-9622065, and by JGC Corporation (Japan), Symix Corporation/Pritsker Division, and Rockwell Software.

REFERENCES

Andradóttir, S. 1998. Simulation optimization. Chapter 9 in J. Banks, ed. *Handook of Simulation: Principles, Methodology, Advances, Applications, and Practice.* John Wiley and Sons, New York.

Banks, J., J. S. Carson, B. L. Nelson, D. M. Nicol. 2001. *Discrete Event System Simulation*, 3rd ed. Prentice Hall, Upper Saddle River, NJ.

Bazaraa, M. S., H. D. Sherali, C. M. Shetty. 1993. *Nonlinear Programming: Theory and Algorithms.* John Wiley and Sons, New York.

Bechhofer, R. E., T. J. Santner, D. Goldsman. 1995. *Design and Analysis for Statistical Selection, Screening and Multiple Comparisons*. John Wiley and Sons, New York.

Benson, D. 1997. Simulation modeling and optimization using ProModel. S. Andradóttir, K. J. Healy, D. H. Withers, B. L. Nelson, eds. *Proc. 1997 Winter Simulation Conf.*, 587–593.

Boesel, J. 2000. Adaptively choosing the best procedure for selecting the best system. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proc. 2000 Winter Simulation Conf.*, 537–543.

——, B. L. Nelson, N. Ishii. 2003. A framework for simulation-optimization software. *IIE Trans.* **35**(3) 221–229.

Chen, C-H. 1996. A lower bound for the correct subset-selection probability and its application to discrete-event simulations. *IEEE Trans. Automatic Control* **41**(8) 1227–1331.

Chick, S. E., K. Inoue. 2001. New two-stage and sequential procedures for selecting the best simulated system. *Oper. Res.* **49** 732–743.

Dai, L. 1996. Convergence properties of ordinal optimization comparison in simulation of discrete event dynamic systems. *J. Optim. Theory Appl.* **91** 363–388.

Fu, M. C. 2002. Optimization for simulation: Theory vs. practice. *INFORMS J. Comput.* **14**(3) 192–215.

——, S. Andradóttir, J. S. Carson, F. Glover, C. R. Harrell, Y. C. Ho, J. P. Kelly, S. M. Robinson. 2000. Integrating optimization and simulation: Research and practice. J. A. Joines, R. R. Barton, K. Kang, P. A. Fishwick, eds. *Proc. 2000 Winter Simulation Conf.*, 610–616.

Glover, F., J. P. Kelly, M. Laguna. 1996. New advances and applications of combining simulation and optimization. J. M. Charnes, D. J. Morrice, D. T. Brunner, J. J. Swain, eds. *Proc. 1996 Winter Simulation Conf.*, 144–152.

Inoue, K., S. E. Chick, C-H. Chen. 1999. An empirical evaluation of several methods to select the best system. *ACM Trans. Model. Comput. Simulation* **9** 381–407.

Nelson, B. L., D. Goldsman. 2001. Comparisons with a standard in simulation experiments. *Management Sci.* **47** 449–463.

——, F. J. Matejcik. 1995. Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Sci.* **41** 1935–1945.

——, J. Swann, D. Goldsman, W. Song. 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Oper. Res.* **49** 950–963.

Rinott, Y. 1978. On two-stage selection procedures and related probability-inequalities. *Comm. Statist.—Theory and Methods* **A7** 799–811.

Yakowitz, S., P. L'Ecuyer, F. Vazquez-Abad. 2000. Global stochastic optimization with low-dispersion point sets. *Oper. Res.* **48** 939–950.