Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

A NEW FRAMEWORK FOR PARALLEL RANKING & SELECTION USING AN ADAPTIVE STANDARD

Linda Pei
Barry L. Nelson

Department of Industrial Engineering & Management Sciences
Northwestern University
Evanston, IL 60208-3119, USA

Susan Hunter

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907, USA

ABSTRACT

When we have sufficient computational resources to treat a simulation optimization problem as a ranking & selection (R&S) problem, then it can be "solved." R&S is exhaustive search—all feasible solutions are simulated—with meaningful statistical error control. High-performance parallel computing promises to extend the R&S limit to even larger problems, but parallelizing R&S procedures in a way that maintains statistical validity while achieving substantial speed-up is difficult. In this paper we introduce an entirely new framework for R&S called Parallel Adaptive Survivor Selection (PASS) that is specifically engineered to exploit parallel computing environments for solving simulation optimization problems with a very large number of feasible solutions.

1 INTRODUCTION

In operations research, stochastic simulation is used to answer three basic types of questions about systems: *feasibility* (whether a system works), *sensitivity* (whether a system is robust to uncontrollable and unpredictable factors), and *optimization* (which systems lead to good performance). This paper addresses simulation optimization (SO), which is broadly concerned with eliminating "bad" systems and identifying "good" ones. SO problems fall under the broader category of *stochastic optimization problems*, which are difficult to solve even in the more structured contexts considered by mathematical programming. In SO, less is known or assumed about the objective function, which usually takes the form of an expected value that can only be estimated with statistical error by running simulation experiments at different feasible solutions; we refer to the feasible solutions as *systems*.

When the number of systems is finite and "small," simulating each system is a manageable and effective approach to solving SO problems. This approach recasts SO as a statistical problem of controlling error across multiple systems' performance estimates. Methods for controlling this error are called Ranking and Selection (R&S); see Kim and Nelson (2006) for a survey. One advantage of using R&S is that it eliminates "bad" systems and identifies "good" systems globally, without requiring knowledge of an underlying structure in the feasible space. R&S methods are also easy to use and usually provide a probabilistic optimality or efficiency guarantee. Thus, even for problems with underlying and exploitable spatial structure, R&S remains a go-to method whenever it is computationally feasible.

The number of systems considered "computationally feasible" in an R&S problem has increased over time, rising to 10,000 or more with recent advances in R&S methodology and the advent of parallel computing technology. In particular, obtaining the required simulation replications in parallel can achieve significant gains in computational efficiency, and can make simulation optimization problems over extremely large numbers of systems tractable. Despite the conceptual simplicity of obtaining simulation replications in parallel, designing R&S procedures for large numbers of systems that fully utilize a parallel computing platform is a nontrivial task; see Hunter and Nelson (2017) for a survey of parallel R&S. In essence,

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

comparisons between systems are fundamental to finding the "best" system, and obtaining valid system comparisons on a parallel computing platform introduces a tension between simulation and logistical efficiency; we use the term "logistical efficiency" to refer to effective coordination of computing resources. From a simulation point of view, frequently comparing system performances often reduces the total number of simulation replications required to determine the best system, which is efficient. From a logistical point of view, frequently comparing system performances often requires information synchronization across parallel processors, which is inefficient. Balancing simulation and logistical efficiencies is a crucial aspect of solving ever-larger R&S problems.

To explore these key parallel R&S issues in more detail, consider a general "master-worker" parallel computing framework, in which there are $p+1$ parallel processors. The p "worker" processors complete "jobs" assigned and coordinated by one "master" processor. As in the framework by Hunter and Nelson (2017), jobs consists of tasks such as obtaining simulation replications from one or more systems, computing summary statistics, and performing calculations that compare systems. Calculations that compare systems require *coupled operations*, which are defined by Hunter and Nelson (2017) as calculations that require the output data from two or more systems. *Fully coupled operations* require the output data from all systems still in contention, such as determining the estimated-best system among all remaining systems, and may employ all p processors. Thus, although p jobs can be processed simultaneously, in contrast to a serial setting, the need for coupled operations in R&S often becomes a bottleneck that prevents a parallel R&S procedure from running p times as fast as a serial one.

While coupled operations sometimes may be performed asynchronously, maintaining the validity of common R&S statistical guarantees often forces coupling and synchronization to go together, which implies the bottleneck. For example, updating key statistics on the master processor "on the fly" as simulation replications are completed may introduce statistical issues such as bias (see, e.g., Ni et al. 2013). Frequentist procedures usually provide a guaranteed probability of selecting a system that is the best or is within a predefined optimality gap from the best. These guarantees are called *probability of correct selection* (PCS) or *probability of good selection* (PGS) guarantees. Bayesian procedures usually either maximize the posterior PCS, or minimize some expected loss between the selected system and the best possible system. In all cases, because "best" is a relative measure, coupled comparisons that maintain statistical validity are important. The easiest way to maintain statistical validity during coupled operations is to force processors to synchronize. In a multi-processor computer architecture, coupled operations that maintain statistical validity often force the master to wait for multiple worker processors to complete their jobs, and for worker processors to wait for the master to receive and start new jobs.

