Foundations of Ranking & Selection for Simulation Optimization

Barry L. Nelson

Abstract In addition to his voluminous and profound research accomplishments, Pierre L'Ecuyer is an extraordinary educator; this includes expository talks and papers, especially in the area of pseudorandom-number generation. This paper is written in that same spirit, covering the foundations of ranking & selection for simulation optimization; simulation optimization is also an area of exceptional accomplishment for Pierre.

1 Introduction

Suppose that we have the ability to simulate $k = 4$ different system designs that use redundancy to be resistant to system failure. Let $Y(x)$ be the time to failure of design type $x = 1, 2, 3, 4$. Your job, as the analyst, is to use the simulation to find $x^\star = \text{argmax}_x \text{E}[Y(x)]$, the system design leading to the largest mean time to failure. How would you do this?

The field of ranking & selection (R&S) provides procedures that "solve" problems of this type. Features we might like in a R&S procedure include controlling the number of simulation replications automatically; providing statistical guarantees of correctness; being appropriate for large as well as small numbers of systems, k; the facility to exploit modern parallel computing; and to do all of this computationally and statistically efficiently.

The field of *simulation optimization* (SO)—of which R&S is a part—attacks stochastic optimization problems in which the objective function is some property of the output of a stochastic, often dynamic and nonstationary, simulation. Critically, the property of interest can only be estimated by simulating instances (feasible solutions, system designs), and those simulations may be computationally expensive. All SO algorithms are subject to three

Barry L. Nelson
Northwestern University, Evanston, IL USA, e-mail: nelsonb@northwestern.edu

sources of error: They may fail to simulate the optimal solution; they may fail to recognize the best solution that was simulated; and they may report an optimistic (biased) estimate of the performance of the solution that they do select. R&S is the only class of SO algorithms that controls all three sources of error, but at the cost of simulating *all* system designs: R&S procedures are exhaustive SO algorithms designed specifically to control statistical error.

R&S originated with Robert Bechhofer (Cornell) and Shanti Gupta (Purdue) in the 1950s to address biostatistics problems such as finding the most efficacious of three drug treatments and a placebo. See Bechhofer et al. (1995) and Gupta and Panchapakesan (2002). The problem characteristics assumed by early R&S procedures include a small number of treatments, k; normally distributed responses; relatively equal (maybe even known) variances; and a requirement to be easy to implement, for instance by applying treatments to batches of subjects rather than sequentially (e.g., one subject at a time and waiting for the results before deciding the next treatment to apply).

At the 1983 Winter Simulation Conference David Goldsman (Georgia Tech) presented a tutorial on R&S (Goldsman, 1983), and organized a session with Bechhofer and Gupta, arguing that R&S was useful for optimizing simulated systems. The simulation community quickly embraced this paradigm, but had more expansive objectives than the founders, including much larger numbers of "treatments" (simulated system designs) k; non-normal (nominal) output data; significantly unequal variances across systems; and intentionally induced dependence across systems due to the use of common random numbers. In addition, since data are generated by computer simulations that are easily controlled, simulation researchers and practitioners were not concerned with how complex or sequential the R&S procedure is as long as it is effective (selects the best system design) and computationally efficient (generates as few simulation replications as possible, since the simulation was assumed to be more computationally expensive than the overhead of the R&S procedure).

R&S has been a theoretical and practical success for simulation: There is supporting theory, including asymptotic regimes for non-normal data and effective use of "statistical learning." Further, R&S has been routinely applied to real problems, partly because R&S procedures are included in commercial simulation software. Of course there is a R&S problem-size limit, since all system designs must be simulated. Therefore, much of the research effort in R&S for simulation has been dedicated to extending this limit via enhanced statistical efficiency to reduce simulation effort and parallel computing to speed up execution. See Kim and Nelson (2006b) and Chen et al. (2015) for earlier surveys.

This paper is a significant extension of Nelson (2018), and a companion to the online masterclass "Ranking & Selection for Simulation Optimization" at `http://users.iems.northwestern.edu/~nelsonb/RSMasterclass.html`. The web site contains `R` code for all of the R&S procedures described here along with slides, videos and self-paced exercises supporting this tutorial. The purpose of the masterclass and this paper is to

present foundations and broad themes in R&S for SO, rather than details or new results. In Section 2 we set up the R&S problem. Section 3 describes the "normal means" case, the most widely studied and solved R&S problem. Exploiting parallel computing in R&S is discussed in Section 4. Some formulations beyond normal means are presented in Section 5. Finally, in Section 6 we briefly contrast R&S with the related field of multi-armed bandits.

2 Set Up

For much of the paper the following set up applies. The true system performance parameters (which are unknown) are $\mu(1) \leq \mu(2) \leq \cdots \leq \mu(k-1) \leq \mu(k)$, and we refer to system k, or any system tied with system k, as "the best." For system x we can estimate $\mu(x)$ with a consistent estimator; for instance, when $\mu(x)$ is the expected value we may employ the sample mean of $n(x)$ replications:

$$\bar{Y}(x) = \frac{1}{n(x)} \sum_{j=1}^{n(x)} Y_j(x)$$

where $Y_j(x)$ is the jth independent and identically distributed (i.i.d.) replication from system design x. We will focus on selecting the best mean, but consider other performance measures in Sections 5–6. The R&S procedure ultimately returns something like $\widehat{x}^\star = \text{argmax}_{x \in \{1,2,\ldots,k\}} \bar{Y}(x)$ as the selected system. We consider two categories of objectives for the R&S procedure:

Fixed Precision:

Simulate until a prespecified level of inference is achieved, ideally a probability of correct selection (PCS), defined as $\Pr\{\widehat{x}^\star = k\} \geq 1 - \alpha$. Since this can be computationally impossible, for instance if there are ties for the best, a compromise such as one of the following is accepted, where $\delta > 0$ is a user-specified parameter:

- **Indifference zone:** $\text{PCS} = \Pr\{\widehat{x}^\star = k \mid \mu(k) - \mu(k-1) \geq \delta\} \geq 1 - \alpha$, where "$\mid \mu(k) - \mu(k-1) \geq \delta$" indicates that the guarantee is only for problems in which the means satisfy this inequality. That is, the best system is highly likely to be selected when there is at least a minimum separation between the best and second-best system.
- **Good selection:** $\text{PGS} = \Pr\{\mu(k) - \mu(\widehat{x}^\star) \leq \delta\} \geq 1 - \alpha$. That is, a system with no more than a specified optimality gap is highly likely to be selected.
- **Top m:** $\Pr\{\widehat{x}^\star \in [k, k-1, \ldots, k-m+1]\} \geq 1 - \alpha$. That is, one of the m best systems is highly likely to be selected.

- **Subset:** Find $\widehat{\mathcal{S}} \subseteq \{1, 2, \ldots, k\}$ such that $\Pr\{k \in \widehat{\mathcal{S}}\} \geq 1 - \alpha$. That is, a subset (ideally small) is returned that is highly likely to contain the best system.

These are typically *frequentist* guarantees to be achieved as efficiently as possible.

Fixed Budget:

Obtain as strong an inference as possible within a given computation budget, often formulated as minimizing some expected loss for the chosen system design, $\mathrm{E}[\mathcal{L}(\widehat{x}^\star, k)]$:

- **0-1 Loss:** Minimize the posterior probability of incorrect selection, $\Pr\{\widehat{x}^\star \neq k|\mathcal{H}\}$.
- **Opportunity cost:** Minimize the posterior expected optimality gap, $\mathrm{E}[\mu(k) - \mu(\widehat{x}^\star)|\mathcal{H}]$.

The inference is typically *Bayesian* in nature, and $\mathcal{H}$ includes the entire history of simulation runs performed and outputs obtained until the budget is exhausted. We will consider both fixed-precision and fixed-budget perspectives in this chapter.

3 The Normal Means Case

The most widely studied case assumes that from system x we can obtain $Y_1(x), Y_2(x), \ldots$ that are i.i.d. normally distributed with mean $\mu(x)$ and variance $\sigma^2(x)$, denoted $\mathrm{N}(\mu(x), \sigma^2(x))$. Further it may be possible to induce $\mathrm{Cov}(Y(x), Y(x')) \neq 0$ if we use common random numbers. Since so much research effort has been expended on this problem, it is reasonable to ask, is normally distributed output actually relevant for simulation problems? Fortunately, the answer is frequently "yes." Each output Y is often the average of *many* more basic outputs, e.g., daily average customer waiting time is the average of many individual customers' waiting times. Also, the sample sizes prescribed by R&S procedures are often large, so we can group or "batch" outputs to obtain approximate normality. And many normal-means procedures are asymptotically valid for non-normal data, as discussed in Section 3.8.

Initially we will assume that we can only simulate one system at a time, and then later we parallelize simulations. One-system-at-a-time procedures are often observation-efficient, but may not be computationally efficient in parallel.

