Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Parallel Adaptive Survivor Selection

Linda Pei, Barry L. Nelson, Susan R. Hunter

To cite this article:
Linda Pei, Barry L. Nelson, Susan R. Hunter (2022) Parallel Adaptive Survivor Selection. Operations Research

Published online in Articles in Advance 24 Aug 2022

. https://doi.org/10.1287/opre.2022.2343

informs.
http://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–19
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Methods

Parallel Adaptive Survivor Selection

Linda Pei,[a,*] Barry L. Nelson,[a] Susan R. Hunter[b]

[a] Department of Industrial Engineering & Management Sciences, Northwestern University, Evanston, Illinois 60208; [b] School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907
*Corresponding author

Contact: lindapei2016@u.northwestern.edu, https://orcid.org/0000-0002-4282-6930 (LP); nelsonb@northwestern.edu, https://orcid.org/0000-0002-1325-2624 (BLN); susanhunter@purdue.edu, https://orcid.org/0000-0003-0881-0895 (SRH)

Received: September 19, 2020
Revised: January 30, 2021; September 3, 2021; April 26, 2022
Accepted: June 21, 2022
Published Online in Articles in Advance: August 24, 2022

Area of Review: Simulation

https://doi.org/10.1287/opre.2022.2343

Abstract. We reconsider the ranking and selection (R&S) problem in stochastic simulation optimization in light of high-performance, parallel computing, where we take "R&S" to mean any procedure that simulates *all* systems (feasible solutions) to provide some statistical guarantee on the selected systems. We argue that when the number of systems is very large, and the parallel processing capability is also substantial, then neither the standard statistical guarantees such as probability of correct selection nor the usual observation-saving methods such as elimination via paired comparisons or complex budget allocation serve the experimenter well. As an alternative, we propose a guarantee on the *expected false elimination rate* that avoids the curse of multiplicity and a method to achieve it that is designed to *scale computationally with problem size and parallel computing capacity*. To facilitate this approach, we present a new mathematical representation, prove small-sample and asymptotic properties, evaluate variations of the method, and demonstrate a specific implementation on a problem with over $1{,}100{,}000$ systems using only 21 parallel processors. Although we focus on inference about the best system here, our parallel adaptive survivor selection framework can be generalized to many other useful definitions of "good" systems.

Funding: This work was supported by the National Science Foundation [Grants CMMI-1537060 and CMMI-1554144].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2022.2343.

Keywords: stochastic simulation • parallel simulation optimization • ranking and selection • master-worker

1. Introduction

We consider optimization over k simulated systems (feasible solutions) evaluated by their mean performances $\mu_1, \mu_2, \ldots, \mu_k$, using methods that simulate *all* systems and provide some *statistical guarantee* of finding good ones. This context is the well-worn territory of ranking and selection (R&S), which is important in simulation research and practice because it is virtually the only class of simulation optimization methods for which strong finite-time, global statements are possible in a non-Bayesian setting; that is, without having to commit to a prior distribution on $\mu_1, \mu_2, \ldots, \mu_k$.

We are interested in situations where the number of systems k is so large that simulating all systems is only possible if we can simulate multiple systems simultaneously with $p \gg 1$ parallel processors. Hunter and Nelson (2017) provide an overview of parallel R&S; they show that the parallel R&S procedures created to date try to extend or load-balance methods created for $p = 1$ processor, which inherently leads to compromises. Three prominent examples are Ni et al. (2017), Zhong and Hong (2022), and Luo et al. (2015); the first two use divide-and-conquer strategies and load balancing, whereas the latter achieves a relaxed statistical requirement by restricting comparisons to special points in time. *We adopt a new paradigm by first rethinking the objectives of R&S when k is very large and then by designing a procedure especially suited for* $p \gg 1$. Unlike the R&S procedures cited in Hunter and Nelson (2017), our approach is inefficient for small k and $p = 1$ but gains computational efficiency as both increase. We achieve these gains by focusing on overall computation rather than just the "number of observations" and by leveraging information provided by all k systems collectively rather than comparisons among systems individually.

Our *parallel adaptive survivor selection* (PASS) framework controls the rate at which "good" systems are eliminated by comparing all systems marginally to a statistically improving standard learned globally. In this setting, we argue that *expected false elimination rate* is a practically relevant criterion and that it leads to procedures that scale with the number of systems k and the number of processors p. PASS algorithms can rapidly reduce a very large number of systems to a relatively small number, which is useful in and of itself, and can also be the front end to an efficient follow-on procedure for small k.

In this paper, we present the theoretical underpinnings of bi-PASS, which is a particular version of the PASS framework, and describe a Python/MPI implementation. We illustrate the effectiveness of our bi-PASS implementation on an R&S problem with more than $1,100,000$ systems using 21 parallel processors. To the best of our knowledge, this is the largest problem ever attempted with so few processors. Pseudo-code for bi-PASS and an empirical comparison of bi-PASS against competitors is provided in the companion paper (Pei et al. 2020). Although our implementation does not yet reflect optimally tuned software, it provides a proof-of-concept for bi-PASS performance.

The purpose of this paper is to advance bi-PASS theory and provide a rigorous computational model that captures parallel dynamics. The companion paper (Pei et al. 2020) is devoted to careful empirical comparisons versus competitors and detailed pseudocode. This paper is organized as follows: Section 2 describes the context of our work. Section 3 contains the mathematical preliminaries motivating the PASS approach, whereas Section 4 contains a theoretical framework for bisection-PASS (bi-PASS) and key theorems. Section 5 introduces a framework for analyzing the computational intricacies for bi-PASS and information discrepancies in a master-worker computing framework, and Section 6 provides additional theoretical results and practical strategies for this computational environment. Section 7 lists essential details of our Python/MPI implementation. Section 8 summarizes the evaluation against competitors in Pei et al. (2020) and introduces new empirical results, whereas Section 9 closes the paper with discussion.

2. Background

R&S procedures, whether frequentist or Bayesian in philosophy, usually search for the best system or a system tied with the best. They provide a statistical guarantee of correctly selecting a system that has a sufficiently small optimality gap, correctly including the best system within a subset that ideally contains many fewer than k systems, or minimizing some measure of suboptimality for the chosen system. For overviews see Hunter and Nelson (2017), Kim and Nelson (2006b), and Frazier (2010). *All these approaches suffer in some way as k increases because their family-wise statistical statements encompass all k systems, and they usually focus on reducing the number of observations rather than overall computation time.*

In a practical application, when k is in the thousands to millions, it seems highly likely that

- There are many systems that are acceptably good relative to the goals of the experimenter, and a huge number that are not.
- What constitutes "acceptably good" can be defined in terms of a known value, denoted μ^+, the unknown means $\mu_1, \mu_2, \ldots, \mu_k$, or both.
- The systems are created by taking combinations of some more basic decision variables.

When large k arises because the systems are created from combinations of decision variables, it is tempting to use methods that search the feasible region and exploit the spatial structure without simulating all feasible solutions. Creation of search methods has been an active research area in stochastic simulation, and many algorithms exist. See, for instance, Pasupathy and Ghosh (2013) for a survey, the proceedings of any Winter Simulation Conference (www.informs-sim.org) for the latest research, and Salemi et al. (2019) and Semelhago et al. (2021) for a well-tested algorithm. Such methods are required for problems beyond the R&S computational limit. However, all search methods to date are susceptible to the three errors in simulation optimization noted by Nelson and Pei (2021): failure to simulate the optimal solution, failure to recognize the best solution actually simulated, and returning a biased estimate of the performance of the selected solution. R&S has the potential to statistically control all three errors, in addition to being relatively easy to implement in parallel. Of course, a continuous-decision-variable approximation is possible for some discrete-decision-variable problems, but not all; see Fu (2002), who includes real-world case studies of staffing problems. Another example is Hoffman et al. (2018), in which one of L preventative maintenance scores are assigned to d machines, so the solution space has L^d feasible solutions. Finally, the dimension of the decision variable is a significant issue for search methods, whereas dimension does not matter at all for R&S, only the number of feasible solutions k. For all these reasons, it is sensible to treat a simulation optimization problem as an R&S problem when computationally feasible to do so, and PASS makes much larger problems computationally feasible with a smaller number of parallel processors required.

Our first departure from typical R&S is designating a system as "good" if its mean is better than some "standard" $\mu^\star = s(\mu_1, \mu_2, \ldots \mu_k, \mu^+)$ for some known function $s(\cdot)$. Although this approach may seem related to comparisons with a standard (Nelson and Goldsman 2001, Kim 2005, Xie and Frazier 2013), we use the standard very differently as explained later.

Our second departure is adopting the statistical objective of controlling the *rate* at which we eliminate systems that are better than the standard. What we call the *expected false elimination rate* (EFER) can be controlled marginally and therefore does not suffer as k increases. This error-rate approach has had a significant impact on large-scale statistical inference more generally (Benjamini and Hochberg 1995, Efron 2012). In the simulation community, Singham and Szechtman (2016) are the closest analogy, and there is a connection between our work and Fan et al. (2016) on indifference-zone free R&S. However, EFER is not the same as controlling the probability of a good selection (PGS; Eckman and Henderson 2018), which is still a family-wise probability statement and requires specifying an allowable optimality gap.

Our final departure is to avoid any direct comparisons among pairs of systems. Although paired comparisons are the foundation for many observation-efficient R&S procedures, they can be a computational bottleneck for parallel implementations. Instead, we compare the systems *individually* to an estimate $\widehat{\mu}$ of $\mu^\star$ that is learned *collectively*. This approach is somewhat similar to the problem of feasibility determination for stochastic constraints (Szechtman and Yücesan 2008, Batur and Kim 2010).

R&S problems and methods bear some resemblance to multiarmed bandit problems and methods, with "simulating a system" corresponding to "pulling an arm." Although it is difficult to cover all aspects of this huge research area succinctly, bandit problems commonly emphasize online learning with bounded costs and rewards and strategies that minimize accumulated regret. R&S problems are offline system design problems typically assuming normally distributed performance and focusing on computational efficiency while selecting good systems. Regret-minimizing algorithms can be inefficient at identifying the single best system (Bubeck et al. 2009, Russo 2020), underscoring the significant differences in methods concerned with regret versus methods concerned with finite-time error guarantees. Villar et al. (2015) show that popular index approaches (Gittins 1979, Whittle 1988) require additional adjustments to control error and have limitations in detecting good systems (statistical power).

There is a thread of the bandit literature addressing best-arm identification (see Jamieson and Nowak 2014 for a survey). However, much of this literature is about playing one arm at a time, which is ill suited for a large-scale parallel setting like ours. Hillel et al. (2013) and Russo and Van Roy 2022) are examples of approaches that focus on identifying "good" arms within ϵ of the best rather than finding an optimal arm, but they both assume arms' rewards are bounded. Russo and Van Roy (2022) establish information theory machinery for a "satisficing" Thompson sampling algorithm identifying ϵ-optimal arms and prove regret bounds for infinite-armed bandit problems.

Parallel bandit approaches commonly differ from the PASS framework in their assumptions and objectives. Hillel et al. (2013) are representative of distributed bandits that consider processor communication tradeoffs and provide bounds on the number of arm pulls per player and the number of communication rounds under the assumption of bounded arm rewards. Li et al. (2017) and Falkner et al. (2018) are examples of parallel approaches that leverage a successive halving algorithm (Karnin et al. 2013, Jamieson and Talwalkar 2016) to "eliminate" the worst performers in each round and devote more effort on better ones in the next round. Successive halving is popular as a subroutine for hyperparameter optimization to address the tradeoff between considering many hyperparameter configurations with short training times versus fewer hyperparameter configurations with longer training times. There are parallel methods such as Desautels et al. (2014) and Kandasamy et al. (2017) that handle function evaluations with stochastic noise, but they use a Gaussian process modeling approach and are concerned with regret bounds.