Despite the need for synchronization, performing such coupled comparisons frequently improves simulation efficiency. In a single-processor computer architecture, "effort" is characterized by the total number of simulation replications needed by the R&S procedure. Employing many coupled or fully coupled operations can help good systems eliminate inferior ones quickly and facilitate continual fine-tuning of the allocation of simulation effort. Many popular R&S methods require pairwise comparisons after each round of simulation replications and are highly efficient from a simulation perspective. However, in a naïvely parallel implementation, the workers will idle while the master carries out $O(k^2)$ pairwise comparisons to determine the next set of jobs; this becomes a prohibitive feat when the number of systems k is very large. Thus, the increased simulation efficiency comes at a cost: coupling implies synchronization, synchronization implies waiting, and waiting diminishes speed-up.

In this paper, we introduce an entirely new perspective on R&S in parallel computing environments. We argue that "correct selection" is not always an appropriate objective, and it does not scale well as the number of feasible solutions k becomes very, very large. We consider a different objective, the *expected false elimination rate* (EFER), which we define in §3. Then, drawing inspiration from the "comparisons with a standard" literature, we propose Parallel Adaptive Survivor Selection (PASS) as a paradigm that adaptively "learns" a standard while controlling the EFER. Controlling the EFER and learning the standard facilitates an approach that effectively avoids synchronized coupled operations, keeping worker processors almost constantly busy, and the master processor largely idle. Thus, PASS is logistically efficient without wasting simulation replications. The PASS paradigm is unlike any other R&S approach in its objective and how it exploits parallel computing.

The paper is organized as follows: Section 2 provides the relevant background and building blocks of PASS. Section 3 introduces the generic PASS framework, and Section 4 specializes it to a particular algorithm we call bi-PASS. Numerical illustrations can be found in Section 5.

2 FOUNDATION OF PASS: COMPARISONS WITH A STANDARD

The inspiration for PASS lies in the class of R&S procedures known as "comparisons with a standard." These procedures compare all system performances with a standard, $\mu^\star$, which is either a known constant or the unknown mean of a benchmark system that must also be simulated. These procedures are interesting to us in the context of PASS because if $\mu^\star$ is a known constant, comparisons with a standard procedures require *no coupling*, which makes their parallel implementation require less synchronization. In this section, we explore background literature on comparisons with a standard procedures, which will form the foundation of PASS. We consider two broad categories of procedures that perform comparisons with a standard: (a) those that select a single best "system," which is either the standard $\mu^\star$ or an alternate system $i \in \{1,2,\ldots,k\}$, and (b) those that select a subset of systems, all of which are estimated to be better than the standard $\mu^\star$. Henceforth, we assume larger system performances are better, such that if we have k systems, k (or a system whose mean is equal to system k's) is the best, and the systems have true mean values $\mu_k \geq \mu_{k-1} \geq \cdots \geq \mu_2 \geq \mu_1$.

Among methods that select a single best system, Nelson and Goldsman (2001) and Kim (2005) provide two-stage and fully sequential procedures, respectively, that guarantee a correct selection with pre-specified probability in the presence of unknown and unequal system variances. The standard is given special status such that it will be selected even when it is tied with the best non-standard system, and a non-standard system will be selected only if it significantly beats the standard and all others. Both procedures employ an indifference-zone framework, guaranteeing that for indifference-zone parameter $\delta > 0$,

$$\Pr\{\text{select the standard } \mid \mu^\star \geq \mu_k\} \geq 1-\alpha,$$
$$\Pr\{\text{select the system } k \mid \mu_k - \mu^\star \geq \delta, \mu_k - \mu_{k-1} \geq \delta\} \geq 1-\alpha,$$

for a user-specified PCS of $1/(k+1) < 1-\alpha < 1$. The procedure by Nelson and Goldsman (2001) also provides confidence bounds on the difference between the mean of each alternative system and the standard, or each alternative system and the best alternative system, depending on the selection decision. Both procedures require assumptions. Specifically, Nelson and Goldsman (2001) require that the difference between each simulation output and its true mean is independent of the true mean, which is an assumption that normally distributed simulation output satisfies. Kim (2005) requires that the simulation output of each system is normally distributed and uses properties of Brownian motion to construct continuation regions that determine when the standard or an alternative system is eliminated.

Among methods that select a subset of systems, there are a variety of approaches. Of particular interest is Singham and Szechtman (2016), who employ a hypothesis testing approach to identify the set of "non-null" systems that are better than the standard. They assume that most of the k systems are "null" and have true means equal to the standard $\mu^\star$. The remaining systems are "non-null" and have true means of $\mu^\star + \delta$, where $\delta > 0$ is an indifference-zone parameter. Instead of guarantees on the probability of correct selection, this framework involves guarantees on the expected false discovery rate (EFDR), which is the probability that a system selected as non-null is actually null. The authors provide a threshold policy for normally distributed systems that assigns each system a p-value based on the hypothesis test, and selects systems with p-values smaller than some cut-off. This policy is based on empirical Bayes methods for large-scale inference (Efron 2012) and a well known algorithm from Benjamini and Hochberg (1995).

Other approaches to select a subset of systems include both the large-deviations and Bayesian approaches of Szechtman and Yücesan (2008) and Xie and Frazier (2013), respectively. Szechtman and Yücesan (2008) derive an asymptotically optimal sample allocation for feasibility determination. Selecting a subset of feasible systems can be seen as an instance of selecting a subset of systems with some performance measure better than a known standard. Xie and Frazier (2013) assume that the standard is a known constant and provide a Bayes-optimal fully sequential sampling policy under a limited ability to simulate, which is modeled through sampling costs or random stopping times. Their procedure uses dynamic programming to find a sample policy that maximizes the value of the accumulated payoff minus the accumulated sampling costs when sampling ends. The accumulated payoff is based on how well the subset of better systems $\{i\colon \mu_i \geq \mu^\star, i = 1,2,\ldots,k\}$ is identified.

