SEQUENTIAL RANKING AND SELECTION PROCEDURES AND SAMPLE COMPLEXITY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Sijia Ma

September 2018

© 2018 Sijia Ma
ALL RIGHTS RESERVED

SEQUENTIAL RANKING AND SELECTION PROCEDURES AND SAMPLE COMPLEXITY

Sijia Ma, Ph.D.

Cornell University 2018

Ranking and selection (R&S) procedures are widely used for selecting the best among a set of candidate systems, where each candidate system is associated with a simulation model. In this thesis, we focus on three aspects on the sample complexity of the R&S problem. First, we develop a method for predicting the sample complexity. Second, we present Envelope Procedure (EP), a R&S procedure that delivers a probably approximately correct selection guarantee, and we provide a high probability upper bound on its sample complexity. We also prove a lower bound on the sample complexity for general R&S procedures. The performance of the EP is demonstrated by numerical experiments. Finally, we discuss some specific aspects and features of the EP in parallel computing environment and the sampling rules.

BIOGRAPHICAL SKETCH

Sijia Ma grew up in Baoding, an old city in China. He spent the first 18 years of his life in that city before he attended Xi'an Jiaotong University, which locates in Xi'an, an even older city in China. After graduating with honors in Information and Computational Sciences, he came to the U.S. for further study. After receiving a M.S. in Scientific Computing from the New York University, he came to Cornell for PhD study.

During the four years spent in Ithaca, Sijia took ski classes every winter and likes skiing in the Greek Peak a lot. He likes the beautiful scenery in this small town, and had a lot of wonderful memories there. He also enjoyed his spring and fall breaks in national parks and skiing resorts. Upon graduating from Cornell, Sijia will move to Bejing, China and begin his career as a machine learning software engineer at Google China AI center.

To my parents and my fiancé.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Shane Henderson, for the patient guidance, consistent encouragement and invaluable advice he has provided throughout my PhD studies. His enthusiasm, humor and immense knowledge have been a great help of my research work. I have been extremely lucky to have him as my advisor.

I would like to extend my sincere thanks to the rest of my committee members: Sidney Resnick and Yudong Chen, for their inspiring comments and insightful questions. Besides them, I would also like to thank Pierre Patie and Madeleine Udell, with whom I had a great pleasure to work as a teaching assistant.

I am also grateful to my colleagues and friends at Cornell, especially David Eckman, Pu Yang, Tiandong Wang and Yuhang Ma, who have always been a source of support and help for my research and graduate student life.

I thank the School of Operations Research and Information Engineering, not only for offering me the environment to undertake this research, but also for giving me the opportunity to meet so many outstanding and interesting people here.

Special thanks go to my fiancé, Zelin Li. This dissertation would not have been possible without her warm love and endless support. Last but not the least, I am very much indebted to my parents, Xiaoduo Zhao and Shiying Ma, who have been offering me unconditionally help throughout my life.

This research was supported, in part, by National Science Foundation grant CMMI 1537394 and Army Research Office grant W911NF-17-1-0094.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background

The simulation optimization (SO) problem integrates stochastic simulation into a nonlinear optimization problem whose objective function is not explicitly available but can be measured with error by Monte Carlo simulation. An example of such a problem is ambulance planning [45]. The decision-makers of ambulance organizations need to decide the locations and schedules of the ambulances in order to minimize a quantile of the response times. For any candidate plan, its performance over a fixed time horizon is analytically intractable. Stochastic simulation can be used to approximately evaluate the objective function in an effort to solve this optimization problem. Such problems arise in a wide variety of areas, e.g., supply chain management [31], transportation, and public health [1, part 4]. For more examples of SO problems, see [23].

Ranking and selection (R&S) problems are a special class of SO problems in which: (1) the number of feasible solutions is finite, (2) no structural properties, e.g., convexity, of the objective are assumed, and (3) the computational budget permits at least some simulation of every system. The goal is to identify the best among a finite set of systems where the performance of each system can only be observed by simulation. This is an important class of problems, since in many

practical applications the structural properties are non-existent or very difficult to verify. The main issue in R&S problems is how to allocate computational budget for simulation of the systems so that a statistically reliable choice of the best system can be determined efficiently. A good R&S procedure achieves a balance between the total running time and the quality of the ultimate selection. For an overview of this area, see [26, 9] for context, and see [4, 21] for book-level treatment.

Many procedures have been proposed for dealing with R&S problems. They can be classified into Bayesian approaches and frequentist approaches. Bayesian approaches usually aim to optimize an objective that penalizes suboptimal choices, as estimated through the posterior distribution. For example, OCBA (Optimal Computing Budget Allocation) [10] allocates a computational budget to maximize approximations of the posterior Probability of Correct Selection (PCS). Other Bayesian procedures [13, 12] allocate samples to minimize the expected opportunity cost, or to maximize a measure of the expected value of information [11].

Frequentist approaches usually provide a certain statistical guarantee on the quality of the selected system irrespective of the unknown problem configuration. An important class of such procedures are indifference-zone (IZ) procedures. IZ procedures originated with [2] and have been well studied; see, e.g., [49, 52, 36, 24, 18]. They guarantee to select the unique best system with at least a prescribed PCS, assuming that the difference between the best and all others is sufficiently large. To be more precise, let μ_i denote the true performance (usually an expectation)

of the ith system. For notational simplicity, suppose that, unknown to the R&S procedure, the systems are indexed so that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$, and System k is the best. A R&S procedure provides a PCS guarantee at level α if

$$\mathbb{P}(I^* = k) \geq 1 - \alpha, \text{ if } \mu_k - \mu_{k-1} > \delta,$$

where I^* is the (random) index of the selected system, the parameter δ is called the IZ parameter, and $1 - \alpha$ is the confidence level. It is natural to require $1 - \alpha > 1/k$, since otherwise the procedure is no better than random guessing. Therefore throughout the thesis we assume that $\alpha < 1 - 1/k$, i.e., α is bounded away from 1 for fixed k. The IZ guarantee only holds when the difference between the best and second-best systems is greater than δ; nothing is guaranteed otherwise. There are also some recent IZ-free frequentist procedures [17] that deliver PCS without imposing an IZ restriction.

A stronger form of guarantee that holds for any configuration of means, and that also implies a PCS guarantee when $\mu_k - \mu_{k-1} > \delta$, is probably approximately correct (PAC) selection, which is also referred to as a probability of good selection (PGS) guarantee in, for example, [46, 47]. It guarantees, with high probability, to select a system whose performance is not too far away from that of the best system, i.e., that

$$\mathbb{P}(\mu_{I^*} > \mu_k - \delta) \geq 1 - \alpha,$$

irrespective of the gap between the best and the other systems. It is not hard to see that a procedure that delivers a PAC guarantee automatically delivers a PCS guarantee, but the converse does not necessarily hold. R&S procedures that pro-

vide PAC guarantees are far less prevalent than those providing PCS guarantees, perhaps due to the difficulty in establishing such guarantees; see [15] for a survey.

The R&S problem is closely related to the multi-armed bandit (MAB) problem. They both originate from the work of [2] and [49]. An important difference is that, in MAB problems, one typically considers the cumulative reward collected by pulling each arm (i.e., obtaining a sample from each system) throughout the experiment, instead of focusing only on the quality of the final decision as in R&S [50]. In pure-exploration instances of the MAB, the goal is to find the best arm /system at the end of the procedure, which is the same overall objective as in R&S problems, but still, different distributional assumptions are made about the underlying arms/systems. Algorithms developed for the bandit problem usually assume bounded or sub-Gaussian simulation outputs with a known bound on the variances. In addition, the algorithms for the MAB problem usually simulate one system at a time. In comparison, the R&S literature typically assumes Gaussian error, uses batching to approximately ensure this when needed, and takes samples from a set of systems in each round.

A focus of the MAB literature is the analysis of the sample complexity. The median elimination procedure of [16] was shown to deliver the PAC guarantee in $\mathcal{O}(k/\delta^2 \log(1/\alpha))$ arm pulls. Since then, the upper bound on the sample complexity for this problem or the no-relaxation form of this problem, i.e., $\delta = 0$, has been successively improved by the work of [33], [27], [34] and [28]. The upper bound on the sample complexity, i.e., how many samples are needed to deliver

such a guarantee, has also been studied in [16] [44] and [28]. These works provide a theoretical analysis of the relation between the sample complexity and the problem configuration. In the R&S literature, there has also been some research on the asymptotic upper bound [17] and lower bound [30]. However, in general this aspect is largely overlooked, and the computational efficiency of the R&S procedures is usually only demonstrated through numerical experiments. We consider the study of the sample complexity as an important issue in the R&S problem since it helps us better understand the procedures and the underlying difficulty of problem instances. In addition, studying the upper and lower bounds suggests where the limits of a R&S procedure lie and the directions for improving the efficiency of existing procedures.

Originally, R&S procedures were only designed for problem instances where the number of systems is small, due to the limited computing capability and the requirement of the statistical guarantee even for the worst-case configuration. With the introduction of sequential R&S procedures, the sample complexity of the R&S procedures has decreased dramatically since samples can be sequentially allocated only to superior systems due to the sequential update of information. As a result, R&S procedures have become more efficient and can be applied to problems of larger scale.

In the past decade, with the significant advances in computing power and the advent of parallel computing, we can aim for solving much larger R&S problems. However, converting the current serial R&S procedures to their parallel

versions can be much more complicated than it seems. The critical question is, can we fit the current serial R&S procedures into a parallel computing environment to achieve high efficiency while preserving statistical validity? As pointed out in [41], the main difference in a parallel computing environment is that "the input and output sequences of observations are different on multiple processors, whereas they are the same on a single processor." For this reason, simply allocating the simulation tasks to different cores without requiring any synchronization may invalidate the statistical guarantee. However, excessive synchronization leads to low efficiency. Furthermore, screening work is in general harmless in serial R&S procedures since its running time is usually negligible compared to that of the simulations, but in a parallel computing environment it can become the bottleneck as discussed in [47]. How to design a parallel R&S procedure is one goal of this thesis.

Predicting sample complexity is another issue that emerges with the introduction of parallel R&S procedures, since the sample complexity determines the amount of computational resources needed for a R&S procedure to terminate in a reasonable time period. This information is unknown to the user of a parallel R&S procedure. If a customer wants to buy some cloud computing service to solve a R&S problem in 48 hours, he needs to predict the sample complexity to determine the number of cores required and hence the cost. However, this task is surprisingly nontrivial. In order to predict the sample complexity, a preliminary sampling stage is needed to estimate the configuration of the problem it is solving. Due to the small sample size of the preliminary stage, small differ-

ences between the system means can be swamped by noise, an effect we call the "means-spreading" phenomenon. Means-spreading leads us to underestimate the running time by overestimating the difference between the best system and the others.

1.2 Contributions

In this thesis, we focus on three aspects of the R&S problem, each of which is a relatively self-contained work. In the following chapters, we make the following contributions.

Predicting the Sample Complexity. We propose a method for predicting the the sample complexity of any given R&S procedure for any given problem. This method effectively reduces the impact of the "means-spreading" phenomenon, so that it provides a better understanding about the true configuration of the problem, and hence a better prediction of the sample complexity. Additional benefits of estimating the configuration of the problem include improving the efficiency of the procedure by wisely determining the values of free parameters, e.g., $\bar{r}$ in [47], and providing a progress bar throughout the procedure for showing the estimated remaining time. Numerical experiments demonstrate the quality of our prediction method.

The Envelope Procedure. We present the Envelope Procedure (EP), a fully sequential procedure that provides a PAC guarantee. It consists of a stopping rule and a sampling rule. The stopping rule is the base of the PAC guarantee and it affords considerable flexibility in designing the sampling rule. Thus, the EP is actually a family of procedures. A high probability upper bound on the sample complexity of the EP with a specific sampling rule is proved, which relates the problem configuration to the sample complexity. We also provide a nearly matching lower bound on the sample complexity that any PAC procedure must satisfy. Computational experiments show that the EP is more efficient than some leading R&S procedures across various problem configurations. We also compare EP with those procedures by realized PAC, and the performance of the EP is even better. A heuristic version of the EP is developed based on this observation and it shows a clear improvement in efficiency over the original EP.

Parallel Envelope Procedure and Sampling Rules. We discuss some specific aspects and features of the EP in a parallel computing environment. Specifically, our parallel EP (1) adopts the master-worker framework, (2) uses a vector-filling method to deal with the random completion issue, (3) considers information updates in sequential job assignment, and (4) eliminates inferior systems for screening efficiency. We combine all of these features together to develop a parallel Envelope Procedure. We also propose the Gap-Minimization sampling strategy designed for serial computing environments, and the No-Waste sampling strategy designed for parallel computing environments. We compare the parallel EP

and the vector-filling KN (VKN) procedure in a computational study in a parallel computing environment and the results show that the parallel EP outperforms the VKN in sample size, wall-clock time, and utilization in various configurations.

CHAPTER 2

PREDICTING THE SAMPLE COMPLEXITY IN RANKING AND SELECTION PROCEDURES

Historically, due to limited computing capability, the interest in R&S problems primarily focuses on small-scale problems where the number of systems and the number of simulations of each system are relatively small. Nowadays, with improvements in computing power, the introduction of cloud computing and the development of efficient algorithms, much larger problems can be solved [47, 41]. However, new issues also emerge as the scale increases. Estimating the sample complexity, i.e., the number of replications of each system required to make a selection is key, because simulation budget is almost always the dominant contributor to running time, and the cost of running a task on the cloud depends on running time. Knowledge of the simulation budget also helps us decide the number of cores needed for the procedure to terminate in a reasonable time period in a parallel computing environment. For these reasons, a methodology for sample complexity estimation for R&S procedures is important for large-scale problems.

The sample complexity of a procedure usually depends heavily on the configuration of the problem it is solving, i.e., the means and variances of these systems, which are unknown. Therefore a preliminary sampling stage is needed in order to get some knowledge about the structure of the problem. Non-trivial issues arise in this process. Since the sample size of the preliminary stage is usually very small, small differences in performances of systems are usually dominated

by noise. With the naïve way of estimating expectations by sample means, we tend to overestimate the differences between the best system and the others, and hence underestimate the running time. An extreme case is the slippage configuration in which all but one system have the same expectation and the other system is only a little better than the others. It is almost impossible to discern such a configuration with small sample sizes, since the sample means will vary. In this chapter we propose a method to better deal with this "means-spreading" phenomenon, so that we have a better understanding about the true configuration of the problem. Our focus throughout the chapter is primarily on estimating the means of the systems, rather than the variances, because the means tend to have a more dramatic effect on running times. For example, in order to distinguish the difference between two normal populations having means μ_1 and μ_2, and variances σ_1^2 and σ_2^2, one can take n replications of each system until a statistically significant difference is seen in the sample means. The sample size n that is needed is on the order of $(\sigma_1^2 + \sigma_2^2)/(\mu_2 - \mu_1)^2$. Therefore, there is greater sensitivity to errors in predictions of means, rather than errors in predictions of variances, explaining our focus on means in this chapter

Based on a limited sample size in the preliminary stage, it is extremely difficult to estimate the configuration of the problem and hence the sample complexity very accurately. Even if the system configuration is given, we still do not know the exact total number of samples due to the randomness of simulation output. However, our goal in estimating the sample complexity is to better allocate computing resources for solving the problem, so we only care about the order of magnitude

of the sample complexity instead of its exact value. For example, we want to be able to predict whether the procedure will require 100 replications or 1000 replications, rather than 100 replications or 110 replications. In the latter case we are unlikely to alter the number of cores allocated.

In this chapter we focus on estimating sample complexity of procedures rather than the more desirable goal of estimating running times, partly because the two are so closely related. Moreover, there are complexities in estimating running times that make them difficult to analyze. For example, synchronization issues, the details of one's parallelization scheme and the implementation of the procedure all can heavily affect the wall-clock running time of procedures running in a parallel computing environment. We thus focus on estimating the simulation budget, taking the pragmatic, but not perfect, view that one can obtain an imperfect estimate of the running time by scaling simulation budgets by observed average running times per replication. This imperfect estimate does not account for the complexities mentioned above, but it can be viewed as an "approximate lower bound" on the running time, which suffices for our purpose.

There is an additional benefit of estimating the configuration of the problem. With a better understanding of the problem based on preliminary samples, we may be able to improve the efficiency of the R&S procedure. For example, for procedures with free parameters, e.g., [47], taking advantage of the knowledge of the configuration helps us determine the values of parameters ($\bar{r}$ in [47]) in such a way as to improve the procedure's efficiency. And for Bayesian procedures, the

prior might be partially based on the estimated problem configuration.

The efficiency of a procedure might be impaired by the extra samples in the preliminary stage. Indeed, for small problems that can be solved in seconds, efficiency is probably not very important, and estimation of the running time might not be necessary. However, for large-scale problems the improvement of efficiency can be large even with the overhead brought by the extra samples. And there may be no need to discard the samples observed in the preliminary stage as long as we are not using the information provided by those samples for the procedure itself, e.g., to determine free parameters for the procedure. Therefore, we can just include those samples in the main body of the procedure and hence nothing is wasted.

Besides estimating the running time before the whole procedure starts, we can also update our estimate of the remaining time during the process to inform the user of progress through some kind of "progress bar.". With more and more data observed, we expect that the estimate will be more and more accurate.

Our algorithm provides a point estimate of the problem configuration and the running time. One might desire more information, through confidence regions or belief distributions on the configuration, for example. Beyond repeatedly running our algorithm, we do not offer a practical method for obtaining such, more complex, information.

2.1 Preliminaries

Suppose we have k simulated systems, from which we can take samples by simulation to evaluate their performance. Let X_{ij} be the j-th sample taken from System i, for $i = 1, \ldots, k$, $j = 1, 2, \ldots$. We assume $\mathbf{X}_i = (X_{ij} : j \geq 1)$ is an i.i.d. sequence of samples drawn from $\mathcal{N}(\mu_i, \sigma_i^2)$, a normal distribution with mean μ_i and variance σ_i^2, and moreover, $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X}_k$ are independent. We denote $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k)$ and $\boldsymbol{\sigma^2} = (\sigma_1^2, \ldots, \sigma_k^2)$ as the expectation vector and variance vector, which together are referred to as the *system configuration*. Our goal is to identify the system with largest expectation with a certain given level of confidence $1 - \alpha$.