Finally, we mention that some bandit approaches deal with known standards and even error-rate control. Haupt et al. (2011) provide a "sparse detection" algorithm that identifies the support of a sparse set of variables, with an asymptotic "false discovery rate" (FDR) guarantee. Jamieson and Jain (2018) propose an algorithm with FDR control for Gaussian systems partitioned by whether they are below or above a known standard, but this algorithm pulls arms one-at-a-time and parallelization is not straightforward. Zhong et al. (2017) propose an asynchronous parallel algorithm that identifies arms above a known standard and is "optimal" (up to a constant factor) in terms of number of arm pulls, but with the assumption of bounded rewards.

We present the PASS framework as a novel perspective on R&S and provide mathematical preliminaries in the next section.

3. Preliminaries

The PASS paradigm uses multiple workers to simulate systems in parallel and eliminate inferior systems that fall below a prespecified standard while providing error-rate control for retaining "good" systems that are at or above the standard. We define a "standard" of acceptable performance for systems, $\mu^\star = s(\mu_1, \mu_2, \ldots, \mu_k, \mu^+)$, in which $s(\cdot)$ is a known function (examples follow), but the systems' true means $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_k$ are unknown to us. We assume that $\mu^\star$ is large and larger means are better, so that system 1 has the best mean. PASS attempts to eliminate inferior systems with means $\mu_i < \mu^\star$ while guaranteeing that good systems with means $\mu_i \geq \mu^\star$ are eliminated at a rate no greater than a prespecified $\alpha \in (0, 1)$.

In this section, we suppose that $\mu^\star$ is known and develop notation and key concepts. When $\mu^\star$ is known, it is easy to create a parallel algorithm with minimal synchronization because only outputs from, say, system i are relevant for deciding if $\mu_i \geq \mu^\star$. In Section 4, we enrich these foundations so that PASS can be implemented with an unknown $\mu^\star$ that is "learned" throughout the algorithm and in a master-worker framework in which processors must pass messages to share information. To define our mathematical setting, we require the following assumptions.

Assumption 1. *For each system $i \in \{1, 2, \ldots, k\}$, the cumulative simulation output up to $t \geq 0$ is $\mathcal{B}_i(t) = \sigma_i \mathcal{W}_i(t) + t\mu_i$, where $0 < \sigma_i < \infty$ and $\mathcal{W}_i(\cdot)$ is a standard Brownian motion.*

Assumption 2. *There is a common known variance for all systems, so that for each $i \in \{1, 2, \ldots, k\}$, $\sigma_i^2 = \sigma^2$ where $0 < \sigma^2 < \infty$ is known.*

Assumption 3. *Given $\mu^\star$ and α, $g : [0, \infty) \to [0, \infty)$ is an increasing function that depends on α, such that for any i,*

$$\Pr\{\mathcal{W}_i(t) + t(\mu_i - \mu^\star) \leq -g(t), \text{ for some } t < \infty \mid \mu_i \geq \mu^\star\} \leq \alpha,$$
$$\Pr\{\mathcal{W}_i(t) + t(\mu_i - \mu^\star) \leq -g(t), \text{ for some } t < \infty \mid \mu_i < \mu^\star\} = 1.$$

For ease of mathematical analysis, Assumption 1 represents the cumulative output of each system as a continuous-time Brownian motion. Assumption 2 states that variances are known and equal across systems, and without loss of generality we take $\sigma^2 = 1$, so that $\mathrm{Var}(\mathcal{B}_i(t)) = t$ and $\mathcal{B}_i(t) = \mathcal{W}_i(t) + t\mu_i$ for all i. We address the realities of unknown and unequal variances later, as well as the use of common random numbers.

The properties of $-g(\cdot)$ in Assumption 3 imply that for each system i individually, the probability that $\mathcal{W}_i(t) + t(\mu_i - \mu^\star)$ ever falls below $-g(t)$ is bounded above by α if i is at least as good as $\mu^\star$, and equals one if i is worse than $\mu^\star$. We use $-g(\cdot)$ to eliminate systems that appear worse than $\mu^\star$. We specify an "error rate" $\alpha \in (0, 1)$, which we later define rigorously.

For our experiments in Section 8, we use $g(t) = \sqrt{[c + \log(t+1)](t+1)}$ from Fan et al. (2016) as the boundary function, where c is a constant chosen to guarantee the EFER α; c is increasing in $1 - \alpha$. This boundary resides in a class of functions whose members are continuously differentiable, grow at a rate in between $O(\sqrt{t \log\log t})$ and $O(t)$, and satisfy Assumption 3. This boundary was originally used in an indifference-zone-free approach that does not require the typical frequentist R&S assumption of the means of the best and second-best system being at least $\delta > 0$ apart. The PASS framework is distinct from the thesis of Fan et al. (2016), but we use the same type of boundary because we do not want to eliminate systems with means equal to $\mu^\star$, and we do not assume anything about the gaps between systems' means and $\mu^\star$. Fan et al. (2016) is a select-the-best procedure that is affected by k because of providing a correct-selection guarantee and the use of pairwise comparisons. In using the Bonferroni inequality to achieve a PCS of α, their boundary (and choice of c) must keep the probability of good systems falling below it at less than $\alpha/(k-1)$, because the best system must be protected against $k - 1$ others. Because PASS achieves an error *rate* of α, we can keep the probability of good systems falling below the boundary at α, and our choice of c is unaffected by k.

To represent the mathematical and computational aspects of PASS, we require distinct notions of the algorithm's "global (wall-clock) time" versus an individual system's "local time." In a run of a PASS algorithm, each system i has its own local time $t_i(\tau) \geq 0$ that depends on the elapsed global time $\tau \geq 0$. We advance system i's local time $t_i(\tau)$ by simulating system i. In practice, we do not continuously observe Brownian motions; instead, we advance each system's local time by obtaining discrete simulation replications. The output from executing the rth independent and identically distributed (i.i.d.) simulation replication of system i is Y_{ir}. For instance, if the Y_{ir} are marginally $\mathrm{N}(\mu_i, 1)$, then $\mathcal{B}_i(n_i) \overset{\mathcal{D}}{=} \sum_{r=1}^{n_i} Y_{ir}$ at integer times $t_i(\tau) = n_i$. Therefore, system i's local time is n_i if we have n_i replications from i. More generally, when the variance is not one, then local time is proportional to n_i.

For each i, $\mathcal{B}_i(t_i(\tau))$ is the accumulated output of system i by local time $t_i(\tau)$. To advance system i's local time to gain more information about μ_i, we simulate system i on a processor, but this requires wall-clock time. Thus, we pay a computational cost in wall-clock time to gain statistical information on a system. Because of the parallel nature of PASS, at a given global time τ, there will typically be multiple systems on different processors being simulated simultaneously and accumulating local time.

In PASS, systems arecompared with the boundary and are eliminated if they fall below it, which means that they are removed from the set of systems under consideration and no longer simulated. For $\tau \geq 0$, let $Q(\tau) \subseteq \{1, 2, \ldots, k\}$ be the set of "contenders," that is, the set of systems not yet eliminated by time τ. The cardinality of this set is nonincreasing in τ. We define

$$Q(\tau) = \{i : S_i(\tilde{\tau}) > -g(t_i(\tilde{\tau})), \text{ for all } \tilde{\tau} < \tau\}, \qquad (1)$$

where each system i's centered partial sum statistic is

$$S_i(\tau) = \mathcal{B}_i(t_i(\tau)) - t_i(\tau)\mu^\star. \qquad (2)$$

In other words, we eliminate system i at global time τ if $S_i(\tau) \leq -g(t_i(\tau))$; that is, its centered partial sum statistic touches the boundary, evaluated at the system's local time. Thus, $Q(\tau)$ is the set of systems that have "survived" elimination up to time τ. We index each $S_i(\cdot)$ and $Q(\cdot)$ by global time. However, whenever we inspect a system i for elimination, we evaluate both system i's cumulative output and the boundary function at system i's *local time* $t_i(\tau)$.

PASS makes elimination decisions in this way to control the EFER. We define the EFER generically because it can apply to various definitions of "good" systems.

Definition 1. Consider a set of "good" systems $\mathcal{G} \subseteq \{1, 2, \ldots, k\}$ and a simulation optimization algorithm that delivers estimates of $\mathcal{G}$ at global time $\tau \geq 0$, denoted $\widehat{\mathcal{G}}(\tau) \subseteq \{1, 2, \ldots, k\}$, for which $\widehat{\mathcal{G}}(\tau') \subseteq \widehat{\mathcal{G}}(\tau)$ when $\tau' > \tau$. Then the EFER at time τ is

$$\text{EFER}(\tau) = \left(1 - \frac{\text{E}|\mathcal{G} \cap \widehat{\mathcal{G}}(\tau)|}{|\mathcal{G}|}\right)$$

and $\text{EFER} = \lim_{\tau\to\infty}\text{EFER}(\tau)$.

Controlling the EFER is a natural objective for very large k because it emphasizes retaining a proportion of good systems relative to the total number of good systems. If τ_{end} is the time at which we stop an optimization run, then controlling the EFER captures our desire for $\mathcal{G}$ and $\widehat{\mathcal{G}}(\tau_{\text{end}})$ to coincide, or at least have most of $\mathcal{G}$ remaining in $\widehat{\mathcal{G}}(\tau_{\text{end}})$. Because $\text{EFER}(\tau_{\text{end}}) \leq \text{EFER}$ for all $\tau_{\text{end}} < \infty$, an algorithm that guarantees an EFER for an infinite time horizon will have no greater error at finite termination times. This property is important because no realistic algorithm will run forever, and we may want to stop or transition to another procedure when the number of surviving systems is small enough.

We now relate EFER to PASS with a known standard. Because systems' true means are unknown, the set $\mathcal{G}$ is also unknown. In PASS, the set of good systems is $\mathcal{G} = \{i : \mu_i \geq \mu^\star\}$ and the estimate of this set is the set of contenders, so that $\widehat{\mathcal{G}}(\tau) = Q(\tau)$ for $\tau \geq 0$.

Under Assumptions 1–3 and Equations (2) and (1), but with no assumptions about whether systems are simulated independently of each other, or how large $|\mathcal{G}|$ or k is, we have that $\text{EFER} \leq \alpha$. Provided that the local time for all systems increases without bound as $\tau \to \infty$, we also have that $\lim_{\tau\to\infty} Q(\tau) \subseteq \mathcal{G}$ with a probability of one. The key is the boundary function in Assumption 3 that ensures that each $i \in \mathcal{G}^c$ is eliminated with a probability of one as τ increases and the probability of false elimination is controlled marginally for each $i \in \mathcal{G}$.

Thus far, we have assumed $\mu^\star$ is known but have not elaborated on how to choose it. There are many practically meaningful ways to define $\mu^\star$, including $\mu^\star = \mu_1$, in which only the best (or ties) are acceptable; $\mu^\star = \mu_b$, $2 \leq b \leq k-1$, in which systems as good as the bth best are acceptable; or $\mu^\star = \min\{\mu_1, \mu^+\}$, in which systems better than a known value μ^+, or μ_1 if there are no such systems, are acceptable. In this paper we focus on $\mu^\star = \mu_1$, since it is the closest to the usual R&S formulation. However, we expect that standards such as $\mu^\star = \mu_b$ and $\mu^\star = \min\{\mu_1, \mu^+\}$ are more relevant in practice when k is very large.

Of course, $\mu^\star$ will not be known in realistic problems, which motivates the need to *learn* $\mu^\star$ efficiently but in a way that does not compromise the EFER; the remainder of the paper concerns this subject. See Online Appendix EC.1 for how the following theory can be extended to other standards beyond $\mu^\star = \mu_1$. For a discussion of the differences among EFER guarantees and more traditional R&S guarantees such as probability of correct selection or good selection, see Online Appendix EC.2.

4. Theoretical Framework for Parallel Adaptive Survivor Selection

In this section, we introduce a theoretical framework and EFER results for bisection-PASS (bi-PASS) for the case when $\mu^\star = \mu_1$, so that the standard is the true best mean. However, unlike in Section 3, we do not assume that $\mu^\star$ is known. Instead, we estimate the true standard, and our estimate $\widehat{\mu}$ is the current sample mean of the outputs of the contenders, which we define as the not-yet-eliminated systems. Thus, bi-PASS is similar to a bisection search in that $\widehat{\mu}$ estimates the average (true) mean of the survivors, which (ideally) increases toward $\mu^\star$ as unacceptable systems are eliminated. We define $\widehat{\mu}$ rigorously in Section 4.1.