Finally, in related literature on factor screening, Wan et al. (2010) adapt the fully sequential comparison-with-a-standard procedure of Kim (2005) to a simulation factor screening setting in which sequential bifurcation is used to detect "important" system factors. Factors are defined to be important if the effect that they have on a given performance measure is greater than some standard. The sequential bifurcation method is designed to a guarantee an EFDR, which corresponds to the expected number of factors incorrectly flagged as important.

We emphasize here two observations from the comparisons-with-a-standard literature that are important for PASS: (a) When simulated systems are compared with a known-value standard and *and not with each other*, then the comparison problems completely decouple; and (b) when the goal is to control an error rate, such as EFDR, then there is little or no statistical efficiency penalty *per system* as the number of systems k grows. Stated differently, the error rate is controlled marginally, rather than jointly. PCS, on the other hand, is a joint statement and the simulation effort per system increases with k, which is a serious issue when k is very large.

3 THE PASS PARADIGM

Recall that our goal is create a new R&S paradigm that decouples the operations within R&S procedures so that these procedures can run efficiently on a parallel computing platform and scale to very large numbers of systems. Although the comparisons-with-a-standard literature provides naturally decoupled procedures, there may exist no such standard, or the user may have no idea whether there are any systems better than the standard. That is, the standard may be "soft." Thus we propose PASS as a way to "learn the standard," by adaptively updating a standard to preserve some systems identified by the user as important, such as the top-b systems. In addition to learning the standard, we also control a different form of error than the traditional PCS or PGS. We instead control the EFER, which enables us to provide a statistical guarantee at every simulation replication n through the procedure. In what follows, we first discuss the EFER. Then, we describe how we learn the standard with PASS.

3.1 EFER of Good Systems

To formally define the EFER in the context of PASS, we require a few definitions. First, our building block statistic for PASS is a centered partial sum of the outputs from system i

$$S_i(n) = \sum_{j=1}^{n}(Y_{ij} - \mu^\star) = \sum_{j=1}^{n} Y_{ij} - n\mu^\star, \tag{1}$$

where $Y_{i1}, Y_{i2}, \ldots$ are identically distributed (μ_i, σ_i^2) for $i = 1,2,\ldots,k$, and as in the context of comparisons with a standard, $\mu^\star$ is the *standard.* (We discuss the definition of $\mu^\star$ for PASS in more detail later, but for now we can think of it as a known standard.) Recall that we assume a bigger mean is better, and *henceforth, we assume* $\mu_k > \mu_{k-1} > \cdots > \mu_1$. This assumption is for simplicity in stating results, and strict inequality is not actually necessary. In addition, we will exploit an increasing function $c_i(n) > 0$ with the property that for a given $0 < \alpha < 1$ and any i,

$$\Pr\{S_i(n) \leq -c_i(n),\ \text{some } n < \infty \mid \mu^\star \leq \mu_i\} \quad \leq \quad \alpha, \tag{2}$$

$$\Pr\{S_i(n) \leq -c_i(n),\ \text{some } n < \infty \mid \mu^\star > \mu_i\} \quad = \quad 1. \tag{3}$$

Properties (2)–(3) state that the probability that the partial sum $S_i(n)$ ever slips below $-c_i(n)$ is bounded by α if system i's mean is at least as good as the standard, and is 1 otherwise. This will be true for each system i individually. When the output data are i.i.d. normally distributed random variables, we can obtain $c_i(\cdot)$ from results in Fan et al. (2016) for either variances known or unknown.

Now consider a procedure that eliminates system i if $S_i(n) \leq -c_i(n)$ for some step $n < \infty$, and let $\mathcal{G} = \{i \colon \mu^\star \leq \mu_i\}$ be the set of "good" systems, which we assume is not empty. Further, let $\hat{\mathcal{E}}_n \subset \{1,2,\ldots,k\}$ be the set of systems that are eliminated at step n, and let $\hat{\mathcal{E}}(\infty) = \cup_{n=1}^{\infty} \hat{\mathcal{E}}_n$, the set of systems ever eliminated. Then we define the EFER to be

$$\text{EFER} := \frac{\mathrm{E}[|\mathcal{G} \cap \hat{\mathcal{E}}(\infty)|]}{|\mathcal{G}|},$$

so that $|\mathcal{G} \cap \hat{\mathcal{E}}(\infty)|$ is the number of good systems ever eliminated. Under our assumptions about $c_i(\cdot)$, and with no assumptions about whether the systems are simulated independently of each other, or how large $|\mathcal{G}|$ or k is, we have that EFER $\leq \alpha$. Further, notice that EFER $\leq \alpha$ implies EFER$(n) \leq \alpha$ for all $n \geq 1$, where the definition of EFER(n) is identical to that of EFER except with $\hat{\mathcal{E}}(\infty)$ replaced by $\hat{\mathcal{E}}(n) = \cup_{j=1}^{n} \hat{\mathcal{E}}_j$. To see this, let $\mathbb{I}_i(n)$ be an indicator variable that equals 1 if system i is eliminated by sample size n, and let $\mathbb{I}_i$ be an indicator variable that equals 1 if system $i \in \mathcal{G}$ is ever eliminated. Then using Equation (2), for all $n \geq 1$,

$$\mathrm{E}[|\mathcal{G} \cap \hat{\mathcal{E}}(n)|] = \mathrm{E}[\textstyle\sum_{i \in \mathcal{G}} \mathbb{I}_i(n)] \leq \mathrm{E}[\sum_{i \in \mathcal{G}} \mathbb{I}_i] = \sum_{i \in \mathcal{G}} \Pr\{\text{system } i \text{ ever eliminated}\} \leq |\mathcal{G}|\alpha.$$

In addition, all systems in $i \in \mathcal{G}^c$ are eliminated with probability 1 as $n \to \infty$.