3.1 The Indifference-zone (IZ) Formulation

One of the most well-known IZ procedures is due to Rinott (1978):
Rinott's Procedure

1. Choose confidence level $1-\alpha$, initial sample size $n_0 \geq 2$ and indifference zone parameter $\delta > 0$. Set $h = h(k, 1-\alpha, n_0)$, a constant that depends on the number of systems, desired confidence level and the initial sample size.
2. For each system $x = 1, 2, \ldots, k$ do the following:
   a. Simulate n_0 replications and compute the sample variance $S^2(x)$.
   b. Compute $N(x) = \left\lceil \frac{h^2 S^2(x)}{\delta^2} \right\rceil$
   c. Simulate $\max\{0, N(x) - n_0\}$ additional replications from system x.
   d. Compute the sample mean of all $N(x)$ replications, $\bar{Y}(x)$.
3. Choose $\widehat{x}^\star = \text{argmax}_x \bar{Y}(x)$.

Rinott's procedure assumes that the outputs are i.i.d. normally distributed, have unknown and possibly unequal variances, and are independent across systems. The last assumption implies using distinct random number seeds for each system's simulation. Rinott guarantees

$$\text{PCS} = \Pr\{\widehat{x}^\star = k \mid \mu(k) - \mu(k-1) \geq \delta\} \geq 1 - \alpha.$$

Below we will outline how Rinott-like procedures provide this guarantee. The parameter δ is often interpreted as the "smallest practically significant difference."

Rinott is easy to implement, and because it requires no coordination among systems it is easy to parallelize. However, it is pessimistic: it assumes the means are in the "slippage configuration" $\mu(1) = \mu(2) = \cdots = \mu(k-1) = \mu(k) - \delta$. This pessimisim leads to more simulation than necessary to achieve the desired PCS for many problems in which the means are more favorably spaced. What happens if there are other good (closer than δ) systems? It turns out that Rinott also has a $1-\alpha$ good selection guarantee, which means selecting a system within δ of the best; this happy fact was not known until more recently (Nelson and Matejcik, 1995).

Notice that the sample size $N(x)$ grows as h^2/δ^2. How does $h(k, 1-\alpha, n_0)$ grow with the number of systems k? Answer: too fast to be practical for really large k, so other strategies (described later in this section) are needed for that case.

Rinott-like procedures achieve their guarantee based on some version of the following argument. Since we assume $\mu(k) - \mu(x) \geq \delta$, $x \neq k$, we have

$$
\begin{aligned}
&\Pr\left\{\bar{Y}(k) > \bar{Y}(x)\right\} \\
&= \Pr\left\{\bar{Y}(k) - \bar{Y}(x) > 0\right\} \\
&= \Pr\left\{\bar{Y}(k) - \bar{Y}(x) - [\mu(k) - \mu(x)] > -[\mu(k) - \mu(x)]\right\} \\
&\geq \Pr\left\{\bar{Y}(k) - \bar{Y}(x) - [\mu(k) - \mu(x)] > -\delta\right\}.
\end{aligned}
$$

The statistic $\bar{Y}(k) - \bar{Y}(x) - [\mu(k) - \mu(x)]$ has mean 0, so we can find the number of replications needed to provide the desired probability guarantee considering only δ and the variances.

This formulation—where we want PCS $\geq 1-\alpha$ when $\mu(k)-\mu(x) \geq \delta$ and we assume the slippage configuration—has been dominant in frequentist R&S because it frees the probability statement from dependence on the true means. There are two challenges: When $\mu(k) - \mu(x) \gg \delta$ the slippage assumption does not exploit it to gain efficiency, which is particularly critical when k is large. And when $\mu(k) - \mu(x) < \delta$ for some inferior system x, we would like a "good selection" guarantee, which Rinott provides, but this is not the case for all IZ procedures; see Section 3.6.

3.2 R&S Based on "Statistical Learning"

The following ideas for R&S are based (formally or informally) on Bayesian reasoning. See Frazier (2012) for a more complete tutorial.

Frequentist reasoning goes like this: $\mu(1), \mu(2), \ldots, \mu(k)$ are *fixed* performance measures and probability statements (e.g., PCS, PGS) are with respect to repeated independent experiments on the same problem. Bayesian reasoning starts from the premise that we have uncertainty about the problem itself (e.g., which system is the best) that we characterize via a prior probability distribution, and we then reduce our uncertainty by running simulation experiments and updating our prior distribution to a (more informative) posterior (after experiment) distribution using Bayes' rule. Typically the experiment-then-posterior-updating cycle is done repeatedly for many iterations.

In R&S our prior on the true means, and perhaps additional aspects, of the problem is

$$
\underbrace{\mu(1), \ldots, \mu(k)}_{\text{your problem}} \qquad \sim \qquad \underbrace{M(1), \ldots, M(k).}_{\text{r.v.'s with a "prior" distribution}}
$$

After observing $(x, Y_j(x))$, we update our knowledge based on the conditional ("posterior") distribution of $[M(1), \ldots, M(k)]$ given the entire history, denoted by $\mathcal{H}$. A generic, fixed-budget, Bayesian R&S procedure is given

below. In this procedure $x^{(j)}$ denotes the system we choose to simulate on iteration j of the procedure.

Generic Bayesian R&S

1. For $x \in \{1, 2, \ldots, k\}$, set $n(x) = 0$, $\bar{Y}(x) = \text{null}$, $\mathcal{H}_0 = \emptyset$, $j = 0$.
2. $x^{(j)} = \pi(\mathcal{H}_j)$ and simulate $Y_{j+1}(x^{(j)})$ [policy $\pi(\cdot)$ based on the posterior distribution].
3. Update $n(x^{(j)}) = n(x^{(j)}) + 1$ and $\bar{Y}(x^{(j)}) = \dfrac{1}{n(x^{(j)})} \displaystyle\sum_{i:x^{(i)}=x^{(j)}} Y_{i+1}(x^{(i)})$ $\mathcal{H}_{j+1} = \mathcal{H}_j \cup \left\{(x^{(j)}, Y_{j+1}(x^{(j)}))\right\}$.
4. If the budget is exhausted then return $\widehat{x}^\star = \text{argmax}_x \bar{Y}(x)$, otherwise $j = j + 1$ and go to 2.

Clearly the key aspect is the policy $\pi(\cdot)$. Often the policy is expressed as some sort of "acquisition function" a, for instance

$$\pi(\mathcal{H}) = \text{argmax}_{x \neq \widehat{x}^\star}\ a(x, \widehat{x}^\star) = \text{argmax}_{x \neq \widehat{x}^\star}\ \mathrm{E}\left[\max\left\{0, M(x) - M(\widehat{x}^\star)\right\} \middle| \mathcal{H}\right] \tag{1}$$

which is the system design with the largest posterior expected value of improvement over the current sample best. Ideally a is chosen to learn "optimally," meaning as efficiently as possible, but the policy also has to be computable, which often means it cannot look too many steps ahead.

Gaussian processes provide a very useful framework for this sort of approach, often based on two fundamental results:

1. If $Z \sim \mathrm{N}(0,1)$ then $\mathrm{E}\left[\max\{0, \mu + \sigma Z\}\right] = \mu\Phi\left(\dfrac{\mu}{\sigma}\right) + \sigma\phi\left(\dfrac{\mu}{\sigma}\right)$ where Φ and ϕ are the cdf and density of Z, respectively.
2. If $(Z_1, Z_2) \sim \mathrm{BVN}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$ then $Z_1 \sim \mathrm{N}(\mu_1, \sigma_1^2)$, and

$$Z_1 | Z_2 = z \sim \mathrm{N}\underbrace{\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(z - \mu_2),\ \sigma_1^2(1 - \rho^2)\right)}_{\text{"learning"}}.$$

The acquisition function in (1) is known as the *complete expected improvement* (CEI) policy (Salemi et al., 2019). When the posterior is the normal distribution, then using the first fundamental fact we have

$$\begin{aligned}\mathrm{CEI}(x, \widehat{x}^\star) = (m(x) - m(\widehat{x}^\star))\Phi\left(\frac{m(x) - m(\widehat{x}^\star)}{\sqrt{\mathrm{Var}(x, \widehat{x}^\star)}}\right) \\ + \sqrt{\mathrm{Var}(x, \widehat{x}^\star)}\phi\left(\frac{m(x) - m(\widehat{x}^\star)}{\sqrt{\mathrm{Var}(x, \widehat{x}^\star)}}\right)\end{aligned}$$

where $m(x) = \mathrm{E}(M(x)|\mathcal{H})$, $\mathrm{Var}(x, \widehat{x}^\star) = \mathrm{Var}(M(x) - M(\widehat{x}^\star)|\mathcal{H})$. The second fact can be exploited to compute the means and variances, conditional on $\mathcal{H}$.

The CEI policy has been shown empirically to make rapid progress toward the best system.

3.3 A Convergence-rate Perspective

Suppose that the best system is unique: $\mu(k) > \mu(k-1)$. Then as long as all the $n(x) \to \infty$, even if not all equal, we will eventually correctly select $\widehat{x}^\star = k$ due to the strong law of large numbers. But what is the best way to get to ∞? For the purposes of this section it will be useful to employ the notation $\bar{Y}_x(n(x))$ for the sample mean of $n(x)$ replications from system x, $\mu_x = \mu(x)$ and $\sigma_x = \sigma(x)$, and further to let $n(x) = \beta_x N$ where $\beta_x \geq 0$, $\sum_x \beta_x = 1$ and N is the total replication budget. The question then becomes, what choice of $\beta_1, \beta_2, \ldots, \beta_k$ makes $\lim_{N\to\infty} \Pr\{\widehat{x}^\star \neq k\}$ go to 0 the fastest?