The ability to provide a global statistical guarantee is a strong motivation for solving a simulation optimization problem via R&S. In the case of bi-PASS, the desired guarantee is $\text{EFER} \leq \alpha$. Here, we provide some theoretical justification in a normally distributed output setting. Section 4.1 contains the key theorem that a synchronized version of bi-PASS in which all systems have known, common variance, attains the desired false elimination rate even if the termination time is infinite. In Section 4.2, we prove that this result also holds when systems have a common but unknown variance. These consequences are significant because they demonstrate that under some conditions, using a careful estimation of an unknown standard does not sacrifice the EFER guarantee that a known standard provides. Although elimination decisions are not completely decoupled as in PASS with a known standard, comparison with an estimated standard still avoids pairwise comparisons and full synchronization, and therefore can be parallelized efficiently.

Although this section's setting does not capture all intricacies of executing bi-PASS, the result supports the observed empirical performance. Pei et al. (2018, 2020) showcase empirically that a practical implementation of bi-PASS attains the EFER guarantee and performs well against R&S competitors, and Section 8 provides further empirical support. In practice, we must assign a system to a processor to advance its local time (at the cost of global wall-clock time), and the processor must somehow share its information (e.g., using message passing). These computational aspects in a parallel algorithm implementation are deferred to Section 5. We provide an asymptotic result about bi-PASS with out-of-sync information in Section 6.

4.1. EFER for Synchronized bi-PASS with Common Known Variance

We introduce a result for a synchronized version of bi-PASS based on the formulation in Section 3. As before, we suppose Assumptions 1–3 hold. We introduce additional assumptions for a more-structured setting which also allows us to make some notation simplifications in this section.

Assumption 4 (Equal Means Configuration). *Let* $\mu_1 = \mu_2 = \cdots = \mu_k = \mu$.

Assumption 5 (Synchronized Local Time). *Each system* $i \in \{1,2,\ldots,k\}$ *has common local time* $t_i(\tau) = t(\tau)$ *for all global time* τ, *where* $t(\cdot)$ *is nondecreasing.*

Assumption 6. *For each system* $i, j \in \{1,2,\ldots,k\}$, $i \neq j$, $\mathcal{B}_i(\cdot)$ *and* $\mathcal{B}_j(\cdot)$ *are independent.*

We use Assumption 4 because for a given common variance, the equal means configuration is the most difficult setting for the EFER because any elimination is a false elimination. Without loss of generality, we take $\mu = 0$, so that $\mathcal{B}_i(t) = \sigma \mathcal{W}_i(t)$ for all $i = 1,2,\ldots,k$. Assumption 5 is "synchronized local time," which means that each contender accumulates local time at the same rate. Because all systems are in sync, we omit dependence on global time and simply use t to index each Brownian motion. Assumption 6 allows us to fully characterize the joint distribution of the contenders' centered partial sum statistics when using an estimated standard. Later in Section 6.3, we discuss the use of common random numbers, which violates this assumption. In Online Appendix EC.3, we prove the following.

Theorem 1. *Let Assumptions 1–6 hold. Recall that, without loss of generality, we assume* $\mu = 0$ *and* $\sigma^2 = 1$, *and omit notational dependence on global time. For* $t = 0,1,2,\ldots$ *and* $i \in \{1,2,\ldots,k\}$, *define*

$$S_i(t) = \mathcal{B}_i(t) - t\widehat{\mu}(t), \tag{3}$$

where

$$\widehat{\mu}(t) = \frac{1}{|Q(t)|}\sum_{j \in Q(t)} \frac{\mathcal{B}_j(t)}{t} \tag{4}$$

and

$$Q(t) = \{i : S_i(\tilde{t}) > -g(\tilde{t}),\ \text{for all } \tilde{t} = 0,1,2,\ldots \text{ such that } \tilde{t} < t\}. \tag{5}$$

Then

$$\Pr\{S_i(t) \le -g(t),\ \text{for some } t = 0,1,2,\ldots\} \le \alpha. \tag{6}$$

Theorem 1 applies to inspection at integer times $t = 0,1,2,\ldots,$ which is natural when output are replication results. However, the proof in Online Appendix EC.3 holds for any discrete times $t = 0, \Delta t, 2\Delta t, \ldots$ with $\Delta t > 0$.

4.2. EFER for Synchronized bi-PASS with Common Unknown Variance

Section 4.1 provides an EFER result for bi-PASS when systems have a common known variance. In this section, we describe adjustments for handling common but unknown variances.

With common known variance, Theorem 1 assumes unit variance without loss of generality because we can always transform a driftless Brownian motion to obtain a standard Brownian motion. Here, we discuss this transformation. Let $\mathcal{W}(\cdot)$ be a standard Brownian motion and let $\sigma^2 > 0$ be a finite constant. If $\mathcal{B}(t) = \sigma\mathcal{W}(t)$ for $t \ge 0$, then $\mathcal{B}(t)/\sigma^2$ has the same distribution as $\mathcal{W}(t/\sigma^2)$, which has unit variance and time scale t/σ^2. See Resnick (1992) for Brownian motion scaling properties. This argument establishes the following consequence of Theorem 1.

Corollary 1. *Let Assumptions 1–6 hold. For* $t = 0,1,2,\ldots$ *and* $i \in \{1,2,\ldots,k\}$, *define*

$$S_i(t) = \frac{\mathcal{B}_i(t) - t\widehat{\mu}(t)}{\sigma^2}, \tag{7}$$

where

$$\widehat{\mu}(t) = \frac{1}{|Q(t)|}\sum_{j \in Q(t)} \frac{\mathcal{B}_j(t)}{t} \tag{8}$$

and

$$Q(t) = \{i : S_i(\tilde{t}) > -g(\tilde{t}/\sigma^2),\ \text{for all } \tilde{t} = 0,1,2,\ldots \text{ such that } \tilde{t} < t\}. \tag{9}$$

Then

$$\Pr\{S_i(t) \le -g(t/\sigma^2),\ \text{for some } t = 0,1,2,\ldots\} \le \alpha.$$

Corollary 1 introduces the transformation needed to preserve the EFER in the setting of common known variance not equal to one. In Corollary 1, each system's centered partial sum statistic is the same as in the unit variance case of Equation (3) except for division by σ^2. The argument for the boundary function is also divided by σ^2. We refer to these modifications as "scaling." Under the special case of $\sigma^2 = 1$, the notation in Theorem 1 and Corollary 1 coincide.

Exploiting Corollary 1, we now describe modifications for unknown variances. First, we show that Theorem 1 holds for common unknown variance using a more generous boundary that accounts for more uncertainty than in Assumption 3. Standard R&S conditioning arguments, as in Fan et al. (2016), can be used to adjust the boundary for estimated variance.

Assumption 7. *Given $\alpha \in (0,1)$ and nonnegative integer d, let $g_d : [0,\infty) \to [0,\infty)$ be an increasing function depending on α and d, such that for a standard Brownian motion $\mathcal{W}(\cdot)$,*

$$\mathrm{E}\left[\Pr\left\{\mathcal{W}(t) \le -g_d(td/X)\frac{X}{d},\ \textit{for some}\quad t < \infty \,\middle|\, X\right\}\right] \le \alpha, \tag{10}$$

where X is a chi-squared random variable with d degrees of freedom independent of $\mathcal{W}(\cdot)$.

Theorem 2. *Let Assumptions 1 and 4–7 hold. Assume that each system i has common variance $\sigma_i^2 = \sigma^2 < \infty$. Let $\widehat{\sigma}^2$ be a sample variance estimator of σ^2 such that $\widehat{\sigma}^2/\sigma^2 \overset{\mathcal{D}}{=} X/d$, where X is a chi-squared random variable with d degrees of freedom that is independent of $\mathcal{B}_i(\cdot)$ for each $i = 1,2,\ldots,k$. For $t = 0,1,2,\ldots$ and $i \in \{1,2,\ldots,k\}$, define $S_i(t), \widehat{\mu}(t)$, and $Q(t)$, as in Corollary 1, but with $\widehat{\sigma}^2$ in place of σ^2. Then $\Pr\left\{S_i(t) \le -g_d(t/\widehat{\sigma}^2),\ \textit{for some}\ t = 0,1,2,\ldots\right\} \le \alpha$.*

Proof. We have

$$\begin{aligned}
&\Pr\left\{S_i(t) \le -g_d(t/\widehat{\sigma}^2),\ \text{for some}\quad t = 0,1,2,\ldots\right\}\\
&\quad = \Pr\left\{\frac{\mathcal{B}_i(t) - t\widehat{\mu}(t)}{\sigma^2}\frac{d}{X} \le -g_d(td/(X\sigma^2)),\right.\\
&\qquad\qquad \left.\text{for some}\quad t = 0,1,2,\ldots\right\}\\
&\quad = \mathrm{E}\left[\Pr\left\{\frac{\mathcal{B}_i(t) - t\widehat{\mu}(t)}{\sigma^2} \le -g_d(td/(X\sigma^2))\frac{X}{d},\right.\right.\\
&\qquad\qquad \left.\left.\text{for some}\quad t = 0,1,2,\ldots \,\middle|\, X\right\}\right]\\
&\quad \le \mathrm{E}\left[\Pr\left\{\mathcal{W}(t/\sigma^2) \le -g_d(td/(X\sigma^2))\frac{X}{d},\right.\right.\\
&\qquad\qquad \left.\left.\text{for some}\quad t = 0,1,2,\ldots \,\middle|\, X\right\}\right]\\
&\quad \le \alpha.
\end{aligned}$$

The second-to-last line follows from a nearly identical application of the proof of Theorem 1 in Online Appendix EC.3. Lemma EC.6 in Online Appendix EC.3 also holds when $-g(t)$ is replaced by $-a_1 \cdot g(-a_2 t)$ for some constants $a_1, a_2 > 0$. □

When σ^2 is unknown, a natural estimator is the pooled sample variance

$$\widehat{\sigma}^2_{\text{pooled}} = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_0 - 1}\sum_{r=1}^{n_0}(Y_{ir} - \bar{Y}_{i\cdot})^2 = \frac{1}{k}\sum_{i=1}^{k}\widehat{\sigma}_i^2, \tag{11}$$

where $Y_{i\cdot} = n_0^{-1}\sum_{r=1}^{n_0} Y_{ir}$, $\widehat{\sigma}_i^2$ is the sample variance of system i, and n_0 is the initial number of replications obtained from each system. The pooled sample variance $\widehat{\sigma}^2_{\text{pooled}}$ is the common "scaling factor" for each system. In the normally distributed output case, the pooled sample variance satisfies the assumptions of Theorem 2 and $k(n_0 - 1)\widehat{\sigma}^2_{\text{pooled}}/\sigma^2$ has a chi-squared distribution with $k(n_0 - 1)$ degrees of freedom. However, we expect the number of systems k to be so large that a degrees-of-freedom adjustment is unnecessary and the $\widehat{\sigma}^2_{\text{pooled}}$ could be treated as "known."

In a general setting in which variances are unknown and unequal, choosing a scaling factor for each system is less straightforward. In Section 6.2, we discuss strategies for practical implementation, and verify their effectiveness empirically in Section 8.

5. Model for Computational bi-PASS

In this section, we develop notation to model how bi-PASS is implemented in practice. We refine the notation in Section 3 to capture the essential features of bi-PASS with an estimated standard in a master-worker computing framework. Otherwise, the setting is the same as Section 3, including Assumptions 1–3, and we let the common known variance be $\sigma^2 = 1$. As before, for each system i, $t_i(\tau)$ is i's local time at global time $\tau \ge 0$.

To be general, we consider a parallel computing environment in which processors do not have shared memory and must communicate information exclusively via message passing. We assume a master-worker computing framework for communication, in which workers exchange messages with a single master, but do not communicate directly with other workers. The master delegates "jobs" to p workers. When the master assigns a job to a worker, it must send a message to that worker with the information necessary to complete the job, and a worker that has completed a job must report back with a message to the master and include any newly acquired information.