3.2 Learning the Standard with PASS

Notice that evaluation of, say, systems i and ℓ with respect to the standard $\mu^\star$ using Equation (1) is completely decoupled. In fact, if we had $p \geq k$ worker processors so that we could run all simulations in parallel, then all partial sum tests could be executing at the same time with no coordination. Of course, this begs three questions: What is our SO objective? How do we set a value of $\mu^\star$ to achieve that objective? And how do we stop the evaluation, since systems better than the standard are unlikely ever to be eliminated, by design? Different versions of PASS provide different answers, as we illustrate below, and in the SO problems of most interest to us, we expect $k \gg p$.

We begin by describing a simple Parallel Survivor Selection (PSS) algorithm that does not adapt and that terminates when a computation budget is exhausted; that is, the standard $\mu^\star$ is provided as an input and there is no termination criterion other than exhausting the "computational resources" (which we intentionally specify vaguely). We then describe how to make PSS adaptive to different objectives. We avoid specifying any specific parallel computing environment other than there being p parallel processors to which we can assign "jobs" and a master processor that handles any necessary coordination. The PSS algorithm is listed as Algorithm 1 below. Note the complete absence of coupled operations.

What can be said about the set of surviving systems Q at termination? It will contain some, if not all, good systems $i \in \mathcal{G}$, with EFER less than or equal to α. As the budget is finite, it may contain undesirable systems $i \in \mathcal{G}^c$, but if the budget increases without bound, such systems will be eliminated from Q with probability 1. Interestingly, these properties will also hold for a *moving* standard $\mu(n)$, as shown in the following Theorem 1.

Theorem 1 Let $\mu(n)$ be a real-valued function for which $\mu(n) \leq \mu^\star$ for all $n = 1,2,\ldots$, and $\mu(n) \to \mu^\star$ as $n \to \infty$. Under the assumption that $\mathcal{G}$ is nonempty, and that $\mu^\star$ is replaced by $\mu(n_i)$ in PSS, then PSS guarantees

1. EFER $\leq \alpha$.
2. $\Pr\{i \in \mathrm{Q} \mid i \in \mathcal{G}^c\} \to 0$ as budget $\to \infty$.

Algorithm 1: Parallel Survivor Selection (PSS)

```
Input: known standard $\mu^\star$, replication increment $\Delta \ge 1$, and a computational budget
Initialize: set of available worker processors $\mathsf{W} \leftarrow \{1,2,\dots,p\}$, ordered list of surviving systems
  $\mathsf{Q} \leftarrow \{1,2,\dots,k\}$, $n_i \leftarrow 0$ and $S_i(0) \leftarrow 0$ for all $i \in \mathsf{Q}$
repeat on master
    while an available worker exists in $\mathsf{W}$ do
        remove the next system $i \in \mathsf{Q}$ and assign to available worker $w \in \mathsf{W}$
        on worker $w$
            receive directions and statistics from master, initialize: $j \leftarrow 0$
            repeat
                $j \leftarrow j+1$
                simulate system $i$ to obtain $Y_{i,n_i+j}$ and update $S_i(n_i+j)$
            until $j = \Delta$ or $S_i(n_i+j) \le -c_i(n_i+j)$
            communicate statistics and elimination status of system $i$ to master
    while message queue nonempty do
        receive statistics and elimination status from worker $w$ regarding system $i$
        if system $i$ is not eliminated then return it to the queue $\mathsf{Q} = \mathsf{Q} \cup \{i\}$
        release worker $w$ to available workers $\mathsf{W} = \mathsf{W} \cup \{w\}$
until the computational budget is consumed
return $\mathsf{Q}$
```

Proof: Conclusion 1 holds because for each i

$$\Pr\left\{\sum_{j=1}^{n_i}(Y_{ij}-\mu(n_i)) \le -c_i(n_i),\ \text{some } n_i<\infty\right\} \le \Pr\left\{\sum_{j=1}^{n_i}(Y_{ij}-\mu^\star) \le -c_i(n_i),\ \text{some } n_i<\infty\right\},$$

since $\mu(n_i) \le \mu^\star$ for all n_i. The result then follows from (2).