One way to answer this question is via a large-deviation principle (LDP). Let $Z_1, Z_2, \ldots, Z_N$ be i.i.d. (μ, σ^2). If Z has finite log moment generating function, then for $z > \mu$

$$\lim_{N\to\infty} \frac{1}{N} \ln[\Pr\{\bar{Z}(N) > z\}] = -I(z)$$

where $I(\cdot)$ is a *rate function* that depends on the distribution of Z. This LDP can be interpreted as

$$\Pr\{\bar{Z}(N) > z\} \approx e^{-NI(z)} \text{ for large } N.$$

Translating to R&S, we want to choose $\beta_1, \beta_2, \ldots, \beta_k$ to maximize the smallest of the rates of decay of the pairwise probabilities of incorrect selection (PICS)

$$\text{PICS}_x = \Pr\{\bar{Y}_x(\beta_x N) - \bar{Y}_k(\beta_k N) > 0\} \approx \exp(-NI(0, \beta_x, \beta_k))$$

where $I(0, \beta_x, \beta_k)$ indicates that the rate function depends on the allocation β_x, β_k. Glynn and Juneja (2004) showed that if the outputs are normally distributed then the LDP rate-optimal allocation satisfies

$$\left(\frac{\beta_k}{\sigma_k}\right)^2 = \sum_{x\neq k} \left(\frac{\beta_x}{\sigma_x}\right)^2$$

$$\frac{(\mu_x - \mu_k)^2}{\frac{\sigma_x^2}{\beta_x} + \frac{\sigma_k^2}{\beta_k}} = \frac{(\mu_{x'} - \mu_k)^2}{\frac{\sigma_{x'}^2}{\beta_{x'}} + \frac{\sigma_k^2}{\beta_k}}, \quad \forall x, x' \neq k.$$

Unfortunately, this expression involves quantities that we do not know, and just plugging in estimates does not give the best possible rate (things get harder for unknown distributions because estimating LDP rates is difficult).

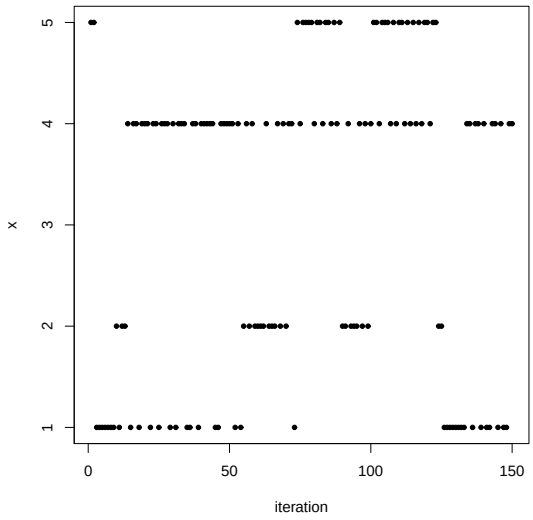

Fig. 1 Illustration of allocations from mCEI in a R&S problem with $k = 5$ systems. Each • represents a replication.

Fortunately, Chen and Ryzhov (2019) showed that a slight modification of the CEI policy from the previous section, called mCEI, is asymptotically equivalent to the rate-optimal allocation! This result is remarkable because CEI comes from unrelated reasoning: the Bayes-optimal allocation of the next simulation run if that run will be your last. Figure 1 illustates the mCEI procedure's allocation in a five-system problem over 150 iterations. Notice that system $x = 4$, which is the true best, also receives the most replications.

Another popular policy that is in the same spirit as mCEI is optimal computer budget allocation (OCBA), which is derived through a Bayesian-inspired approximation to the posterior PCS. OCBA uses plug-in estimates and nonlinear optimization to allocate batches of replications. Although it does not achieve the rate-optimal allocation in the limit, it is quite effective empirically. See Chen and Lee (2011).

3.4 Doing Better than "Rate Optimal"

The asymptotically optimal allocation focuses on the endgame, as sample sizes get large, and is not necessarily the best allocation for *finite* N. After all, we do not need to drive PICS to 0 to be highly confident of selecting

the best. Further, in the rate-optimal allocation all $\beta_x > 0$, which means that all systems remain in play until we stop, which may imply a lot of computational overhead on each step, especially if k is large. Also, the rate-optimal allocation does not provide a way to do fixed-precision stopping. And finally, one-system-at-a-time allocation is becoming less and less attractive as it becomes easier and easier to simulate $p > 1$ systems or replications in parallel.

Often (especially when k is large) there are many bad systems we can completely eliminate from further consideration quickly. This is one way to beat rate-optimal for finite N. There are two basic strategies:

- **Screen & select:** Get a small number of replications from all system designs, create a subset $\widehat{\mathcal{S}}$ that still contains the best, then apply an efficient R&S procedure to the remainder. This usually requires splitting the α error between subset and selection so that $\Pr\{k \in \widehat{\mathcal{S}}\} \geq 1 - \alpha/2$.
- **Continuous screening:** Iteratively replicate, eliminate, replicate, eliminate and so on until one system remains. This usually requires tracking all pairwise comparisons and controling the overall error via (for instance) the Bonferroni inequality. But even for a single pairwise comparison we need results that allow "multiple looks" at the data for continuous screening.

3.4.1 Screening

We begin with a basic subset selection (screening) procedure from Nelson et al. (2001) for systems simulated independently:

Basic subset selection

1. Simulate $n(x) \geq 2$ replications from system x, set $t(x) = t_{(1-\alpha)^{\frac{1}{k-1}},\ n(x)-1}$ the $(1-\alpha)^{\frac{1}{k-1}}$ quantile of the t distribution with $n(x) - 1$ degrees of freedom, for $x = 1, 2, \ldots, k$.
2. Calculate the sample means $\bar{Y}(x)$ and sample variances

$$S^2(x) = \frac{1}{n(x) - 1} \sum_{j=1}^{n(x)} \left(Y_j(x) - \bar{Y}(x)\right)^2$$

   for $x = 1, 2, \ldots, k$, and also for all $x \neq x'$ compute

$$W(x, x') = \left(t(x)^2 \frac{S^2(x)}{n(x)} + t(x')^2 \frac{S^2(x')}{n(x')} \right)^{1/2}.$$

3. Form the subset

$$\widehat{\mathcal{S}} = \left\{ x \colon \bar{Y}(x) \geq \bar{Y}(x') - W(x, x') \text{ for all } x' \neq x \right\}.$$

The following reasoning is behind many subset selection procedures:

$$\begin{aligned}
&\Pr\{k \in \widehat{\mathcal{S}}\} \\
&= \Pr\left\{\bar{Y}(k) \geq \bar{Y}(x) - W(k,x),\ x \neq k\right\} \\
&= \Pr\left\{\bar{Y}(k) - \bar{Y}(x) - [\mu(k) - \mu(x)] \geq -W(k,x) - [\mu(k) - \mu(x)],\ x \neq k\right\} \\
&\geq \Pr\left\{\bar{Y}(k) - \bar{Y}(x) - [\mu(k) - \mu(x)] \geq -W(k,x),\ x \neq k\right\}.
\end{aligned}$$

Notice that the statistic $\bar{Y}(x) - \bar{Y}(x') - [\mu(x) - \mu(x')]$ has mean 0 for all $x \neq x'$, allowing the $W(x, x')$'s to be derived to give the desired probability based only on their variances. The survivors of subset selection can then be passed on to something like an IZ R&S procedure; see for instance, Nelson et al. (2001).

3.4.2 Fully Sequential Screening

The downside of using subset selection for screening, then applying IZ R&S to the survivors to select the best, is that the effectiveness of subset selection depends on the choice of sample size $n(x)$, and a good choice of $n(x)$ depends on the true means and variances of the outputs, which are unknown. A natural generalization is to do many rounds of subset selection, perhaps only stopping when there is one system remaining. Fully sequential, eliminating procedures do just that. Many such procedures are built on modeling the simulation output process as Brownian motion, a continuous-time, continuous-state stochastic process we review next.

Let $\{\mathcal{B}(t);\ t \geq 0\}$ be *standard Brownian motion (BM)*. Then

1. $\mathcal{B}(0) = 0$.
2. $\{\mathcal{B}(t);\ t \geq 0\}$ is almost surely continuous.
3. $\{\mathcal{B}(t);\ t \geq 0\}$ has independent increments: $\mathcal{B}(t) \perp \mathcal{B}(t+s) - \mathcal{B}(t)$.
4. $\mathcal{B}(t) - \mathcal{B}(s) \sim \mathrm{N}(0, t-s),\ 0 \leq s \leq t$.