In bi-PASS, we can think of the master-worker framework and jobs as a multiserver queueing network that is closed but leaky. Each job uniquely represents a contender. The master and workers are the servers, and jobs (contenders) are customers that cycle between being served by the master and being served by a worker. At initialization, no systems have been eliminated and the set of contenders consists of all k systems, so that there are also k jobs, one for each contender. Each worker serves a job i by advancing local time on system i and reporting new information on system i to the master. The master serves job i by receiving and processing this new information on system i. Processing the new information on system i includes determining whether the system should be sent to a worker to accumulate more local time, or whether the system should be eliminated, in which case the job corresponding to system i is eliminated. Optionally, a worker may also be allowed to eliminate the system it is simulating. We refer to this option as "worker elimination."

Algorithm 1 (Pseudocode Sketch of bi-PASS Master Loop)

```
1.  Listen for incoming worker messages
2.  if Worker $w$ sends job $i$ completion message then
3.      Update system $i$'s info: cumulative output and
        local time [and elimination status]
4.      Add $i$ to job queue according to some criteria
        [unless eliminated by worker]
5.  while Worker $w$ has no job do
6.      Remove first system from front of job queue
        (say, system $j$)
7.      if $j$ can be eliminated then
8.          Remove $j$ from contenders
9.          Update estimated standard
10.     else
11.         Assign $j$ to worker $w$
12.         Send cumulative output, local time, and run
            length $\Delta_j$ [and estimated standard and incre-
            ment $\delta_j$]
13. Return to Step 1
```

Algorithm 2 (Pseudocode Sketch of bi-PASS Worker Loop)

```
1.  Listen for job assignment message from master
2.  if Master sends job $j$ assignment message then
3.      while Additional $\Delta_j$ local time not yet accumulated
        do
4.          Accumulate local time
5.          [Inspect system $j$ for elimination after each
            increment $\delta_j$]
6.      Send job $j$ completion message to master in-
        cluding system $j$'s info: new cumulative output
        and local time [and elimination status]
7.  Return to Step 1
```

In Algorithms 1 and 2, we list the main loop for the master and for each worker as an outline for the structure of bi-PASS. Sections 5.1 and 5.2 provide explicit notation capturing the contenders, the estimated standard, and elimination conditions in a parallel environment. The loops begin after an initial n_0 observations are collected on each system. We describe what each worker does to complete a job and report its results back to the master, as well as what the master does to complete a job and assign a new job to an idle worker. In general, message-passing time increases with the size of the message, and bi-PASS facilitates relatively speedy communication since only a few scalar quantities are communicated between the master and a worker within a message. In brackets, we include additional steps and information needed when worker elimination is allowed. As discussed in Section 5.2, when worker elimination is allowed, a worker processing job i also iteratively inspects for elimination of system i in local time increments of $\delta_i > 0$. See Pei et al. (2020) for complete pseudocode.

In bi-PASS, there is a one-to-one correspondence between each job and contender. This property implies that bi-PASS avoids statistical bias problems created when replications from the *same* system are parallelized, as described in Heidelberger (1988) and Luo et al. (2015). Additionally, each job asks a worker to carry out the same set of operations, so that any worker can take any job and job assignment is simple for the master. It is also the case that if the systems are actually the same simulation model with different parameters or decision variables, then the master can preload the model on each worker and only pass a system's unique parameters when instructing a worker to simulate that system. Workers are either performing a simulation job or are retired, and workers do not need to wait for any other workers to start a job. This property is in contrast to algorithms like the state-of-the-art parallel good selection procedure (GSP) from Ni et al. (2017), in which there are different types of jobs and workers switch between "simulation" jobs and "screening" jobs.

Next, we introduce notation to capture discrepancies in the information that is accessible to a worker simulating system i and the information available to the master. Workers only report to the master upon a job completion. Although the quantity $\mathcal{B}_i(t_i(\tau))$ is well defined at all global times $\tau \geq 0$, the master might not have access to this information at time τ.

For each system i, define

$$a_i(\tau) = \begin{cases} 1, & \text{if system } i \text{ has been assigned to a} \\ & \quad \text{worker at global time } \tau \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

If $a_i(\tau) = 0$ at global time τ, then system i is either in the master queue or has been eliminated. We let $a_i(\tau) = 1$ if the master and a worker are communicating about system i at time τ, so that message-passing is included in the time i is assigned to a worker.

For each system i, define

$$l_i(\tau) = \sup\{\tilde{\tau} \le \tau : a_i(\tilde{\tau}) = 0\} \tag{13}$$

as the most recent ("last") global time before τ that system j was *not* assigned to a worker; thus, $t_i(l_i(\tau))$ represents the most recent local time for system i that is accessible to the master. We emphasize here that $l_i(\tau)$ is a global time, whereas $t_i(l_i(\tau))$ is a local time. Because only workers can advance systems' local times, we have $t_i(l_i(\tau)) \le t_i(\tau)$ for all global time τ and for all systems i. For any global time τ, we say that the master processor is "current" on system i if $t_i(l_i(\tau)) = t_i(\tau)$ and say that the master processor is "out-of-date" on system i otherwise. If the master is out-of-date on system i, then this implies that $a_i(\tau) = 1$, and the worker assigned to i has advanced i's local time but not yet reported new information on system i to the master.

Equations (12) and (13) are important for describing the value of the estimated standard that the master (and potentially workers) use to inspect for elimination, explained in the following Sections 5.1 and 5.2, respectively. At any global time τ, the master only has access to $\mathcal{B}_i(t_i(l_i(\tau)))$ for each system i. Therefore, the master must base the estimated standard and set of contenders at time τ on $\mathcal{B}_i(t_i(l_i(\tau)))$ for each system i. To ensure that all quantities are well defined, we assume that each master computation, such as inspecting a system for elimination or assigning a job to a worker, takes at least $\epsilon > 0$ global time, as would be the case in a real implementation.

5.1. Master-Only Elimination

Here we describe the centered partial sum statistic used to inspect a system for elimination in the "master-only elimination" setting, in which the master inspects contender i for elimination when it reaches the head of the master queue and at least one worker is idle. In other words, the master inspects contender i for elimination immediately before it is (potentially) assigned. Workers only simulate systems and do *not* inspect them for elimination.

For each system i, let $C_i(\tau)$ be the global elimination inspection times for system i up to but not including time τ. Then $C_i(\tau)$ is a set of stopping times defined as

$$C_i(\tau) = \{\tilde{\tau} < \tau : \text{ system } i \text{ is inspected for elimination}\}. \tag{14}$$

Because master computations take at least $\epsilon > 0$ global time, at most one system is inspected for elimination at any time, and the number of times a system is inspected for elimination is countable.

The master maintains the set of contenders and the estimated standard based on the information the master has available. We append [m] to Q to emphasize that $Q^{[\mathrm{m}]}(\tau)$ is the set of contenders that the master maintains based on information available to the master at global time τ. More precisely, we define $Q^{[\mathrm{m}]}(\tau)$ as

$$Q^{[\mathrm{m}]}(\tau) = \left\{i : S_i(\tilde{\tau}) > -g(t_i(\tilde{\tau})), \text{ for all } \tilde{\tau} \in C_i(\tau)\right\}, \tag{15}$$

where we define system i's centered partial sum statistic $S_i(\tilde{\tau})$ in Equation (16).

Suppose that the master inspects system i for elimination at global time $\tilde{\tau}$. Then $a_i(\tilde{\tau}) = 0$ because i is not on a worker and the master calculates system i's centered partial sum statistic as

$$S_i(\tilde{\tau}) = \mathcal{B}_i(t_i(\tilde{\tau})) - t_i(\tilde{\tau}) \underbrace{\frac{1}{|Q^{[\mathrm{m}]}(\tilde{\tau})|} \sum_{j \in Q^{[\mathrm{m}]}(\tilde{\tau})} \frac{\mathcal{B}_j(t_j(l_j(\tilde{\tau})))}{t_j(l_j(\tilde{\tau}))}}_{\text{estimated standard } \hat{\mu}(\tilde{\tau})}. \tag{16}$$

Putting this together in an explicit elimination condition, if $\tilde{\tau}$ is an inspection for system i, then the master eliminates system i if $S_i(\tilde{\tau}) \le -g(t_i(\tilde{\tau}))$. The master compares system i's cumulative output to an estimated standard computed from the master's version of the set of contenders. However, the master can only use the last reported cumulative output of each contender for the estimated standard, because the master cannot access new information about contenders assigned to workers until the workers complete their jobs and report back.

5.2. Master and Worker Elimination

Next consider the "master and worker elimination" setting, which is the same as earlier with the addition that workers also inspect systems for elimination. This modification allows for more systems to be eliminated faster. When job i is assigned to a worker, the worker advances local time and inspects system i for elimination in discrete increments of local time $\delta_i \in (0, \Delta_i)$, until a predetermined run length $\Delta_i > 0$ has been accumulated or until i is eliminated, whichever comes first. For noneliminated systems, the master continues to inspect for elimination immediately before it is potentially assigned.

Let $C_i(\tau)$ be the set of system i's global inspections up to time τ. Now $C_i(\tau)$ contains times that a worker inspects i, because system i is inspected for elimination by *both* the master and (in discrete increments of δ_i local time) by any worker that is simulating it. As before, let $Q^{[\mathrm{m}]}(\tau)$ be the set of contenders, containing systems that have not been eliminated by global time τ from the master's perspective. However, now, in the construction of each system's centered partial sum statistic, $Q^{[\mathrm{m}]}(\cdot)$ takes an argument that accommodates worker elimination.

If system i is inspected for elimination at global time $\tilde{\tau}$, its centered partial sum statistic is

$$S_i(\tilde{\tau}) = \mathcal{B}_i(t_i(\tilde{\tau})) - t_i(\tilde{\tau}) \underbrace{\frac{1}{|Q^{[\mathrm{m}]}(l_i(\tilde{\tau}))|} \sum_{j \in Q^{[\mathrm{m}]}(l_i(\tilde{\tau}))} \frac{\mathcal{B}_j\Big(t_j(l_j(\tilde{\tau}) \wedge l_i(\tilde{\tau}))\Big)}{t_j(l_j(\tilde{\tau}) \wedge l_i(\tilde{\tau}))}}_{\text{estimated standard} \hat{\mu}(\tilde{\tau})}. \tag{17}$$

When $a_i(\tilde{\tau}) = 0$, so that the master is inspecting i for elimination, the formula derived in Equation (16) is a special case of the formula in (17). To see this, suppose that the master inspects system i for elimination at time $\tilde{\tau}$. Because $a_i(\tilde{\tau}) = 0$, we have $l_i(\tilde{\tau}) = \sup\{\tau' \le \tilde{\tau} : a_i(\tau') = 0\} = \tilde{\tau}$, which also implies that $l_j(\tilde{\tau}) \wedge l_i(\tilde{\tau}) = l_j(\tilde{\tau})$.

Now consider $a_i(\tilde{\tau}) = 1$, meaning that i is assigned to some worker w. Worker w inspects system i using the estimated standard value communicated by the master at the time of job assignment. This estimated standard value stays constant throughout the job because worker w does not have access to current information on the master or other workers while it is completing job i. The $l_j(\tilde{\tau}) \wedge l_i(\tilde{\tau})$ term reflects this information discrepancy. The minimum operator ensures that the estimated standard value stays constant or "frozen." Even if other workers complete jobs and report back to the master while worker w is in progress on its job i, worker w does not have access to this new information. As before, if $\tilde{\tau}$ is an inspection for system i, then i is eliminated if $S_i(\tilde{\tau}) \le -g(t_i(\tilde{\tau}))$.

The "frozen" estimated standard in worker elimination complicates our ability to guarantee the EFER, as discussed in Online Appendix EC.4, although we observe no deleterious effects in our experiments in Section 8.

5.3. Worker-Only Elimination

We omit development of the "worker-only elimination" setting, in which only workers but not the master inspect for elimination. We see no practical advantage of this setting, even though it fits under the general framework of Equation (17). Because the master must manage the estimated standard anyway, the master's inspection is a simple "if" statement with no additional computation. When a master eliminates a system, this action avoids an unnecessary job assignment and thus saves the worker overhead of message passing and setting up the simulation.