For Conclusion 2, let $\tilde{\mu} = \max\{\mu_i \colon i \in \mathcal{G}^c\}$. Thus, $\tilde{\mu} < \mu^\star$. Then $\mu(n_i) \to \mu^\star$ implies that there exists $\tilde{n}$ large enough that $\mu(n_i) \in (\tilde{\mu}, \mu^\star]$ for all $n_i \ge \tilde{n}$. For any value in this interval the result holds by (3). □

Theorem 1 suggests that—in the absence of knowledge of a desirable $\mu^\star$—we could be adaptive and somehow work our way toward a $\mu^\star$ that will achieve a desired objective; we call this PASS. Generically, let the standard $\mu^\star$ be a function of the true means: $\mu^\star = g(\mu_1, \mu_2, \dots, \mu_k)$. There are many useful mappings, depending on the context, including the following:

$$\mu^\star = \mu_k \tag{4}$$
$$\mu^\star = \mu_{k-b+1} \tag{5}$$
$$\mu^\star = \min\{\mu^\dagger, \mu_{k-b+1}\}. \tag{6}$$

Objective (4) is the most like a typical R&S procedure in which only the single best is desired; (5) applies when all of the top b systems are considered "good;" while (6) states that *all* systems with means as large as some known value $\mu^\dagger$ are of interest, if there are any, and otherwise only the b best are. The challenge is that since $\mu_1, \mu_2, \dots, \mu_k$ are unknown we need to *learn* the value of $\mu^\star$ from the simulation outputs in a way that does not destroy the desirable properties of PSS in Theorem 1. For concreteness we focus on (4) in the remainder of the paper, and exhibit one such approach in the next section.

4 BI-PASS ALGORITHM

We adopt the definition in (4) for the desired standard, $\mu^\star$, which means we want the probability of falsely eliminating the best system to be less than or equal to α, while ideally all of the other $k-1$ systems are eliminated before exhausting the budget. The strategy of the bisection-PASS (bi-PASS) algorithm is to set the standard at the midpoint, by value, of the surviving systems and work to eliminate those whose mean is lower than the midpoint; this then raises the midpoint, and eliminations continue like a bisection search. Therefore, the ideal midpoint $\bar{\mu}$ when Q is the set of surviving systems is

$$\bar{\mu}_{\text{ideal}} = \frac{1}{|\mathsf{Q}|}\sum_{i\in\mathsf{Q}} \mu_i.$$

Notice that as long as Q contains system k then $\bar{\mu}_{\text{ideal}} \le \mu_k$; and if $\mathsf{Q} = \{k\}$ then $\mu^\star = \mu_k$.

Since the true means are not known, the bi-PASS algorithm employs

$$\widehat{\mu}^\star = \frac{1}{|\mathsf{Q}|}\sum_{i\in\mathsf{Q}} \bar{Y}_i(n_i),$$

which is the average of the sample means of the surviving systems, to mimic $\bar{\mu}_{\text{ideal}}$. Updating the standard $\widehat{\mu}^\star$ is a fast, simple job to be done by the master processor based on the output data reported back by the worker processors. The bi-PASS algorithm is listed in Algorithm 2; it is similar to the PSS algorithm, except that the standard is updated.

Algorithm 2: Bisection Parallel Adaptive Survivor Selection (bi-PASS) for $\mu^\star = \mu_k$

```
Input: replication increment $\Delta \ge 1$, and a computational budget
Initialize: set of available worker processors $\mathsf{W} \leftarrow \{1,2,\ldots,p\}$, ordered list of surviving systems
    $\mathsf{Q} \leftarrow \{1,2,\ldots,k\}$, $n_i \leftarrow 0$ and $\bar{S}_i(0) \leftarrow 0$, $\mathcal{S}_i \leftarrow 0$ for all $i \in \mathsf{Q}$, and $\widehat{\mu}^\star \leftarrow -\infty$
repeat on master
    while an available worker exists in $\mathsf{W}$ do
        remove the next system $i \in \mathsf{Q}$ and assign to available worker $w \in \mathsf{W}$
        on worker $w$
            receive directions and statistics from master, initialize: $j \leftarrow 0$
            repeat
                $j \leftarrow j+1$
                simulate system $i$ to obtain $Y_{i,n_i+j}$
                update $\mathcal{S}_i \leftarrow \mathcal{S}_i + Y_{i,n_i+j}$
                update $\bar{S}_i(n_i+j) \leftarrow \mathcal{S}_i - (n_i+j)\widehat{\mu}^\star$
            until $j = \Delta$ or $\bar{S}_i(n_i+j) \le -c(n_i+j)$
            communicate statistics and elimination status of system $i$ to master
    while message queue nonempty do
        receive statistics and elimination status from worker $w$ regarding system $i$
        if system $i$ is not eliminated then return it to the queue $\mathsf{Q} = \mathsf{Q} \cup \{i\}$
        possibly update $\widehat{\mu}^\star = \sum_{i\in\mathsf{Q}} \bar{Y}_i(n_i)/|\mathsf{Q}|$
        release worker $w$ to available workers $\mathsf{W} = \mathsf{W} \cup \{w\}$
until the computational budget is consumed
return $\mathsf{Q}$
```

While a full mathematical analysis of bi-PASS is not yet available, we present results for a stylized version that suggests why it works (see the empirical results in Section 5 below). The stylized version of bi-PASS employs the following simplifying assumptions:

Assumption 1. *The system output data are normally distributed with common, known variance* σ^2.

Assumption 1 implies that $c_i(\cdot)$ from Fan et al. (2016) no longer depends on i, so it will be denoted by $c(\cdot)$. The function $c(\cdot)$ does depend on the value of σ^2, but does not depend on the means or on k.