An important generalization is BM with *drift* δt defined as $\mathcal{B}(t;\delta) = \mathcal{B}(t) + \delta t$. Therefore, $\sigma\mathcal{B}(t;\delta/\sigma) = \sigma\mathcal{B}(t) + \delta t$ has drift δt and variance $\sigma^2 t$. The relationship between BM with drift and R&S is as follows: Consider the sum of pairwise differences between the best system k and some other system x: $D_x(r) = \sum_{j=1}^{r}(Y_j(k) - Y_j(x))$, with $\sigma_{kx}^2 = \mathrm{Var}(Y_j(k) - Y_j(x))$, $\delta_{kx} = \mu(k) - \mu(x)$, and all outputs normally distributed. Then

$$\{D_x(r);\ r = 1, 2, \ldots\} \stackrel{\mathcal{D}}{=} \{\sigma_{kx}\mathcal{B}(r; \delta_{kx}/\sigma_{kx});\ r = 1, 2, \ldots\}. \tag{2}$$

That is, we can represent the cumulative pairwise-differences of two system's outputs (one being the best) as scaled Brownian motion with positive drift but monitored only at integer times. The following fundamental result relates the crossing times and probabilities of Brownian motion observed continuously, and only at integer times:

Theorem 1 (Jennison et al. (1982)) *Suppose $\delta > 0$, and we have a continuous function $g(t) \geq 0$ for all $t \geq 0$. Let*

$$T_d = \min\{r\colon |\mathcal{B}(r;\delta)| \geq g(r),\ r = 1, 2, \ldots\}$$

$$T_c = \min\{t\colon |\mathcal{B}(t;\delta)| \geq g(t),\ t \geq 0\}.$$

Then $T_c \leq T_d$ a.s. and $\Pr\{\mathcal{B}(T_d;\delta) \leq -g(T_d)\} \leq \Pr\{\mathcal{B}(T_c;\delta) \leq -g(T_c)\}$.

Thus, if crossing $-g(t)$ is an undesirable event—such as causing us to eliminate the true best system—then such an event is even less likely if we only observe the process at integer times. A lot is known about the probability of BM crossing boundaries of the form $\pm g(t)$. This, along with Theorem 1 facilitates desiging regions that control the probability of a selection error.

The relationship in (2) applies to synchronized, pairwise differences. Hong (2006) noted that the BM model can also extend to unequal samples sizes on a non-integer time scale via

$$\left[\frac{\sigma^2(k)}{n(k)} + \frac{\sigma^2(x)}{n(x)}\right]^{-1} \left[\bar{Y}(k) - \bar{Y}(x)\right] \stackrel{\mathcal{D}}{=} \mathcal{B}\left(\left[\frac{\sigma^2(k)}{n(k)} + \frac{\sigma^2(x)}{n(x)}\right]^{-1}; \mu(k) - \mu(x)\right). \tag{3}$$

Illustration: Paulson's Procedure

Because fully sequential, eliminating procedures have been so important in R&S we take a deep dive into Paulson's Procedure (Paulson, 1964), a fully sequential IZ procedure for known, common variance.

Paulson's Procedure

0. Set $\mathcal{S} = \{1, 2, \ldots, k\}$, choose $\lambda \in (0, \delta)$, set $a = \frac{\sigma^2}{\delta - \lambda} \ln\left(\frac{k-1}{\alpha}\right)$ and set $r = 0$.
1. Set $r = r + 1$. Simulate $Y_r(x),\ \forall x \in \mathcal{S}$.
2. Mark systems $\ell \in \mathcal{S}$ for elimination if

$$\min_{x \in \mathcal{S}} \left\{\sum_{j=1}^{r} (Y_j(\ell) - Y_j(x))\right\} < \min\{0, -a + \lambda r\}.$$

3. Remove all marked systems from $\mathcal{S}$.
4. If $|\mathcal{S}| = 1$ then stop and select system $\mathcal{S}$ as best; else go to Step 1.

Paulson's procedure tries to be *observation efficient* by attempting to eliminate systems after *each* additional replication. Notice that elimination decisions are highly coordinated, and require looking at $\binom{|\mathcal{S}|}{2}$ pairwise differences. Paulson guarantees $\Pr\{\text{select } k \mid \mu(k) - \mu(k-1) \geq \delta\} \geq 1 - \alpha$, but the guarantee is not clear when there are systems closer than δ. The extension to

unknown and unequal variances is not hard; as an illustration the case of unknown common variance σ^2 will be presented later. The procedure ends by or before step $N+1 = \lfloor a/\lambda \rfloor + 1$.

The large-deviation result supporting Paulson is as follows:

Theorem 2 *Suppose $Z_1, Z_2, \ldots$ are i.i.d.* $\mathrm{N}(\Delta, \sigma^2)$ *with $\Delta < 0$. Then for any constant $a > 0$*

$$\Pr\left\{\sum_{j=1}^{r} Z_j > a \text{ for some } r < \infty\right\} \le \exp\left(\frac{2\Delta a}{\sigma^2}\right).$$

Notice that since $\Delta < 0$ we expect the sum to drift *down*; this large deviation result bounds the probability it drifts *up* more than a. In the IZ formulation, we believe that $Y_j(x) - Y_j(k)$ has negative drift of at least $-\delta$ for all $x \neq k$. Attacking the pairwise differences we would like to choose a to obtain

$$\Pr\{k \text{ eliminated}\} \le \sum_{x=1}^{k-1} \Pr\{x \text{ eliminates } k\} = \sum_{x=1}^{k-1} \Pr\{\mathrm{ICS}_x\} \le \alpha.$$

Proof: We consider the probability that system $x \neq k$ incorrectly eliminates system k in isolation.

$$\begin{aligned}
\Pr\{\mathrm{ICS}_x\} &\le \Pr\left\{\sum_{j=1}^{r}(Y_j(k) - Y_j(x)) < -a + \lambda r \text{ some } r \le N+1\right\} \\
&= \Pr\left\{\sum_{j=1}^{r}(Y_j(x) - Y_j(k) + \lambda) > a \text{ some } r \le N+1\right\} \\
&\le \Pr\left\{\sum_{j=1}^{r}(Y_j(x) - Y_j(k) + \lambda) > a \text{ some } r < \infty\right\} \\
&\le \exp\left(\frac{2(\mu(x) - \mu(k) + \lambda)a}{2\sigma^2}\right) \le \exp\left(\frac{(-\delta + \lambda)a}{\sigma^2}\right) = \frac{\alpha}{k-1}
\end{aligned}$$

where the last step follows because we set $a = \dfrac{\sigma^2}{\delta - \lambda} \ln\left(\dfrac{k-1}{\alpha}\right)$. A common choice for the slope is $\lambda = \delta/2$.

There are a number of ways of improving on Paulson's Procedure, including (a) accomodating unknown and unequal variances (see Section 3.7); (b) exploiting tighter Brownian-motion large-deviation results (notice the result we used protected system k for all $r < \infty$; see Kim and Nelson (2001)); (c) facilitating variance-dependent sampling so that systems with low variance need to be simulated less (see Hong (2006)); (d) providing a PGS guarantee for when $\mu(k) - \mu(k-1) < \delta$ (see Section 3.6); (e) avoiding breaking up into

paired comparisons and using Bonferroni's inequality (see Dieker and Kim (2012)); and (f) exploiting common random numbers (see Section 3.5).

3.5 Common Random Numbers

R&S procedures that employ pairwise comparisons can often be "sharpened" by using common random numbers (CRN) because

$$\mathrm{Var}(Y(x) - Y(x')) = \mathrm{Var}(Y(x)) + \mathrm{Var}(Y(x')) - 2\,\mathrm{Cov}(Y(x), Y(x'))$$

and CRN tends to make $\mathrm{Cov}(Y(x), Y(x')) > 0$ (Nelson and Pei, 2021). However, to fully realize the CRN effect requires $n(x) = n(x')$ so that replications can be paired.

As an illustration, the impact of CRN on subset selection (Section 3.4.1) is that

$$W(x, x') = \left(t(x)^2 \frac{S^2(x)}{n(x)} + t(x')^2 \frac{S^2(x')}{n(x')} \right)^{1/2}$$

becomes

$$W(x, x') = \left(t^2 \frac{S^2(x, x')}{n} \right)^{1/2}$$

where $S^2(x, x') = S^2(x) + S^2(x') - 2\widehat{\mathrm{Cov}}(x, x')$ and $\widehat{\mathrm{Cov}}(x, x')$ is the estimated covariance. Thus, positive covariance should make it more difficult for inferior systems to remain in the subset because the boundary is tighter. Similarly, the impact on Paulson's Procedure (Section 3.4.2) with equal, known variance σ^2 and CRN-induced correlation $\rho > 0$ is that the elimination boundary has intercept

$$a = \frac{\sigma^2(1-\rho)}{\delta - \lambda} \ln\left(\frac{k-1}{\alpha}\right) \text{ rather than } a = \frac{\sigma^2}{\delta - \lambda} \ln\left(\frac{k-1}{\alpha}\right).$$

Again, positive covariance should make it more difficult for inferior systems to remain in the subset because the elimination boundary is narrower.

Simulation languages have random number "streams" that map to starting seeds that are *very* far apart; therefore, we can assign a unique stream to each random process and replication to enhance the impact of CRN (L'Ecuyer et al., 2002; Nelson and Pei, 2021).

3.6 "Good Selection"

The IZ-PCS paradigm $\mathrm{PCS} = \Pr\{\widehat{x}^\star = k \mid \mu(k) - \mu(k-1) \geq \delta\} \geq 1 - \alpha$ has been the most widely adopted in practice. Typically, δ is chosen as the

"smallest practically significant difference," which may not be close to the *actual* differences $\mu(k) - \mu(x)$. In fact when k is large we expect several "good" systems, and very many inferior ones. Thus, guaranteed probability of good selection

$$\text{PGS} = \Pr\{\mu(k) - \mu(\widehat{x}^\star) \leq \delta\} \geq 1 - \alpha$$

is more meaningful than PCS because it can be interpreted as an acceptable bound on the optimality gap.