6. Analysis of Computational bi-PASS

In this section, we discuss theoretical results and practical considerations for a computational implementation of bi-PASS in the master-worker computing framework of Section 5. We present an asymptotic result showing that even when contenders' observations are out-of-sync, so that at any global time contenders may have different local times from each other, any problematic statistical effects are mitigated when the number of systems k is large. We also discuss strategies for estimating scaling factors when variances are unknown and unequal and present results suggesting that common random numbers should not be used with bi-PASS.

6.1. Asymptotic EFER for bi-PASS

We consider the asymptotic behavior of bi-PASS, as outlined in Section 5 and Algorithms 1 and 2, as the number of systems $k \to \infty$. This setting is relevant as bi-PASS is designed for large k. We introduce additional assumptions for a result that shows that under equal means, known and equal variances, and no elimination, the EFER guarantee holds asymptotically despite systems being out of sync.

Assumption 8. *Assume that*

(a) *There is an infinite sequence of problems indexed by k,*

$$\{(\mathcal{B}_i^k(t), t_i^k(\tau)) : t, \tau \ge 0, i = 1, 2, \ldots, k\}_{k=1}^{\infty},$$

where each $\mathcal{B}_i^k(\cdot)$ is an independent Brownian motion with unknown mean μ_i and known variance σ_i^2. System 1 is the best, so that $\mu_i \le \mu_1 < \infty$ for all $i \in \{1, 2, \ldots\}$. There are constants σ_L, σ_U such that $0 < \sigma_L \le \sigma_i \le \sigma_U < \infty$ for all i. Let $\Pr_k$ denote the probability with respect to problem k's stochastic processes $\{(\mathcal{B}_i^k(t), t_i^k(\tau)) : t, \tau \ge 0, i = 1, 2, \ldots, k\}$.

(b) *For each $i \in \{1, 2, \ldots, k\}$ and $k \in \{1, 2, \ldots\}$, system i's local time $t_i^k(\tau)$ is a nondecreasing, continuous, and deterministic function with $t_i^k(0) = 0$ and $t_i^k(\tau) \to \infty$ as $\tau \to \infty$.*

(c) *For each $i, j \in \{1, 2, \ldots, k\}$ and $k \in \{1, 2, \ldots\}$, there is a $\tau_0 \in (0, \infty)$ such that $t_j^k(\tau_0) \ge t_0 > 0$, and a constant r such that $t_i^k(\tau)/t_j^k(\tau) < r < \infty$ for $\tau \ge \tau_0$.*

Remark 1. Letting $k \to \infty$ is only meaningful if the number of workers p also increases; otherwise, some systems might never be simulated. Assumption 8(a) is one way of representing the requirement that every system is simulated, and one possible way to achieve this is by having $p/k \to \gamma \in (0, 1)$ as $k \to \infty$, where γ might be interpreted as the number of workers per system.

Assumption 8(b) establishes that each system's local time continues advancing with respect to global time; that is, each system is simulated infinitely often. This setting is sensible, because we are not eliminating systems in this asymptotic environment. The bounds in Assumption 8(c) ensure that as $k \to \infty$, each system's local time does not get arbitrarily far ahead of any other system's local time. This bound is in fact enforceable in practice, as in the implementation in Section 7, in which we move contenders with significantly less local time to the front of the master's job queue.

Next, we state Theorem 3, and its proof is given in Online Appendix EC.5.

Theorem 3. *Let Assumptions 2, 3, 6, and 8 hold, and assume equal means $\mu_i = \mu$ and equal variances $\sigma_i^2 = \sigma^2$ for all i, and without loss of generality assume that $\mu = 0$ and $\sigma^2 = 1$.*

For $\tau \ge 0$ and $k \in \{1,2,\dots\}$, define $S_1^k(0) = 0$ and

$$S_1^k(\tau) = \mathcal{B}_1^k(t_1^k(\tau)) - \frac{t_1^k(\tau)}{k}\sum_{i=1}^{k}\frac{\mathcal{B}_i^k(t_i^k(\tau))}{t_i^k(\tau)}$$

$$= t_1^k(\tau)\left[\left(\frac{k-1}{k}\right)\frac{\mathcal{B}_1^k(t_1^k(\tau))}{t_1^k(\tau)} - \frac{1}{k}\sum_{i=2}^{k}\frac{\mathcal{B}_i^k(t_i^k(\tau))}{t_i^k(\tau)}\right].$$

Let $\mathcal{W}(\cdot)$ denote a standard Brownian motion. Consider $m > 0$ fixed local times $0 < c_1 < c_2 < \cdots < c_m < \infty$, and for each $k \in \mathbb{N}$, let $\tau_1^k < \tau_2^k < \cdots < \tau_m^k$ be the corresponding global times such that $t_1^k(\tau_\ell^k) = c_\ell$ for $\ell \in \{1,2,\dots,m\}$. Then

$$\lim_{k\to\infty} \Pr_k\{S_1^k(\tau_\ell^k) \le -g(c_\ell) \text{ for some } \ell = 1,2,\dots,m\}$$

$$= \Pr\{\mathcal{W}(c_\ell) \le -g(c_\ell) \text{ for some } \ell = 1,2,\dots,m\} \le \alpha.$$

Because system 1 is arbitrary, Theorem 3 provides an asymptotic EFER guarantee. The theorem's setting involves using the average of the sample means from all systems as an estimated standard to inspect system 1 for elimination at a fixed set of local time points. In practice, these fixed local time points are proportional to system 1's run length Δ_1. With a suitable boundary function and an infinite sequence of "problems" consisting of sets of independent Brownian motions with known and equal variances, the aforementioned estimated standard behaves asymptotically like a known standard, despite systems having different local times. Provided that there is a limit on how far out of sync the other systems can be, the impact of being out of sync vanishes as we consider larger and larger problems.

Although this result is certainly not a complete asymptotic justification of bi-PASS, it suggests why we have not detected problems empirically. This result does not include elimination of other systems and subsequent statistical effects on the estimated standard.

Remark 2. Our conditions imply that each system's global-time-to-local-time function $t_i^k(\cdot)$ can depend on i and problem size k, but not on the simulation output. When systems are not eliminated, assuming that each $t_i^k(\cdot)$ is deterministic is not dramatically different from reality.

Finally, Theorem 3 also holds when means are unequal and when variances are known yet unequal, provided that the variances of the systems remain within bounds. This result is stated later, and its proof in Online Appendix EC.5 is analogous to the proof of Theorem 3.

Corollary 2. *Let Assumptions 3, 6, and 8 hold. For $\tau \ge 0$ and $k \in \{1,2,\dots\}$, define*

$$S_1^k(\tau) = \frac{1}{\sigma_1^2}\left[\mathcal{B}_1^k(t_1^k(\tau)) - \frac{t_1^k(\tau)}{k}\sum_{i=1}^{k}\frac{\mathcal{B}_i^k(t_i^k(\tau))}{t_i^k(\tau)}\right].$$

Consider some fixed local times $0 < c_1 < c_2 < \cdots < c_m < \infty$ and for each $k \in \mathbb{N}$, let $\tau_1^k < \tau_2^k < \cdots < \tau_m^k$ be the corresponding global times such that $t_1^k(\tau_\ell^k) = c_\ell$ for $\ell \in \{1,2,\dots, m\}$. Then

$$\limsup_{k\to\infty} \Pr_k\{S_1^k(\tau_\ell^k) \le -g(c_\ell/\sigma_1^2) \text{ for some } \ell = 1,2,\dots,m\}$$

$$\le \Pr\{\mathcal{W}(c_\ell/\sigma_1^2) \le -g(c_\ell/\sigma_1^2) \text{ for some } \ell = 1,2,\dots,m\} \le \alpha.$$

6.2. Scaling Factors for Unknown and Unequal Variances

Previous results in Sections 4.1 and 4.2 provide theoretical conclusions for synchronized bi-PASS with common variance. The results of Section 6.1 show that when variances are unequal but known, using each system's true variance as its scaling factor upholds the EFER guarantee asymptotically under certain conditions, even when systems are out of sync. In practice, when variances are unequal and unknown, the situation is not straightforward. For master-only elimination and master-and-worker elimination, Equations (16) and (17) must be divided by some "scaling factor" $\tilde{\sigma}_i^2$ for each system i, and for the elimination-inspection conditions, the boundary $-g(t_i(\tau))$ becomes $-g(t_i(\tau)/\tilde{\sigma}_i^2)$. We consider three approaches for scaling factors:

1. **Pooled scaling factor.** As a common *average* variance, use the same scaling factor $\widehat{\sigma}_{\text{pooled}}^2$ for each system, as in Equation (11). If the variances do not differ too widely then we expect this approximation to perform well with a common known-variance boundary $g(\cdot)$.
2. **Batch means.** Replace Y_{ir} with differently sized batch means from each system so that the batch means have approximately equal target variance σ_c^2. That is, the basic observations from system i become $\bar{Y}_{ir}(b_i) = b_i^{-1}\sum_{h=(r-1)b_i+1}^{rb_i} Y_{ir}$, $r = 1,2,\dots$ such that $\text{Var}(\bar{Y}_{ir}(b_i)) \approx \sigma_c^2$. Again, a common known-variance boundary $g(\cdot)$ is used. Goldsman and Nelson (1990) applied a similar idea in the context of multiple comparisons.
3. **Custom scaling factor.** Each system $i \in \{1,2,\dots,k\}$ uses its own scaling factor $\widehat{\sigma}_i^2$, for example, its individual sample variance. If n_0 is common, then the same degrees-of-freedom adjusted $g_d(\cdot)$ is appropriate for all systems, but due to scaling, boundaries will be

evaluated at different local times $n/\widehat{\sigma}_i^2$ for $n = 1, 2, \ldots$ for each system i. This approach ensures the desired EFER for a known standard $\mu^\star$. This adjustment will be approximately correct if the variance of the estimated standard $\widehat{\mu}$ is effectively zero, as we expect when k is very large (initially) or n is very large (eventually) when the number of contenders is small.

Approach 1 is the simplest but can be risky if variances are significantly unequal. Approach 2 heuristically adjusts for this possibility while staying in an "equal variance" paradigm. Approach 3 slightly increases algorithm complexity but stays robust across different configurations of variances by giving each system its own scaling factor. We evaluate Approaches 1 and 3 in Section 8.

6.3. Common Random Numbers

Common random numbers (CRNs) is perhaps the most well-known and most often-used variance reduction technique (Nelson and Pei 2021), reducing the variance of the *difference* between two estimators by inducing positive correlation. This is important because pairwise differences are the foundation for many R&S procedures. Because CRN is the default experiment design in most simulation software, it may be applied intentionally or inadvertently. In this section, we explore the merits of using CRN in bi-PASS *and conclude that CRN is not generally recommended.*

For the analysis that follows, we compare independent simulations of k systems to the best-case scenario for CRN. Instead of using an independent Brownian motion $\mathcal{W}_i(t)$ for each system, we use a *common* Brownian motion $\mathcal{W}(t)$. In the best-case scenario for CRN, systems are perfectly correlated, so that $\mathcal{B}_i^{\mathrm{CRN}}(t) = \sigma_i \mathcal{W}(t) + \mu_i t$ for all $i = 1, 2, \ldots, k$; that is, the stochastic component of each system's simulation output is the same $\mathcal{W}(t)$.

We first consider the synchronized setting of Section 4.1 and then later examine the impact of being out of sync. In the synchronized setting, let

$$\begin{aligned} S_i^{\mathrm{CRN}}(t) &= \mathcal{B}_i^{\mathrm{CRN}}(t) - \frac{1}{k}\sum_{j=1}^{k} \mathcal{B}_j^{\mathrm{CRN}}(t) \\ &= (\sigma_i - \bar{\sigma})\mathcal{W}(t) + (\mu_i - \bar{\mu})t, \end{aligned}$$

where $\bar{\sigma} = \sum_{j=1}^{k} \sigma_j / k$ and $\bar{\mu} = \sum_{j=1}^{k} \mu_j / k$. If all variances are equal to σ^2, then this expression reduces to $S_i^{\mathrm{CRN}}(t) = (\mu_i - \bar{\mu})t$, which has $\mathrm{EFER} = 0$ for system 1 and any tied with it, and eliminates all inferior systems with a probability of one if we update $\bar{\mu}$ after eliminations. This result suggests a benefit to CRN, but the general case suggests otherwise.