There are two primary forms of frequentist probabilistic guarantees: the probability of correct selection (PCS) and probably approximately correct (PAC) selection. For a given confidence level $1 - \alpha$, the PCS guarantee means that the probability of choosing the best system (the system with the highest expectation) is no less than $1 - \alpha$, if the difference between the expectations of the best and others is greater than or equal to δ, the *indifference-zone parameter*. The PAC guarantee does not require such an assumption on the configuration of the problem. It means that the probability of selecting a system with expectation within δ of the best is no less than $1 - \alpha$.

In practice simulation outputs may not be normally distributed, and then these probabilistic guarantees may not necessarily hold. A commonly used strategy is to draw a batch of samples at a time and take their mean as one output X_{ij}, so

that the central limit theorem (CLT) ensures that the outputs are approximately normally distributed, and the guarantees might then hold approximately.

The sampling rule of a R&S procedure specifies how to allocate samples to each of the k systems, either deterministically or adaptively based on the samples that have already been observed. The R&S procedure terminates when the observed statistics are sufficient to deliver the pre-specified probabilistic guarantee. The sampling rule and stopping rule are designed to ensure the validity of the procedure, and together determine the efficiency of the R&S procedure, i.e., how many samples are needed to identify the best system.

Most efficient R&S procedures adopt an elimination strategy, i.e., identifying inferior systems to eliminate during the procedure as more samples are observed, so that those eliminated systems will not be considered as candidates and will no longer be sampled. At the end of the procedure, all but one system are eliminated, and the sole survivor is selected as the best.

2.2 Estimating the Total Number of Samples

R&S procedures essentially are composed of three tasks: (1) deciding the number of samples to draw from each system, (2) running simulations, and (3) screening (making elimination decisions), which includes computing and comparing statistics based on the simulation results. Usually, the computational cost for task (2)

dominates, especially for large-scale problems, which is our focus in this article. Therefore, the essential question to ask in estimating the running time is how many samples are needed for the procedure to terminate and how long one replication takes.

The number of samples depends on the configuration of the given problem, i.e., the expectation vector $\boldsymbol{\mu}$ and the variance vector $\boldsymbol{\sigma^2}$, which are both unknown. Therefore we use a preliminary stage to estimate the system configuration; see Section 4. For now we just assume that we have already obtained an estimate of the system configuration, based on which we present two approaches for estimating the total number of samples: (1) a first-order estimate, and (2) simulating the simulation process. The first approach simplifies the problem by ignoring the fluctuation in elimination time due to randomness in samples. The second approach simulates the whole R&S procedure, but instead of simulating the systems to get samples, it generates normally distributed samples directly from the estimated problem configuration.

In the preliminary stage, we estimate both the system configuration and the average simulation time $\bar{T}$, i.e., the amount of time spent taking one sample. Then the estimated total running time $\hat{T}_{\text{total}}$ is given by

$$\hat{T}_{\text{total}} = \hat{N}\bar{T},$$

where $\hat{N}$ is the estimated total number of samples. When the simulation time varies from system to system, we record $\bar{T}_i$'s separately for each System i, and

then

$$\hat{T}_{\text{total}} = \sum_{i=1}^{k} \hat{N}_i \bar{T}_i,$$

where $\hat{N}_i$ is the estimated number of samples from System i.

In a parallel computing environment, other issues such as latency and synchronization also affect scalability and wall-clock time. Both highly depend on the algorithm itself, its implementation and the configuration of the problem. We ignore those issues, so that we get what might be viewed as an approximation of a lower bound on the running time when we convert our estimation to the parallel computing environment directly by dividing the total running time by the number of cores. In general, one might first estimate the total running time with our methodology, and second, decide the number of cores needed to finish the job in an acceptable time.

2.2.1 First-order estimate

Under our assumptions the accumulated sums of the system outputs $(Y_i(n) : n \geq 0)$ with

$$Y_i(n) = \sum_{j=1}^{n} X_{ij}, i = 1, \ldots, k,$$

or the differences between pairs of such accumulated sums are random walks with normally distributed increments. R&S procedures explicitly [36, 24] or implicitly [18] define a continuation region, whereby a system is eliminated once the corresponding random walk hits the boundary $f(n, \theta)$ of the continuation region,

where the parameter θ can involve statistics of the outputs, e.g., sample variances or accumulated sums of other systems. Hence, estimating N_i, the total sample size of System i, is equivalent to the problem of estimating a first-passage time of the corresponding random walk.

The expected first-passage time for simple cases (e.g., constant and linear boundaries) are well-understood, but other than that, closed-form formulae for few boundaries are known [32]. For mathematical tractability, we simplify the problem to a first-order estimate, which means replacing the random walks $(Y_i(n), n \geq 0)$ by linear functions $(y_i(n) = n\mu_i, n \geq 0)$. In other words, we ignore the fluctuation of the random walks when estimating the first-passage time.

Why is this reasonable? The law of the iterated logarithm [7, 54] establishes that when $\sigma_i > 0$,

$$\limsup_{n\to\infty} \frac{|Y_i(n) - \mu_i n|}{\sigma_i \sqrt{n \log\log n}} = \sqrt{2}, \quad \text{a.s.}$$

Therefore for systems with large total sample size N_i, it is rare to see fluctuations go beyond the order of $\sqrt{n \log\log n}$ in magnitude, and the drift μ_i dominates. Therefore, relatively speaking, the performance of those systems will be close to their expected performance. When we focus on large-scale problems, most of the samples are contributed by those systems with large N_i, and so this simplification is reasonable. In addition, R&S procedures are designed to eliminate inferior systems based on their expectations irrespective of their variances, and so continuation regions are designed to be large enough to identify the average performance, irrespective of fluctuations, with high probability. Hence, it seems reasonable to

neglect the randomness in determining when the random walk exits the continuation region. We will later see in numerical experiments that this seemingly rough simplification usually gives good results.

2.2.2 Simulation of the simulation process

Based on the estimated system configuration, we can simply run the whole R&S procedure for the original problem, replacing simulation by samples from a normal distribution with our estimated expectations and variances. In other words, we "simulate" the whole R&S procedure. Actually this strategy has been used before to test R&S procedures, e.g., [36] and [18].

Usually R&S procedures spend a majority of their running time on simulations, which is skipped in this approach. Therefore we expect it to be faster than the procedure itself, which is essential. The screening time of some R&S procedures is $O(k^2)$, for example, the original KN procedure [36] and [24]. For those procedures, this simulation approach may take a long time for large-scale problems, compared to those procedures with screening times that are linear [18] or approximately linear [47] in the number of systems. In these $O(k^2)$ procedures, we recommend using the first-order estimate approach if k is large.

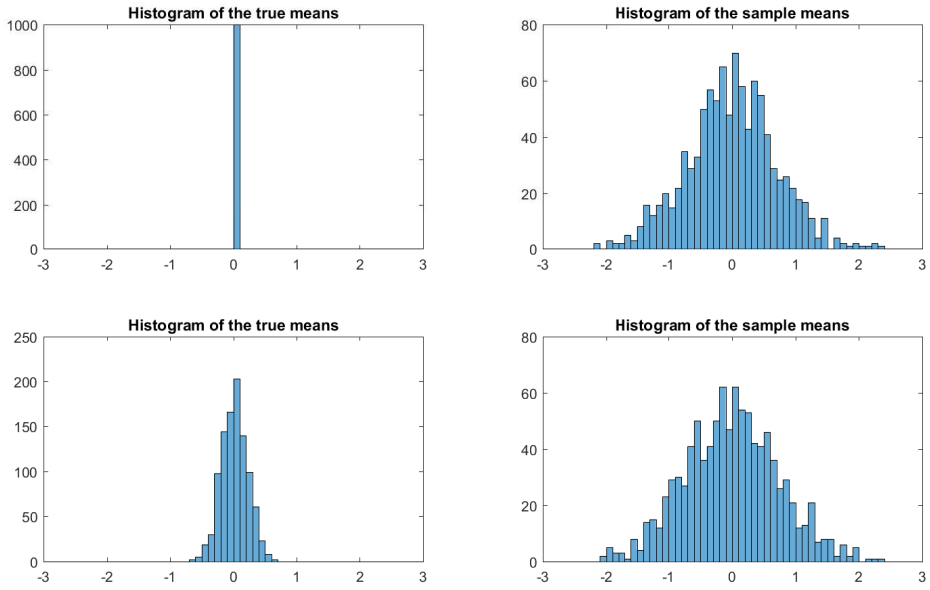

Figure 2.1: Histograms of expectations and sample means under the following configurations: SC (row 1) and RPI (row 2). $k = 1000$, $\delta = 0.1$, $\sigma_i^2 = 25$, $n_0 = 50$.

2.3 Estimation of the Ordered Expectations

2.3.1 Issues with Naive Estimation

As we see from the previous section, the estimation of the total number of samples N depends on the configuration $(\boldsymbol{\mu}, \boldsymbol{\sigma^2})$ of the problem. The configuration is unknown, so we use a preliminary stage in which we sample each system n_0 times. For most R&S procedures that deal with unknown variances, there is such a stage for estimating variances. We can just make use of those samples taken for

estimating variances for our preliminary stage so that no extra samples should be taken. However, if the estimate of running time will be used for determining the value of some free parameters in the R&S procedure, then usually those samples have to be dropped in order to ensure the statistical guarantee.

Recall that our goal is to answer questions like, how many samples are needed for the best system to eliminate the second best one. Therefore, instead of estimating the μ_i's, we only care about estimating the ordered expectations $\mu_{(i)}$'s without the need to know their identities. For σ_i^2's, we use sample variances as our estimators and pair them, in the natural way, with the ordered expectations by the order of the corresponding sample means. For example, if the sample mean $\bar{X}_i(n_0)$ of System i is the second largest, we pair its sample variance S_i^2 with the estimated second largest ordered expectation $\hat{\mu}_{(2)}$. The same pairing strategy can be used for the average simulation time $\bar{T}_i$ as well.

With n_0 samples from each of the systems, the most straightforward estimators of their expectations are the sample averages $\bar{X}_i(n_0), i = 1, 2, \ldots, k$. Each sample average is an unbiased estimator of the expectation, but the estimated expectations tend to "spread out", so the ordered sample averages are not unbiased estimators of the ordered expectations. An extreme case is the slippage configuration (SC), in which $\mu_1 = \delta$ $\mu_i = 0$, $\forall i \neq 1$, where $\sigma_i^2 = \sigma^2$ are equal. The sample means are independent and normally distributed with variance σ^2/n_0, hence the differences between their true means are exaggerated. We call this phenomenon "means-spreading". Histograms of expectations and sample means are given as

illustration in Figure 2.1. The first row is for the SC, and the second row is for the random problem instances (RPI) configuration, where $\mu_i \sim \mathcal{N}(0, 4\delta^2)$ and are independent in i. As we can see, the sample means of both configurations in the right-hand panels are more spread out than their corresponding true expectations in the left-hand panels.

In general, for any configuration of the problem,

$$\mathbb{E} \max_i \bar{X}_i \geq \max_i \mathbb{E}\bar{X}_i = \max_i \mu_i, \text{ and}$$

$$\mathbb{E} \min_i \bar{X}_i \leq \min_i \mathbb{E}\bar{X}_i = \min_i \mu_i,$$

suggesting that means-spreading is ubiquitous. (We will later characterize the difference between $\mathbb{E} \max_i \bar{X}_i$ and $\max_i \mu_i$.) It is therefore reasonable to expect that we are being overly optimistic if we treat the sample means as true expectations, since the advantage of the best system over other systems is magnified, leading to an underestimation of $\hat{N}$. For configurations in which the expectations are close to each other, e.g., SC, the estimated total number of samples $\hat{N}$ can be tremendously smaller than what is needed, as we will see in numerical experiments.

The impact of means-spreading depends on how clustered the true expectations are relative to their true variances. If the expectations are very close to each other and the variances are large, then the sample means will be very spread-out. In general, for a fixed δ, the smaller the differences between the expectations and the larger the variances, the harder the problem, in the sense that it usually takes more samples to find the best system.

2.3.2 Regression Method for Estimating the Ordered Expectations

We have seen that recovering the ordered expectations solely from the sample means $\bar{X}_i(n_0)$ can perform badly. Fortunately, we have the full process: $(X_{ij}, i = 1, 2, \ldots, k, n = 1, 2, \ldots, n_0)$ and hence $(\bar{X}_i(n), i = 1, 2, \ldots, k, n = 1, 2, \ldots, n_0)$. Our goal is to understand the relation between ordered sample means and the ordered true expectations, especially those around $\mathbb{E}\max_i \bar{X}_i$ versus $\max_i \mu_i$.

First consider the case where all the μ_i are the same. Assuming all X_i's are independent and normally distributed, the distribution function of $M = M(n) = \max_{i=1,\ldots,k} \bar{X}_i(n)$ is given by

$$\begin{aligned} F_M(x) &= P(M(n) \le x) \\ &= \prod_{i=1}^{k} P(\bar{X}_i(n) \le x) \\ &= \prod_{i=1}^{k} \Phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right). \end{aligned}$$

The probability density function of M is then

$$\begin{aligned} f_M(x) &= \frac{d}{dx} \prod_{i=1}^{k} \Phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right) \\ &= \sum_{i=1}^{k} \left[\prod_{j\neq i} \Phi\left(\frac{x-\mu}{\sigma_j/\sqrt{n}}\right)\right] \phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}}\right) \frac{1}{\sigma_i/\sqrt{n}}. \end{aligned}$$

Therefore

$$\begin{aligned}\mathbb{E}M(n) &= \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x-\mu}{\sigma_j/\sqrt{n}} \right) \right] \phi\left(\frac{x-\mu}{\sigma_i/\sqrt{n}} \right) \frac{1}{\sigma_i/\sqrt{n}} dx \\ &= \mu + \frac{1}{\sqrt{n}} \int_{-\infty}^{\infty} y \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{y}{\sigma_j} \right) \right] \phi\left(\frac{y}{\sigma_i} \right) \frac{1}{\sigma_i} dy \quad (\text{let } y = \frac{x-\mu}{1/\sqrt{n}}) \\ &= \mu + \frac{C}{\sqrt{n}}\end{aligned}$$

where

$$C = C(\boldsymbol{\sigma}) = \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x}{\sigma_j} \right) \right] \phi\left(\frac{x}{\sigma_i} \right) \frac{1}{\sigma_i} dx$$

is a constant that does not depend on n.

The value of C is usually unknown since we do not know the variances σ_i. However, we do not need to calculate it explicitly since we only care about the expectation μ. For each n, the corresponding largest sample mean $M(n)$ is known from the data, and so we can apply linear regression to estimate μ, using the model $M(n) = \mu + Cx(n)$ with $x(n) = n^{-1/2}$ to get an unbiased estimator of μ, which is the estimate of the intercept. An illustration of the regression method is given in Figure 2.2. In this example, the intercept of the fitted line, which is also the estimated $\hat{\mu}$, is 0.02, which is very close to its true value 0.

For general configurations, we similarly have

$$F_M(x) = \prod_{i=1}^{k} \Phi\left(\frac{x-\mu_i}{\sigma_i/\sqrt{n}} \right),$$

and

$$f_M(x) = \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x-\mu_j}{\sigma_j/\sqrt{n}} \right) \right] \phi\left(\frac{x-\mu_i}{\sigma_i/\sqrt{n}} \right) \frac{1}{\sigma_i/\sqrt{n}},$$

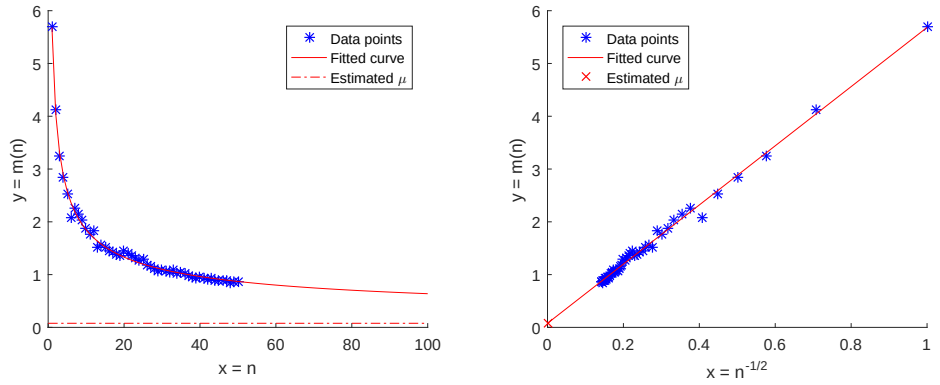

Figure 2.2: Regression method to estimate μ when all $\mu_i = \mu = 0$ and $\sigma_i^2 = \sigma^2 = 4$ are equal. Here $k = 1000$ and $n_0 = 50$. The estimated $\hat{\mu} = 0.08$.

so that

$$\begin{aligned}
\mathbb{E}M(n) &= \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x - \mu_j}{\sigma_j / \sqrt{n}} \right) \right] \phi\left(\frac{x - \mu_i}{\sigma_i / \sqrt{n}} \right) \frac{1}{\sigma_i / \sqrt{n}} dx \\
&\quad \left(\text{let } y = \frac{x - \mu_{(k)}}{1/\sqrt{n}} \right) \\
&= \mu_{(k)} + \frac{1}{\sqrt{n}} \int_{-\infty}^{\infty} y \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{y - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) \right] \phi\left(\frac{y - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) \frac{1}{\sigma_i} dy \\
&= \mu_{(k)} + \frac{D(n)}{\sqrt{n}}
\end{aligned}$$

where

$$\begin{aligned}
D(n) &= D(n, \boldsymbol{\mu}, \boldsymbol{\sigma}) \\
&= \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) \right] \phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) \frac{1}{\sigma_i} dx
\end{aligned}$$

In this case, the slope $D(n)$ is not constant. Moreover, its expression is compli-

cated and mathematically intractable. What if we simply treat it as constant in n and apply linear regression as with the case of identical expectations? Denoting $\hat{\mu}_{(k)}$ as our estimator of $\mu_{(k)}$ by linear regression, the following theorem can help us understand the relation between $\hat{\mu}_{(k)}$ and $\mu_{(k)}$.

Theorem 1. *$D(n)$ is non-negative, non-increasing in n, and*

$$\lim_{n\to\infty} D(n, \boldsymbol{\mu}, \boldsymbol{\sigma}) = C(\tilde{\boldsymbol{\sigma}}),$$

where the components of $\tilde{\boldsymbol{\sigma}}$ are those σ_i's whose corresponding μ_i's are equal to $\mu_{(k)}$.