Empirical experience suggests that procedures with an IZ-PCS guarantee also provide a PGS guarantee; however, counterexamples can be created. IZ procedures *without elimination* (e.g., Rinott) can often be shown to guarantee PGS, but elimination makes proving PGS difficult. An excellent comprehensive reference is Eckman and Henderson (2018). A condition that insures both PCS and PGS is stated in the following theorem:

Theorem 3 (Nelson and Matejcik (1995)) *Suppose a R&S procedure creates estimators* $\widehat{\mu}(1), \widehat{\mu}(2), \ldots, \widehat{\mu}(k)$ *that guarantee* $\Pr\{\widehat{\mu}(k) > \widehat{\mu}(i),\ \forall i \neq k \mid \mu(k) - \mu(k-1) \geq \delta\} \geq 1 - \alpha$. *Then if*

$$\begin{pmatrix} \widehat{\mu}(k) \\ \widehat{\mu}(k-1) - \mu(k-1) + (\mu(k) - \delta) \\ \\ \vdots \\ \widehat{\mu}(1) - \mu(1) + (\mu(k) - \delta) \end{pmatrix}$$

has the same distribution as estimators would have had in the corresponding slippage configure problem, then the procedure also guarantees $\text{PGS} \geq 1 - \alpha$.

Normally distributed output procedures like Rinott that do not adapt to the sample means often satisfy the conditions of this theorem. Unfortunately, lack of adaptation also tends to lead to inefficiency.

Zhong and Hong (2018) make an adjustment to Paulson's procedure so that it provides a good-selection guarantee. Recall that Paulson eliminates system ℓ if for some other system x we have $\sum_{j=1}^{r}(Y_j(\ell) - Y_j(x)) < -a + \lambda r$. Instead, Zhong and Hong (2018) use the condition $\sum_{j=1}^{r}(Y_j(\ell) - Y_j(x) + \delta) < -a + \lambda r$. Notice that when $\mu(k) - \mu(\ell) < \delta$, the sum of differences $\sum_{j=1}^{r}(Y_j(\ell) - Y_j(k) + \delta)$ still has positive drift. Thus, good systems should survive to the end, where Zhong and Hong (2018) then select the sample-best system.

A Bayesian "good selection" R&S procedure stops when it has collected enough output so that there is a system $\widehat{x}^\star$ for which

$$\Pr\{M(\widehat{x}^\star) > M(x) - \delta,\ \forall x \neq \widehat{x}^\star \mid \mathcal{H}\} \geq 1 - \alpha.$$

This is computable under some assumptions, but if not then it can be approximated or bounded. The interpretation is that "With probability at least

$1-\alpha$ the *random problem* from your space of priors is one for which the *fixed system* $\widehat{x}^\star$ is good." This contrasts with the frequentist perspective: The *random system* $\widehat{x}^\star$ chosen by the procedure has probability at least $1-\alpha$ of being good for this *fixed problem* (Eckman and Henderson, 2018).

3.7 Unknown Variances

As a general rule, neither known nor equal variances can be assumed in simulation R&S problems. For procedures that break into pairwise differences the variance of each pairwise difference can be estimated separately, which is also helpful for using CRN.

A useful result that sits behind many R&S procedures is this: If $Z_1, Z_2, \ldots, Z_{n_0}$ are i.i.d. $\mathrm{N}(\mu, \sigma^2)$ then $\bar{Z}$ is independent of S^2. Thus, using a "first-stage" S^2 to calibrate the additional simulation needed does not introduce bias. If done cleverly, we can derive the PCS *conditional* on S^2 and then uncondition. Not surprisingly, using estimated σ^2 increases E(sample size) relative to known variance (Mukhopadhyay and Solanky, 1994).

Illustration: Unknown Common Variance Paulson

Recall in Paulson that we set $\lambda = \delta/2$ and $a = \dfrac{2\sigma^2}{\delta}\ln\left(\dfrac{k-1}{\alpha}\right)$, assuming σ^2 was known. Suppose we estimate σ^2 from an initial simulation of n_0 replications from each system by

$$S^2 = \frac{1}{k(n_0-1)}\sum_{x=1}^{k}\sum_{j=1}^{n_0}(Y_j(x)-\bar{Y}(x))^2.$$

We will exploit two useful facts:

$$\frac{k(n_0-1)S^2}{\sigma^2} \sim \chi^2_d \text{ with } d = k(n_0-1) \text{ and } \mathrm{E}\left[\exp(t\chi^2_d)\right] = (1-2t)^{-d/2}$$

when χ^2_d is a chi-squared random variable with d degrees of freedom. The approach we take is to set $a = \eta S^2/\delta$ and see what η needs to be to get the desired PCS.

In the proof of Paulson's procedure we used a large-deviation result to show that for *fixed* a and $\lambda = \delta/2$

$$\Pr\{\mathrm{ICS}_x\} \le \exp\left(-\frac{\delta}{2\sigma^2}a\right).$$

With $a = \eta S^2/\delta$, to obtain $\Pr\{\mathrm{ICS}_x\} \le \alpha/(k-1)$ we need η to satisfy

$$\Pr\{\mathrm{ICS}_x\} = \mathrm{E}\left[\Pr\{\mathrm{ICS}_x \mid S^2\}\right] \le \mathrm{E}\left[\exp\left(-\frac{\delta}{2\sigma^2}\frac{\eta S^2}{\delta}\right)\right]$$
$$= \mathrm{E}\left[\exp\left(\underbrace{-\frac{\eta}{2d}}_{t}\underbrace{\frac{dS^2}{\sigma^2}}_{\chi^2_d}\right)\right] = \left(1-\frac{-2\eta}{2d}\right)^{-d/2} = \frac{\alpha}{k-1}.$$

Solving for η gives

$$\eta = \left(\frac{\alpha}{k-1}\right)^{-2/d} - 1.$$

Notice that the independence of $\bar{Y}$ and S^2 is critical.

Paulson is great for illustrating concepts, but the limitation to equal variances and no common random numbers makes it rarely used in simulation. There are many descendants, with one of the most statistically efficient and robust being KN (Kim and Nelson, 2001), which uses a tighter Brownian motion result; allows unequal variances and CRN; has been shown to be asymptotically valid for non-normal output data (discussed below); and has been implemented in commercial simulation languages and in parallel.

3.8 A Note on Asymptotic Analysis

Asymptotic analysis of R&S procedures is useful in at least three contexts:

1. Establishing that a procedure will work when core assumptions such as normality are violated (typically as $\delta \to 0$ in a way that also makes the problem harder).
2. Comparing the efficiency of procedures that are difficult to evaluate in finite samples (typically as $1-\alpha \to 1$ so that behavior becomes deterministic).
3. Comparing the efficiency of procedures with estimated variances relative to their known-variance counterparts (typically as $\delta \to 0$ drives $n_0 \to \infty$).

Setting 1 helps explain why normal-theory IZ procedures seem to work well more generally, while Setting 2 is often the only way (other than empirically) to compare procedures that eliminate systems.

For Setting 1 a *meaningful* limit is essential: If $\mu(k)-\mu(x)$ is fixed, then as we let $\delta \to 0$ for procedures with sample size proportional to $1/\delta^2$, we have PCS $\to 1$ for almost any kind of data by the strong law of large numbers. Kim and Nelson (2006a) let $\mu(k) = \mu$ and $\mu(x) = \mu - \delta$ for $x \neq k$. Notice that as $\delta \to 0$ the sample size goes to ∞ but the problem itself also gets harder. Is this a relevant setting? If $\delta \gg \mu(k)-\mu(x)$ then any system design is acceptable. If $\delta \ll \mu(k)-\mu(x)$ then a procedure will tend to simulate so many replications that it will select the best. Thus $\mu(x) = \mu(k) - \delta$ is the critical regime.

For R&S procedures based on Brownian motion, a key tool for asymptotic analysis via Setting 1 is

Theorem 4 (Donsker's Theorem) *If $Y_1, Y_2, \ldots$ are i.i.d. (μ, σ^2) with $\sigma^2 < \infty$ then as $N \to \infty$*

$$\frac{\sum_{j=1}^{\lfloor Nt \rfloor} Y_j - Nt\mu}{\sigma\sqrt{N}} \xrightarrow{\mathcal{D}} \mathcal{B}(t), \ 0 \le t \le 1.$$

The usual Central Limit Theorem drops out at $t = 1$. Donsker's Theorem goes further, stating that very general i.i.d. output processes, standardized the right way, look like Brownian motion as we get more and more data. In many IZ R&S procedures we can take $Y_j = (Y_j(x) - Y_j(x'))$, and letting $\delta \to 0$ drives the sample size to ∞ when $N \propto 1/\delta^2$.

4 Parallel R&S

The future of simulation, and certainly simulation optimization, is parallel computing. Simulation languages have already been redesigned to run in the cloud, where computer time is "rented." For instance, the commercial product Simio automatically exploits multi-core/multi-thread personal computers, and its portal version can recruit up to 10,000 processors from Microsoft Azure to run simulations in parallel.

