Challenges in Applying Ranking and Selection after Search

David Eckman
Cornell University, ORIE
dje88@cornell.edu

Shane Henderson
Cornell University, ORIE
sgh9@cornell.edu

November 14, 2016

This work is supported by the National Science Foundation under grants DGE–1144153 and CMMI–1537394.
Motivation

Large-scale problems in simulation optimization:

- Optimize a function observed with simulation noise over a large number of systems.
- Simulation budget only allows for testing a subset of candidate systems.
- Ultimately choose a system as the “best”.

Goal

A finite-time statistical guarantee on the quality of the chosen system relative to the other candidate systems.

- Not interested in asymptotic convergence rates.
- Not interested in finding global optimum.
Approach

1. Identify a set of candidate systems via search.
   - Identify systems as the search proceeds, using observed replications.
   - E.g. random search, stochastic approximation, simulated annealing, Nelder-Mead, tabu search

2. Run a ranking-and-selection (R&S) procedure on the candidate systems.

R&S procedures can safely be used to “clean-up” after search when only new replications are used in Step 2.

Research question

In Step 2, can we reuse the search replications from Step 1 and still preserve the guarantees of the R&S procedure?
Prior work assumes that it is safe to reuse past search replications in making selection decisions:

**After** search

**Within** search
- Pichitlamken, Nelson, and Hong (2006)
- Hong and Nelson (2007)
Our Findings

High-level results
- Reusing search data can result in reduced probability of correct selection (PCS).
- In certain cases, this leads to violated PCS guarantees.

Main findings should extend to selection procedures for non-normal data, e.g. multi-armed bandits in full-exploration.
1 Introduction

2 R&S after Search

3 Search Data

4 Experiments
R&S Procedures

Procedures for sampling from a set of systems in order to ensure a statistical guarantee, typically with respect to selecting the best system.

Typical assumptions:
- Replications are i.i.d. normal, independent across systems.
- Fixed set of $k$ systems with configuration $\mu$.

The space of configurations is divided into two regions:
- Preference Zone (PZ($\delta$)): the best system is at least $\delta$ better than all the others.
- Indifference Zone (IZ($\delta$)): complement of PZ($\delta$).
R&S Guarantees

- Correct Selection (CS): selecting the best system.
- Good Selection (GS): selecting a system strictly within $\delta$ of the best.

Guarantees for a fixed configuration $\mu$

$\mathbb{P}({\text{CS}}) \geq 1 - \alpha$ for all $\mu \in \text{PZ}(\delta)$,  \hspace{2cm} (PCS)

$\mathbb{P}({\text{GS}}) \geq 1 - \alpha$ for all $\mu$,  \hspace{1cm} (PGS)

for $1/k < 1 - \alpha < 1$ and $\delta > 0$.

PGS guarantee is similar to PAC guarantees of multi-armed bandit problems in full-exploration setting.
When the set of candidate systems $\mathcal{X}$ is randomly determined by search, what types of guarantees should we hope for?

**Overall guarantee**

$$\mathbb{P}(\text{GS after Search}) \geq 1 - \alpha.$$ 

**Guarantee conditioned on $\mathcal{X}$**

$$\mathbb{P}(\text{GS after Search} \mid \mathcal{X}) \geq 1 - \alpha \quad \text{for all } \mathcal{X}.$$
PCS Guarantees after Search

Overall guarantee

\[ \mathbb{P}(\text{CS after Search} \mid \mu(\mathcal{X}) \in \text{PZ}(\delta)) \geq 1 - \alpha, \]

Guarantee conditioned on \( \mathcal{X} \)

\[ \mathbb{P}(\text{CS after Search} \mid \mathcal{X}) \geq 1 - \alpha \quad \text{for all } \mathcal{X} \text{ s.t. } \mu(\mathcal{X}) \in \text{PZ}(\delta), \]

Indifference-zone formulation for PCS is ill-suited for the purposes of R&S after search. PGS is a more worthwhile goal.
Example for $k = 3$
What’s the Problem with Search Data?

Observation

The *identities* of returned systems depend on the *observed performance* of previously visited systems.

Search replications are conditionally dependent given the sequence of returned systems.
Adversarial Search (AS)

How AS works:
- If best system looks best $\rightarrow$ add a $\delta$-better system.
- If best system doesn’t look best $\rightarrow$ add a $\delta$-worse system.

Intuition
Weaken future correct decisions and make it hard, if not impossible, to reverse incorrect decisions.

All configurations returned are in $\text{PZ}(\delta) \Rightarrow \text{PCS} = \text{PGS}$.

AS doesn’t satisfy our definition of search, but can still be used for near-worst-case analysis.
Test R&S procedures in two settings:

1. After AS, reusing search data.
2. Slippage configuration (SC):

\[ \mu[i] = \mu[k] - \delta \quad \text{for all } i = 1, \ldots, k - 1. \]

(*PCS in the SC is a lower bound on PCS in PZ(\(\delta\))*)

Estimate overall PCS over 10,000 macroreplications.

Set \(1 - \alpha = 0.95\), \(\delta = 1\), \(\sigma^2 = 1\), and \(n_0 = 10\).
Selection: Bechhofer

![Graph showing PCS vs No. of Returned Systems (k)]

- Bechhofer
- PCS
- No. of Returned Systems (k)
- AS
- SC
Selection: Rinott
Subset-Selection: Modified Gupta

Modified Gupta

Empirical PCS

No. of Returned Systems (k)

1.0

0.9

0.8

0.7

0.6

0.5

0.4

0.3

0.2

0.1

0

0

200

400

600

800

1000

AS

SC
Subset-Selection: Screen-to-the-Best

![Graph showing Screen-to-the-Best](image.png)

- **Empirical PCS** as a function of the **No. of Returned Systems (k)**
- Comparison between **AS** and **SC** approaches

**Introduction**

**R&S after Search**

**Search Data**

**Experiments**
Maximize $\lceil \log_2 x \rceil$ on the interval $[1/16, 16]$.

- Start at $x_1 = 0.75$ and take $n_0 = 10$ replications.
- Choose a new system uniformly at random from within $\pm 1$ of best-looking system.
A Realistic Search Example

Modified Gupta for Realistic Search

Probability of Selection Event

No. of Returned Systems (k)

PCS | \( \mu \) in PZ(1)

PGS
Conclusions

Main take aways

Care should be taken when reusing search replications in R&S procedures. **Efficiency** at the expense of a **statistical guarantee**. For practical problems, reusing search data is likely fine.

Open questions:

- Does dependent search data cause issues with R&S procedures that use common random numbers?
- **Can R&S procedures be designed to safely reuse search replications?**