Assumption 2. *The standard* $\widehat{\mu}^\star = \widehat{\mu}^\star(n)$ *is updated continually and in sync, so that the standard for system k at replication n is*

$$\widehat{\mu}^\star(n) = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n}\sum_{j=1}^{n} Y_{ij} \tag{7}$$

and bi-PASS monitors $\bar{S}_i(n) = \sum_{j=1}^{n}(Y_{ij} - \widehat{\mu}^\star(n)) = \sum_{j=1}^{n} Y_{ij} - n\widehat{\mu}^\star(n)$ *for* $i = 1, 2, \ldots, k$.

Assumption 3. *Eliminated systems continue to be simulated to update the standard* $\widehat{\mu}^\star(n)$ *in Equation* (7).

Theorem 2 Under Assumptions 1–3, bi-PASS guarantees that $\Pr\{\bar{S}_k(n) \le -c(n),\ \text{some } n < \infty\} \le \alpha$.

Proof: By straightforward algebra

$$\begin{aligned}
\bar{S}_k(n) &= \sum_{j=1}^{n}(Y_{kj} - \widehat{\mu}^\star(n)) = \sum_{j=1}^{n}\left(Y_{kj} - \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n}\sum_{\ell=1}^{n} Y_{i\ell}\right) \\
&= \sum_{j=1}^{n}\left(\frac{k-1}{k}Y_{kj} - \frac{1}{k}\sum_{i=1}^{k-1} Y_{ij}\right) = \sum_{j=1}^{n} W_j,
\end{aligned}$$

where the W_j are i.i.d. normal with mean $\mu_k - \bar{\mu} > 0$ and variance $\frac{k-1}{k}\sigma^2$. But

$$\begin{aligned}
\Pr\left\{\sum_{j=1}^{n} W_j \le -c(n)\right\} &= \Pr\left\{\sqrt{\frac{k}{k-1}}\sum_{j=1}^{n} W_j \le -\sqrt{\frac{k}{k-1}}c(n)\right\} \\
&\le \Pr\left\{\sqrt{\frac{k}{k-1}}\sum_{j=1}^{n} W_j \le -c(n)\right\} \qquad (8) \\
&\le \alpha,
\end{aligned}$$

where (8) follows because $\sqrt{\frac{k}{k-1}} > 1$. Notice that $\sqrt{\frac{k}{k-1}}W_j$ are i.i.d. normal with mean $\sqrt{\frac{k}{k-1}}(\mu_k - \bar{\mu}) > 0$ and variance σ^2, so the final inequality follows by the design of $c(\cdot)$. □

Remark: Analysis of the stylized model suggests that even with a statistically learned standard it is possible to guarantee the desired EFER, in this case for the best system, system k. Since the boundary function $c(\cdot)$, or more generally functions $c_i(\cdot)$ when variances are unequal, do not depend on the number of systems, once eliminations do occur we essentially have a R&S problems with a smaller value of k; the result above is valid for any $k \ge 1$.

Remark: The literature on multi-armed bandits, especially those that provide a probably approximately correct (PAC) bound, is closely related to R&S. The *median elimination procedure* in Even-Dar et al. (2002) bears a superficial resemblance to our bi-PASS algorithm, in that at the end of a sequential stage all systems (arms) whose means are less than the median of the sample means are eliminated. However, the guarantee is a good selection and not an error rate guarantee, the stages are fully coupled, and the simulation (arm) output is Bernoulli.

5 EMPIRICAL STUDY

In this section we summarize the results of some experiments to compare the performance of bi-PASS to PSS. Our focus is on the loss of efficiency from having to "learn" the standard $\mu^\star = \mu_k$ relative to knowing it, which is (of course) not possible in practice but is possible in a controlled study. Stated differently, PSS represents a bound on the efficiency attainable by a PASS algorithm, and we want to see how close bi-PASS gets.

5.1 Description of the Experiments

Rather than implement Algorithms 1 and 2 on a parallel computing platform, we create a discrete-event simulation that represents an environment with $p+1$ processors as a closed-loop system of two tandem queues: a single-server queue mimicking the master, and a single-buffer-p-server queue mimicking the workers. There are initially k "customer" entities in the worker queue that represent the k systems. Entity i carries information corresponding to system i, including its current cumulative sum of observations $\sum_{j=1}^{n_i} Y_{ij}$, number of obtained observations n_i, and a local copy of the standard; the standard is $\mu^\star = \mu_k$ for PSS, and $\widehat{\mu}^\star$ for bi-PASS.

At any given time a worker processor is simulating at most one system. After a system is simulated by a worker, it exits the queueing loop if it has been eliminated, or proceeds to the master queue if not. After a system is processed by the master, it exits the loop if the updated standard eliminates it, or continues to the worker queue if not. This pattern continues until a prespecified simulation clock time, representing the budget, is attained. The processing time for the master is 1 time unit if it updates the standard, and 0 otherwise. The time per simulation replication on a worker is exponentially distributed with mean $\beta > 0$ time units; thus, it is defined relative to the master. We do not model message passing, data transfer or initialization time separately.

An *experiment* is defined by setting the following factors: The number of systems k, along with the configuration of the true means $\{\mu_i\colon i = 1,2,\ldots,k\}$; the common known variance σ^2; the number of parallel processors $p+1$; the replication batch size Δ; the mean time per simulation replication β; and the total time budget T. In all cases the output data are normally distributed. For each experiment we run 100 macroreplications of PSS (Algorithm 1) and bi-PASS (Algorithm 2).