The availability of cheap, easy-to-use parallel computing greatly extends the R&S limit in terms of problem size, k. However, since one may have to pay for the service, the focus of "efficiency" in R&S shifts from being observation-efficient to being computationally efficient as measured by wall-clock or rental time. Thus, it may be acceptable to waste simulation-generated replications to avoid idling processors and get the R&S problem solved faster. Factors such as heterogeneous processors, communication delays, processor failures, etc. that may disrupt the usual synchronization in R&S procedures now become relevant.

To describe parallel R&S we assume a master-worker paradigm: 1 Master process performing calculations and generating new jobs, and p Worker processes executing simulation and calculation jobs. While not the only possible architecture, it is a common one. The following framework for parallel R&S is based on Hunter and Nelson (2017).

We represent a R&S procedure as a sequence of jobs generated by the Master, $\mathcal{J} = \{J_j : 1 \le j \le M\}$, where Job j is an ordered list

$$J_j \equiv \{\underbrace{(\mathcal{Q}_j, \Delta_j, \mathcal{U}_j)}_{\text{simulate}}, \underbrace{(\mathcal{P}_j, \mathcal{C}_j)}_{\text{calculate}}\}.$$

The components of job j are

- $\mathcal{Q}_j \subseteq \{1, 2, \ldots, k\}$ indices of systems to be simulated;
- $\Delta_j = \{\Delta_{xj}\}$ how many replications to take from each system $x \in \mathcal{Q}_j$;
- $\mathcal{U}_j$ (optional) the assigned block of random numbers;
- $\mathcal{C}_j$ the list of non-simulation calculations or operations to perform; and
- $\mathcal{P}_j$ the list of jobs that must complete before executing the calculation $\mathcal{C}_j$.

Using this computational paradigm, most of the (non-parallel) R&S procedures presented so far look something the Nominal R&S Procedure below.

Nominal R&S Procedure

1. Until fixed-precision or fixed-budget ending condition reached, do
2. For $\ell = 1, 2, \ldots$

   a. Execute simulation jobs for non-eliminated or active systems:

   $$J_\ell = [\{(\text{system } i, 1 \text{ rep}), (\emptyset)\}, \ldots, \{(\text{system } j, 1 \text{ rep}), (\emptyset)\}, \ldots]$$

   b. Execute a comparison job using the results from Step 2a.

   $$J'_\ell = \{(\emptyset), (\text{all jobs in } J_\ell, \mathcal{C}_\ell)\}$$

   where $\mathcal{C}_\ell$ performs calculations on all non-eliminated or active systems.

The nominal procedure enforces many of the assumptions necessary for both small-sample and asymptotic analysis by "synchronized coupling." To directly parallelize it, the Master could spread job J_ℓ out among the p workers, but then many workers may be idle while waiting for the coupled Step 2b to complete. Such issues do not arise in a single-processor setting.

Why not just use the outputs from the simulation jobs as soon as they complete, rather than waiting? Luo et al. (2015) address this question, and show that new statistical issues arise. Recall there are $p + 1$ processors, consisting of 1 Master and p Workers, and suppose that all simulation jobs are assigned by the Master to a Worker in round robin fashion as follows: system $1, 2, \ldots, k, 1, 2, \ldots$. An eliminated system is removed from the remaining list. Let $Z_j(x)$ be the *input sequence*—the result of jth replication from alternative x *requested* by the Master—with execution and communication time $T_j(x)$. Similarly, let $Y_j(x)$ be the corresponding *output sequence*, meaning the jth output from alternative x *returned* to the Master. If the R&S procedure uses each output as soon as it is available to the Master, then the following new statistical issues arise.

1. The procedure is working with random sample sizes at each comparison step, rather than prescribed numbers of replications.
2. The $Y_j(x)$, $j = 1, 2, \ldots$ are not i.i.d. To see this, suppose $k = 1$, $Z_j(x) = T_j(x) \sim \text{Expon}(\mu(x))$. Then it can be shown that $\text{E}(Y_j(x)) = \mu(x)\left(1 - \left(1 - \frac{1}{p}\right)^j\right)$ because jobs with short execution times return to the master sooner.

3. There is a subtle dependence among systems' outputs caused by elimination of some systems impacting the number of replications of other systems.

Thus, parallelization takes some careful thought, not only from a computer science point of view, but also with respect to the statistical validity of the R&S procedure.

4.1 New Measures of Efficiency

How do we define "efficiency" in this new parallel paradigm?

- Let $0 < T_j < \infty$ be the wall-clock time Job J_j finishes, so that the ending time of the procedure is $T_e(\mathcal{J}) = \max_{j=1,2,\ldots,M} T_j$.
- Let $c(p, s)$ be the cost to purchase p processors for s time units.

With these definitions we can define revised "efficient" objectives.

Fixed Precision:

Achieve a statistical guarantee while being cost efficient:

$$\begin{aligned} \text{minimize}_{p,\mathcal{J}} \quad & \mathrm{E}[\beta_t \underbrace{T_e(\mathcal{J})}_{\text{time}} + \beta_c \underbrace{c(p, T_e(\mathcal{J}))}_{\text{cost}}] \\ \text{s.t.} \quad & \Pr\{ \underbrace{G(\widehat{x}^\star, k)}_{\text{good event}} \} \geq 1 - \alpha. \end{aligned}$$

Fixed Budget:

Minimize a loss for the selected system within a budget:

$$\begin{aligned} \text{minimize}_{p,\mathcal{J}} \quad & \mathrm{E}[\underbrace{\mathcal{L}(G^c(\widehat{x}^\star, k), \mathcal{J})}_{\text{loss from bad event}}] \\ \text{s.t.} \quad & \underbrace{c(p, T_e(\mathcal{J}))}_{\text{cost}} \leq b. \end{aligned}$$

Notice that for both fixed-precision and fixed-budget formulations, the decision variables are the number of processors to rent p and the jobs to execute $\mathcal{J}$. For fixed precision it is possible that we would only have one of β_t or β_c to be non-zero, depending on whether the time to reach a decision or the rental cost to reach a decision is most important.

Table 1 Selected parallel R&S literature.

R&S Procedure	**Load Balancing (Standard Assumptions)**	**Comparison Timing (Relaxed Assumptions)**
Fixed-Precision	Simple Divide and Conquer (Chen, 2005) Vector-Filling Procedure (Luo et al., 2015) Good Selection Procedure (Ni et al., 2017) Strategic Updating (Zhong and Hong, 2021)	Asymptotic Parallel Selection (Luo et al., 2015) bi-PASS (Pei et al., 2020)
Fixed-Budget	Parallel OCBA (Luo et al., 2000) Asynchronous OCBA/KG (Kamiński and Szufel, 2018)	bi-PASS (Pei et al., 2020)

To the best of our knowledge no R&S procedure has yet been created that directly attacks one of these formulations. Table 1 cites much of the existing literature on parallel R&S, divided into fixed-budget vs. fixed-precision, and load balancing to enforce standard non-parallel assumptions vs. a uniquely parallel paradigm. Consider for instance Luo et al. (2015), and its "phantom clock." The underlying procedure is KN, a fully sequential procedure that uses pairwise sums of differences. Luo et al. (2015) note that even when the input and output sequences are not the same, if the procedure only makes comparisons at times t when $\sum_{j=1}^{t}(Y_j(k) - Y_j(x)) = \sum_{j=1}^{t}(Z_j(k) - Z_j(x))$ then the order of return from the workers does not matter. Strictly enforcing this is load balancing. Instead, Luo et al. (2015) insert a "phantom job" at the end of each round-robin job cycle $(1, 2, \ldots, k, \text{ph}, 1, 2, \ldots, k, \text{ph}, \ldots)$, and then only compare systems when a phantom job returns to the Master. They show that by doing this the sums of differences are only out of sync by an asymptotically negligible amount.

4.2 New Objectives

Does insuring a prespecified PCS or PGS continue to make sense if k is very large? For instance, if $k >$ 1,000,000 systems, is it sensible or even computationally feasible to identify the single best or near-best with high probability? In such a problem we expect many bad systems, but also a lot of good ones. Trying to achieve PCS or PGS, which are family-wide statements, in such a setting runs counter to approaches in large-scale statistical inference of controlling "error rates." (Efron, 2012). Specifically, to control PCS requires more and more effort per system as k increases. As we argue below, rates such as "false discovery" can be attained with little or no "k effect."

But why apply R&S for such large-k problems anyway? Surely so many systems arise from combinations of more basic decision variables, which suggests using a search algorithm rather than exhaustive simulation. However, from a practical perspective, the key is to *actually solve the problem* in some effective way. As discussed earlier, R&S is the only SO technique that can control all sources of error. Therefore, if parallel computing extends the R&S limit enough to encompass a problem, it makes sense to use it.

This motivates consideration of new goals for parallel R&S:

- More scalable—but still useful and understandable—error control than PCS or PGS. As an example we discuss *Expected False Elimination Rate (EFER)*, which is the fraction of good systems eliminated.
- Avoid procedures with coupled operations and synchronization, to facilitate parallelization. As an example we describe parallel adaptive survivor selection (PASS), and a specific instance bisection-PASS (bi-PASS).