Remark: The dimension of $\tilde{\boldsymbol{\sigma}}$ will usually be smaller than k. For example, if there are three systems with expectation equal to $\mu_{(k)}$, then the dimension of $\tilde{\boldsymbol{\sigma}}$ is 3, and implicit in our notation is the the fact that we evaluate C with $k = 3$.

We use stochastic domination to prove this theorem. Recall that X is said to be *smaller than X' in the usual stochastic order* (denoted by $X \leq_{\text{st}} X'$) if

$$F'(x) \leq F(x) \quad \forall x \in \mathbb{R},$$

where F and F' are the cumulative distribution function of X and X', respectively.

PROOF. (Theorem 1)

Firstly, since

$$\mu_{(k)} + \frac{D(n)}{\sqrt{n}} = \mathbb{E} \max_{i=1,\ldots,k} \bar{X}_i(n) \geq \max_{i=1,\ldots,k} \mathbb{E}\bar{X}_i(n) = \mu_{(k)},$$

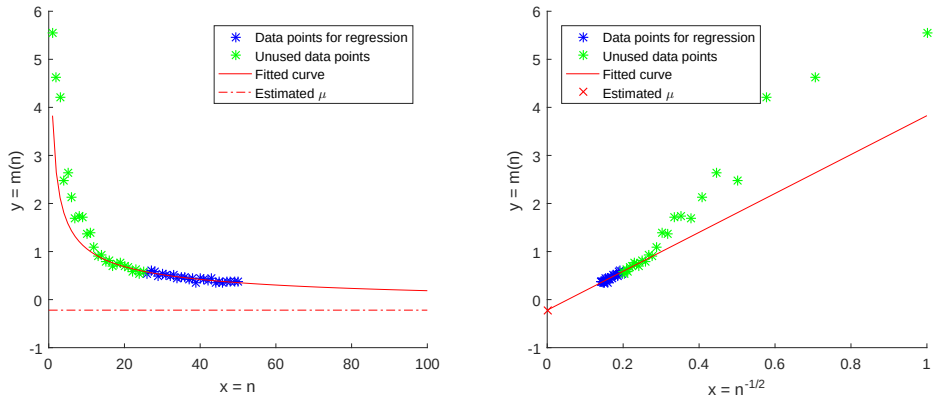

Figure 2.3: Regression method to estimate $\mu_{(k)}$ when all $\sigma_i^2 = 4$ are equal and μ_i are independently and normally distributed with standard deviation equal to 0.2. We subtract $\max_i \mu_i$ from $\mu_i, i = 1, 2, \ldots, k$ to ensure that $\mu_{(k)} = 0$. Here $k = 1000$ and $n_0 = 50$. Only the last 25 data points are used for regression. The estimated $\hat{\mu}_{(k)} = -0.22$.

$D(n) \geq 0$. Secondly,

$$D(n) = \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j \neq i} \Phi\left(\frac{x - \sqrt{n}(\mu_j - \mu_{(k)})}{\sigma_j} \right) \right] \phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right) \frac{1}{\sigma_i} dx$$

$$= \mathbb{E}\tilde{M},$$

where $\tilde{M} = \max_{i=1,\ldots,k} \tilde{X}_i$, $\tilde{X}_i \sim \mathcal{N}(\sqrt{n}(\mu_i - \mu_{(k)}), \sigma_i)$. The distribution function of $\tilde{M}$

$$F_{\tilde{M}(x)} = \prod_{i=1}^{k} \Phi\left(\frac{x - \sqrt{n}(\mu_i - \mu_{(k)})}{\sigma_i} \right)$$

is non-decreasing in n for all x since Φ is an increasing function and $\mu_i - \mu_{(k)} \leq 0$, $\forall i$, so the corresponding random variables are non-increasing in the usual stochastic order. Therefore, $D(n) = \mathbb{E}\tilde{M}$ is non-increasing. See (1.A.7) of [53].

In addition, since

$$\lim_{n\to\infty} \Phi\left(\frac{x-\sqrt{n}(\mu_j-\mu_{(k)})}{\sigma_j}\right) = \begin{cases} \Phi\left(\frac{x}{\sigma_j}\right), & \mu_j = \mu_{(k)} \\ 1, & \mu_j < \mu_{(k)} \end{cases}$$

$$\lim_{n\to\infty} \phi\left(\frac{x-\sqrt{n}(\mu_i-\mu_{(k)})}{\sigma_i}\right) = \begin{cases} \phi\left(\frac{x}{\sigma_i}\right), & \mu_i = \mu_{(k)} \\ 0, & \mu_j < \mu_{(k)} \end{cases}$$

$$\begin{aligned} \lim_{n\to\infty} D(n,\boldsymbol{\mu},\boldsymbol{\sigma}) &= \lim_{n\to\infty} \int_{-\infty}^{\infty} x \sum_{i=1}^{k} \left[\prod_{j\neq i} \Phi\left(\frac{x-\sqrt{n}(\mu_j-\mu_{(k)})}{\sigma_j}\right)\right] \phi\left(\frac{x-\sqrt{n}(\mu_i-\mu_{(k)})}{\sigma_i}\right)\frac{1}{\sigma_i}dx \\ &= \int_{-\infty}^{\infty} x \sum_{i:\mu_i=\mu_{(k)}} \left[\prod_{j\neq i,\mu_j=\mu_{(k)}} \Phi\left(\frac{x}{\sigma_j}\right)\right] \phi\left(\frac{x}{\sigma_i}\right)\frac{1}{\sigma_i}dx \\ &= C(\tilde{\boldsymbol{\sigma}}), \end{aligned}$$

where the components of $\tilde{\boldsymbol{\sigma}}$ are those σ_i's whose corresponding μ_i's are equal to $\mu_{(k)}$. The second equality can be justified by the Monotone Convergence Theorem since $D(n,\boldsymbol{\mu},\boldsymbol{\sigma})$ is monotonically decreasing in n.

□

By Theorem 1, the slope $D(n)$ is non-negative, non-increasing with n and hence non-decreasing with $\frac{1}{n}$. Since our goal is to estimate the intercept of the linear function, in other words, the limit of $\mathbb{E}M(n)$ when $n \to \infty$, if we treat $D(n)$ as a constant, our estimator $\hat{\mu}_{(k)}$ underestimates the true highest expectation $\mu_{(k)}$. The larger n, the closer $D(n)$ to its limit, so using $M(n)$ with larger n in regression leads to lower bias. However, fewer data points brings higher variance, so there is a tradeoff between bias and variance over the number of data

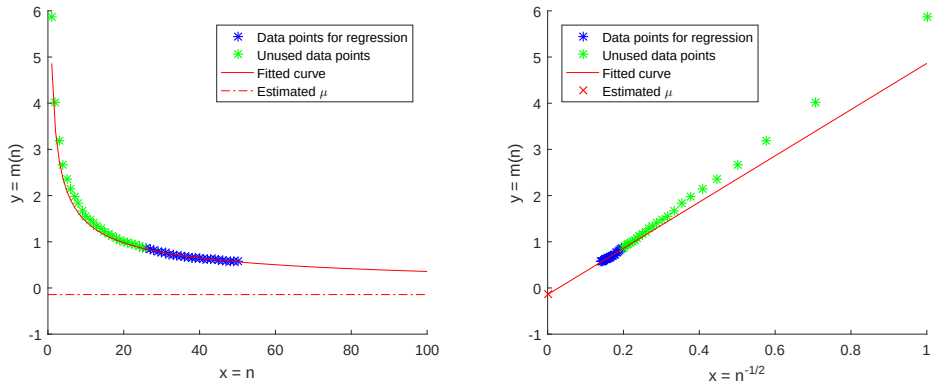

Figure 2.4: Regression method to estimate $\mu_{(k)}$ when all $\sigma_i^2 = 4$ are equal and μ_i are normally distributed with standard deviation equal to 0.2, and $\mu_{(k)} = 0$. Here $k = 1000$ and $n_0 = 50$. Only the last 25 data points are used for regression. The estimated $\hat{\mu}_{(k)} = -0.14$.

points used for regression. A practical rule of thumb that has had reasonable performance in our experiments is to use the second half of the data points, e.g., $(M(n), n = 26, 27, \ldots, 50)$ for $n_0 = 50$. An illustration is given in Figure 2.3, where the estimated $\hat{\mu}_{(k)} = -0.22$.

In our linear regression model, we use $M(n)$ in place of $\mathbb{E}M(n)$. However, using only one observation leads to high variance, especially for the case where n is small. If we have more observations of $M(n)$, we can obtain a better estimate of $\mathbb{E}M(n)$ and hence make our estimation on $\mu_{(k)}$ more reliable. Limited to the sample size of the preliminary stage, we only have n_0 samples for each system. However, noticing that all the samples are generated independently and there is no specified order, we can use bootstrapping, e.g., we can shuffle the samples by randomly reordering them for each system to generate more data points $M(n)$'s

for our regression, to make the estimation more stable. Denoting the number of shuffles by n_{shuffle}, for each $j = 1, 2, \ldots, n_{\text{shuffle}}$, we randomly reorder the samples for each system independently and compute the largest sample mean for the first n samples $M^{(j)}(n)$, $n = 1, 2, \ldots, n_0$. Define $\bar{M}(n) = \frac{1}{n_{\text{shuffle}}} \sum_{j=1}^{n_{\text{shuffle}}} M^{(j)}(n)$. Using $\bar{M}(n)$ instead of $M(n)$ gives a more accurate estimator of $\mathbb{E}M(n)$ and hence a more stable $\hat{\mu}_{(k)}$; see Figure 2.4.

The same idea applies to all of the ordered expectations $\mu_{(i)}$, $i = 1, 2, \ldots, k$. For each i, letting $M_i = M_i(n) = \bar{X}_{(i)}(n)$ be the i-th smallest sample mean, its probability density function is given by

$$\sum_{r \in \mathcal{R}} \left[\prod_{j_1=1}^{i-1} \Phi\left(\frac{x - \mu_{r(j_1)}}{\sigma_{r(j_1)}/\sqrt{n}} \right) \right] \left[\prod_{j_2=i+1}^{k} \left(1 - \Phi\left(\frac{x - \mu_{r(j_2)}}{\sigma_{r(j_2)}/\sqrt{n}} \right) \right) \right] \phi\left(\frac{x - \mu_{r(i)}}{\sigma_{r(i)}/\sqrt{n}} \right) \frac{1}{\sigma_{r(i)}/\sqrt{n}},$$

where $\mathcal{R}$ is the set of permutations of $1, 2, \ldots, k$. Using the similar change of variable $y = \frac{x - \mu_i}{1/\sqrt{n}}$, we get

$$\mathbb{E}M_i(n) = \mu_{(i)} + \frac{D_i(n)}{\sqrt{n}},$$

Hence, we can estimate $\mu_{(i)}$ with linear regression as we did for $M = M_k$, although $D_i(n)$ is not necessarily monotone in n.

Unfortunately, $\hat{\mu}_{(i)}$, the estimate of the ith order statistics, may not be monotonically increasing in i. Therefore, in practice we further process the estimates $\hat{\mu}_{(i)}$ by taking $\hat{\mu}_{(i)}^{\text{M}} = \max_{j=1,\ldots,i} \hat{\mu}_{(j)}$ to enforce monotonicity.

We are estimating the ordered means, so we want the variance of the differences between systems to be small. In practice we recommend using common random numbers (CRN) for the initial stage to enhance the accuracy of the esti-

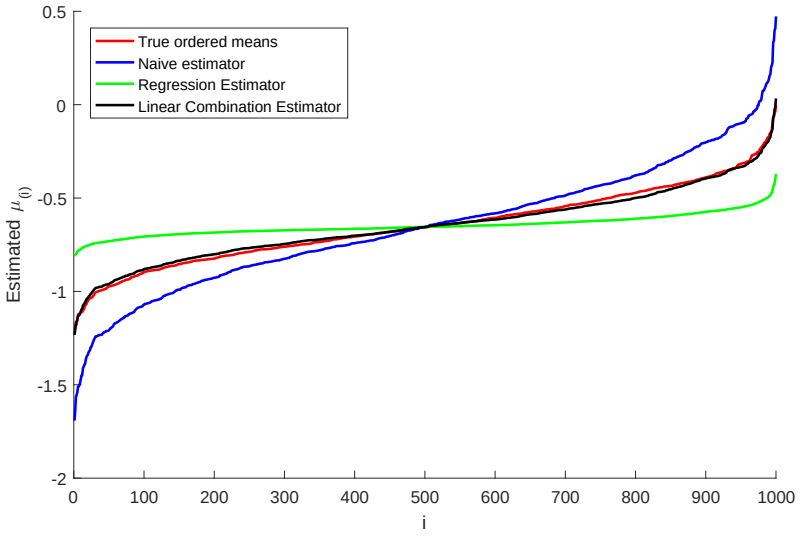

Figure 2.5: Ordered expectations and the values of estimators. The red curve is the true ordered means, the blue curve is the naive estimator m_i, the green curve is the regression estimator $\hat{\mu}_{(i)}$ and the black curve is the linear combination estimator $\tilde{\mu}_{(i)}$. All $\sigma_i^2 = 4$ are equal and μ_i are independent and normally distributed with standard deviation equal to 0.2. Here $k = 10000$ and $n_0 = 50$. The coefficient $\rho = 0.48$.

mation. When CRN is employed, in the shuffling step, instead of shuffling each system's replications independently, we shuffle the replications across all systems together. More specifically, if $\mathbf{X}^j = (X_{ij} : i = 1, 2, \ldots, k)$, we randomly select from $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^{n_0}$.

2.3.3 Convex Combination of Naive and Regression Estimators

As stated previously, $\hat{\mu}_{(k)}$ underestimates the true highest expectation $\mu_{(k)}$. Similarly, we can prove that $\hat{\mu}_{(1)}$ overestimates the true lowest expectation $\mu_{(1)}$. In general, intuitively and empirically, the estimated ordered expectations are condensed relative to the true configuration, which means that the estimated expectations will be closer to each other than the true expectations.

On the other hand, mean-spreading tells us that the ordered sample means are not condensed, but rather, spread out. That inspires us to combine these two consistent estimators to construct a less biased estimator. Here we consider the linear combination (LC) estimator

$$\hat{\boldsymbol{\nu}} = \rho\hat{\boldsymbol{\mu}}^{\mathrm{M}} + (1-\rho)\boldsymbol{M},$$

where $\hat{\boldsymbol{\nu}}$ is our new estimator of the ordered expectations, $\hat{\boldsymbol{\mu}}^{\mathrm{M}} = (\hat{\mu}^{\mathrm{M}}_{(1)}, \ldots, \hat{\mu}^{\mathrm{M}}_{(k)})^T$ is the vector of the monotone regression estimators of the ordered expectations, $\boldsymbol{M} = (M_1(n_0), \ldots, M_k(n_0))^T$ is the vector of ordered sample means of the first n_0 samples, and ρ is a coefficient. In order to find an appropriate value of ρ for given vectors $\hat{\boldsymbol{\mu}}^{\mathrm{M}}$ and $\boldsymbol{M}$, we want to be able to measure how close $\hat{\boldsymbol{\nu}}$ is to the true ordered expectations, but these are unknown. However, we can take advantage of the ordered sample means in the sense that if two expectation vectors are close, then the ordered sample means they generate should also be close. For two ordered sample mean vectors $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$, we define the distance between them to be

the L_2 norm of their difference

$$\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{y}}\| = \left[\sum_{i=1}^{k} (\bar{x}_{(i)} - \bar{y}_{(i)})^2\right]^{1/2}.$$

We enumerate ρ from 0 to 1 in steps of length 0.02 to find the best coefficient. For each ρ, we first compute the corresponding $\tilde{\boldsymbol{\nu}}$, and generate n_{gen} sample means independently based on it, with the corresponding $\hat{\sigma^2}_i/\sqrt{n_0}$ as variance, then compute the average distance from those generated ordered sample means to the true ordered sample means $\boldsymbol{M}$. In our experiments we set $n_{\text{gen}} = 100$. Finally we select the ρ that minimizes this value, and take the corresponding $\tilde{\boldsymbol{\nu}}$ as our final estimator. As illustrated in Figure 2.5, comparing to the true ordered means, the naive estimator is more spread-out and the regression estimator is more condensed, while the linear combination of these two estimators with optimal coefficient $\rho = 0.4$ appears to be very close to the true values.

One might be tempted to simply set $\rho = 0.5$, which may be more efficient. However, the optimal ρ can be close to 0 when the means of the systems are spread-out, or close to 1 when the means of the systems are close to one another. In these cases, setting $\rho = 0.5$ does not yield accurate predictions. In addition, as we will show in numerical experiments, the time spent on predicting runtimes, including the time spent calculating a good choice of ρ, is negligible compared to the total running time of the R&S procedure.

Thus, we have a final estimator that appears to work much better than the original naive estimator. The whole algorithm is summarized below.

Estimation of the ordered expectations

1. Generate n_0 samples $X_{ij}, j = 1, 2, \ldots, n_0$ from each system $i = 1, 2, \ldots, k$. Compute $\bar{X}_i(n_0)$ and $\hat{\sigma^2}_i$ for each i. Let $M_i(n_0) = \bar{X}_{(i)}(n_0)$ and $r(i)$ be the corresponding index, i.e., $M_i(n_0) = \bar{X}_{r(i)}(n_0)$.

2. For $j = 1, \ldots, n_{\text{shuffle}}$,

   (a) For $i = 1, \ldots, k$,

      i. Shuffle $X_{ij}, j = 1, \ldots, n_0$ and get $X_{i\ell}^{(j)}, \ell = 1, \ldots, n_0$, where ℓ is the new index after shuffling.

      ii. Compute $\bar{X}_i^{(j)}(n) = \frac{1}{n} \sum_{\ell=1}^{n} X_{i\ell}^{(j)}$ for $n = 1, \ldots, n_0$.

   (b) Compute the ordered sample means $M_i^{(j)}(n) = \bar{X}_{(i)}^{(j)}(n)$ for $i = 1, \ldots, k$, $n = 1, \ldots, n_0$.

3. Compute $\bar{M}_i(n) = \frac{1}{n_{\text{shuffle}}} \sum_{j=1}^{n_{\text{shuffle}}} M_i^{(j)}(n)$ for $i = 1, \ldots, k$, $n = 1, \ldots, n_0$.