Since bi-PASS uses the average of the sample means of the surviving systems as the standard, we require Δ observations to be obtained from all systems before the standard is updated for the first time and any eliminations can occur. After completing this "first-round" requirement the standard is computed and all systems will have the same local copy when they return to the worker queue to be processed for the second time. From then on the local standards may differ due to asynchronous updating. PSS, on the other hand, uses the true mean of the true best system as the known standard. Nevertheless, we also impose the first-round requirement to represent the need to estimate σ^2 in practice. For PSS the local copy of the standard is always $\mu^\star = \mu_k$. Since the standard is not updated by the master, the processing time of the master is always 0 for PSS. For both bi-PASS and PSS we use the boundary function $c(t) = \sqrt{[\gamma + \log(t+1)](t+1)}$ and choose the constant $\gamma = 10.4$ based on a table from Fan et al. (2016) to be appropriate for a known-variance case.

We compare the performance of bi-PASS and PSS by analyzing their ability to protect the true best system, system k, and ability to eliminate the inferior systems $1,2,\ldots,k-1$. We examine each method's empirical false elimination rate (FER) and average surviving subset size at termination time T. For a particular macroreplication we say that a false elimination occurs if the best system is eliminated, therefore the empirical FER is the proportion of macroreplications in which false eliminations occur.

We consider the following factor settings: $k = 10{,}000$ systems with means $\mu_i = \sqrt{i}$ for $i = 1,2,\ldots,k$, employing $p+1 = 101$ processors with per replication mean compute time $\beta = 50$, and nominal EFER $\alpha = 0.005$. We let the time budget T be 1 million, 10 million, and 100 million time units, representing small, medium, and large-budget cases. We let the batch size Δ be 10 or 100 replications to examine the effect of many short runs vs. fewer long runs of each system on each return to the workers. We consider output variance $\sigma^2 = 1$ and $\sigma^2 = 10$. We also tested two different initial orderings of the systems in the worker queue: from best to worst (BtoW) and from worst to best (WtoB). In a real problem we expect the order to be scrambled; however, by employing these extreme cases we can examine the effect of initial ordering.

5.2 Experiment Results

Out of 2400 macroreplications (24 experiments × 100 macroreplications each), we observed only 1 false elimination for PSS; and we observed no false eliminations for PASS out of 2400 macroreplications. This result suggests that an EFER ≤ 0.005 was achieved.

Table 1 displays the results for surviving subset size. When the output variance σ^2 is larger, the surviving subset sizes are also larger. This trend demonstrates that the boundary function is more cautious about eliminating systems when there is more variability. In all configurations, the surviving subset size for PASS was larger than PSS, in most cases roughly twice as large. This difference in subset size is the cost of learning the standard as opposed to having perfect knowledge, as using the average of the sample means of the surviving systems as the standard eliminates systems less aggressively compared to using the true value of the best. A doubling of the subset size seems quite reasonable.

When the budget T is small and the batch size Δ is large, both bi-PASS and PSS had large surviving subsets compared to the same budget with the smaller batch size. This effect happens because the number

Table 1: Empirical performance of bi-PASS vs. PSS

				bi-PASS	**PSS**
Variance	**Initial Order**	Δ	T	**Subset Size**	**Subset Size**
$\sigma^2 = 1$	WtoB	10	10^6	13.6	6.7
			10^7	4.8	2.5
			10^8	1.9	1.0
		100	10^6	131.6	113.2
			10^7	5.0	2.5
			10^8	2.0	1.0
$\sigma^2 = 1$	BtoW	10	10^6	13.9	6.9
			10^7	4.6	2.5
			10^8	2.0	1.0
		100	10^6	5709.8	170.3
			10^7	5.0	2.6
			10^8	2.0	1.0
$\sigma^2 = 10$	WtoB	10	10^6	40.8	19.3
			10^7	13.3	6.7
			10^8	4.7	2.4
		100	10^6	363.2	162.2
			10^7	13.5	6.6
			10^8	4.7	2.5
$\sigma^2 = 10$	BtoW	10	10^6	42.45	19.2
			10^7	13.3	6.6
			10^8	4.7	2.5
		100	10^6	5770.1	268.9
			10^7	14.0	6.8
			10^8	4.7	2.5

of observations will not be very evenly distributed at termination time T when T is small and the batch size Δ is large. In this case, we expect that many inferior, but surviving systems that could have been eliminated by just a few additional observations are still in the worker queue while waiting for large simulation batches for other systems to complete.

The large-batch-size effect becomes even more prominent when the worker queue is preloaded with systems from best to worst. Since superior systems have a lower probability of being eliminated, they are more likely to survive for most or all of their Δ replications compared to inferior systems. In this case, bi-PASS has a dramatically larger surviving subset than PSS, because the faster that inferior systems are eliminated, the faster that the bi-PASS standard increases. There is a compounding effect that having a large number of inferior systems surviving prevents the standard from increasing, which causes inferior systems to be eliminated slowly. When the budget is sufficiently large to accommodate a large batch size, there is no longer the problem of inferior systems not getting the chance to be eliminated. These results indicate the potential for choosing the batch size Δ adaptively.

To further illustrate how learning the standard affects elimination of inferior systems, Figure 1 plots the surviving subset size as a function of computation time for the case of $\sigma^2 = 1$, $\Delta = 10$ and WtoB for the first 2,500,000 time units of computer budget, averaged across 10 macroreplications. bi-PASS, with an estimated standard, tracks PSS, which exploits knowing the true value of the best, quite effectively after an initial lag.

