Selecting the Best Simulated System: Thinking Differently About an Old Problem

Barry L. Nelson

Abstract The methods known collectively as "ranking & selection" have been a theoretical and practical success story for the optimization of simulated stochastic systems: they are widely used in practice, have been implemented in commercial simulation software, and research has made them more and more statistically efficient. However, "statistically efficient" has meant minimizing the number of simulation-generated observations required to make a selection, or maximizing the strength of the inference given a budget of observations. Exploiting high-performance computing, and specifically the capability to simulate many feasible solutions in parallel, has challenged the ranking & selection paradigm. In this paper we review the challenge and suggest an entirely different approach.

1 Introduction

A generic stochastic simulation optimization (SO) problem has the form

Maximize E[Simulated Performance]
Subject to: Resource constraints

The types of simulations that are the focus of this paper are dynamic, often nonstationary, and may be computationally expensive to execute. SO is difficult because the lack of a mathematical expression for, or even a deterministic numerical method to evaluate, E[Simulated Performance], implies that algorithms must make progress by *estimating* the performance of specific feasible solutions. This leads to the three sources of error in SO:

1. The SO algorithm never simulates the optimal solution.
2. The SO algorithm does not recognize the best feasible solution it simulated.

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

3. The estimated performance of the sample-best solution returned by the SO algorithm is biased.

This paper addresses methods collectively known as Ranking & Selection (R&S). R&S originated with Bechhofer [2] and Gupta [7] in the 1950s for biostatistics and industrial applications, such as evaluating the efficacy of three drug treatments and a placebo. Typical problem characteristics included a small number of treatments k; normally distributed responses; relatively equal (maybe even known) variances; and a requirement to be easy to implement (e.g., since human subjects were involved). At the 1983 Winter Simulation Conference Goldsman [6] presented a tutorial on R&S and organized a session with both Bechhofer and Gupta, arguing that R&S was useful for optimizing simulated systems as well.

Since 1983 R&S has been an area of intense theoretical and practical interest in stochastic simulation. However, simulators were interested in problems with different characteristics:

- *Much* larger numbers of "treatments" (system designs) k.
- Possibly non-normal (nominal) simulation output data.
- Significantly unequal variances across system designs.
- Intentionally induced dependence across the outputs of simulated system designs due to Common Random Numbers (CRN).
- Highly sequential procedures to reduce the number of expensive simulation runs required to select the best system.

R&S has been a theoretical and practical success for simulation, including innovative theory; asymptotic regimes for non-normal data; and effective use of concepts from "statistical learning." Further, R&S is routinely applied in real problems and is included in many commercial software packages. The appeal of R&S is that it can control all three SO errors:

1. R&S is exhaustive, simulating all feasible solutions, so the optimal solution is always simulated.
2. R&S is explicitly concerned with recognizing the best solution with statistical confidence.
3. R&S may provide confidence intervals on the true performance of the selected solution.

Thus, it is desirable to turn a SO problem into a R&S problem if at all possible, and high-performance computing, and in particular parallel computing, would seem to facilite treating problems with larger and larger numbers of feasible solutions as R&S problems. Unfortunately, nearly all the methodological developments in R&S assume single-processor computing, and define "cost" as synonymous with the number of simulated observations. The topic of this paper is how parallel computing changes (nearly) everything, and a suggestion for how to think differently.

Remark 1. There is a connection between R&S and multi-arm bandit (MAB) problems that will not be explored here, other than to say that the objectives of MAB and R&S are often different (e.g., MAB minimizes regret); the MAB focus is online