When at least some variances are unequal, $\mathrm{Var}(S_i^{\mathrm{CRN}}(t)) = (\sigma_i - \bar{\sigma})^2 t$. If instead we simulate k independent Brownian motions and form our centered partial sum statistic $S_i^{\mathrm{IND}}(t) = \mathcal{B}_i(t) - \frac{1}{k}\sum_{j=1}^{k} \mathcal{B}_j(t)$, then it is easy to derive that

$$\mathrm{Var}(S_i^{\mathrm{IND}}(t)) = \left[\left(\frac{k-1}{k}\right)^2 \sigma_i^2 + \frac{1}{k^2}\sum_{j \neq i} \sigma_j^2\right] t.$$

Simple algebra shows that $\mathrm{Var}(S_i^{\mathrm{CRN}}(t)) < \mathrm{Var}(S_i^{\mathrm{IND}}(t))$ if and only if

$$\sigma_i > \frac{1}{2}\left(\frac{\sum_{j \neq i} \sigma_j}{k-1}\right)\left(1 - \frac{\sum_{j \neq i} \sigma_j^2}{(\sum_{j \neq i} \sigma_j)^2}\right).$$

As an easy-to-interpret special case, suppose that $\sigma_1 = \cdots = \sigma_{i-1} = \sigma_{i+1} = \cdots \sigma_k = \sigma \neq \sigma_i$. Then CRN is effective if

$$\sigma_i > \frac{\sigma}{2}\left(\frac{k-2}{k-1}\right) \approx \frac{\sigma}{2}$$

for k large. *Thus, not all systems benefit from CRN if variances are substantially unequal.*

Unfortunately, even if variances are equal or nearly so, making elimination decisions with out-of-sync local times, as we fully expect to do, can penalize CRN. To show this, we assume all $\sigma_i = 1$ and, without loss of generality, that all $\mu_i = 0$. We focus on system i at time $t_1 > t_0 > 0$, where t_0 is the local time for systems $2, 3, \ldots, k$. Consider the statistic

$$S_i^{\mathrm{CRN}}(t_1) = \mathcal{B}_i^{\mathrm{CRN}}(t_1) - \frac{t_1}{k}\left(\sum_{j \neq 1}^{k} \frac{\mathcal{B}_j^{\mathrm{CRN}}(t_0)}{t_0} + \frac{\mathcal{B}_i^{\mathrm{CRN}}(t_1)}{t_1}\right).$$

Tedious algebra yields

$$\begin{aligned} \mathrm{Var}(S_i^{\mathrm{CRN}}(t_1)) &= \left(\frac{k-1}{k}\right)^2 \left[t_1 - t_0 + \left(1 - \frac{t_1}{t_0}\right)^2 t_0\right] \\ &\xrightarrow{k \to \infty} t_1 - t_0 + \left(1 - \frac{t_1}{t_0}\right)^2 t_0 > t_1 \end{aligned}$$

for t_1 large enough. On the other hand, in the same setting but with all systems simulated independently,

$$\begin{aligned} \mathrm{Var}(S_i^{\mathrm{IND}}(t_1)) &= \left(\frac{k-1}{k}\right)^2 \left[t_1 - t_0 + \left(1 + \frac{t_1^2}{(k-1)t_0^2}\right) t_0\right] \\ &\xrightarrow{k \to \infty} t_1 - t_0 + t_0 = t_1. \end{aligned}$$

Thus, CRN can inflate variance when systems have substantially out-of-sync local times, even if variances are equal.

Remark 3. If one wants to use CRN in bi-PASS, assignment of pseudorandom number streams is straightforward because at any given time in the algorithm there is only one simulation job per contender. Then the first time a system is simulated, it is assigned well-separated streams, say $1, 2, \ldots, u$; the second time it is simulated, it uses streams $u+1, u+2, \ldots, 2u$; and so on.

This maintains synchronization across systems but guarantees independent replications. A pseudorandom number generator from L'Ecuyer et al. (2002) is well suited for this setting.

7. Implementation of bi-PASS

Here we discuss the bi-PASS implementation that we use for numerical experiments. Our practical algorithm is straightforward to deploy in a parallel, message-passing setting. Deriving the "optimal" implementation is beyond the scope of this paper and depends on the computing architecture. Our implementation choices provide promising results.

To implement bi-PASS in a master-worker framework, we use the academic and industry standard for parallel communication, the message passing interface (MPI). We specifically use the popular open-source library, Open MPI, and its Python implementation `mpi4py`, but emphasize that the structure of the code is platform and programming language independent. We do not explicitly manage a message queue because such mechanics are handled by Open MPI. We do not assume shared memory to avoid dependence on a particular computer architecture. Other communication schemes are possible but are not discussed here.

The bi-PASS algorithm consists of an initial Stage 0 requiring simulation to estimate systems' variances and a sequential elimination Stage 1. Detailed and complete pseudocode is available in the companion paper (Pei et al. 2020), but we provide a brief overview here as well.

In Stage 0, the master creates a queue of k jobs and assigns jobs one at a time to an available worker. Each job i corresponds to simulating system i for a prespecified n_0 observations and computing the running sum of output and estimated sample variances for each system and then reporting this information back to the master. After all jobs are completed and the job queue is empty, the master computes scaling factors for each system and computes an initial standard before the next stage. Thus, all systems must have n_0 observations before elimination is allowed.

At the beginning of Stage 1, the master creates a queue of k jobs and sends jobs one-at-a-time to workers until each worker is busy. Until a stopping condition occurs, the algorithm then obeys the loop structure introduced in Section 5 and Algorithms 1 and 2. In summary, Stage 1 consists of sequential elimination: The master "serves" each job in its queue by either eliminating the corresponding system or assigning it to a worker, and workers "serve" assigned jobs by advancing local time on the corresponding system and potentially inspecting the system for elimination.

For elimination decisions, we use the boundary $g_{n_0-1}(t) = \sqrt{[c_{\alpha,n_0-1} + \log(t+1)](t+1)}$, where c_{α,n_0-1} is a constant chosen so that the algorithm attains a desired false elimination rate of α with an initial sample size of n_0 for variance estimation. We use this boundary because it possesses the necessary properties of Assumption 3 and because there is a precedent for using it in the indifference-zone-free R&S literature (Fan et al. 2016). Pei et al. (2020) describe a Monte Carlo procedure for estimating c_{α,n_0-1} for given α and n_0. Constructing a boundary that is "optimal" in some sense is beyond the scope of this paper but is a topic for future work.

Recall from Algorithms 1 and 2 that Δ_i is system i's run length, which is proportional to the amount of local time to simulate on i when i is assigned to a worker. We set $\Delta_i = \Delta$ for all i so that Δ is the common run length for all systems. As in the experiments of Pei et al. (2020), we add the modification that when a job corresponding to system i completes, then i is placed at the front of the master queue if it is 2Δ or more replications behind any other system. Otherwise, the system is placed in the back of the master queue. We add this small modification to ameliorate out-of-sync local times.

If there are more worker processors than contenders, we avoid creating new jobs for "excess" workers and instead retire them to preserve the one-job-per-contender property. Although this may be inefficient for utilization, in practical settings, we do not expect bi-PASS to run for very long after the first worker is retired. In fact, if bi-PASS is used as a screening step before another R&S procedure, a natural switchover point is when the number of contenders reaches p.

Our algorithm as written does not have a defense against processor failure. However, intermittently saving the master's copy of statistics, that is, the running sums, scaling factors, and contenders, allows users to resume an interrupted bi-PASS trial from the most-recent inspection.

8. Empirical Performance

In this section, we review previous comparisons of bi-PASS versus key competitors and report two sets of experiments, all using the algorithm structure and implementation choices detailed in Section 7. We also briefly describe our computing architecture. In Section 8.1, we run extensive trials on a carefully controlled collection of small problems with 1,000 systems to understand how characteristics of the problem and bi-PASS implementation affect algorithm performance. In Section 8.2, we deploy bi-PASS on a realistic problem with more than 1 million systems.

Pei et al. (2020) is a companion paper specifically created to cover code details and empirical evaluation. It provides a thorough comparison of bi-PASS versus its closest competitor, the GSP, which is a large-scale parallel R&S algorithm. bi-PASS does not have a natural competitor because its elimination *rate* guarantee is different from the norm of family-wise statements in R&S. It also does not make sense to run bi-PASS until one contender remains because bi-PASS inefficient with a small number of contenders. Nevertheless, we include such comparisons for completeness. As noted in Section 2, a common assumption for parallel stochastic multiarmed bandit algorithms is bounded observations, and many methods provide guarantees about regret but lack assurances about finite-time selection (or elimination) probabilities or rates.

Pei et al. (2020) compare three algorithms: bi-PASS, a synchronized version of GSP similar to the Spark implementation in Ni et al. (2017), and a parallel version of subset selection based on Nelson et al. (2001); they also discuss computational tradeoffs among the algorithms. On a 22, 500 system problem with 101 processors, using the same total number of replications that GSP used to select a near-optimal solution, bi-PASS and subset selection narrowed the contenders to about 10 and 200, respectively. On a 216, 600 system problem with 101 processors, bi-PASS and GSP ran until a stopping condition of each contender (excluding eliminated systems) accumulated 100 replications. At this termination point, bi-PASS had on average about 2,300 contenders with 4 million total replications, outperforming GSP, which delivered about 3,100 contenders with 5.7 million total replications in twice the wall-clock time. In both experiments, bi-PASS did not make any false eliminations, showing support for the EFER guarantee and our implementation choices. Both experiments had systems with nonnormal output data, demonstrating that bi-PASS is robust to nonnormality, at least to some extent.

This section's experiments used Northwestern University's Optimization and Statistical Learning (OSL) cluster, which runs a Linux CentOS 7.4 operating system. We use two compute nodes, each with two 40-core Intel(R) Xeon(R) Gold 6148 processors with 256 GB of RAM and 1 TB of hard drive memory. No other processes executed during our experiments, which prevented processor competition for resources and corruption of wall-clock time statistics.

8.1. bi-PASS Configuration Experiments

In this section, we evaluate different implementation choice for bi-PASS. We discuss 36 test configurations created from different combinations of two types of scaling factors, whether worker elimination is allowed, and various configurations of means and variances. Each configuration has $k = 1{,}000$ systems and $p + 1 = 11$ processors. Each trial stops when each contender (system that has not yet been eliminated) amasses at least $1{,}000\Delta$ observations or until there is only one contender remaining, whichever occurs first.

We set the initial sample size $n_0 = 10$, common run length after the n_0 stage to $\Delta_i = \Delta = 100$ for each system i, and EFER parameter $\alpha = 0.05$. We consider "custom" scaling factors and "pooled" scaling factors, as we describe in Section 6.2. We use the boundary function from Section 7 with the constant $c_{\alpha,n_0-1} = 8.6$ based on the aforementioned α and n_0 parameters.

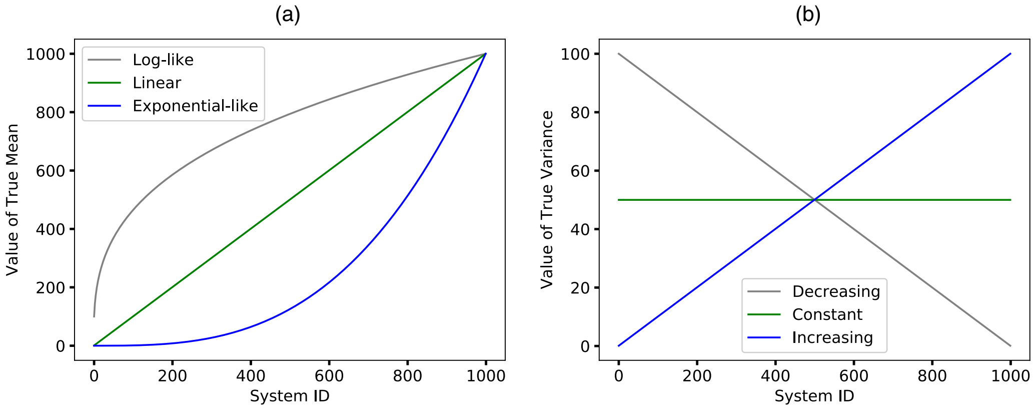
We consider master-only elimination and both master and worker elimination, as described in Section 5. When worker elimination is permitted, a worker completing job i inspects system i for elimination after each additional simulation replication. We are interested in efficiency and the EFER, because worker elimination *can* compromise EFER guarantees due to workers using an estimated out-of-date standard that does not incorporate any new information from systems that have been simulated since job assignment.