For some known constant $\mu^\star$, that we call the "standard," let $S_x(n) = \sum_{j=1}^{n}(Y_j(x) - \mu^\star) = \sum_{j=1}^{n} Y_j(x) - n\mu^\star$. Suppose we have a non-decreasing function $g_{x,\alpha}(\cdot) \geq 0$ with the property that

$$\Pr\{S_x(n) \leq -g_{x,\alpha}(n), \text{ some } n < \infty\} \begin{cases} \leq \alpha, \ \mu(x) \geq \mu^\star \\ = 1, \ \mu(x) < \mu^\star. \end{cases}$$

Finally, let $\mathcal{G} = \{x\colon \ \mu(x) \geq \mu^\star\}$, which we refer to as the "good systems." Consider the following parallel procedure:

Parallel Survivor Selection (PSS)

1. Given a standard $\mu^\star$, an increment Δn and a budget.
2. Let $\mathsf{W} = \{1, 2, \ldots, p\}$ be the set of available workers; $\mathcal{Q} = \{1, 2, \ldots, k\}$ the set of surviving systems; and $n(x) = 0$ for all $x \in \mathcal{Q}$.
3. Until the budget is consumed, do
   a. While there is an available worker in W, do in parallel:
      i. Remove a system $x \in \mathcal{Q}$ and assign to available worker $w \in \mathsf{W}$
      ii. $j = 1$
      iii. while $j \leq \Delta n$
           Simulate $Y_{n(x)+j}(x)$.
           If $S_x(n(x) + j) \leq -g_{x,\alpha}(n(x) + j)$ then eliminate system x and break loop.
           Else $j = j + 1$.
      iv. If x not eliminated then return to $\mathcal{Q} = \mathcal{Q} \cup \{x\}$ and $n(x) = n(x) + \Delta n$.
      v. Release worker w to available workers W.
4. Return $\mathcal{Q}$.

Pei et al. (2018) show that EFER $= \mathrm{E}[|\mathcal{G} \cap \mathcal{Q}^c|]/|\mathcal{G}| \leq \alpha$. That is, the expected fraction of good systems eliminated by the procedure is no greater than α. Critically, the function g depends only on x and α, but *not* k.

The generic boundary function $g(\cdot)$ needs to insure that driftless Brownian motion ($\mu(x) = \mu^\star$) crosses with probability no more than the EFER α, while Brownian motion with negative drift ($\mu(x) < \mu^\star$) crosses with probability 1. Fan et al. (2016) note that driftless Brownian motion grows to ∞ at rate $O(\sqrt{t \log\log(t)})$, while BM with negative drift goes to $-\infty$ at rate $O(t)$. Thus $g(\cdot)$ needs to be between these two; they suggest $g(t) = \sqrt{[c + \log(t+1)](t+1)}$, then tuning c to get the desired error control α. The function $g_{x,\alpha}$ also includes time scaling by σ_x^2 or an estimate of it.

PSS requires no coupling and keeps the workers constantly busy; it could perhaps be made more efficient by making Δn depend on the system. Of course, the decoupling occurs because the standard $\mu^\star$ is known. However, the EFER is still controlled at $\leq \alpha$, and elimination still occurs with probability 1, if we replace $\mu^\star$ by $\mu^\star(n) \leq \mu^\star$ where $\mu^\star(n) \uparrow \mu^\star$, because a system eliminated by a smaller standard would also have been eliminated by a larger standard, and a system protected from a larger standard would also be protected from a smaller one. This suggests trying to *learn* a standard that achieves our objectives empirically, which is called Parallel *Adaptive* Survivor Selection.

Generically, we define the standard to be $\mu^\star = s(\mu_1, \mu_2, \ldots, \mu_k, \mu^+)$. Some examples of possibly interesting standards are

- Protect the best or ties: $\mu^\star = \mu_k$.
- Protect the top m: $\mu^\star = \mu_{k-m+1}$.
- Protect the best and everything as good as a known μ^+: $\mu^\star = \min\{\mu^+, \mu_k\}$.

The key is to *learn* the standard's value in a way that still avoids coupling and does not affect the EFER. bi-PASS uses the standard

$$\bar{\mu} = \frac{1}{|\mathcal{Q}|} \sum_{x \in \mathcal{Q}} \bar{Y}(x)$$

the average of the sample means of the current survivors. Thus, the standard acts like a bisection search. Under some conditions it can be shown that the EFER is still $\leq \alpha$. Notice also that updating $\bar{\mu}$ is fast for the Master, and can occur whenever replications are returned from the Workers.

4.3 Parting Thoughts

Computer science issues really matter in parallel R&S: There is not one, unique parallel architecture, and customizations can be valuable. Message passing via MPI is conceptually easy, but unexpected behavior can occur, and passing messages does take time. Processors may be heterogeneous, and

results can be lost. Memory may be shared or not. The overhead to load a simulation onto a processor can be substantial, so one also needs to consider the fixed cost to set up a simulation, as well as marginal time per replication. And management of pseudorandom numbers can be tricky, e.g., if we want to use CRN.

However, when a simulation optimization problem can be treated as a R&S problem then it can be "solved" and all three errors can be controlled. High-performance, parallel computing extends the "R&S limit" but introduces new statistical and computational problems. Standard assumptions may be violated, and "cost" no longer equals the number of replications.

5 Other Formulations

Although our focus has been on the best-mean problem with normally distributed outputs, the R&S literature is much broader. Many R&S procedures have been created for specific non-normal data; e.g., Poisson. R&S procedures have also been created for other performance measures; e.g., probabilities and quantiles. Selecting the system that is *most likely* to be the best is called "multinomial selection," which may make sense for one-shot decisions. Selecting the best system better than a *standard* (either system or constant value) has also received attention. See Bechhofer et al. (1995).

The Holy Grail is a R&S procedure that works for virtually any performance measure (mean, probability, quantile) and output data distribution (normal, non-normal). Let $\theta(x)$ be the generic performance measure, and $\widehat{\theta}(x)$ a point estimator. Two insights make an omnibus procedure possible:

1. If we can construct estimators $\widehat{\theta}(x)$ of parameters $\theta(x)$ such that

$$\Pr\left\{\widehat{\theta}(x) - \widehat{\theta}(k) - (\theta(x) - \theta(k)) \le \delta,\ \forall x \ne k\right\} \ge 1 - \alpha \qquad (4)$$

   then

$$\text{PGS} = \Pr\{\theta(k) - \theta(\widehat{x}^\star) \le \delta\} \ge 1 - \alpha.$$

2. Given a sample of output data, we can estimate the achieved probability in (4) using *bootstrapping*, and then increase the sample size until it is $\ge 1 - \alpha$.

Suppose we have N replications from each of the k systems, and let $\widehat{x}^\star = \text{argmax}_x \widehat{\theta}(x)$, the sample best. Then the bootstrap estimate of PGS based on B bootstrap samples is

$$\widehat{\text{PGS}} = \frac{1}{B}\sum_{b=1}^{B} \prod_{x \ne \widehat{x}^\star} \mathcal{I}\left\{\widehat{\theta}^{(b)}(x) - \widehat{\theta}^{(b)}(\widehat{x}^\star) - \left[\widehat{\theta}(x) - \widehat{\theta}(\widehat{x}^\star)\right] \le \delta\right\}$$

where $\widehat{\theta}^{(b)}(x)$ comes from independent bootstrap samples of size N and $\mathcal{I}(\cdot)$ is the indicator function. The omnibus procedure is to increase N (generate more simulation output) until this bootstrap estimate is $\geq 1-\alpha$. Lee and Nelson (2016) showed this approach to be asymptotically valid under very mild conditions on the data as $\delta \to 0$.

As an illustration suppose $\theta(x)$ is the mean.

Simulation output: $[Y_1(x), \ldots, Y_N(x)] \to \bar{Y}(x),\ x = 1, 2, \ldots, k$, with $\widehat{x}^\star = \text{argmax}_x \bar{Y}(x)$ the current sample best with N replications.

Bootstrap: Resample the simulation outputs B times with replacement to get $\left[Y_1^{(b)}(x), \ldots, Y_N^{(b)}(x)\right] \to \bar{Y}^{(b)}(x),\ x = 1, 2, \ldots, k,\ b = 1, 2, \ldots, B$.

Estimate PGS:

$$\widehat{\text{PGS}} = \frac{1}{B}\sum_{b=1}^{B} \prod_{x \neq \widehat{x}^\star} \mathcal{I}\left\{\bar{Y}^{(b)}(x) - \bar{Y}^{(b)}(\widehat{x}^\star) - \left[\bar{Y}(x) - \bar{Y}(\widehat{x}^\star)\right] \leq \delta\right\}.$$

Notice that we can incorporate CRN by bootstrapping *vectors* of replications, where each vector contains one replication from each of the k systems generated using CRN.

6 Multi-armed Bandits

"Multi-armed bandit" is a slang name for slot machines. To play a slot machine one inserts a coin or token and pulls the machine's mechanical "arm." Typically the coin is lost (negative reward), but occasionaly the machine "pays out" a positive reward. A slot machine player would like to find the machine among many with the highest payout while losing as little money as possible. In the state of Illinois the percentage payback from slot machines in 2017 ranged from 89%–92.5%, so in the long run you lose (thus the name "bandit"). But the slot machine paradigm provides a structure for thinking about optimal sequential decision making.