4. Apply linear regression to calculate $\hat{\mu}_{(i)}$ using $\bar{M}_i(n), n = \lceil \frac{n_0}{2} \rceil + 1, \ldots, n_0$, for $i = 1, \ldots, k$.

5. Enforce monotonicity by computing $\hat{\mu}_{(i)}^{\text{M}}$.

6. For $\rho = 0, 0.02, \ldots, 1$,

   (a) Compute $\hat{\boldsymbol{\nu}} = \rho \hat{\boldsymbol{\mu}}^{\text{M}} + (1 - \rho)\boldsymbol{M}$.

   (b) For $j = 1, \ldots, n_{\text{gen}}$,

      i. Generate $\bar{Y}_i \sim \mathcal{N}(\hat{\nu}_i, \hat{\sigma^2}_{r(i)}/\sqrt{n_0})$ for $i = 1, \ldots, k$, independently.

ii. Compute $D_j(\rho)$, the L_2 distance between $(\bar{Y}_{(1)}, \bar{Y}_{(2)}, \ldots \bar{Y}_{(k)})^T$ and $\boldsymbol{M} = (M_1(n_0), M_2(n_0), \ldots, M_k(n_0))^T$.

(c) Compute the average L_2 distance $D(\rho) = \frac{1}{n_{\text{gen}}} \sum_{j=1}^{n_{\text{gen}}} D_j(\rho)$.

7. Find $\rho_{\text{opt}} = \text{argmin}_\rho D(\rho)$ and let $\hat{\boldsymbol{\nu}} = \rho_{\text{opt}} \hat{\boldsymbol{\mu}}^{\text{M}} + (1 - \rho_{\text{opt}}) \boldsymbol{M}$ be our estimate of the ordered true means. The estimate of the corresponding variances is $(\hat{\sigma^2}_{r(1)}, \hat{\sigma^2}_{r(2)}, \ldots, \hat{\sigma^2}_{r(k)})^T$.

2.3.4 Statistical Properties of The Estimators

In this section we give some theoretical support for the estimators. Specifically, we establish consistency and identify the rate of convergence of the naive estimator $M_i(n_0)$, the regression estimator $\hat{\mu}_{(i)}$ and hence the LC estimator $\hat{\nu}_i$. For simplicity, we analyze simpler forms of the estimators that exploit neither the shuffling of samples for generating more data points, nor the enforcing of monotonicity. In proving these results, we relax the normality assumption, since the results hold more generally. To simplify indexing notation, in this section we assume, unbeknownst to our procedure, that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$. We impose different subsets of the following conditions in the results to follow.

A0 Means are unique: $\mu_1 < \mu_2 < \cdots < \mu_k$.

A1 Sequences are iid with finite mean: $\mathbf{X}_i = (X_{ij} : j \geq 1)$ is an i.i.d. sequence with finite mean μ_i, for each $i = 1, \ldots, k$.

A2 Finite second moments: $\mathbb{E}(X_{i1}^2) < \infty$, so that the variance, σ_i^2, is finite, $i = 1, 2, \ldots, k$.

A3 Finite fourth moments: $\mathbb{E}(X_{i1}^4) < \infty$, $i = 1, 2, \ldots, k$.

A4 Mutual independence: $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X}_k$ are independent.

Proposition 1. *Fix $i \in \{1, 2, \ldots, k\}$. If A0, A1 hold, then the naive estimator $M_i(n)$ is a strongly consistent estimator of μ_i, for each $i = 1, 2, \ldots, k$. If A0–A2 hold, then we have the central limit theorem*

$$\sqrt{n}(M_i(n) - \mu_i) \Rightarrow \sigma_i N(0, 1)$$

as $n \to \infty$. If A0–A3 hold, then the mean squared error of $M_i(n)$ is $\mathcal{O}(n^{-1})$.

Notice that we assume neither normality nor mutual independence; the result is more general, and covers these special cases. The assumption of finite fourth moments for the mean-squared error result is potentially stronger than needed, but permits a straightforward proof, and is standard; see, e.g., [20]. Here we assume the uniqueness of means, A0, but we relax that condition in Proposition 2 in the appendix, leading to qualitatively similar results that are more complicated to establish. For example, the limiting distribution in the central limit theorem changes.

PROOF. Let $\bar{X}_i(n) = n^{-1} \sum_{j=1}^{n} X_{ij}$ be the nth sample mean from System i. This does not necessarily equal $M_i(n)$, since $M_i(n)$ is the ith smallest sample mean. These two quantities agree on the event, $B_i(n)$ say, where $\bar{X}_i(n)$ is the ith smallest

of the sample means $\bar{X}_1(n), \ldots, \bar{X}_k(n)$, i.e., it is ranked correctly, and may differ otherwise.

To establish consistency, the strong law of large numbers and the fact that k is finite ensure that $\max_{q=1}^{k} |\bar{X}_q(n) - \mu_q| \to 0$ as $n \to \infty$ a.s. Hence, since the μ_qs are unique, the event $B_i(n)$ occurs eventually a.s., i.e., there exists a (random) N such that for all $n \geq N$, $B_i(n)$ occurs a.s. Moreover, on the event $B_i(n)$, $M_i(n) = \bar{X}_i(n)$.

To establish the central limit theorem, let $\mathbb{1}(B)$ denote the indicator of the event B that equals 1 on the event and 0 otherwise. As noted above, $\mathbb{1}(B_i(n)) = 1$ for sufficiently large n (for $n \geq N$) a.s., so that the indicator of the complement, $\mathbb{1}(B_i^c(n)) = 0$ for $n \geq N$ a.s.. Define $E_i(n) = \sqrt{n}\,(M_i(n) - \mu_i)$. Then

$$E_i(n) = \sqrt{n}(\bar{X}_i(n) - \mu_i)\mathbb{1}(B_i(n)) + E_i(n)\mathbb{1}(B_i^c(n)),$$

and the result now follows from the ordinary central limit theorem and Slutsky's theorem.

To establish the mean-squared error result, we write

$$E_i(n) = \sqrt{n}(\bar{X}_i(n) - \mu_i) + \sqrt{n}(M_i(n) - \bar{X}_i(n))\mathbb{1}(B_i^c(n)).$$

The inequality $(a + b)^2 \leq 2a^2 + 2b^2$ gives

$$\mathbb{E}(E_i(n)^2) \leq 2n\mathbb{E}(\bar{X}_i(n) - \mu_i)^2 + 2n\mathbb{E}\left[(M_i(n) - \bar{X}_i(n))^2\mathbb{1}(B_i^c(n))\right]. \tag{2.1}$$

The first term in (2.1) equals $2\sigma_i^2$. For the second term, the Cauchy-Schwarz inequality, together with the fact that $M_i(n)$ is equal to $X_J(n)$ for some (random) J

gives

$$
\begin{aligned}
&2n\mathbb{E}\left[\left(M_i(n)-\bar{X}_i(n)\right)^2 \mathbb{1}(B_i^c(n))\right] \\
&\leq 2n\mathbb{E}\left[\left(M_i(n)-\bar{X}_i(n)\right)^4\right]^{1/2} \mathbb{P}(B_i^c(n))^{1/2} \\
&\leq 2n\left(\sum_{j\neq i}\mathbb{E}\left[\left(\bar{X}_j(n)-\bar{X}_i(n)\right)^4\right]^{1/2}\right)\left(\sum_{j>i}\mathbb{P}(\bar{X}_j(n)-\bar{X}_i(n)<0)+\sum_{j<i}\mathbb{P}(\bar{X}_j(n)-\bar{X}_i(n)>0)\right)^{1/2}.
\end{aligned}
$$

We used the fact that $B_i^c(n) \subseteq \left\{\bigcup_{j>i}\{\bar{X}_j(n)-\bar{X}_i(n)<0\} \cup \bigcup_{j<i}\{\bar{X}_j(n)-\bar{X}_i(n)>0\}\right\}$. The term $\mathbb{E}\left[\left(\bar{X}_j(n)-\bar{X}_i(n)\right)^4\right]$, by direct calculation, is $\mathcal{O}(1)$. Chebyshev's inequality applied to the remaining terms yields the bound

$$
\begin{aligned}
&2n\mathcal{O}(1)\left(\sum_{j\neq i}\frac{\operatorname{var}\left(\bar{X}_j(n)-\bar{X}_i(n)\right)}{(\mu_j-\mu_i)^2}\right)^{1/2} \\
&= 2n\mathcal{O}(1)\left(\sum_{j\neq i}\frac{\operatorname{var}\left(X_j-X_i\right)}{n(\mu_j-\mu_i)^2}\right)^{1/2} \\
&= \mathcal{O}(1),
\end{aligned}
$$

concluding the proof. □

Turning now to the regression estimator, let $x_j = j^{-1/2}$ and $y_j = M_i(j)$ for $j \geq 1$. Assume that n is even, for notational simplicity. Define

$$\bar{x} = \frac{1}{n/2}\sum_{j=n/2+1}^{n} x_j,$$

so that $\bar{x}$ is a mean of the x_j values. Similarly, define $\bar{y}, \overline{xy}$ and $\overline{x^2}$, so that, for example,

$$\overline{xy} = \frac{1}{n/2}\sum_{j=n/2+1}^{n} x_j y_j.$$

The regression estimator is the y intercept of the estimated regression line, and can thus be written as

$$\hat{\mu}_i(n) = \frac{\overline{x^2}\bar{y} - \bar{x}\overline{xy}}{\overline{x^2} - \bar{x}^2}. \tag{2.2}$$

Theorem 2. *Fix $i \in \{1, 2, \ldots, k\}$. If A1 holds, then the regression estimator $\hat{\mu}_i(n)$ is a strongly consistent estimator of μ_i as $n \to \infty$. If A0-A2 hold, then it satisfies the central limit theorem*

$$\sqrt{n}(\mu_i(n) - \mu_i) \Rightarrow \eta N(0, 1),$$

where the variance constant η^2 is approximately $8.4144\sigma_i^2$. If A1-A3 hold then its mean squared error is $\mathcal{O}(n^{-1})$.

We assume uniqueness of the means to establish the central limit theorem because the argument is involved; we conjecture that a version of the central limit theorem holds without that assumption.

PROOF. To establish consistency, write

$$\hat{\mu}_i(n) = \bar{y} + \frac{\bar{x}^2(\overline{xy}/\bar{x} - \bar{y})}{\overline{x^2} - \bar{x}^2}. \tag{2.3}$$

From Proposition 2 in the appendix, we know that under A1, $M_i(n) \to \mu_i$ as $n \to \infty$ a.s. It immediately follows that $\bar{y} \to \mu_i$ and $\overline{xy}/\bar{x} \to \mu_i$ as $n \to \infty$ a.s. For two real-valued sequences $(a_n : n \geq 1)$ and $(b_n : n \geq 1)$ we write $a_n \sim b_n$ if $a_n/b_n \to 1$ as

$n \to \infty$. Approximating sums by integrals,

$$\begin{aligned}
\bar{x} &= \frac{2}{n} \sum_{j=n/2+1}^{n} j^{-1/2} \sim \frac{2}{n} \int_{n/2}^{n} x^{-1/2}\, dx = \frac{\beta_1}{\sqrt{n}}, \\
\overline{x^2} &= \frac{2}{n} \sum_{j=n/2+1}^{n} j^{-1} \sim \frac{\beta_2}{n}, \\
\overline{x^2} - \bar{x}^2 &\sim \frac{\beta_3}{n}, \text{ and} \\
\frac{\bar{x}^2}{\overline{x^2} - \bar{x}^2} &\sim \beta_1^2/\beta_3,
\end{aligned}$$

where the constants $\beta_1 = 4 - 2\sqrt{2}$, $\beta_2 = 2\ln(2)$, and $\beta_3 = \beta_2 - \beta_1^2$. These observations, together with the representation (2.3), yield consistency.

We prove the mean-squared error result before the central limit theorem. Recall that we defined E_j so that $y_j = \mu_i + \frac{E_j}{\sqrt{j}}$. Using this definition in (2.2),

$$\begin{aligned}
\hat{\mu}_i(n) &= \frac{\overline{x^2}\bar{y} - \bar{x}\overline{xy}}{\overline{x^2} - \bar{x}^2} \\
&= \frac{\overline{x^2}\left[\mu_i + \frac{2}{n}\sum_{j=n/2+1}^{n} j^{-1/2} E_j\right] - \bar{x}\left[\bar{x}\mu_i + \frac{2}{n}\sum_{j=n/2+1}^{n} j^{-1} E_j\right]}{\overline{x^2} - \bar{x}^2} \\
&= \mu_i + \sum_{j=n/2+1}^{n} \frac{2\overline{x^2} j^{-1/2} - 2\bar{x} j^{-1}}{n(\overline{x^2} - \bar{x}^2)} E_j \\
&= \mu_i + \sum_{j=n/2+1}^{n} c_{n,j} E_j, \qquad (2.4)
\end{aligned}$$

where $c_{n,j} = \mathcal{O}(n^{-3/2})$ using the order-of-magnitude results above.

Thus, using the fact that $\mathbb{E}(E_n^2) = \mathcal{O}(1)$ from (A.1) in the proof of Proposition 2

in the appendix, (2.4) yields

$$\begin{aligned}\mathbb{E}[(\hat{\mu}_i(n) - \mu_i)^2] &= \mathbb{E}\left[\sum_{j=n/2+1}^{n} c_{n,j} E_j\right]^2 \\ &= \mathcal{O}(1) \sum_{j=n/2+1}^{n} \sum_{m=n/2+1}^{n} c_{n,j} c_{n,m} \\ &= \mathcal{O}(1)\mathcal{O}(n^2)\mathcal{O}(n^{-3}) \\ &= \mathcal{O}(n^{-1})\end{aligned}$$

giving the mean-squared error result.

Turning to the central limit theorem, by consistency and the fact that the means are assumed unique, there exists (random) N such that for all $j \geq N$, $M_i(j) = \bar{X}_i(j)$. Thus, $\mathbb{1}(n/2 \geq N)$ converges to 1 almost surely, and (2.4) gives

$$\sqrt{n}(\hat{\mu}_i(n) - \mu_i) = \mathbb{1}(n/2 \geq N)\sqrt{n} \sum_{j=n/2+1}^{n} c_{n,j} E_j + \mathbb{1}(n/2 < N)\sqrt{n}(\hat{\mu}_i(n) - \mu_i).$$

The second term in this expression is eventually equal to 0 and the indicator in the first expression is eventually equal to 1. It therefore suffices to focus attention on the summation in the first expression. Moreover, on the event $\{n/2 \geq N\}$, $E_j = \sqrt{j}(\bar{X}_i(j) - \mu_i)$ for $j \geq n/2$, and then (on that event),

$$\begin{aligned}\sqrt{n} \sum_{j=n/2+1}^{n} c_{n,j} E_j &= \sum_{j=n/2+1}^{n} c_{n,j} \sqrt{nj}(\bar{X}_i(j) - \mu_i) \\ &= \sum_{j=n/2+1}^{n} c_{n,j} \sqrt{n/j} \sum_{m=1}^{j} (X_i(m) - \mu_i) \\ &= \sum_{m=1}^{n} \left(\sum_{j=\max(n/2+1,m)}^{n} c_{n,j} \sqrt{n/j} \right) (X_i(m) - \mu_i) \\ &= \sum_{m=1}^{n} a_{n,m}(X_i(m) - \mu_i)\end{aligned}$$

for appropriately defined constants $a_{n,m}$, where we changed the order of summation in the second-to-last step. We can now apply the Lindeberg-Feller Central Limit Theorem; see, e.g., Theorem 27.2 of [5]. Verifying the Lindeberg condition is straightforward; it follows exactly the same lines as the discussion of (27.9) in that text. The central limit theorem we desire therefore follows using the magnitude of the constants $a_{n,m}$ that we determine next, and has variance given by

$$\sigma_i^2 \lim_{n\to\infty} \sum_{m=1}^{n} a_{n,m}^2. \tag{2.5}$$

It remains to obtain the expression for the variance. To that end,

$$\begin{aligned} a_{n,m} &= \sum_{j=\max(n/2+1,m)}^{n} c_{n,j}\sqrt{n/j} \\ &= 2\sqrt{n} \sum_{j=\max(n/2+1,m)}^{n} \frac{\overline{x^2}j^{-1} - \bar{x}j^{-3/2}}{n(\overline{x^2} - \bar{x}^2)} \\ &\sim \frac{2\sqrt{n}}{\beta_3} \sum_{j=\max(n/2+1,m)}^{n} (\beta_2 n^{-1} j^{-1} - \beta_1 n^{-1/2} j^{-3/2}). \end{aligned}$$

Using an integral to asymptotically approximate the sum shows that for $m \le n/2$,

$$a_{n,m} \sim \frac{2\beta_2 \ln(2) - 4\beta_1(\sqrt{2}-1)}{\beta_3\sqrt{n}} = \frac{\beta_4}{\sqrt{n}},$$

and for $m > n/2$,

$$a_{n,m} \sim \frac{2}{\beta_3\sqrt{n}} \left(\beta_2 \ln(n/m) - 2\beta_1\left(\sqrt{n/m} - 1\right)\right).$$

We now use an integral once more to obtain the asymptotic form of (2.5). This step involves considerable but tedious algebra. A symbolic manipulator yields the expression

$$\begin{aligned} \sigma_i^2[&-512\sqrt{2} + 2208\sqrt{2}\ln^2(2) - 904\sqrt{2}\ln^3(2) + 80\sqrt{2}\ln^4(2) + 2688\ln^2(2) - 1392\ln^3(2) \\ &+ 84\ln^4(2) - 4\ln^5(2) - \ln(2)(384\sqrt{2} - 256) - 1024]/(\ln^2(2) - 24\ln(2) + 16)^2, \end{aligned}$$

which evaluates to $8.4144\sigma_i^2$. □

The observation in Theorem 2 that the asymptotic variance of the regression estimator (that appears in the central limit theorem) is larger than that of the naive estimator should not be too surprising. Indeed, the regression estimator is designed to combat the bias in the naive estimator, and bias-reduced estimators usually entail increased variance. Moreover, the regression estimator is essentially a weighted average of the samples, and with non-uniform weights one would expect an increase in variance. The fact that the asymptotic variance only involves σ_i^2 and not the variance of the other systems is natural; the asymptotic analysis depends on the fact that asymptotically the systems are correctly ordered by their sample means, and on that event the regression estimator is based only on samples from the ith system.