Three different configurations of the means and three different configurations of the variances are illustrated in Figure 1. Each system i has output that is $\mathrm{N}(\mu_i, \sigma_i^2)$ and independent from other systems. In each means configuration, larger indices correspond to larger means, and system k (system 1,000) has the best and largest true mean of 1,000. We refer to systems relatively far away from k as "inferior" and relatively close to k as "better." Under the log-like means configuration, in which there are many high-performing systems and they are close together, $\mu_i = \sqrt[3]{i}\cdot(1{,}000)^{2/3}$ for each i. Under linear means, $\mu_i = i$. Under exponential-like means, in which there are many bad systems, and they are close together, $\mu_i = i^3/1{,}000^2$. Under the decreasing variances configuration, better systems have low variances and $\sigma_i^2 = 100 - (i-1)/10$ for system i. Under equal variances, $\sigma_i^2 = 50$. Under increasing variances, better systems have high variances and $\sigma_i^2 = i/10$.

Table 1 summarizes the average total effort and the sum of replications across all systems at trial termination, with standard errors based on 1,000 macroreplications for different testing configurations. Each estimated total effort has a standard error no greater than 1.3%. We only report the total effort because, in each trial for each test configuration, bi-PASS correctly identifies the single best by eliminating all other systems *before* the other termination criteria of each contender accumulating 1,000Δ observations. We observe a false elimination rate of exactly 0. The estimated EFER suggests that on these relatively modest problems, bi-PASS is conservative.

Adding worker elimination reduces the total effort required compared with master-only elimination because systems are inspected more often. Worker elimination decreases total effort by about 4.8% under exponential-like means and 7.6% under log-like means, suggesting

Figure 1. (Color online) Configurations of Means and Variances



Notes. (a) Three cases for true means. (b) Three cases for true variances.

that worker elimination is more helpful when there are many high-performing systems. Frequently inspecting better contenders for elimination is more helpful than frequently inspecting inferior ones, because inferior contenders generally stay far below the estimated standard and are easier to eliminate anyway. In all cases, on average, adding worker elimination decreases total effort by 6.1% compared with master-only elimination. In our tests, worker elimination does not appear to violate the EFER, likely because of an adequate distance between true means of better systems in all configurations.

The impact of pooled scaling depends on the distribution of variances. Under decreasing variances, pooled scaling increases total effort by about 29% relative to custom scaling. Here, it is harder to eliminate better systems because their centered partial sum statistics use an overly conservative scaling factor that is greater than their true variances. Under increasing variances, pooled scaling decreases total effort by about 19% relative to custom scaling because of better systems having higher true variances than the scaling factor and thus being easier to eliminate. One downside of pooled scaling is that good systems are at a higher risk for getting falsely eliminated if their scaling factors are less than their true variances, although we do not observe this behavior in our tests. On average across all cases, we find little difference in total effort between the two scaling factors.

Based on these results, we adopt master-only elimination and custom scaling factors in Section 8.2 because of their lower false elimination risk and because they do not use any a priori information on the distribution of means and variances.

8.2. Large-Scale Project Planning Problem Experiments

This section illustrates empirical bi-PASS performance on a large-scale problem with 1,184,040 feasible solutions (systems). The problem is to reduce the time to complete a stochastic activity network (SAN) based on a real-world model of U.S. Food and Drug Administration (FDA) research planning (Kwak and Jones

Table 1. Average Total Replications Across All Systems at Trial Termination, with Standard Errors Based on 1,000 Trials, and Termination Occurring When There Is a Single Contender Remaining or Each Contender Has at Least 1,000Δ Observations, Whichever Comes First

	Master-only elimination, custom scaling				Master and worker elimination, custom scaling		
	Decreasing var	Constant	Increasing		Decreasing var	Constant	Increasing
Log means	10.4e4 ± 1e3	15.8e4 ± 1e3	21.3e4 ± 2e3	Log means	9.1e4 ± 6e2	1.5e5 ± 1e3	2.0e5 ± 2e3
Linear	8.0e4 ± 9e2	8.6e4 ± 7e2	9.3e4 ± 7e2	Linear	7.8e4 ± 5e2	8.2e4 ± 5e2	8.7e4 ± 5e2
Exp	6.1e4 ± 6e2	6.2e4 ± 7e2	6.4e4 ± 8e2	Exp	5.8e4 ± 5e2	5.9e4 ± 5e2	6.0e4 ± 5e2
	Master-only elimination, pooled scaling				Master and worker elimination, pooled scaling		
	Decreasing var	Constant	Increasing		Decreasing var	Constant	Increasing
Log means	1.6e5 ± 8e2	1.6e5 ± 1e3	1.5e5 ± 1e3	Log means	1.5e5 ± 4e2	1.4e5 ± 7e2	1.4e5 ± 1e3
Linear	9.0e4 ± 8e2	8.6e4 ± 7e2	8.6e4 ± 7e2	Linear	8.1e4 ± 5e2	8.1e4 ± 6e2	8.1e4 ± 5e2
Exp	6.2e4 ± 7e2	6.2e4 ± 7e2	6.2e4 ± 7e2	Exp	5.9e4 ± 5e2	6.0e4 ± 5e2	5.9e4 ± 5e2

Note. All sample paths observed terminate with one contender remaining and no false eliminations.

1978). Paths through the network connect milestones (nodes) via research activities (arcs), and activities must be completed in sequence. Each activity has an uncertain completion time. The project begins at an initial milestone and completes when the last milestone is reached, so that the overall project completion time is the time required to complete the longest path through the network. See section 3.4 in Nelson and Pei (2021) for more background on SANs.

In the FDA problem there are 28 activities, and 7 of them can be chosen for a "speed up" by investing additional resources; the choice of 7 was ours to obtain a problem of roughly one million options. The goal is to find the seven activities that minimize the expected (mean) value of project completion time. Using the negative of the project completion time to transform the problem to maximization, the objective becomes

$$\max_{x_1,\ldots,x_{28}} \mathrm{E}\left(\max_{j\in\{1,\ldots,12\}} -\sum_{i\in \mathsf{P}_j} A_i(x_i)\right)$$

$$\text{s.t.} \sum_{i=1}^{28} x_i = 7,$$

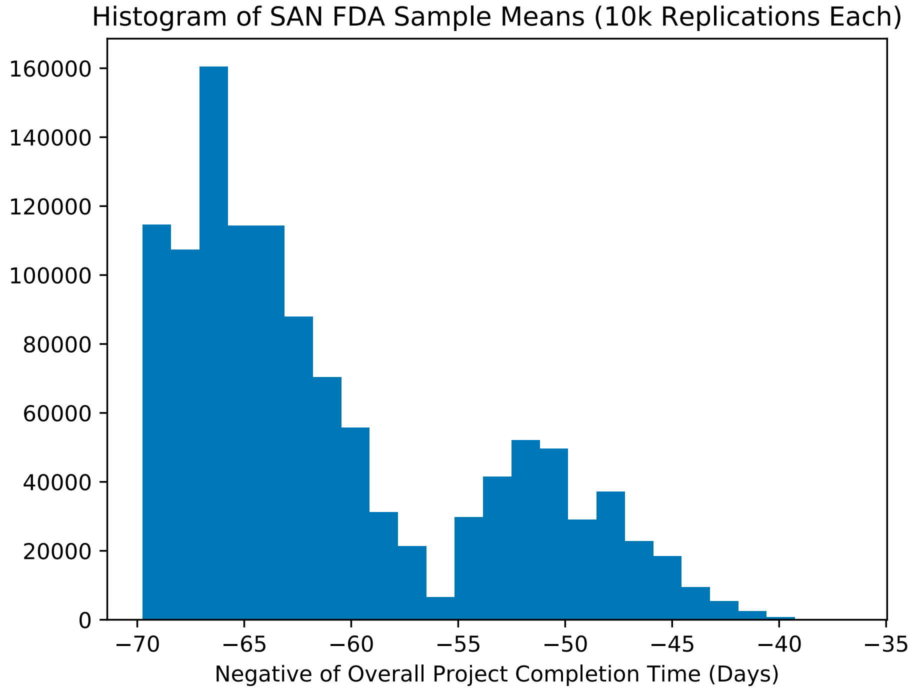
where $x_i \in \{0,1\}$ for $i = 1,2,\ldots,28$, P_j is a set of indices of activities on path j, and $A_i(x_i)$ is activity i's stochastic completion time. We use the same network as Kwak and Jones (1978), which has 12 paths, detailed in their paper.

The default time for activity i is $A_i(0)$ days, having a Gamma distribution with mean given in Kwak and Jones (1978). A few activities were simply logical connections with a duration of zero in the original problem, so we made them Gamma distributed with mean 0.1 day for our version. Setting $x_i = 1$ represents allocating additional resources to activity i. The reduced time for activity i is $A_i(1)$, having a Gamma distribution with mean equal to half of the mean of $A_i(0)$. For both $A_i(0)$ and $A_i(1)$ their coefficient of variation is 0.1.

Figure 2 displays a histogram of the estimated expected project completion time for all $k = 1,184,040$ feasible solutions, based on an expensive simulation of 10,000 i.i.d. replications per solution. The estimated difference between the best and second-best system is 0.1 days, and the best system has a sample mean of 36.61 days estimated with a standard error less than 0.01.

The FDA problem underscores the usefulness of R&S in general, and bi-PASS in particular, for situations that are untenable for other methods. The FDA problem is not amenable to a continuous approximation, has a highly nonlinear objective function, and lacks an exploitable spatial structure. Although it is a network, the network structure cannot be exploited in a response surface or adaptive random search method because even "close" systems can have distinctly different objective function values because of having different longest paths. Also, because dimension is irrelevant to bi-PASS, our parametrization of decision variables

Figure 2. (Color online) Histogram of FDA SAN Sample Means for All $k = 1,184,040$ Feasible Solutions



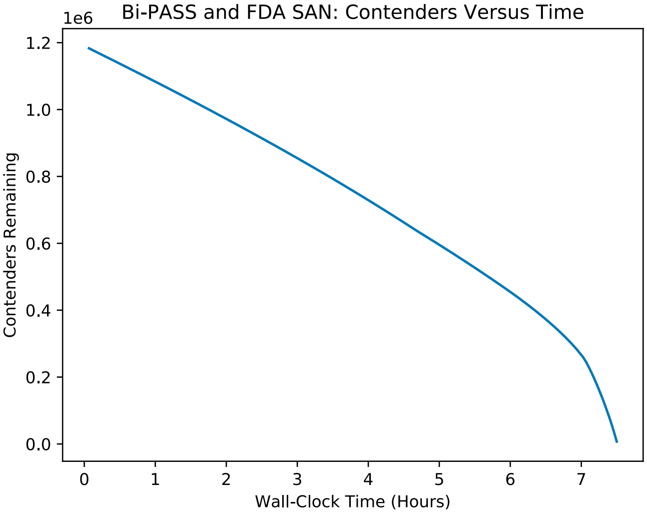
has the same difficulty as a problem with, say, a thousand binary variables.

The experiment design is as follows: We use $p+1=21$ processors, initial sample size $n_0=10$, common run length $\Delta=100$ replications, EFER parameter $\alpha=0.05$, master-only elimination, and custom scaling factors, based on our experiments in the previous section. The boundary function in Section 7 is used with the same constant $c_{\alpha,n_0-1}=8.6$ as in Section 8.1. The constant is the same because α and n_0-1 are the same, even though here we have more than a million additional systems. This underscores a large-scale advantage of bi-PASS: the statistical guarantee is on elimination *rate*, so the boundary does not depend on the number of systems.