Multi-armed bandit (MAB) procedures address the problem of learning via experimentation which of k possible decisions leads to the greatest accumulated reward. Clearly there is a connection between R&S and MAB, and the procedures look similar, but they are not the same. The usual objective of MAB is to minimize "regret" (defined below) when making repeated decisions, while R&S attempts to identify the best system to implement. Most MAB procedures are intended for online use, while R&S is offline simulation optimization. MAB and R&S have different standards for "good procedure performance" and different assumptions about the reward/output data. R&S procedures tend to be more willing to waste observations on inferior systems so as to reduce the *overall* number of observations needed to make a correct selection, while MAB, which accumulates rewards, attempts to avoid

Table 2 MAB definitions of "regret."

Regret	$R_n = \max_x \sum_{t=1}^{n} Y_t(x) - \sum_{t=1}^{n} Y_t(I_t)$
Expected regret	$r_n = \mathrm{E}(R_n)$
Pseudo-regret	$\bar{r}_n = \max_x \mathrm{E}\left[\sum_{t=1}^{n} Y_t(x) - \sum_{t=1}^{n} Y_t(I_t)\right]$

the regret of choosing decisions with suboptimal rewards while searching for the best decision. A good overview reference is Jamieson and Nowak (2014). There is no denying that "multi-armed bandit" is a cooler name than "ranking & selection," but both have their roles.

In a bit more detail, "online" means making decisions in real time, with a stochastic reward after each decision, while "offline" means running a computer experiment to select a system and then implementing the selection in the real world. There is no reward associated with the R&S experiment, although there is a computational cost for running simulation.

The term "regret" refers to the shortfall in rewards that are obtained relative to what could have been attained by making the best decision, while PCS refers to getting the best choice in the end, not how one gets there. MAB tends to evaluate procedures via their probability *complexity*, while R&S evaluates procedures via their *finite-time effort*. MAB tends to assume sub-Gaussian (even bounded) reward distributions; R&S often assumes normally distributed output. MAB typically assumes a finite budget, and R&S often desires fixed precision.

In the classical stochastic MAB formulation, using our R&S notation, the decisions or "arms" are $x \in \{1, 2, \ldots, k\}$, with unknown reward distribution F_x having expected value $\mu(x)$ for making decision x. Let I_t be the decision chosen on opportunity t, and $Y_t(I_t) \sim F_{I_t}$ the associated stochastic reward. Using this notation, Table 2 defines regret, expected regret, and pseudo-regret. Loosely, the goal is to pick a policy for selecting I_t that minimizes regret or expected regret; since neither of these is achievable, pseudo-regret is a stand-in. One well-known MAB procedure is the upper confidence bound (UCB) policy: At the end of decision opportunity t, construct an UCB for each decision's mean, $\mu(x)$. On opportunity $t+1$, play the arm with the largest UCB. This is sometimes referred to as "optimism in the face of uncertainty" since one selects the decision with the largest apparent upside.

Clearly all forms of regret are non-decreasing in the number of decisions n that one makes; a good MAB procedure tries to have regret increase at the slowest possible rate. A building block result for pseudo-regret is the following:

$$\begin{aligned}\bar{r}_n &= n\mu(k) - \sum_{t=1}^{n} \mathrm{E}(\mu_{I_t}) \\ &= n\mu(k) - \sum_{x=1}^{k} \mu(x)\mathrm{E}(\#\text{ times played arm } x \text{ thru turn } n) \\ &= \sum_{x=1}^{k} (\mu(k) - \mu(x))\mathrm{E}(\#\text{ times played arm } x \text{ thru turn } n).\end{aligned}$$

One then derives an upper bound on the rate at which the E(# times played arm x thru turn n) increases as the number of decisions n increases for $x \neq k$. This bounds the rate at which $\bar{r}_n$ increases. Note that this bound is neither an estimate of the pseudo-regret $\bar{r}_n$ nor a statistical guarantee. But it does say that as you play you accumulate regret no faster than the derived rate.

MAB procedures are frequently quite simple to implement, which makes them attractive, and of course many problems require online solutions (e.g., if you do not have the luxury of a simulation model of the world). However, for simulation optimization used to design systems, the R&S formulation tends to be more efficient and attacks the relevant objective.

Acknowledgements

Special thanks to David Eckman and Linda Pei for comments on the structure and specifics of this tutorial, as well as to the two referees for their careful reports. This research was supported by National Science Foundation Grant Number DMS-1854562.

References

Bechhofer R, Santner T, Goldsman D (1995) Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons. John Wiley and Sons, New York

Chen CH, Lee LH (2011) Stochastic Simulation Optimization: An Optimal Computing Budget Allocation. World Scientific, Singapore

Chen CH, Chick SE, Lee LH, Pujowidianto NA (2015) Ranking and selection: Efficient simulation budget allocation. In: Fu M (ed) Handbook of Simulation Optimization, Springer, pp 45–80

Chen EJ (2005) Using parallel and distributed computing to increase the capability of selection procedures. In: Proceedings of the 2005 Winter Simulation Conference, IEEE, pp 723–731

Chen Y, Ryzhov IO (2019) Complete expected improvement converges to an optimal budget allocation. Advances in Applied Probability 51(1):209–235

Dieker A, Kim SH (2012) Selecting the best by comparing simulated systems in a group of three when variances are known and unequal. In: Proceedings of the 2012 Winter Simulation Conference, IEEE, pp 490–496

Eckman DJ, Henderson SG (2018) Guarantees on the probability of good selection. In: Proceedings of the 2018 Winter Simulation Conference, IEEE, pp 351–365

Efron B (2012) Large-scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction. Cambridge University Press

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

Frazier P (2012) Tutorial: Optimization via simulation with Bayesian statistics and dynamic programming. In: Proceedings of the 2012 Winter Simulation Conference, IEEE, pp 79–94

Glynn P, Juneja S (2004) A large deviations perspective on ordinal optimization. In: Proceedings of the 2004 Winter Simulation Conference, IEEE, pp 577–585

Goldsman D (1983) Ranking and selection in simulation. In: Proceedings of the 1983 Winter Simulation Conference, IEEE, pp 387–393

Gupta SS, Panchapakesan S (2002) Multiple Decision Procedures: Theory and Methodology of Selecting and Ranking Populations. SIAM, Philadelphia

Hong LJ (2006) Fully sequential indifference-zone selection procedures with variance-dependent sampling. Naval Research Logistics 53(5):464–476

Hunter SR, Nelson BL (2017) Parallel ranking and selection. In: Advances in Modeling and Simulation, Springer, New York, pp 249–275

Jamieson K, Nowak R (2014) Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In: 48th Annual Conference on Information Sciences and Systems (CISS), IEEE, pp 1–6

Jennison C, Johnstone IM, Turnbull BW (1982) Asymptotically optimal procedures for sequential adaptive selection of the best of several normal means. In: Statistical Decision Theory and Related Topics III, Elsevier, pp 55–86

Kamiński B, Szufel P (2018) On parallel policies for ranking and selection problems. Journal of Applied Statistics 45(9):1690–1713

Kim SH, Nelson BL (2001) A fully sequential procedure for indifference-zone selection in simulation. ACM Transactions on Modeling and Computer Simulation (TOMACS) 11(3):251–273

Kim SH, Nelson BL (2006a) On the asymptotic validity of fully sequential selection procedures for steady-state simulation. Operations Research 54(3):475–488

Kim SH, Nelson BL (2006b) Selecting the best system. In: Henderson SG, Nelson BL (eds) Handbooks in Operations Research and Management Science, vol 13, Elsevier, pp 501–534

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

Lee S, Nelson BL (2016) General-purpose ranking and selection for computer simulation. IIE Transactions 48(6):555–564

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

Luo YC, Chen CH, Yücesan E, Lee I (2000) Distributed web-based simulation optimization. In: Proceedings of the 2000 Winter Simulation Conference, IEEE, pp 1785–1793

Mukhopadhyay N, Solanky TK (1994) Multistage Selection and Ranking Procedures: Second Order Asymptotics. CRC Press, Boca Raton, Florida

Nelson BL (2018) Selecting the best simulated system: Thinking differently about an old problem. In: International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer, New York, pp 69–79

Nelson BL, Matejcik FJ (1995) Using common random numbers for indifference-zone selection and multiple comparisons in simulation. Management Science 41(12):1935–1945

Nelson BL, Pei L (2021) Foundations and Methods of Stochastic Simulation: A First Course, 2nd edn. 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. Operations Research 49(6):950–963

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

Paulson E (1964) A sequential procedure for selecting the population with the largest mean from k normal populations. The Annals of Mathematical Statistics pp 174–180

Pei L, Hunter SR, Nelson BL (2018) A new framework for parallel ranking & selection using an adaptive standard. In: Proceedings of the 2018 Winter Simulation Conference, IEEE, pp 2201–2212

Pei L, Nelson BL, Hunter SR (2020) Evaluation of bi-PASS for parallel simulation optimization. In: Proceedings of the 2020 Winter Simulation Conference, IEEE, pp 2960–2971

Rinott Y (1978) On two-stage selection procedures and related probability-inequalities. Communications in Statistics - Theory and Methods 7(8):799–811

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

Zhong Y, Hong LJ (2018) Fully sequential ranking and selection procedures with PAC guarantee. In: Proceedings of the 2018 Winter Simulation Conference, IEEE, pp 1898–1908

Zhong Y, Hong LJ (2021) Knockout-tournament procedures for large-scale ranking and selection in parallel computing environments. Operations Research Forthcoming