ACKNOWLEDGEMENTS

Hunter's research was partially supported by the National Science Foundation under Grant Number CMMI-1554144. Nelson and Pei's research was partially supported by the National Science Foundation under Grant Number CMMI-1537060.

REFERENCES

Benjamini, Y., and Y. Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing". *Journal of the Royal Statistical Society, Series B* 57(1):289–300.

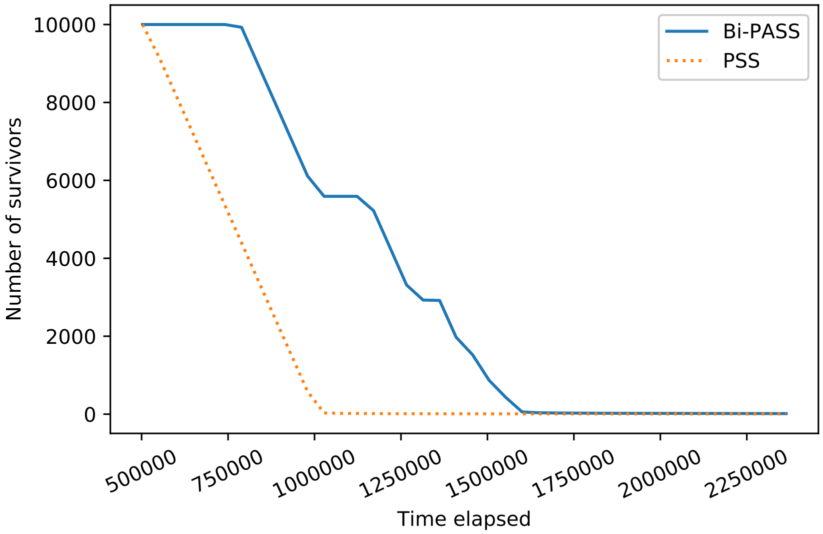

Figure 1: Number of survivors vs. computer time of bi-PASS vs. PSS for the first 2,500,000 time units of the computer budget when $\sigma^2 = 1$ and $\Delta = 10$.

Efron, B. 2012. *Large-scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, Volume 1. Cambridge, United Kingdom: Cambridge University Press.

Even-Dar, E., S. Mannor, and Y. Mansour. 2002. "PAC bounds for Multi-Armed Bandit and Markov Decision Processes". In *International Conference on Computational Learning Theory*, edited by J. Kivinen and R. H. Sloan, 255–270. New York: Springer.

Fan, W., L. J. Hong, and B. L. Nelson. 2016. "Indifference-Zone-Free Selection of the Best". *Operations Research* 64(6):1499–1514.

Hunter, S. R., and B. L. Nelson. 2017. "Parallel Ranking and Selection". In *Advances in Modeling and Simulation*, edited by A. Tolk et al., Chapter 12, 249–275. New York: Springer.

Kim, S.-H. 2005. "Comparison with a Standard via Fully Sequential Procedures". *ACM Transactions on Modeling and Computer Simulation* 15(2):155–174.

Kim, S.-H., and B. L. Nelson. 2006. "Selecting the Best System". In *Handbooks in Operations Research and Management Science*, edited by S. Henderson and B. L. Nelson, Chapter 13, 501–534. New York: Elsevier.

Nelson, B. L., and D. Goldsman. 2001. "Comparisons with a Standard in Simulation Experiments". *Management Science* 47(3):449–463.

Ni, E. C., S. R. Hunter, and S. G. Henderson. 2013. "Ranking and Selection in a High Performance Computing Environment". In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy et al., 833–845. Piscataway, New Jersey: IEEE.

Singham, D. I., and R. Szechtman. 2016. "Multiple Comparisons with a Standard using False Discovery Rates". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder et al., 501–511. Piscataway, New Jersey: IEEE.

Szechtman, R., and E. Yücesan. 2008. "A New Perspective on Feasibility Determination". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason et al., 273–280. Piscataway, New Jersey: IEEE.

Wan, H., B. E. Ankenman, and B. L. Nelson. 2010. "Improving the Efficiency and Efficacy of Controlled Sequential Bifurcation for Simulation Factor Screening". *INFORMS Journal on Computing* 22(3):482–492.

Xie, J., and P. I. Frazier. 2013. "Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard". *Operations Research* 61(5):1174–1189.

AUTHOR BIOGRAPHIES

LINDA PEI is a Ph.D. student in the Department of Industrial Engineering and Management Sciences at Northwestern University. Her research interests are simulation optimization and simulation analytics. Her e-mail address and website are lindapei2016@u.northwestern.edu and http://lindapei.me, respectively.

BARRY L. NELSON is the Walter P. Murphy Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University. He is a Fellow of INFORMS and IIE. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems, and he is the author of *Foundations and Methods of Stochastic Simulation: A First Course*, from Springer. His e-mail address and website are nelsonb@northwestern.edu and http://www.iems.northwestern.edu/~nelsonb/, respectively.

SUSAN R. HUNTER is an assistant professor in the School of Industrial Engineering at Purdue University. Her research interests include Monte Carlo methods and simulation optimization, especially in the presence of multiple performance measures. Her e-mail address is susanhunter@purdue.edu, and her website is http://web.ics.purdue.edu/~hunter63/.