The LC estimator is a convex combination of the naive and linear regression estimators, where the weight ρ is random and is obtained from the same data used to construct the estimators. Accordingly, the results for consistency and mean-squared error are easily established. If the estimate of ρ converges a.s. as the sample size goes to infinity, as would be the case if it is fixed at the outset for example, then a central limit theorem for the LC estimator can also be established [51], but we don't provide that result since it is a straightforward extension of the results we have established.

Corollary 1. *Under A1, the LC estimator $\hat{\nu}_i$ is a strongly consistent estimator of μ_i. Under A1-A3, its mean-squared error is $\mathcal{O}(n^{-1})$.*

Table 2.1: Problem configurations used in numerical experiments.

Name	Expectation	Variances
Slippage Configuration (SC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = \sigma^2 = 4$
SC, increasing variances (SC-INC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = (1 + \frac{2i}{k})^2$
SC, decreasing variances (SC-DEC)	$\mu = [\delta, 0, \ldots, 0]$	$\sigma_i^2 = (3 - \frac{2i-2}{k})^2$
Monotone decreasing means (MDM)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = \sigma^2 = 4$
MDM, increasing variances (MDM-INC)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = (1 + \frac{2i}{k})^2$
MDM, decreasing variances (MDM-DEC)	$\mu_i = -\delta(i-1)$	$\sigma_i^2 = (3 - \frac{2i-2}{k})^2$
Random problem instances 1 (RPI1),	$\mu_i \sim \mathcal{N}(0, 4\delta^2)$, i.i.d.	$\sigma_i^2 = \sigma^2 = 4$
Random problem instances 2 (RPI2),	$\mu_i \sim \mathcal{N}(0, 25\delta^2)$, i.i.d.	$\sigma_i^2 = \sigma^2 = 4$
RPI, heterogeneous variances (RPI-HET),	$\mu_i \sim \mathcal{N}(0, 25\delta^2)$, i.i.d.	$\sigma_i^2 = (1 + \frac{2i}{k})^2$

2.4 Numerical Experiments

In this section we demonstrate the performance of our estimates of the simulation budget as measured through the total number of samples. We use three efficient R&S procedures: the KN procedure [36] , the Bayes-inspired indifference zone (BIZ) procedure [18] and the Good Selection Procedure (GSP) [47]. The KN procedure is highly influential and might be considered the state-of-the-art for indifference zone R&S [6]. It compares all pairs of systems and eliminates inferior ones when certain random walks leave a triangular continuation region, until one

system remains. The BIZ procedure has a tight worst-case preference-zone PCS bound for the known-variance case, and is empirically more efficient than KN on a variety of problems. The GSP is designed for a parallel computing environment, and it provides a PAC guarantee. Since we are only using the samples taken in the preliminary stage for simulation budget prediction, those samples are reused as initial samples in the R&S procedures. All the source code for our experiments is available in the open-source repository [42].

2.4.1 Comparison of Naive Estimation and Linear Combination Estimation

We demonstrate the difference in performance of the naive estimator and LC estimator on 9 standard test configurations: SC, SC-INC, SC-DEC, MDM, MDM-INC, MDM-DEC, RPI1, RPI2, RPI1-HET, with three problem sizes: $k = 100, 500, 2000$. These test configurations (and their acronyms) are described in Table 2.1. For all configurations and problem sizes, the indifference zone constant $\delta = 0.1$ and the confidence level $1-\alpha = 0.95$. We compare and illustrate the accuracy of the estimators by the ratios of the estimator and the true value of the total number of samples N. The results do not show significant difference between the three procedures, BIZ, KN and GSP, so we only visualize the results for BIZ through histograms of the ratios under different configurations with different k, as shown in Figure 2.6, 2.7 and 2.8. The results in the histograms come from the *simulation of the simulation*

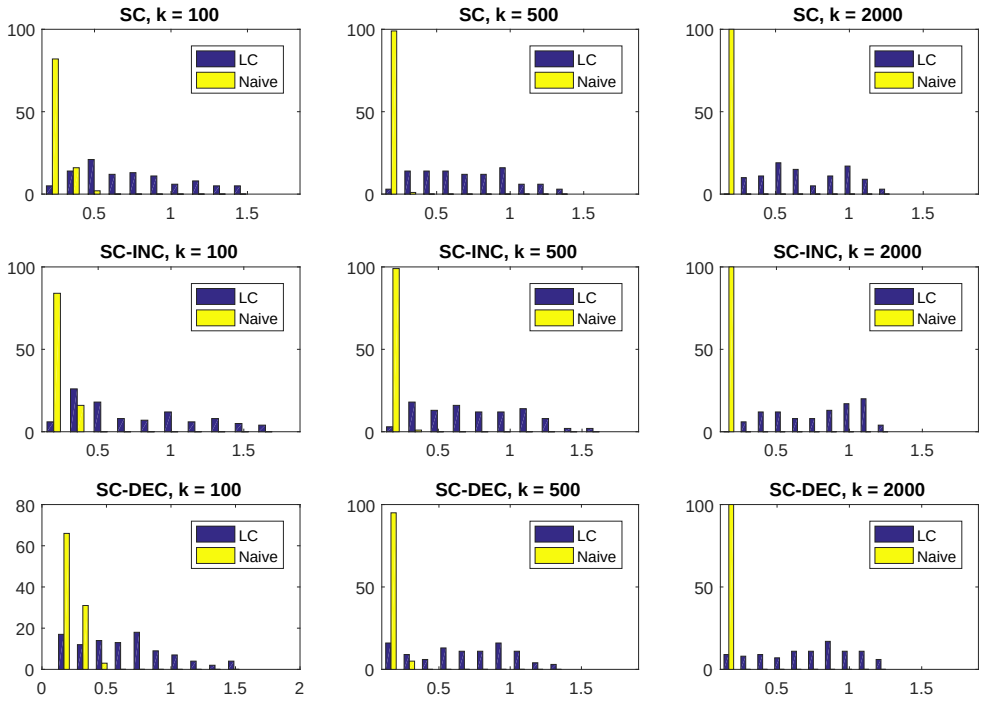

Figure 2.6: Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: SC (row 1); SC-INC (row 2) and SC-DEC (row 3) with k = 100 (column 1); 500 (column 2) and 2000 (column 3).

process approach. The average ratio for all procedures and both *first-order estimate* and *simulation of the simulation process* approaches are given in Tables **??** and **??** in the appendix.

For the slippage configurations in Figure 2.6, the LC estimator performs consistently much better than the naive estimator. We see that we usually under-

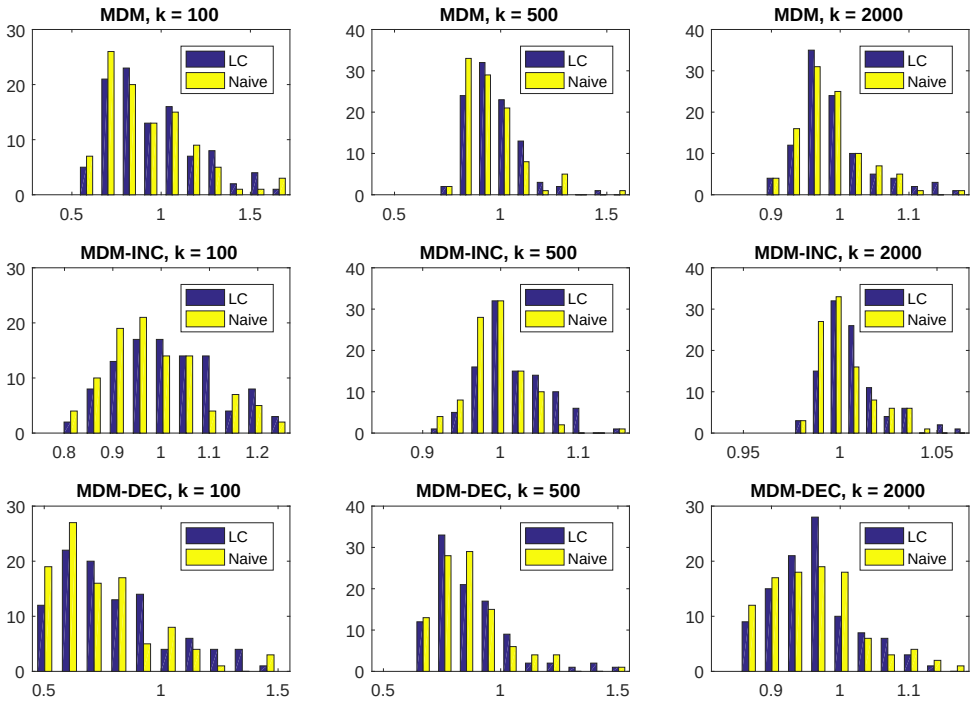

Figure 2.7: Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: MDM (row 1); MDM-INC (row 2) and MDM-DEC (row 3) with $k = 100$ (column 1); 500 (column 2) and 2000 (column 3).

estimate the number of samples, perhaps because the slippage configuration is difficult to recover. However, there is still a huge difference between the performances of the naive estimator and linear combination estimator. The LC estimator can achieve around 70% accuracy (the ratio of the estimator to the true value), and so we expect that it can get the correct order of magnitude, which is enough for determining the number of cores or how much computing resource is needed in

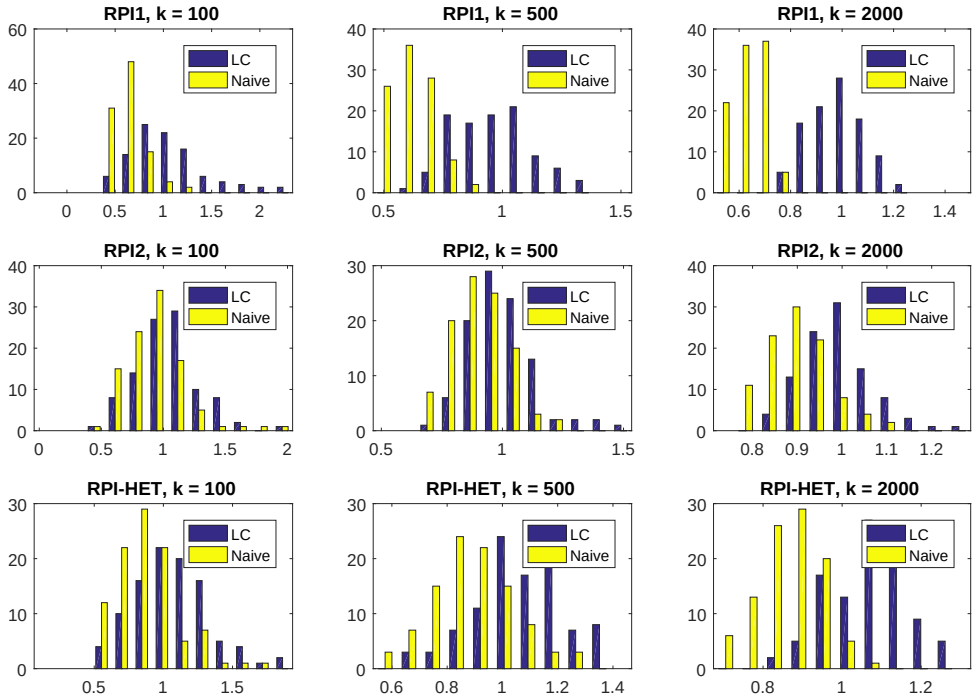

Figure 2.8: Histogram of ratios between estimation and true value of N. Test on BIZ under the following configurations: RPI1 (row 1); RPI2 (row 2) and RPI-HET (row 3) with k = 100 (column 1); 500 (column 2) and 2000 (column 3).

practice. In comparison, the value of the naive estimator is usually around 10% to 20% of the true value, and that can result in an incorrect order of magnitude.

For monotone decreasing means configurations in Figure 2.7, both estimators perform well. The reason is that the means are already very clearly separated, so the "mean-spreading" phenomenon is not severe in this case. In practice, we ex-

pect that both estimators should have good performance for configurations with spread-out means.

For the random problem instances in Figure 2.8, the LC estimator has a good performance as well while the naive estimator sometimes diverges from the truth. In those random instances when there are many good systems not too far away from the best, the LC estimator is still able to recover the original configuration, while the naive estimator might be confused. The ratios between the LC estimator and the true values are usually between 0.8 and 1.2, especially for large k. In comparison, the naive estimator tends to underestimate.

In practice, we rarely see a real-world problem that is as difficult as the slippage configuration. Usually the means are more spread out than the slippage configuration but more condensed than the MDM configuration. In general, the LC estimator should be able to accurately reflect the true configuration without being too "optimistic" or "pessimistic" about the running time.

The LC estimator performs better when k is large for condensed configurations (SC family and RPI1), since with more information provided by more systems, it is able to more accurately estimate the configuration. This is desirable since we mainly focus on large-scale problems.

As shown in Table **??**, the estimations obtained with the first-order approach are similar to those obtained with the simulation approach, and also have good performance.

2.4.2 An Illustrative Problem

In this section we test our estimation of total number of samples on the "throughput-maximization" problem taken from SimOpt.org [23]. In this problem, we want to solve

$$\max_{x} \mathbb{E}[g(x;\xi)]$$

$$\text{s.t. } r_1 + r_2 + r_3 = R$$

$$b_1 + b_2 = B$$

$$x = (r_1, r_2, r_3, r_4, r_5) \in \{1, 2, \ldots\}^5$$

where the function $g(x;\xi)$ represents the random throughput of a three-station flow line with finite buffer storage in front of Stations 2 and 3, denoted by b_2 and b_3 respectively, and an infinite number of jobs in front of Station 1. The processing times of each job at stations 1, 2, and 3 are independently exponentially distributed with service rates r_1, r_2 and r_3, respectively. The overall objective is to maximize expected steady-state throughput by finding an optimal (integer-valued) allocation of buffer and service rates. The warm-up period consists of 2000 released jobs starting from an empty system and the average throughput is then computed based on the time required for the following 10 jobs. For further details of this problem, see [23].

For our test, we set $R = B = 20$, so that $k = 3249$, and we set $\alpha = 0.05$ and $\delta = 0.1$. It takes 0.008 seconds to get one simulation sample on average, and there is no significant difference in simulation time between different systems. We test

Table 2.2: Means and standard deviations of the ratios of estimation to the true value of the sample sizes for the illustrative problem

n_0	KN		BIZ		GSP	
	mean	std	mean	std	mean	std
10	0.78	0.14	1.41	0.21	0.79	0.18
20	0.82	0.11	1.49	0.17	0.63	0.17
50	0.81	0.08	1.28	0.11	0.67	0.17

Table 2.3: Means and standard deviations of the true sample sizes for the illustrative problem ($\times 10^5$, 2 significant figures)

n_0	KN		BIZ		GSP	
	mean	std	mean	std	mean	std
10	21	1.5	1.9	0.15	180	21
20	13	1.4	2.1	0.10	68	7.0
50	12	0.7	2.9	0.15	41	5.1

our estimation for all three procedures: KN, BIZ and GSP with simulation-of-the-simulation-process approaches. To find out the impact of the initial sample size n_0 on the accuracy of the prediction and the computational effort of the prediction, we tried n_0 = $10, 20$ and 50 for all cases.

The ratios of estimates and true values are given in Table 2.2. All of the esti-

Table 2.4: Average total running time for the R&S procedure and for the prediction of running time for the illustrative problem (seconds, 2 significant figures)

n_0	KN		BIZ		GSP	
	R&S	predicting	R&S	predicting	R&S	predicting
10	1000	7.3	140	1.4	7800	1.7
20	630	9.1	140	1.7	2800	2.0
50	610	12	200	2.1	1700	2.4

mates are reasonably accurate, suggesting that the estimation procedure is practical. Further, for different values of n_0, the results show that there is no significant change in the accuracy of the prediction for this problem as n_0 varies. Table 2.3 gives statistics on the total sample size required by each of the procedures as n_0 varies. As n_0 increases, the total sample size increases only for BIZ, and not by a huge amount, though certainly by a nontrivial amount. For the other procedures, the sample size decreases, due to a more confident prediction of the system variances that is exploited in the procedures. (BIZ uses a heuristic to handle estimated variances.) This suggests that the recommendation for R&S algorithms of using n_0 on the order of 10 or 20 is only a rule of thumb, and other choices may be better. Taking n_0 larger is useful, though not essential, for our regression-based estimation procedure. The choice of $n_0 = 50$ seems reasonable in general.

The computational time spent running the R&S procedure and in predicting

the necessary simulation budget is given in Table 2.4. For all procedures, the time spent on simulations dominates, with the predictions requiring at most a few percent of the time spent running simulations. This model is very simple and getting one replication takes only milliseconds. For more complicated models the ratio between the length of the time for making the prediction and for running the R&S procedure would be even smaller. We conclude that our predictions introduce negligible overhead.

We conclude this section by providing a sense of how the predictions can be used to select the number of cores in a parallel implementation. In doing so we use the mean sample size information in Table 2.3 as if they were the predictions of the simulation budget, and the running-time information in Table 2.4 as if it gave the predicted running time obtained by scaling the predicted simulation budgets by average computational time obtained in the prediction stage. Of course, this means we are using actual running time values rather than the predictions in our explanation, which we do simply because it means we need not provide additional information on the predictions; the decision process is the same. The decision process is simply to choose the number of cores, c say, so that the total computation time divided by c is on the order of the desired wall-clock computation time. An additional check is that the total number of simulation replications needed is large enough relative to c that one would expect a well-implemented R&S procedure to be able to efficiently distribute the computation across the cores. If the number of replications per core is too small, e.g., less than a few hundred, then we should be cautious of the heuristic of dividing total computation by available

cores, which assumes perfect utilization of cores.

The running-times in Table 2.4 for $n_0 = 50$ (our proposed choice of n_0) suggest that both KN and GSP will be time-consuming, requiring on the order of 600 seconds and 1700 seconds of computation respectively. If we want the R&S procedure to conclude within, say, one or two minutes, then we would require on the order of 10 cores and 30 cores for the two procedures, respectively. The number of needed replications (from Table 2.3 is so large that one expects a well-implemented R&S procedure to be able to efficiently use the cores, so we expect that the procedures would be very likely to finish within two minutes. Similarly, if one wanted to finish within a few seconds, then we'd instead need on the order of 600 cores and 2000 cores for the two procedures, respectively. KN requires on the order of 1.2 million replications overall, again suggesting that the recommended number of cores (600) could be efficiently used. This would not be the case if we wanted to finish in a small fraction of a second, however, since then the number of replications per core gets so low that one expects synchronization issues and the like to play a role.