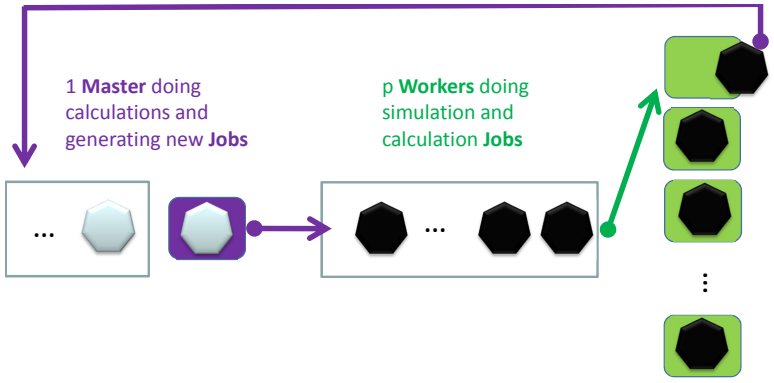

Fig. 1: Master-Worker environment.

decision making, while R&S is always offline; and the two literatures have different standards for what constitutes a "good procedure." See for instance [10].

Remark 2. The particular parallel computing architecture within which we implement R&S matters, but we will not address those details other than to assume that there are $p+1$ processors in a "Master-Worker" environment in which the Master processor performs calculations and decides what jobs to send to the Worker processors; see Figure 1. We define a "job" more precisely later.

Remark 3. While it is possible to treat many SO problems as R&S problems, clearly not all of them can be attacked in this way. We now consider $k = 10{,}000$ systems to be routine, but there are practical problems for which k is several orders of magnitude larger, and can even be uncountably infinite if systems are defined by continuous-valued decision variables. Further, the resource constraints may also be stochastic, requiring simulation to assess feasibility.

2 R&S Basics

For notation, let the true, but unknown, expected values (means) of the k feasible solutions (systems) be denoted by

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_{k-1} \leq \mu_k.$$

We refer to system k, or any system tied with system k, as the best, and of course we do not actually know which system is system k. Let Y_{ij} be the jth output from system i, which has mean μ_i and variance σ_i^2. For system i we can estimate μ_i with a consistent estimator, which for the purpose of this paper is the sample mean of n_i independent and identically distributed (i.i.d.) replications:

$$\bar{Y}_i(n_i) = \frac{1}{n_i}\sum_{j=1}^{n_i} Y_{ij}.$$

The R&S procedure returns something like

$$\widehat{K} = \text{argmax}_{i\in\{1,2,\ldots,k\}}\bar{Y}_i(n_i)$$

as the selected solution, where what the procedure specifies is the values of n_i.

One categorization of R&S procedures is *fixed-precision* vs. *fixed-budget*. For the former, we simulate until a prespecified confidence level is achieved, ideally probability of correct selection (PCS): $\Pr\{\mu_{\widehat{K}} = \mu_k\} \geq 1-\alpha$. Since attaining this goal can be computationally infeasible if, say, the best and second-best systems' means are very close, a compromise is made such as the following:

- **Indifference zone:** $\Pr\left\{\widehat{K} = k \mid \mu_k - \mu_{k-1} \geq \delta\right\} \geq 1-\alpha$
- **Good selection:** $\Pr\left\{\mu_k - \mu_{\widehat{K}} \leq \delta\right\} \geq 1-\alpha$
- **Top** m**:** $\Pr\left\{\widehat{K} \in [k, k-1, \ldots, k-m+1]\right\} \geq 1-\alpha$
- **Subset:** Find $\widehat{S} \subseteq \{1,2,\ldots,k\}$ such that $\Pr\{k \in \widehat{S}\} \geq 1-\alpha$

These are typically *frequentist* guarantees to be achieved as efficiently as possible. Here δ is taken as the smallest difference that is practically relevant.

A fundamental building block for many fixed-precision procedures is the standardized sums of differences:

$$\left[\frac{\sigma_k^2}{n_k} + \frac{\sigma_i^2}{n_i}\right]^{-1}[\bar{Y}_k(n_k) - \bar{Y}_i(n_i)] \stackrel{\mathscr{D}}{\approx} \mathscr{B}_{\mu_k-\mu_i}\left(\left[\frac{\sigma_k^2}{n_k} + \frac{\sigma_i^2}{n_i}\right]^{-1}\right)$$

where $\mathscr{B}_{\mu_k-\mu_i}(t)$ is Brownian motion (BM) with drift $\mu_k - \mu_i$ and the sample sizes n_k and n_i are independent of the sample means. This relationship is true in finite samples if the Y_{ij} are normally distributed (see [8]), and may be true asymptotically for appropriately standardized statistics. Much is known about BM processes crossing various boundaries (see, for instance, [12]), but for the purpose of this paper notice that employing this building block involves $k(k-1)/2$ pairwise comparisons, a number that can become a computational bottleneck when k is large.