Ni et al. (2017) exhibit a Spark version of the GSP solving a 1 million system throughput maximization problem with 480 processors. Because we only use 21 processors in our experiment, we cannot test GSP due to the prohibitive number of operations required. GSP divides systems into "screening groups" distributed across workers, which each require $O((k/p)^2)$ pairwise comparisons—on the order of a billion comparisons—for the first round alone. The need for a relatively small screening group is a significant bottleneck for applying procedures based on pairwise comparisons in large-scale problems, an issue bi-PASS avoids. Zhong and Hong (2022) also use 480 processors for the 1 million system problem.

Figure 3 displays the average number of contenders remaining versus wall-clock time in hours, recorded every additional 1 million total replications, averaged over 10 independent trials of bi-PASS. The contenders-versus-time trajectory traces the progress that bi-PASS makes: As each run progresses, more systems are eliminated and the estimated standard increases. In this FDA problem, the rate of eliminations per hour increases toward the end. We also point out that the contenders-versus-time trajectory is similar across trials: at each recording increment, the standard error of the number of contenders was less than 5% of the mean, and the standard error of the time was less than 0.5% of the mean.

Figure 3. (Color online) Number of Contenders vs. Wall-Clock Time in Hours, Averaged Across 10 i.i.d. Bi-PASS Trials with 20 Workers on the FDA SAN Problem, Recorded Every 1 Million Total Replications



In all trials, bi-PASS eliminated all but the true best system, suggesting adherence to the EFER guarantee. On average, bi-PASS took 7.51 hours and consumed 114.4 million total replications. Both statistics have standard errors less than 0.1% of their means. The results from the FDA problem demonstrate that bi-PASS is a powerful tool, reliably retaining the single best system from 1.2 million systems with only 21 processors within 8 hours. Twenty-one processors is only slightly larger than the current eight-core-two-thread laptop computers that are available for a few hundred dollars.

9. Concluding Remarks

To explore the relationship between bi-PASS and classical R&S, we focus on the standard $\mu^\star=\mu_1$ here. However, we do not envision bi-PASS being used as a fixed-precision procedure that is run until a single system remains. Instead, we see bi-PASS as a fixed-budget screening step before doing something else, such as applying a standard R&S procedure or a search method to a smaller subset of systems. In this setting, an objective such as $\mu^\star=\mu_b$ for some $b>1$ is more appropriate. Using bi-PASS as a screening step is similar to a subset selection procedure (Nelson and Pei 2021) but with a different objective: retaining *all* of the top b systems not just the best.

For convenience, we adopt the lower-boundary function $-g(\cdot)$ from Fan et al. (2016); they use $\pm g(\cdot)$ to create an indifference-zone-free select-the-best R&S procedure. However, this boundary is much more conservative than necessary if we use bi-PASS as a fixed-budget procedure, so future work involves finding less conservative, budget-sensitive boundaries.

Brownian motion (normally distributed) output is a common starting assumption for many indifference-zone, select-the-best-mean R&S procedures. Such procedures can sometimes be shown to be asymptotically valid for more general output distributions (Kim and Nelson 2006a). The asymptotic regime drives the indifference zone parameter $\delta\rightarrow 0$ while also driving the difference between the best and second-best system to zero, which makes sense: harder problems require more output data. We believe that the version of bi-PASS presented here is somewhat more dependent on normality than these indifference-zone procedures in the sense that there is not a natural regime to drive large sample sizes (the procedure is built to preserve EFER even if run forever), and letting the number of systems $k \to \infty$ does not change the marginal distribution of each systems' output. Nevertheless, we observe substantial robustness to nonnormality in extensive experiments, which suggests that there is an appropriate asymptotic explanation for it.

Acknowledgments

The idea of PASS was introduced in Pei et al. (2018), which contained none of the theory, algorithms, or results provided here. The authors thank the associate editor and referees for thoughtful and challenging reviews.

References

Batur D, Kim S-H (2010) Finding feasible systems in the presence of constraints on multiple performance measures. *ACM Trans. Modeling Comput. Simulation* 20(3):13.

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Royal Statist. Soc. B* 57(1):289–300.

Bubeck S, Munos R, Stoltz G (2009) Pure exploration in multi-armed bandits problems. *Proc. Internat. Conf. on Algorithmic Learn. Theory* (Springer, Berlin), 23–37.

Desautels T, Krause A, Burdick JW (2014) Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *J. Machine Learn. Res.* 15:3873–3923.

Eckman DJ, Henderson SG (2018) Guarantees on the probability of good selection. *Proc. Winter Simulation Conf.* (IEEE, New York), 351–365.

Efron B (2012) *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*, vol. 1 (Cambridge University Press, Cambridge, UK).

Falkner S, Klein A, Hutter F (2018) BOHB: Robust and efficient hyperparameter optimization at scale. *Proc. Internat. Conf. on Machine Learn.* (PMLR), 1437–1446.

Fan W, Hong LJ, Nelson BL (2016) Indifference-zone-free selection of the best. *Oper. Res.* 64(6):1499–1514.

Frazier PI (2010) Decision-theoretic foundations of simulation optimization. *Wiley Encyclopedia of Operations Research and Management Sciences* (Wiley, New York).

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

Gittins JC (1979) Bandit processes and dynamic allocation indices. *J. Royal Statist. Soc. B* 41(2):148–164.

Goldsman L, Nelson BL (1990). Batch-size effects on simulation optimization using multiple comparisons with the best. *Proc. 1990 Winter Simulation Conf.* (IEEE, New York), 288–293.

Haupt J, Castro RM, Nowak R (2011) Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Trans. Inform. Theory* 57(9):6222–6235.

Heidelberger P (1988) Discrete event simulations and parallel processing: Statistical properties. *SIAM J. Sci. Statist. Comput.* 9(6): 1114–1132.

Hillel E, Karnin ZS, Koren T, Lempel R, Somekh O (2013) Distributed exploration in multi-armed bandits. *Advances in Neural Information Processing Systems*, vol. 26 (NIPS, Lake Tahoe, NV).

Hoffman M, Song E, Brundage M, Kumara S (2018) Condition-based maintenance policy optimization using genetic algorithms and Gaussian Markov improvement algorithm. Bregon A, Orchard M, eds. *Proc. Annual Conf. of the PHM Soc., vol. 10* (PHM Society, Philadelphia). https://papers.phmsociety.org/index.php/phmconf/issue/view/phm2018 and https://papers.phmsociety.org/index.php/phmconf/article/view/537.

Hunter SR, Nelson BL (2017) Parallel ranking and selection. Tolk A, Fowler J, Shao G, Yücesan E, eds. *Advances in Modeling and Simulation* (Springer, New York), 249–275.

Jamieson KG, Jain L (2018) A bandit approach to sequential experimental design with false discovery control. *Advances in Neural Information Processing Systems*, vol. 31 (NIPS, Montreal).

Jamieson K, Nowak R (2014) Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. *Proc. 48th Annual Conf. on Inform. Sci. and Systems* (IEEE, New York), 1–6.

Jamieson K, Talwalkar A (2016) Non-stochastic best arm identification and hyperparameter optimization. *Artificial Intelligence and Statistics* (PMLR), 240–248.

Kandasamy K, Krishnamurthy A, Schneider J, Poczos B (2017) Asynchronous parallel Bayesian optimisation via Thompson sampling. Preprint, submitted May 25, https://arxiv.org/abs/1705.09236.

Karnin Z, Koren T, Somekh O (2013). Almost optimal exploration in multi-armed bandits. *Proc. Internat. Conf. on Machine Learn.* (PMLR), 1238–1246.

Kim S-H (2005) Comparison with a standard via fully sequential procedures. *ACM Trans. Modeling Comput. Simulation* 15(2):155–174.

Kim S-H, Nelson BL (2006a) On the asymptotic validity of fully sequential selection procedures for steady-state simulation. *Oper. Res.* 54(3):475–488.

Kim S-H, Nelson BL (2006b) Selecting the best system. Henderson S, Nelson BL, eds. *Handbooks in Operations Research and Management Science* (Elsevier, New York), 501–534.

Kwak NK, Jones L (1978) An application of PERT to R&D scheduling. *Inform. Processing Management* 14(2):121–131.

L'Ecuyer P, Simard R, Chen EJ, Kelton WD (2002) An object-oriented random-number package with many long streams and substreams. *Oper. Res.* 50(6):1073–1075.

Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A (2017) Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Machine Learn. Res.* 18(1):6765–6816.

Luo J, Hong LJ, Nelson BL, Wu Y (2015) Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. *Oper. Res.* 63(5):1177–1194.

Nelson BL, Goldsman D (2001) Comparisons with a standard in simulation experiments. *Management Sci.* 47(3):449–463.

Nelson BL, Pei L (2021) *Foundations and Methods of Stochastic Simulation: A First Course*, 2nd ed. (Springer, New York).

Nelson BL, Swann J, Goldsman D, Song W (2001) Simple procedures for selecting the best simulated system when the number of alternatives is large. *Oper. Res.* 49(6):950–963.

Ni EC, Ciocan DF, Henderson SG, Hunter SR (2017) Efficient ranking and selection in parallel computing environments. *Oper. Res.* 65(3):821–836.

Pasupathy R, Ghosh S (2013) Simulation optimization: A concise overview and implementation guide. *INFORMS Tutorials in Operations Research* (INFORMS), 122–150.

Pei L, Nelson BL, Hunter S (2018) A new framework for parallel ranking & selection using an adaptive standard. *Proc. Winter Simulation Conf.* (IEEE), 2201–2212.

Pei L, Nelson BL, Hunter S (2020) Evaluation of bi-PASS for parallel simulation optimization. *Proc. Winter Simulation Conf.* (IEEE), 2960–2971.

Resnick SI (1992) *Adventures in Stochastic Processes* (Springer, New York).

Russo D (2020) Simple Bayesian algorithms for best-arm identification. *Oper. Res.* 68(6):1625–1647.

Russo D, Van Roy B (2022) Satisficing in time-sensitive bandit learning. *Math. Oper. Res.*, ePub ahead of print March 14, https://doi.org/10.1287/moor.2021.1229.

Salemi P, Song E, Nelson BL, Staum J (2019) Gaussian Markov random fields for discrete optimization via simulation: Framework and algorithms. *Oper. Res.* 67(1):250–266.

Semelhago M, Nelson BL, Song E, Wächter A (2021) Rapid discrete optimization via simulation with Gaussian Markov random fields. *INFORMS J. Comput.* 33(3):915–930.

Singham DI, Szechtman R (2016) Multiple comparisons with a standard using false discovery rates. *Proc. Winter Simulation Conf.* (IEEE), 501–511.

Szechtman R, Yücesan E (2008) A new perspective on feasibility determination. *Proc. Winter Simulation Conf.* (IEEE), 273–280.

Villar SS, Bowden J, Wason J (2015) Multi-armed bandit models for the optimal design of clinical trials: Benefits and challenges. *Statist. Sci.* 30(2):199.

Whittle P (1988) Restless bandits: Activity allocation in a changing world. *J. Appl. Probability* 25:287–298.

Xie J, Frazier PI (2013) Sequential Bayes-optimal policies for multiple comparisons with a known standard. *Oper. Res.* 61(5): 1174–1189.

Zhong Y, Hong LJ (2022) Knockout-tournament procedures for large-scale ranking and selection in parallel computing environments. *Oper. Res.* 70(1):432–453.

Zhong J, Huang Y, Liu J (2017). Asynchronous parallel empirical variance guided algorithms for the thresholding bandit problem. Preprint, submitted April 15, https://arxiv.org/abs/1704.04567.

Linda Pei is conducting postdoctoral research at Northwestern University applying simulation optimization techniques to real-world COVID social distancing policies. Her research interests include large-scale parallel simulation optimization and simulation analytics.

Barry L. Nelson is the Walter P. Murphy professor of the Department of Industrial Engineering and Management Sciences at Northwestern University. His research focus is on the design and analysis of computer simulation experiments on models of discrete-event, stochastic systems. He is a Fellow of INFORMS and IIE.

Susan R. Hunter is an associate 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.