Of course, the running time for generating a single replication of this simple example is very small, so the order of the predictions above are modest. For more complex simulation models where the predictions of total computation time are on the order of days (on a single core), the prediction tools outlined in this chapter become more important and relevant. This is especially the case in some cloud-computing settings where the setup time at the start of a computation can be on

the order of minutes. (Our simple heuristic for estimating wall-clock running times ignores setup). In such settings, for wall-clock computation targets that are on the order of days or more, our procedure is highly relevant and worthwhile.

2.5 Conclusions

In this chapter, we explored the problem of predicting the simulation budget of R&S procedures, which is important when dealing with large-scale problems. We presented two approaches for estimating the total number of samples needed for a R&S procedure to terminate, both of which rely on the estimation of the problem configuration, which is not trivial. We develop a linear combination estimator that exhibits excellent performance in a realistic setting, and reasonable performance even in the slippage configuration. Experimental results for both synthetic test problems and a realistic problem together suggest that our approach is effective and sufficient for application.

CHAPTER 3

A SEQUENTIAL SELECTION PROCEDURE DELIVERING A PROBABLY-APPROXIMATELY-CORRECT SELECTION USING CONFIDENCE BANDS

An important class of R&S procedures, indifference-zone (IZ) procedures, provide a certain statistical guarantee on the quality of the selected system. Frequentist IZ procedures originated with [2] and are well studied; see, e.g., [49, 52, 36, 24, 18]. They guarantee to select the unique best system with at least a prescribed probability of correct selection (PCS), assuming that the difference between the best and all others is sufficiently large. To be more precise, let μ_i denote the true performance (usually an expectation) of the ith system. For notational simplicity, suppose that, unknown to the R&S procedure, the systems are indexed so that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$, so that System k is the best. A R&S procedure provides a PCS guarantee at level α if

$$\mathbb{P}(I^* = k) \geq 1 - \alpha, \quad \text{if } \mu_k - \mu_{k-1} > \delta,$$

where I^* is the (random) index of the selected system, the parameter δ is called the indifference-zone parameter, and $1 - \alpha$ is the confidence level. It is natural to require $1 - \alpha > 1/k$, since otherwise the procedure is no better than random guessing. Therefore throughout the chapter we assume that $\alpha < 1 - 1/k$, i.e., α is bounded away from 1 for fixed k. The IZ guarantee only holds when the difference between the best and second-best systems is greater than δ; nothing is guaranteed otherwise. A stronger form of guarantee that holds for any configuration of means, and

that also implies a PCS guarantee when $\mu_k - \mu_{k-1} > \delta$, is probably approximately correct (PAC) selection, which is also referred to as a probability of good selection (PGS) guarantee in, e.g., [46, 47]. It guarantees, with high probability, to select a system whose performance is not too far away from that of the best system, i.e., that

$$\mathbb{P}(\mu_{I^*} > \mu_k - \delta) \geq 1 - \alpha,$$

irrespective of the gap between the best and the other systems. R&S procedures that provide PAC guarantees are far less prevalent than those providing PCS guarantees, perhaps due to the difficulty in establishing such guarantees; see [15] for a survey.

In this chapter we present the Envelope Procedure (EP), a fully sequential procedure providing a PAC guarantee. The EP repeatedly takes samples from systems based on an adaptive sampling rule until a stopping criterion is met. In contrast to elimination-based sequential procedures, e.g., [36, 24, 18], all systems remain in contention until the stopping rule is met, although systems that do not appear to be competitive will be sampled infrequently. The stopping rule of EP is the main factor in ensuring the PAC guarantee, and it affords considerable flexibility in designing the sampling rule. Thus, the EP is really a family of procedures, where the stopping rule is universal but the sampling rule can vary from one procedure to another.

The PAC guarantee of EP is based on the construction of confidence bands. The idea of using confidence bands in R&S procedures is not new. For example,

this idea is also adopted in [39, 40]. A clear difference is that both of those papers consider pairwise differences between systems, while the EP considers events for each single system one at a time, and the methods for constructing confidence bands here are different. To calculate the parameter that determines the width of the bands, and hence the confidence on the quality of the final selection, essentially we need to evaluate the exit probability of a Brownian Motion from certain continuation regions. Computing such probabilities for general regions is mathematically intractable, so we give both bounds and Monte Carlo schemes for our particular choice of continuation region.

In R&S literature, computational efficiency is usually only demonstrated through simulation experiments. We go further by providing an analysis of the sample complexity of the EP. The analysis gives a high-probability upper bound on the total sample size, which helps us better understand the procedure and the underlying difficulty of problem instances. Intuitively, it is hard to distinguish a system from the best one if the difference between their means is small. However, when the difference is smaller than δ there is no need to precisely capture the difference because of our focus on PAC selection. In addition, the problem is difficult if the levels of the noise are high, i.e., the variances are large. Given this intuition, we define a measure of the complexity of a given problem by

$$H_\delta = \sum_{i=1}^{k} \left(\frac{\sigma_i}{\max(\Delta_i, \delta)} \right)^2,$$

where $\Delta_i = \mu_k - \mu_i$, and σ_i^2 is the variance of an observation from the ith system. We provide an explicit relationship between the sample complexity of the EP (with a

specific choice of sampling rule) and H_δ.

To complement this complexity result of the EP, we provide a nearly matching lower bound on the sample complexity that any PAC procedure must satisfy. These upper and lower bounds allow us to conclude that the EP is optimal (from a complexity standpoint that ignores multiplicative constants) up to a logarithmic factor. Related results exist in both the R&S and MAB settings; our lower bound extends a bound in [30] that applies in the IZ setting for a very restricted class of selection procedures. Our result is inspired by a related result in [44] that provides a lower bound for Bernoulli MAB models. [35] gives a general lower bound for the case $\delta = 0$.

The sampling and stopping rules of the EP require negligible computational effort relative to that expended on obtaining simulated samples. This is not always the case for R&S procedures. For example, the screening time of some fully sequential R&S procedures such as [37] and [24] is $\mathcal{O}(k^2)$, where k is the number of systems, since these procedures require pairwise comparisons between surviving systems. (Some recent work [25] reduces the work in pairwise comparisons to an $\mathcal{O}(k)$ operation for the KN procedure, but the ideas do not extend to all R&S procedures.) As stated in [47], this screening time can become prohibitive when k is large, especially when R&S procedures are implemented in a parallel computing environment where large k is the norm.

In addition to a complexity analysis, the EP also has strong performance in computational experiments. Our experiments demonstrate that, across various

problem configurations, EP is more efficient than KN [36], broadly considered a state-of-the-art IZ R&S procedure. Comparing to BIZ [18], an IZ procedure that has a tight lower-bound on PCS for the so-called slippage configuration, EP shows considerable reductions in sample size for configurations with spread-out expectations, especially for larger-scale problems. If we compare EP with KN and BIZ in terms of empirical, i.e. realized, PAC, instead of the lower bound on PAC used to select procedure parameters, EP is even more impressive. Inspired by the gap between the empirical PAC and its lower bound, we develop a heuristic version of the EP that is more efficient than the original EP and the competitor procedures we have tried.

As stated above, the EP is a R&S algorithm and is perhaps best viewed as a contribution to that line of research. Much has been done in this area, dating back to the seminal works of [2] and [22]. See [26, 9] for context, [38] for an introduction to the principles underlying many R&S procedures and, for book-level treatments, see [4, 21]. We also view the EP as a contribution to the best-arm identification problem in the exploratory multi-armed bandit (MAB) setting. That work traces its origins to [3] and [49]. Though the algorithms designed for the pure exploration MAB problems usually have different assumptions on the simulation outputs, e.g., bounded outputs [16] or sub-Gaussian (with a known bound on the scale parameter) outputs [28], the essence of the problems are quite similar. Both R&S problems and pure exploration MAB problems focus on finding the best system (arm) at the end of the procedure, rather than tracking the performance of the arms explored during the exploratory phase as is done in so-called (online)

regret settings. In [29], best-arm algorithms are divided into three classes: action elimination (AE), UCB and LUCB. Though the EP does not strictly belong to any of these classes, its underlying design is inspired by a mixture of them. Thus, we regard this work as a bridge connecting the R&S literature and the best-arm literature.

The rest of the chapter is organized as follows. In §3.1 we formally introduce the formulation and the assumptions of the problem. The EP for the case of known variances is presented in §3.2, in which we describe the procedure, introduce two sampling rules, analyze the statistical validity and sample complexity of the procedure theoretically, and discuss the computation of the parameter η. In §3.3 we discuss the EP for the case of unknown variances in a similar fashion. In §3.4 we present our lower bound on the complexity of any PAC procedure. Then we summarize results from simulation experiments in §3.5, followed by conclusions in §3.6. Two appendices contain an alternative sampling strategy and several supporting results.

3.1 Problem Formulation

Let $\mathcal{S} = \{1, 2, \ldots, k\}$ be a set of k systems. Each system $i \in \mathcal{S}$ is associated with its own simulation model, from which we can generate a sequence of iid real-valued replications $X_{i1}, X_{i2}, \ldots$ For each $i \in \mathcal{S}$, let $\mu_i = EX_{i1}$ be the associated mean, which we assume to be finite. For simplicity of notation, and without loss of generality,

we assume that the systems are indexed so that $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k$. Naturally the decision maker should not be aware of, nor exploit, this indexing.

Throughout this chapter we adopt Assumptions A0 and A1 below. This rules out the use of common random numbers, for example. We selectively adopt Assumption A2 (commonly assumed in the best-arm literature), or A3 (commonly assumed in the R&S literature).

A0 Sequences of i.i.d. samples: $\mathbf{X}_i = (X_{ij} : j \geq 1)$ is an i.i.d. sequence with finite mean μ_i, $\forall i \in \mathcal{S}$.

A1 Independent systems: The sequences $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ are independent.

A2 Sub-Gaussian: $\forall t \in \mathbb{R}$, $\mathbb{E}[\exp(t(X_{ij} - \mu_i))] \leq \exp(\sigma_i^2 t^2/2)$, $\forall_1 \in \mathcal{S}$.

A3 Gaussian: X_{ij} is normally distributed with mean μ_i and variance σ_i^2, i.e., $X_{ij} \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $\forall i \in \mathcal{S}$.

For any $\delta > 0$, System i is said to be δ-optimal if its mean is within δ of the best, i.e., $\mu_i > \mu_k - \delta$, where System k is a best system by our notational convention. A R&S algorithm draws samples from the systems, potentially adaptively based on the samples obtained so far, and terminates and returns a selected system $I^* \in \mathcal{S}$. A R&S algorithm is defined to be a Probably Approximately Correct (PAC) algorithm for given parameters $\delta > 0$, $\alpha \in (0, 1)$ if it 1) always terminates in finite time, and 2) returns a δ-optimal system with probability at least $1 - \alpha$, i.e.,

$$\mathbb{P}(\mu_{I^*} > \mu_k - \delta) \geq 1 - \alpha.$$

Our goal is to devise, for any given $\delta > 0, \alpha \in (0,1)$, a PAC algorithm, under the assumptions A0, A1 and A2.

3.2 Known σ^2 Envelope Procedure

The Envelope Procedure (EP) is a sequential procedure, which means that it takes actions in rounds. In each round t, it either takes samples from a subset of the systems based on the observed samples up to that point, or terminates and returns the selected system. We denote the total number of samples of System i prior to round t as $n_i(t)$, and the number of samples to take from System i in round t as $m_i(t)$, so $n_i(t+1) = n_i(t) + m_i(t)$, where $n_i(0) = n_0, i = 1, 2, \ldots, k$ is the number of samples to take from each system in the initial stage. We let $\bar{X}_i(n) = \frac{1}{n}\sum_{j=1}^{n} X_{ij}$ denote the sample mean of the first n samples of System i.

The key elements of the algorithm are the stopping rule and the sampling rule. The former is a mapping from the observed samples to the decision between continuation and termination, and the latter is a mapping from the observed samples to the allocation of new samples to draw from the systems. The EP accommodates a variety of sampling rules, since the validity (the PAC guarantee) does not rely on the specific choice of it. The sampling rule determines the efficiency of the algorithm. We will also introduce two specific sampling rules in this section. We call our algorithm the Envelope Procedure because for each system we consider a random walk inside an *Envelope*: between the upper and lower confidence bound,

up to a determined time point.

In many scenarios, we can assume that the σ's have known bounds, e.g., the samples are Bernoulli distributed, the support of the distribution is bounded, or there is some known upper bound of the variances. (The scale parameter σ is an upper bound of the standard deviation of the distribution, which is proved in [8].) Here we introduce KEP, the EP with known σ, and later on we will extend the algorithm to the unknown σ version.

3.2.1 The Procedure

KEP

1. Setup. Let $a = 1 - (1-\alpha)^{1/k}$. Compute η, such that

$$\mathbb{P}\left(W_n \leq \eta\sqrt{n},\ \forall n = 1, 2, \ldots, N\right) \geq 1 - a, \tag{3.1}$$

   where

$$N = \left\lceil \left(\frac{2\eta \max_i \sigma_i}{\delta} \right)^2 \right\rceil, \tag{3.2}$$

   and $(W_n, n = 1, 2, \ldots)$ is any random walk with i.i.d. standardized (mean 0, scale factor 1) sub-Gaussian increments. Calculate $N_i = (2\eta\sigma_i/\delta)^2$ for each system i.

2. Initialization. Initialize $t = 0$. Obtain n_0 samples X_{ij}, $j = 1, 2, \ldots, n_0$ from each

system $i = 1, 2, \ldots, k$. Compute sample means $\bar{X}_i(n_0)$ and define

$$i^*(t) = \underset{i=1,2,\ldots,k}{\operatorname{argmax}} \bar{X}_i(n_i(t)), \qquad j^*(t) = \underset{j \neq i^*}{\operatorname{argmax}} \bar{X}_j(n_j(t)) + \frac{\eta\sigma_j}{\sqrt{n_j(t)}}$$

breaking ties arbitrarily. We sometimes omit the index t in i^* and j^* when it does not cause confusion.

3. Stopping Rule. If

$$\bar{X}_{i^*}(n_{i^*}(t)) - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}(t)}} \geq \bar{X}_{j^*}(n_{j^*}(t)) + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*}(t)}} - \delta, \tag{3.3}$$

go to Step 5. Otherwise go to Step 4.

4. Sampling Rule. Compute $m_i(t)$ for $i = 1, 2, \ldots, k$ according to the sampling rule, and take $\min(N_i - n_i(t), m_i(t))$ samples from each system i. Let $t = t + 1$ and update $n_i(t)$, $\bar{X}_i(n_i(t))$, i^* and j^*. Go to Step 3.

5. Termination. Stop and select system $I^* = i^*$ as the best.

We motivate the KEP as follows. For a given constant $\eta > 0$, denote by

$$L_i(n_i) = \bar{X}_i - \frac{\eta\sigma_i}{\sqrt{n_i}}$$
$$U_i(n_i) = \bar{X}_i + \frac{\eta\sigma_i}{\sqrt{n_i}}$$

the lower and upper confidence bound, respectively, which are high probability bounds on the deviation of the empirical mean of System i from its expectation in the specific direction. The width of the confidence bound monotonically decreases

as more samples are drawn. We carefully choose parameters η so that with high probability, the following inequalities

$$L_i(n) \leq \mu_i \text{ for } i = 1, 2, \ldots, k-1, \text{ and}$$
$$U_k(n) \geq \mu_k,$$

hold throughout the whole procedure. The parameter η determines the width of the confidence bounds and hence the statistical validity and sample complexity of the procedure. We will discuss more about the choice of η and its properties in later sections.

In each iteration, we allocate samples to systems according to the sampling rule, take samples and update the upper and lower confidence limits. Once we see that the lower confidence limit $L_{i^*}(n_{i^*}(t))$ of System i^*, the one with the highest sample mean, is no less than the highest $U_j(n_j(t)) - \delta$, $j \neq i^*$, we select System i^* and stop. Since the inequalities for the upper and lower confidence limits should hold with high probability when the procedure terminates, we expect that the selected system i^* is probably a δ-optimal system. We will formally prove this result in §3.2.3.

The behavior of the EP is illustrated in Figure 3.1. The figure originally appeared in [43]. The example has $k = 3$ systems. The sample mean $\bar{X}_i(n_i(t))$ and upper and lower confidence limits $U_i(n_i(t))$ and $L_i(n_i(t))$ for each system i are plotted versus the iterator t. In this example we use the multi-samples strategy discussed in §4.2.1. Samples are drawn from systems iteratively, and the total sample size varies across systems. Due to the sampling rule, the green system has

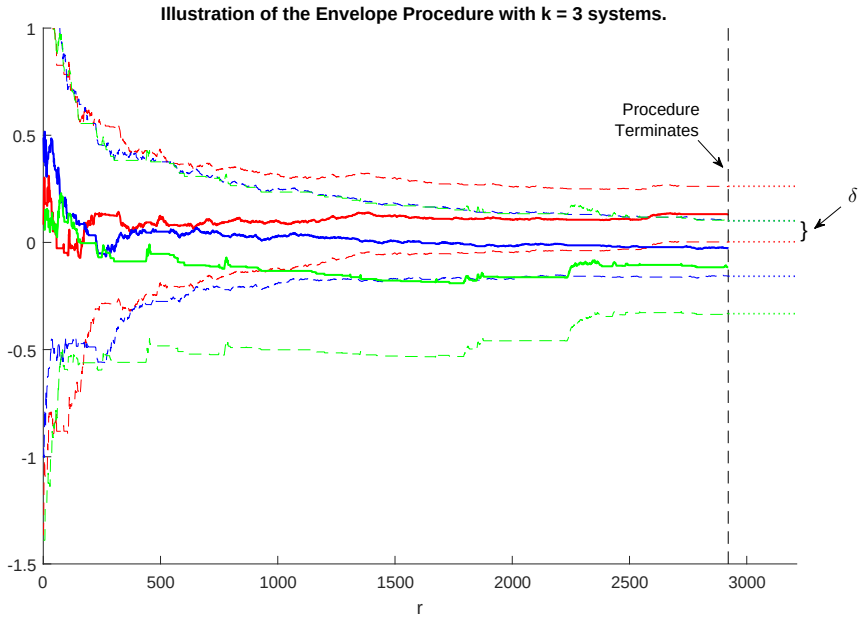

Figure 3.1: Illustration of EP. Systems are distinguished by different colors. The solid lines are $\bar{X}_i(n_i)$'s and the dashed lines are $U_i(n_i)$'s and $L_i(n_i)$'s throughout the procedure. The dotted lines are the final upper and lower confidence bounds when the procedure stops. The black dashed line indicates termination of the procedure when the stopping rule is met.

fewer samples and hence wider confidence interval, since its mean is relatively lower than those of the red one and blue one. As $n_i(t)$ increases for each system, the sample means are converging to their true means and the confidence intervals are shrinking. Finally, when the difference between the lower confidence limit of System i^* (the red one) and the highest upper confidence limit of the others is no more than δ, the procedure stops and we take i^* as the output. Here the final upper confidence limits of the inferior systems (green and blue) are overlapping

(the second dotted line from the top) since they are very close, which is due to our choice of the sampling rule.