For fixed-budget procedures, the goal is to obtain as strong an inference as possible within a fixed computation budget. This is typically formulated as minimizing some expected loss for the chosen solution, $\text{E}[\mathscr{L}(\widehat{K})]$, and the inference is typically *Bayesian*:

0-1 Loss: Maximize posterior PCS
Opportunity cost: Minimize posterior expected optimality gap

The fixed-budget paradigm is to attain information in an optimal, sequential fashion; see Frazier [5]. Tools for doing so include "Expected Improvement" and the "Knowledge Gradient (KG)." For instance if our prior is

$$(\mu_1, \mu_2, \ldots, \mu_k)^\top \sim \text{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

and the simulation output are normal, then we can compute the Complete Expected Improvement of solution i over the current sample best $\widehat{K}$,

$$\mathrm{CEI}(i,\widehat{K}) = \mathrm{E}\left[\max\{0,\mu_i - \mu_{\widehat{K}}\} \mid Y_{ij}\text{'s collected through stage } t\right]$$

from $\mathrm{N}(\boldsymbol{\mu}_t,\ \boldsymbol{\Sigma}_t \mid Y_{ij}\text{'s collected through stage } t)$, the posterior (normal) distribution. Notice that, implemented naively, this statistical learning approach takes only one simulated observation "optimally" at a time, and therefore does not exploit parallelization. In addition, calculation of the posterior distribution and searching for the maximum CEI or KG solution can be numerically challenging for large k.

Remark 4. R&S addresses a more diverse set of problems than selecting the system with the best mean; see [1] for a comprehensive reference.

3 R&S Computation

This section is based on [9].

Instead of thinking in terms of the statistical efficiency of a R&S procedure, here we consider the overall computation involved. All R&S procedures perform simulation *replications* and numerical *calculations*. Therefore, we define a R&S "job" j as the ordered list

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

where

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

We allow $(\mathscr{Q}_j,\Delta_j,\mathscr{U}_j)$ or $(\mathscr{P}_j,\mathscr{C}_j)$ to be null, or for a job to contain multiple simulate and calculate sub-jobs. The random numbers $\mathscr{U}_j$ are important to insure independence or dependence (CRN), if desired. Since we do not discuss CRN here, we suppress the specification of random numbers $\mathscr{U}_j$ from here on.

From the perspective of the jobs required, a generic R&S procedure looks something like this:

Generic R&S Procedure

1. For job $\ell = 1,2,\ldots$ until termination, do
   a. *Simulation jobs*

$$\mathscr{J}_\ell = [\{(\text{system } 1, 1 \text{ rep}),(\emptyset)\},\ldots,\{(\text{system } i, 1 \text{ rep}),(\emptyset)\},\ldots]$$

b. *Comparison jobs*

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

where $\mathscr{C}_\ell$ performs calculations on all (non-eliminated) systems.

2. Report $\widehat{K}$.

This generic model enforces many of the assumptions necessary for both small-sample and asymptotic analysis by "synchronized coupling:" simulate all required replications, perform calculations on the collected output to decide what to simulate next, simulate all required replications, and so on.

Now suppose that we want to parallelize this. Recall that we initially have k systems and $p+1$ processors, 1 Master and p Workers. Perhaps the most natural way to think about adapting the Generic R&S Procedure to this setting is for the Master to maintain a round robin queue of systems from which a replication is needed, and whenever a replication result is returned from some processor the Master assigns another system to it from its queue. Based on the returned replications the Master then makes comparisons, eliminates systems, updates posterior distributions, etc.

The obvious problem with this approach is that when k is very large, the computations required of the Master may be so significant that the p Workers are starved for additional simulation assignments. But there is also a more subtle issue. Define the input sequence and output sequence as follows:

Input sequence: X_{ij} is the jth *requested* observation from system i by the Master, with execution time T_{ij}.
Output sequence: Y_{ij} is the jth *returned* observation to the Master from system i.

The validity of a R&S procedure is established based on properties of the *returned* sequence, which will not be the same as the *requested* sequence when there are $p > 1$ Workers and the execution times are random variables. As shown in [13], this can lead to statistical problems, including random sample sizes, non-i.i.d. outputs from any specific system, and a dependence induced across systems outputs by eliminations, all of which invalidate the statistical guarantees of R&S procedures. Of course $X_{ij} = Y_{ij}$ can be assured by having the Master wait for and reorder the output, insuring the statistical validity but signficantly diminishing the computational efficiency.

This suggests that when we have the capability to simulate in parallel we need to refine our goals for R&S. We now formally define a R&S Procedure as the collection of jobs generated by the Master: $\mathscr{J} = \{J_j \colon 1 \le j \le M\}$, where M is determined by the procedure and may be either random or fixed. Both *wall-clock* ending time of the procedure and the *cost* of purchasing time on $p+1$ processors matter:

- Let $0 < T_j < \infty$ be the wall-clock time job J_j finishes, so

$$T_e(\mathscr{J}) = \max_{j=1,2,\ldots,M} T_j$$

is the ending time of the procedure.

Table 1: Existing parallel R&S procedures

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)	Asymptotic Parallel Selection (Luo et al. 2015)
Fixed-Budget	Parallel OCBA (Luo et al. 2000) Asynchronous OCBA/KG (Kamiński & Szufel 2018)	

- $c(p,s)$ = cost to purchase p processors for s time units.
- $t(p,b)$ = maximum time we can purchase on p processors for budget $\$b$

$$t(p,b) = \max\{s\colon c(p,s) \le b\}.$$

We can now define revised objectives:

Fixed precision: Requires statistical guarantees while being efficient.

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

where β_t and β_c are weights or relative costs; typically one of β_t or β_c is zero and the other is one.

Fixed budget: Provides an efficiency guarantee within a budget.

$$\begin{aligned} \text{minimize}_{p,\mathscr{J}} \quad & \mathrm{E}[\underbrace{\mathscr{L}(G^c(\widehat{K},k), \mathscr{J})}_{\text{loss of bad event}}] \\ \text{s.t.} \quad & \underbrace{t(p,b)}_{\text{processor time}} \le t^\star \end{aligned}$$

where $t^\star$ is the wall-clock-time budget.

To the best of our knowledge, no one has yet formulated a parallel R&S procedure specifically to solve one of these optimization problems. Instead, the procedures shown in Table 1 either try to balance the Master-Worker load in a way that keeps the Workers busy, or they weaken the assumptions behind the Generic R&S Procedure so that it is still (at least asymptotically) valid when $X_{ij} \neq Y_{ij}$.

Remark 5. The clever approaches cited in Table 1 all try to adapt the existing R&S paradigms to the parallel environment. However, if we have, say, $k > 1{,}000{,}000$ systems, then is it sensible to insist on locating the single best/near-best with high probability? We should expect many bad systems, but also a lot of good ones. Guarantees like PCS also run counter to approaches in large-scale statistical inference of controlling "error rates." In fact, to control PCS requires more effort/system as k increases, while error rates such as "false discovery" can be attained with little or no "k effect."

4 Thinking Differently

The section is based on [17].

We want to disassemble the R&S paradigm and start over with the expectation of a very large number of systems k and number of parallel processors $p+1$. Our goals are (a) to provide a more scalable—but still useful and understandable—error control than PCS; and (b) avoid coupled operations and synchronization by exploiting the idea of comparisons with a standard [15]. The result is our Parallel Adaptive Survivor Selection (PASS) framework.