3.2.2 Sampling Rule

The goal of the sampling rule is to achieve high efficiency, i.e., take as few samples as possible before the procedure terminates. Intuitively, we do not want to waste samples on inferior systems. In other words, we should focus the computational effort on those systems that are more likely to be the best system, or its strong competitors. From another point of view, we want the stopping rule to be met as soon as possible, so we should focus on the systems that make a difference to (3.3).

Top-Two Strategy We present a very simple sampling rule, where we only consider the two systems that show up explicitly in (3.3), i.e., Systems i^* and j^*. In round t we draw one sample from $i^*(t)$ and one sample from $j^*(t)$. Formally,

$$m_i(t) = \begin{cases} 1 & \text{if } i = i^*(t) \text{ or } i = j^*(t) \\ 0 & \text{otherwise.} \end{cases}$$

The same sampling rule is also adopted in [33]. The simplicity of this sampling strategy makes it possible to analyze its sample complexity, as in §3.2.3.

The top-two strategy has a good theoretical bound on its sample complexity, as

we will see in §3.2.3. However, when we solve practical problems using stochastic simulation, the number of systems can be large, and thus parallel computing might be adopted for simulation. In such a case, the 2-sample strategy might not be very efficient. We propose some sampling strategies for practical use in both sequential and parallel computing environment in §4.2.

3.2.3 Theoretical Analysis

The theoretical analysis of the algorithm comprises two aspects: statistical validity and sample complexity. The statistical validity is about the PAC guarantee of the algorithm, while the sample complexity is a high-probability worst-case bound on the total sample size of the algorithm before it terminates.

An efficient algorithm should be able to use fewer samples for easier problems. We say a problem configuration is hard when it is difficult to distinguish the best system from other systems. Intuitively, such difficulty for System $i \in \mathcal{S}$ is high when its mean is close to μ_k and its variance is large. We denote by $\Delta_i = \mu_k - \mu_i$, the difference between the means of Systems i and k. In our setting, when a system is δ-optimal, we do not need to know how close it is to the best, so the quantity we care about is $\max(\Delta_i, \delta)$. Hence, we define the complexity of the problem configuration as

$$H_\delta = \sum_{i=1}^{k} \left(\frac{\sigma_i}{\max(\Delta_i, \delta)} \right)^2 .$$

Ideally, an efficient algorithm accommodates problem configurations of different

complexity, and should take fewer samples when the problem is easier. In the following, we denote $\sigma = \max_i \sigma_i$ to simplify the notation.

Theorem 3 below has two parts. The first part states the statistical validity of the procedure, i.e., it has the PAC guarantee, and the result does not rely on the choice of the sampling rule. The second part gives a high probability bound of the sample complexity of the procedure, under a specific choice of the sampling rule.

Theorem 3. *If A0, A1 and A2 hold, then with probability at least* $1 - \alpha$*, (1) KEP returns a* δ*-optimal system, and (2) using the 2-sample strategy, the total sample size is upper bounded by* $16\eta^2 H_\delta$*, where a lower bound on* η^2 *is given in Lemma 2, and that bound is*

$$\mathcal{O}\left(\ln \frac{k}{\alpha} + \ln \ln \frac{\sigma}{\delta}\right), \text{ as } \frac{k}{\alpha} \to \infty \text{ and } \frac{\sigma}{\delta} \to \infty.$$

Proof of (1). We first show that the procedure always terminates in finite time. If $n_{i^*(t)} < N_{i^*}$ or $n_{j^*(t)} < N_{j^*}$, then the algorithm is still drawing samples. Otherwise, we plug $n_{i^*(t)} = N_{i^*} = \left\lceil (2\eta\sigma_{i^*}/\delta)^2 \right\rceil$ and $n_{j^*(t)} = N_{j^*} = \left\lceil (2\eta\sigma_{j^*}/\delta)^2 \right\rceil$ into (3.3), giving

$$\begin{aligned}
\bar{X}_{i^*}(n_{i^*}(t)) - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}(t)}} &\geq \bar{X}_{i^*}(n_{i^*}(t)) - \frac{\delta\eta\sigma_{i^*}}{2\eta\sigma_{i^*}} \\
&= \bar{X}_{i^*}(n_{i^*}(t)) - \frac{\delta}{2} \\
&\geq \bar{X}_{j^*}(n_{j^*}(t)) - \frac{\delta}{2} \\
&= \bar{X}_{j^*}(n_{j^*}(t)) + \frac{\delta\eta\sigma_{j^*}}{2\eta\sigma_{i^*}} - \delta \\
&\geq \bar{X}_{j^*}(n_{j^*}(t)) + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*}(t)}} - \delta.
\end{aligned}$$

It follows that the stopping rule is met. Therefore, the procedure terminates by the time that $n_{i^*(t)} = N_{i^*}$ and $n) j^*(t) = N_{j^*}$.

Under our ascending means assumption, System k is the true best system. Define events

$$\begin{aligned} A_i &= \left\{ \bar{X}_i(n) \le \mu_i + \frac{\eta\sigma_i}{\sqrt{n}}, \forall n \le N \right\} \\ &= \left\{ \frac{n\bar{X}_i(n) - n\mu_i}{\sigma_i} \le \eta\sqrt{n}, \forall n \le N \right\} \\ &= \left\{ W_i(n) \le \eta\sqrt{n}, \forall n \le N \right\}, \end{aligned}$$

for $i = 1, 2, \ldots, k-1$, where $W_i(n)$ is a standard Brownian motion observed at discrete time n. Also, define

$$\begin{aligned} A_k &= \left\{ \bar{X}_k(n) \ge \mu_k - \frac{\eta\sigma_k}{\sqrt{n}}, \forall n \le N \right\} \\ &= \left\{ \frac{n\bar{X}_k(n) - n\mu_k}{\sigma_k} \ge -\eta\sqrt{n}, \forall n \le N \right\} \\ &= \left\{ W_k(n) \ge -\eta\sqrt{n}, \forall n \le N \right\}. \end{aligned}$$

Since $N_i \le N$, $\forall i = 1, 2, \ldots, k$, the inequalities hold throughout the whole procedure. By (3.1),

$$P(A_i) \ge 1 - a, \forall i = 1, 2, \ldots, k.$$

and by assumption A1, the A_i's are independent, so that

$$P(\bigcap_{i=1}^{k} A_i) = \prod_{i=1}^{k} P(A_i) \ge (1-a)^k = 1 - \alpha.$$

Denote the final round as T and the index of the selected system as $i^* = i^*(T)$. By definition of A_i, on the event $\bigcap_{i=1}^{k} A_i$,

$$\mu_i \ge \bar{X}_i(n) - \frac{\eta\sigma_i}{\sqrt{n}}, \forall n < N, \forall i = 1, 2, \ldots, k-1, \tag{3.4}$$

and

$$\mu_k \leq \bar{X}_k(n) + \frac{\eta\sigma_k}{\sqrt{n}}, \forall n \leq N. \tag{3.5}$$

For the selected system i^* and on the event $\bigcap_{i=1}^{k} A_i$, , there are two cases: (i) $i^* = k$, or (ii) $i^* \neq k$.For case (i), a δ-optimal system is selected. In case (ii), since $n_i(T) \leq N$, $\forall i = 1, 2, \ldots, k$, we have

$$\begin{aligned}
\mu_{i^*} &\geq \bar{X}_{i^*}(n_{i^*}(T)) - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}(T)}} \\
&\geq \bar{X}_{j^*}(n_{j^*}(T)) + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*}(T)}} - \delta \\
&\geq \bar{X}_k(n_k(T)) + \frac{\eta\sigma_k}{\sqrt{n_k(T)}} - \delta \\
&\geq \mu_k - \delta,
\end{aligned}$$

where the first and fourth inequalities are due to the definition of A_{i^*} and A_k, the second is due to (3.3), and the third is from the definition of j^*. Therefore i^* is also a δ-optimal system. Hence,

$$\mathbb{P}\left(i^* \text{ is a } \delta\text{-optimal system}\right) \geq P(\bigcap_{i=1}^{k} A_i) \geq 1 - \alpha.$$

□

Before proving the second part of Theorem 3, we present a lemma giving a bound on the hitting probability of the random walk with sub-Gaussian increments for square-root boundaries. The proof is inspired by [28].

Lemma 1. *Let $X_1, X_2, \ldots, X_N$ be i.i.d. standard (with scale parameter 1 and mean 0) sub-Gaussian random variables and denote $W_n = \sum_{j=1}^{n} X_j$. For any $d > 0$ and $\varepsilon > 0$, with*

probability at least $1 - 2\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \exp\left(-(1+\varepsilon)d\right)$,

$$W_n \le (1+\sqrt{\varepsilon})\sqrt{2(1+\varepsilon)dn}, \forall n = 1, 2, \ldots, N.$$

Proof. First, by Lemma 3 in the appendix with $f(\cdot) = 1$,

$$\mathbb{P}\left(W_n \ge \sqrt{2xn}\right) \le \exp\left(-x\right), \forall n, \forall x.$$

We define a sequence of important integers by induction: $u_0 = 1$, $u_{i+1} = \lceil (1+\varepsilon)u_i \rceil$. Then using a union bound

$$\begin{aligned} \mathbb{P}\left(\exists i, \text{ s.t. } u_i \le N : W_{u_i} \ge \sqrt{2(1+\varepsilon)du_i}\right) &\le \sum_{i=1}^{\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil} \mathbb{P}\left(W_{u_i} \ge \sqrt{2(1+\varepsilon)du_i}\right) \\ &\le \left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \exp\left(-(1+\varepsilon)d\right). \end{aligned} \tag{3.6}$$

In what follows we use Hoeffding's maximal inequality,

$$\mathbb{P}\left(\max_{n=1,2,\ldots,m} W_n \ge x\right) \le \exp\left(-\frac{x^2}{2m}\right).$$

This inequality appears in [28], but we have not found a proof anywhere,, so we proved it as Lemma 3 and 4 in the appendix.

Next, consider the time period between u_i and u_{i+1}. By the above inequality,

$$\begin{aligned} P_i &:= \mathbb{P}\left(\exists n \in \{u_i + 1, \ldots, u_{i+1} - 1\} : W_n - W_{u_i} \ge \sqrt{2\varepsilon d u_{i+1}}\right) \\ &= \mathbb{P}\left(\exists n \in \{1, \ldots, u_{i+1} - 1 - u_i\} : W_n \ge \sqrt{2\varepsilon d u_{i+1}}\right) \\ &\le \exp\left(-\frac{2\varepsilon d u_{i+1}}{2(u_{i+1} - 1 - u_i)}\right) \\ &\le \exp\left(-\varepsilon d \frac{1+\varepsilon}{\varepsilon}\right) \\ &= \exp\left(-(1+\varepsilon)d\right). \end{aligned}$$

Similarly, applying a union bound yields

$$\mathbb{P}\left(\exists i,\ \text{s.t. } u_i \le N, \exists n \in \{u_i+1,\dots,u_{i+1}-1\} : W_n - W_{u_i} \ge \sqrt{2\varepsilon d u_{i+1}}\right) \le \sum_{i=1}^{\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil} P_i$$
$$\le \left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \exp\left(-(1+\varepsilon)d\right). \tag{3.7}$$

Let $u_\ell(n) = u_i$ and $u_u(n) = u_{i+1}$ such that $u_i \le n < u_{i+1}$. Combining (3.6) and (3.7), and the fact that $u_u(n) \le n(1+\varepsilon)$, we have that with probability at least $1 - 2\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \exp\left(-(1+\varepsilon)d\right)$, $\forall n = 1, 2, \dots, N$,

$$\begin{aligned} W_n &= \left(W_n - W_{u_\ell(n)}\right) + W_{u_\ell(n)} \\ &\le \sqrt{2\varepsilon d u_u(n)} + \sqrt{2(1+\varepsilon) d u_\ell(n)} \\ &\le (1+\sqrt{\varepsilon})\sqrt{2(1+\varepsilon)dn} \end{aligned}$$

□

Based on the previous lemma, we give a sufficient condition on choosing η, which determines the sample size of the procedure.

Lemma 2. *A sufficient condition for (3.1) and (3.2) is that*

$$\eta^2 \ge 2(1+\sqrt{\varepsilon})^2 \ln\left(\frac{8\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right) = \mathcal{O}\left(\ln\frac{k}{\alpha} + \ln\ln\frac{\sigma}{\delta}\right),\ \textit{as } \frac{k}{\alpha} \to \infty \textit{ and } \frac{\sigma}{\delta} \to \infty.$$

Proof. Let $\eta = (1+\sqrt{\varepsilon})\sqrt{2(1+\varepsilon)d}$. Then, by Lemma 1,

$$\mathbb{P}\left(W_n \le \eta\sqrt{n}, \forall n = 1, \dots, N\right) \ge 1 - 2\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \exp\left(-\frac{\eta^2}{2(1+\sqrt{\varepsilon})^2}\right).$$

Plugging (3.2) into (3.1) and using $\left\lceil \frac{\ln N}{\ln(1+\varepsilon)} \right\rceil \leq \frac{\ln(N(1+\varepsilon))}{\ln(1+\varepsilon)}$, we need

$$\frac{4\ln\left(\frac{2\eta\sigma(1+\varepsilon)}{\delta}\right)}{\ln(1+\varepsilon)} \exp\left(-\frac{\eta^2}{2(1+\sqrt{\varepsilon})^2}\right) \leq a,$$

which is equivalent to

$$\eta \geq (1+\sqrt{\varepsilon})\left(2\ln\left(\frac{4\ln\left(\frac{2\eta\sigma(1+\varepsilon)}{\delta}\right)}{a\ln(1+\varepsilon)}\right)\right)^{\frac{1}{2}}. \tag{3.8}$$

If (3.8) does not hold, then using the fact that $\ln(x) \leq x$,

$$\begin{aligned}\eta &< (1+\sqrt{\varepsilon})\left(2\ln\left(\frac{4\ln\left(\frac{2\eta\sigma(1+\varepsilon)}{\delta}\right)}{a\ln(1+\varepsilon)}\right)\right)^{\frac{1}{2}} \\ &\leq (1+\sqrt{\varepsilon})\left(\frac{8\ln\left(\frac{2\eta\sigma(1+\varepsilon)}{\delta}\right)}{a\ln(1+\varepsilon)}\right)^{\frac{1}{2}} \\ &\leq 4(1+\sqrt{\varepsilon})\left(\frac{\eta\sigma(1+\varepsilon)}{a\delta\ln(1+\varepsilon)}\right)^{\frac{1}{2}}.\end{aligned} \tag{3.9}$$

Hence

$$\eta < \frac{16(1+\sqrt{\varepsilon})^2(1+\varepsilon)\sigma}{a\delta\ln(1+\varepsilon)}.$$

Plug this final inequality in the right-hand side of (3.9) to get

$$\eta < (1+\sqrt{\varepsilon})\left(2\ln\left(\frac{8\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right)\right)^{\frac{1}{2}}.$$

Therefore, we get a sufficient condition for (3.8):

$$\eta^2 \geq f_\eta(\sigma,\delta,a,\varepsilon) = 2(1+\sqrt{\varepsilon})^2\ln\left(\frac{8\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right). \tag{3.10}$$

Then we analyze the asymptotic behavior of f_η when $\sigma/\delta \to \infty$ and $a \to 0$. When σ/δ is large enough and a is close enough to 0, and using the fact that for $A \geq 2$ and $B \geq 2$, $AB \geq A + B$,

$$
\begin{aligned}
f_\eta(\sigma, \delta, a, \varepsilon) &= 2(1+\sqrt{\varepsilon})^2 \ln\left(\frac{8\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right) \\
&= 2(1+\sqrt{\varepsilon})^2 \ln\left(\frac{8\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right) + 8\ln\left(\frac{1}{(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right) \\
&\leq 2(1+\sqrt{\varepsilon})^2 \ln\left(\frac{64\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right)\ln\left(\frac{1}{(a\ln(1+\varepsilon))^{\frac{1}{2}}}\right)}{a\ln(1+\varepsilon)}\right) \\
&\leq 2(1+\sqrt{\varepsilon})^2 \ln\left(\frac{64\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right)}{(a\ln(1+\varepsilon))^{\frac{3}{2}}}\right) \\
&= 2(1+\sqrt{\varepsilon})^2 \left(\frac{3}{2}\ln\left(\frac{16}{\ln(1+\varepsilon)}\right) + \frac{3}{2}\ln\left(\frac{1}{a}\right) + \ln\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right)\right) \\
&= \mathcal{O}\left(\ln\frac{1}{a} + \ln\ln\frac{\sigma}{\delta}\right)
\end{aligned}
$$

Since α is bounded away from 1, $a = 1 - (1-\alpha)^{1/k} = \mathcal{O}(\frac{\alpha}{k})$, so

$$\eta^2 = \mathcal{O}\left(\ln\frac{k}{\alpha} + \ln\ln\frac{\sigma}{\delta}\right)$$

□

Remark: One might attempt to reduce the order of η^2 above by optimizing over ε. As we show in Proposition 3 in the appendix, the order of the solution

to the minimization problem is $\varepsilon = \mathcal{O}\left(\left(\ln\frac{k}{\alpha} + \ln\ln\frac{\sigma}{\delta}\right)^{-1}\right)$, which does not affect the order of η^2.

Now with all of the ingredients given above, we are ready to prove the second part of Theorem 3. for notation simplicity, in what follows we omit the ceiling signs when we refer to the number of samples, which does not hurt the validity of the proof.