Again, let $Y_{i1}, Y_{i2}, \ldots$ be i.i.d. with mean μ_i and from here on we assume $\mu_k > \mu_{k-1} > \cdots > \mu_1$. For some known constant $\mu^\star$ that we refer to as the *standard*, let

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

We will employ a non-decreasing function $c_i(\cdot)$ with the property that

$$\Pr\{S_i(n) \le -c_i(n), \text{ some } n < \infty\} \begin{cases} \le \alpha & \mu_i \ge \mu^\star \\ = 1 & \mu_i < \mu^\star. \end{cases}$$

For normally distributed output such functions can be derived from the results in [4]. Finally, let $\mathscr{G} = \{i\colon\ \mu_i \ge \mu^\star\}$, the set of systems as good or better than the standard $\mu^\star$, which we assume is not empty; if it is empty then there is no false elimination. For any algorithm, let $\mathscr{E}$ be the set of systems that the algorithm decides are are not in $\mathscr{G}$ when they actually are. Then we define the expected false elimination rate for the algorithm as $\text{EFER} = \text{E}[|\mathscr{E}|]/|\mathscr{G}|$.

Before tackling the case of unknown $\mu^\star$, consider the following algorithm:

Parallel Survivor Selection (PSS)

1. given a standard $\mu^\star$, an increment $\Delta n \ge 1$ and a budget
2. let $\mathsf{W} = \{1, 2, \ldots, p\}$ be the set of available Workers; $\mathsf{I} = \{1, 2, \ldots, k\}$ the set of surviving systems; and $n_i = 0$ for all $i \in \mathsf{I}$.
3. until the budget is consumed

a. while an available Worker in W, do in parallel:
   i. remove next system $i \in \mathsf{I}$ and assign to available Worker $w \in \mathsf{W}$
   ii. $j = 1$
   iii. while $j \leq \Delta n$
        simulate Y_{i,n_i+j}
        if $S_i(n_i+j) \leq -c_i(n_i+j)$ then eliminate system i and break loop
        else $j = j+1$
   iv. if i not eliminated then return to $\mathsf{I} = \mathsf{I} \cup \{i\}$
   v. release Worker w to available Workers W

4. return I

Notice that PSS requires no coupling and keeps the Workers constantly busy. And from the properties of $c(\cdot)$, PSS maintains EFER $\leq \alpha$ and, if run forever, will eliminate all systems with means $< \mu^\star$. Further, the EFER is still controlled at $\leq \alpha$ and elimination of systems not in $\mathscr{G}$ still occurs with probability 1, if we let Δn_i depend on the system i, and we replace $\mu^\star$ by $\mu(n) \leq \mu^\star$ where $\mu(n) \to \mu^\star$. This is the case 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 that in the practical case in which $\mu^\star$ is *unknown* we may be able to *learn* the standard in a way that still that achieves our objectives; we call this Parallel *Adaptive* Survivor Selection.

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

- Protect the best: $\mu^\star = \mu_k$, which we focus on here.
- Protect the top b: $\mu^\star = \mu_{k-b+1}$.
- Protect best and everything as good as some known value μ^+: $\mu^\star = \min\{\mu^+, \mu_k\}$.

We want to learn the standard's value in a way that still avoids synchronized coupling but does not compromise the EFER.

Consider PSS but with the adaptive standard

$$\bar{\mu} = \frac{1}{|\mathsf{I}|} \sum_{i \in I} \bar{Y}_i(n_i)$$

which is the average of the sample means of the current survivors. Thus, the adaptive standard acts like a bisection search. We call algorithm PSS with this standard bi-PASS. Under some conditions, including normally distributed output, we can show that the EFER for system k is still $\leq \alpha$ [17]. Thus, we can achieve nearly uncoupled parallelization and controlled EFER with an unknown standard. When $\mu^\star = \mu_k$ this means the chance that we eliminate the best system is $\leq \alpha$. However, since EFER is controlled marginally, α can be set even smaller than the traditional $\alpha = 0.1, 0.05, 0.01$ values with little penalty on efficiency and greater protection for system k.

5 Conclusions

When a simulation optimization problem can be treated as a R&S problem then it can be "solved" with statistical guarantees: that is, all three SO errors can be controlled. High-performance, parallel computing extends the "R&S limit" to larger problems, but introduces new statistical and computational challenges, including violation of standard assumptions and "cost" not being captured by the number of observations. The PASS framework introduced here replaces guarantees like PCS that do not scale well with k, with EFER which does, while at the same time making it easier to achieve "embarassingly parallel" speed up by comparing each system only to an adaptive standard, rather than to each other.

Acknowledgements Portions of this work were supported by NSF Grant CMMI-1537060 and SAS Institute.

References

1. Bechhofer, R., Santner, T., Goldsman, D.: Designing Experiments for Statistical Selection, Screening, and Multiple Comparisons, New York: J. Wiley & Sons (1995)
2. Bechhofer, R.E.: A single-sample multiple decision procedure for ranking means of normal populations with known variances. The Annals of Mathematical Statistics **25**(1), 16–39 (1954)
3. Chen, E.J.: Using parallel and distributed computing to increase the capability of selection procedures. In: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, J.A. Joines (eds.) Proceedings of the 2005 Winter Simulation Conference, pp. 723–731. Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ (2005)
4. Fan, W., Hong, L.J., Nelson, B.L.: Indifference-zone-free selection of the best. Operations Research **64**(6), 1499–1514 (2016)
5. Frazier, P.I.: Decision-theoretic foundations of simulation optimization. In: Wiley encyclopedia of operations research and management sciences. Wiley, New York (2010)
6. Goldsman, D.: Ranking and selection in simulation. In: S. Roberts, J. Banks, B. Schmeiser (eds.) Proceedings of the 1983 Winter Simulation Conference, pp. 387–393 (1983). URL http://informs-sim.org/wsc83papers/1983_0017.pdf
7. Gupta, S.S.: On some multiple decision (selection and ranking) rules. Technometrics **7**(2), 225–245 (1965)
8. Hong, L.J.: Fully sequential indifference-zone selection procedures with variance-dependent sampling. Naval Research Logistics **53**(5), 464–476 (2006)
9. Hunter, S.R., Nelson, B.L.: Parallel ranking and selection. In: A. Tolk, J. Fowler, G. Shao, E. Yücesan (eds.) Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences, pp. 249–275. Springer International Publishing (2017). DOI 10.1007/978-3-319-64182-9_12. URL https://doi.org/10.1007/978-3-319-64182-9_12
10. Jamieson, K., Nowak, R.: Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In: Information Sciences and Systems (CISS), 2014 48th Annual Conference on, pp. 1–6. IEEE (2014)
11. Kamiński, B., Szufel, P.: On parallel policies for ranking and selection problems. Journal of Applied Statistics **45**(9), 1690–1713 (2018)
12. Kim, S.H., Nelson, B.L.: A fully sequential procedure for indifference-zone selection in simulation. ACM Transactions on Modeling and Computer Simulation **11**(3), 251–273 (2001)

13. Luo, J., Hong, L.J., Nelson, B.L., Wu, Y.: Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. Operations Research **63**(5), 1177–1194 (2015). DOI 10.1287/opre.2015.1413
14. Luo, Y.C., Chen, C.H., Yücesan, E., Lee, I.: Distributed web-based simulation optimization. In: J.A. Joines, R.R. Barton, K. Kang, P.A. Fishwick (eds.) Proceedings of the 2000 Winter Simulation Conference, pp. 1785–1793. Institute of Electrical and Electronics Engineers, Inc., Piscataway, NJ (2000). DOI 10.1109/WSC.2000.899170
15. Nelson, B.L., Goldsman, D.: Comparisons with a standard in simulation experiments. Management Science **47**(3), 449–463 (2001)
16. Ni, E.C., Ciocan, D.F., Henderson, S.G., Hunter, S.R.: Efficient ranking and selection in parallel computing environments. Operations Research **65**(3), 821–836 (2017)
17. Pei, L., Hunter, S., Nelson, B.L.: A new framework for parallel ranking & selection using an adaptive standard. In: M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, B. Johansson (eds.) Proceedings of the 2018 Winter Simulation Conference, pp. 2201–2212. IEEE Press (2018)