Proof of Part (2) of Theorem 3. Throughout the proof suppose we are on the event $\bigcap_{i=1}^{k} A_i$, and then (3.4) and (3.5) hold. Denote the set of all δ-optimal systems as $\mathcal{G}$ (good systems), and that of all others $\mathcal{B}$ (bad systems). Further, for System $i \in \mathcal{G}$, we say it is active at time t if $n_i(t) < N_i$, and inactive if $n_i(t) = N_i$. We do not define active or inactive for System $i \in \mathcal{B}$, so when we say a system is active or inactive that also implies that it is a good system.

We count the total number of rounds before termination, since the total sample size is at most twice the total number of rounds. Split all of the rounds t into the following subsets (see Table 3.1): (I) at least one of $i^*(t)$ and $j^*(t)$ are active, (II) $j^*(t) \in \mathcal{B}$ and $i^*(t)$ is not active (i.e., $i^*(t) \in \mathcal{B}$ or $i^*(t)$ is inactive), and (III) $j^*(t)$ is inactive and $i^*(t) \in \mathcal{B}$. This is a partition of the rounds since it if both $i^*(t)$ and $j^*(t)$ is inactive, the procedure terminates by our previous proof.

For Case (I), there are at most $\sum_{i\in\mathcal{G}} N_i = \sum_{i\in\mathcal{G}} \left(\frac{2\eta\sigma_i}{\delta}\right)^2$ rounds, since at least one active system gets one sample in such a round and an active system i becomes inactive once $n_i(t) = N_i$.

Table 3.1: Partition of the possible cases of rounds.

	$j^*(t)$ is active	$j^*(t)$ is inactive	$j^*(t) \in \mathcal{B}$
$i^*(t)$ is active	I	I	I
$i^*(t)$ is inactive	I	Termination	II
$i^*(t) \in \mathcal{B}$	I	III	II

For Case (II), there are two sub-cases: (i) $i^*(t) \neq k$ and (ii) $i^*(t) = k$ and System k is inactive. If (i), then by the definition of $i^*(t)$, $j^*(t)$, (3.4) and (3.5),

$$\begin{aligned}
\mu_{j^*(t)} + \frac{2\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} &\geq \bar{X}_{j^*(t)}(n_{j^*(t)}(t)) + \frac{\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} \\
&\geq \bar{X}_k(n_k(t)) + \frac{\eta\sigma_k}{\sqrt{n_k(t)}} \\
&\geq \mu_k,
\end{aligned}$$

which implies $n_{j^*(t)}(t) \leq \left(\frac{2\eta\sigma_{j^*(t)}}{\Delta_{j^*(t)}}\right)^2$.

If (ii), if the procedure has not terminated yet, then by the stopping rule,

$$\bar{X}_k(n_k(t)) - \frac{\eta\sigma_k}{\sqrt{n_k(t)}} \leq \bar{X}_{j^*(t)}(n_{j^*(t)}(t)) + \frac{\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} - \delta.$$

Since System k is inactive, $n_k(t) = N_k = \left(\frac{2\eta\sigma_k}{\delta}\right)^2$, so

$$\begin{aligned}\mu_{j^*(t)} + \frac{2\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} - \delta &\geq \bar{X}_{j^*(t)}(n_{j^*(t)}(t)) + \frac{\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} - \delta \\ &\geq \bar{X}_k(n_k(t)) - \frac{\eta\sigma_k}{\sqrt{N_k}} \\ &\geq \mu_k - \frac{2\eta\sigma_k}{\sqrt{N_k}} \\ &= \mu_k - \delta,\end{aligned}$$

which implies $n_{j^*(t)}(t) \leq \left(\frac{2\eta\sigma_{j^*(t)}}{\Delta_{j^*(t)}}\right)^2$.

Therefore, in either sub-case (i) or (ii), $n_{j^*(t)}(t) \leq \left(\frac{2\eta\sigma_{j^*(t)}}{\Delta_{j^*(t)}}\right)^2$, so there are at most $\sum_{i\in\mathcal{B}}\left(\frac{2\eta\sigma_i}{\Delta_i}\right)^2$ rounds in case (II).

For Case (III), because $i^*(t) \neq k$, by definition of $j^*(t)$,

$$\bar{X}_{j^*(t)}(n_{j^*(t)}(t)) + \frac{\eta\sigma_{j^*(t)}}{\sqrt{n_{j^*(t)}(t)}} \geq \bar{X}_k(n_k(t)) + \frac{\eta\sigma_k}{\sqrt{n_k(t)}} \geq \mu_k,$$

and since $j^*(t)$ is inactive, $n_{j^*(t)}(t) = N_{j^*(t)}$, so

$$\bar{X}_{j^*(t)}(n_{j^*(t)}(t)) \geq \mu_k - \frac{\eta\sigma_{j^*(t)}}{\sqrt{N_{j^*(t)}}} = \mu_k - \frac{\delta}{2}.$$

Therefore, by (3.4) and the definition of $i^*(t)$,

$$\begin{aligned}\mu_{i^*(t)} + \frac{\eta\sigma_{i^*(t)}}{\sqrt{n_{i^*(t)}(t)}} &\geq \bar{X}_{i^*(t)}(n_{i^*(t)}(t)) \\ &\geq \bar{X}_{j^*(t)}(n_{j^*(t)}(t)) \\ &\geq \mu_k - \frac{\delta}{2},\end{aligned}$$

which implies $n_{i^*(t)}(t) \leq \left(\frac{\eta\sigma_{i^*(t)}}{\Delta_{i^*(t)} - \frac{\delta}{2}}\right)^2 \leq \left(\frac{2\eta\sigma_{i^*(t)}}{\Delta_{i^*(t)}}\right)^2$ since $\Delta_{i^*(t)} \geq \delta$. Therefore there are at most $\sum_{i\in\mathcal{B}}\left(\frac{2\eta\sigma_i}{\Delta_i}\right)^2$ rounds in case (III).

Summarizing the analysis for (I), (II) and (III) yields that the total number of rounds is at most

$$\sum_{i\in\mathcal{G}}\left(\frac{2\eta\sigma_i}{\delta}\right)^2+\sum_{i\in\mathcal{B}}\left(\frac{2\eta\sigma_i}{\Delta_i}\right)^2+\sum_{i\in\mathcal{B}}\left(\frac{2\eta\sigma_i}{\Delta_i}\right)^2\le 2\sum_{i=1}^{k}\left(\frac{2\eta\sigma_i}{\max(\Delta_i,\delta)}\right)^2\le 8\eta^2 H_\delta.$$

Combining with Lemma 2 completes the proof.

□

3.2.4 On Computing the Parameter η

Computing the Bound on η

The parameter η determines the width of the confidence bounds. The statistical validity of the EP requires that η should be large enough so that the probability that the sample mean of any system will cross the boundary is small. On the other hand, the total sample size is roughly proportional to η^2, so for the sake of efficiency we should not set η too large. In other words, we want to find the smallest η that satisfies (3.1).

For general sub-Gaussian samples, we can replace the inequality in (3.8) by equality and solve the fixed point problem

$$\eta = f_1(\eta) = (1+\sqrt{\varepsilon})\left(2\ln\left(\frac{4\ln\left(\frac{2\eta\sigma(1+\varepsilon)}{\delta}\right)}{a\ln(1+\varepsilon)}\right)\right)^{\frac{1}{2}}$$

through iteration,

$$\eta_{n+1} = f_1(\eta_n)$$

with any reasonable starting point η_0. In practice this iteration usually converges in several steps, but we do not attempt to prove that, because ε plays a role only in our complexity analysis, where any positive fixed value suffices. In our implementations, we compute η in a different way.

Computing η Numerically

Instead of computing a bound on η, one can also find an approximate value of η with Monte Carlo. Based on our analysis in §3.2.3, η is roughly linear in $(\ln \ln N)^{1/2}$, so the efficiency of the algorithm will not be heavily affected if we choose N slightly larger than what is needed. For given parameters α, k (hence a) and N, we can use Monte Carlo, simulating random walks with normally distributed increments and approximating η by the $1-a$ sample quantile of $\max_{n \le N} W_n / n^{1/2}$.

Here we fix $\alpha = 0.05$, and take $k = 10, 100, 1000$ and 10000, $N = 10^3, 10^4, 10^5$. The results are shown in Table 3.2. For fixed k (or equivalently, fixed a), as N increases by orders of magnitude, the corresponding η only increases by a small amount. For example, when $k = 1000$, η only increases by 1% when N increases from 10^4 to 10^5, which only causes an increase in the total number of samples by 2%, agreeing with our analysis above. Therefore, in practice we can first estimate

a large enough upper bound for η, and find the smallest N that satisfies (3.2) in Table 3.2 to get the value of η.

Table 3.2: Simulated η for different values of k and N.

k	$N = 10^3$	$N = 10^4$	$N = 10^5$
10	3.58	3.69	3.77
100	4.20	4.30	4.37
1000	4.73	4.83	4.88
10000	5.22	5.28	5.35

From simulation results, we can fit a function that yields a reasonable value of η. Based on (3.8), we consider functions of the following form:

$$\eta = \left(c_0 + c_1 \ln \left(\frac{\ln \frac{c_2 \sigma}{\delta}}{a} \right) \right)^{\frac{1}{2}}. \tag{3.11}$$

To use the simulation results, by (3.2), we let $\sigma/\delta = N^{1/2}/(2\eta)$. Fitting the above function to the simulation results, we get $c_0 = -0.318$, $c_1 = 2.114$ and $c_2 = 3.231$. Using the fitted function, we recalculate η for given k and N, and get Table 3.3. The values in Table 3.2 and Table 3.3 are very close, indicating a good fit.

Table 3.3: Fitted η for different values of k and N.

k	$N = 10^3$	$N = 10^4$	$N = 10^5$
10	3.59	3.69	3.77
100	4.20	4.29	4.36
1000	4.73	4.82	4.88
10000	5.21	5.28	5.35

3.3 Unknown σ^2 Envelope Procedure

Usually the scale parameters or the variances of the systems are unknown in advance and there is no explicit bound on them. Same regularity assumption is essential [19]; as in much R&S literature, in this section we assume A3, i.e., normally distributed samples. We present two EP variants with unknown σ: UEP, which is a two-stage procedure, where the the first stage provides variance estimates, and UEPu, in which we keep updating the sample variances throughout the procedure. These two procedures can be seen as generalizations of the KEP since the structure and intuition behind them is similar.

3.3.1 The Procedure

UEP

1. Initialization. Initialize $t = 0$. Obtain n_0 samples X_{ij}, $j = 1, 2, \ldots, n_0$ from each system $i = 1, 2, \ldots, k$. Compute sample means $\bar{X}_i(n_0)$ and sample variances S_i^2.

2. Setup. Let $a = 1 - (1 - \alpha)^{1/k}$. Compute η, such that

$$\mathbb{E}\left[F_{n_0-1}\left(\frac{(n_0-1)Z^2}{\eta^2}\right)\right] \leq a, \tag{3.12}$$

where

$$N = \left\lceil \left(\frac{2\eta \max_i S_i}{\delta}\right)^2 \right\rceil \text{ and } Z = \max_{n=1,\ldots,N} \frac{W_n}{\sqrt{n}}, \tag{3.13}$$

$(W_n, n = 1, 2, \ldots)$ is a random walk with i.i.d. standardized Gaussian increments and F_{n_0-1} is the cumulative distribution function of the chi-squared distribution with $n_0 - 1$ degrees of freedom. Calculate $N_i = (2\eta S_i/\delta)^2$ for each system i.

3. Stopping Rule. If

$$\bar{X}_{i^*}(n_{i^*}(t)) - \frac{\eta S_{i^*}}{\sqrt{n_{i^*}(t)}} \geq \bar{X}_j(n_{j^*}(t)) + \frac{\eta S_{j^*}}{\sqrt{n_{j^*}(t)}} - \delta,$$

where

$$i^*(t) = \underset{i=1,2,\ldots,k}{\operatorname{argmax}} \bar{X}_i(n_i(t)), \qquad j^*(t) = \underset{j \neq i^*}{\operatorname{argmax}} \bar{X}_j(n_j(t)) + \frac{\eta S_j}{\sqrt{n_j(t)}},$$

go to Step 5. Otherwise go to Step 4.

4. Sampling Rule. Compute $m_i(t)$ for $i = 1, 2, \ldots, k$ according to a sampling rule, and take $\max(N_i - n_i(t), m_i(t))$ samples from each system i. Let $t = t + 1$ and update $n_i(t)$, $\bar{X}_i(n_i(t))$, i^* and j^*. Go to Step 3.

5. Termination. Stop and select system $I^* = i^*$ as the best.

The UEP takes an additional parameter n_0, which is the sample size of the initial stage to obtain the sample variances of the systems. In practice we recommend a value between 10 and 100. If n_0 is too small then the boundary can be too large due to large uncertainty in the variance estimates, while if n_0 is too large then many samples can be wasted. In our experiments we set $n_0 = 20$ or 50. After the initial stage, UEP is similar to KEP, replacing the known variances σ_i^2 by the estimated variance S_i^2, for each $i = 1, 2, \ldots, k$.

We also develop a heuristic procedure, UEPu, based on UEP, in which variance estimators are updated as samples are gathered. UEPu is heuristic in the sense that it does not offer a formal PAC guarantee. We expect that UEPu is more efficient than UEP because the variance estimates become more and more accurate as the procedure runs. Experimental results in §3.5 support this conjecture.

UEPu

1. Initialization. Initialize $t = 0$. Obtain n_0 samples X_{ij}, $j = 1, 2, \ldots, n_0$ from each system $i = 1, 2, \ldots, k$. Compute sample means $\bar{X}_i(n_0)$ and sample variances $S_i^2(n_0)$.
2. Setup. Let $a = 1 - (1 - \alpha)^{1/k}$. Compute η, such that

$$\mathbb{P}\left(\frac{W_n}{\sqrt{n}S_n} \leq \eta, \forall n = 1, 2, \ldots, N\right) \geq 1 - a,$$

where

$$N = \left\lceil \left(\frac{2\eta \max_i S_i(n_0)}{\delta} \right)^2 \right\rceil,$$

$(W_n, n = 1, 2, \ldots)$ is a random walk with i.i.d. standardized Gaussian increments and $(S_n, n = 1, 2, \ldots)$ is the sample variance of W_n. Calculate $N_i = (2\eta S_i(n_0)/\delta)^2$ for each system i.

3. Stopping Rule. If

$$\bar{X}_{i^*}(n_{i^*}(t)) - \frac{\eta S_{i^*}(n_{i^*}(t))}{\sqrt{n_{i^*}(t)}} \geq \bar{X}_j(n_{j^*}(t)) + \frac{\eta S_{j^*}(n_{j^*}(t))}{\sqrt{n_{j^*}(t)}} - \delta,$$

where

$$i^*(t) = \underset{i=1,2,\ldots,k}{\operatorname{argmax}} \bar{X}_i(n_i(t)), \qquad j^*(t) = \underset{j \neq i^*}{\operatorname{argmax}} \bar{X}_j(n_j(t)) + \frac{\eta S_j}{\sqrt{n_j(t)}},$$

go to Step 5. Otherwise go to Step 4.

4. Sampling Rule. Compute $m_i(t)$ for $i = 1, 2, \ldots, k$ according to the sampling rule, and take $\max(N_i - n_i(t), m_i(t))$ samples from each system i. Let $t = t + 1$ and update $n_i(t)$, $\bar{X}_i(n_i(t))$, $S_i^2(n_i(t))$, i^* and j^*. Go to Step 3.

5. Termination. Stop and select system $I^* = i^*$ as the best.

For the sampling rule for both UEP and UEPu , we can use modified versions of the 2-sample strategy and multi-samples strategy, where the variances σ_i^2 are replaced by the estimated variances S_i^2. We omit the details since they are similar to those described in §3.2.2.

3.3.2 Theoretical Analysis

Theorem 4. *If A0, A1 and A3 hold, then with probability at least* $1-\alpha$*, (1) UEP returns a* δ*-optimal system, and (2) using the 2-sample strategy, the total sample size is* $\mathcal{O}\left(H_\delta \frac{1}{n_0^2}\left(\frac{\alpha}{2^{n_0/2}k}\right)^{2/n_0}\right)$.

Proof. Using the same notation as in Theorem 3, we define events

$$\begin{aligned} A_i &= \left\{\bar{X}_i(n) \le \mu_i + \frac{\eta s_i}{\sqrt{n}}, \forall n \le N\right\} \\ &= \left\{\frac{n\bar{X}_i(n) - n\mu_i}{\sigma_i} < \eta\sqrt{n}\cdot\frac{S_i}{\sigma_i}, \forall n < N\right\} \\ &= \left\{W_i(n) < \eta\sqrt{n}\cdot\frac{1}{\sqrt{n_0-1}}\cdot\frac{\sqrt{n_0-1}S_i}{\sigma_i}, \forall n < N\right\} \\ &= \left\{\frac{W_i(n)}{\sqrt{n}} < \eta\cdot\sqrt{\frac{\chi_i^2}{n_0-1}}, \forall n < N\right\} \end{aligned}$$

for $i = 1, 2, \ldots, k-1$, where $W_i(n)$ is a standard Brownian motion observed at discrete time n, and $(\chi_i^2 : 1 \le i \le k)$ are independent χ^2 random variables with $n_0 - 1$ degrees of freedom that are independent of $(W_i(n) : 1 \le i \le k, 1 \le n \le N)$. Also, define

$$\begin{aligned} A_k &= \left\{\bar{X}_k(n) > \mu_k - \frac{\eta S_k}{\sqrt{n}}, \forall n < N\right\} \\ &= \left\{\frac{W_k(n)}{\sqrt{n}} > -\eta\cdot\sqrt{\frac{\chi_k^2}{n_0-1}}, \forall n < N\right\} \end{aligned}$$

Following the same argument as in the proof of Theorem 3, on the event $\bigcap_{i=1}^k A_i$, the inequalities hold throughout the whole procedure. By (3.12), $\forall i = 1, 2, \ldots, k$,

letting $Z_i = \max_{n=1,\ldots,N} W_i/n^{1/2}$,

$$\begin{aligned}
P(A_i) &= P\left(Z_i < \eta \cdot \sqrt{\frac{\chi_i^2}{n_0 - 1}}\right) \\
&= P\left(\chi_i^2 > \frac{(n_0 - 1)Z_i^2}{\eta^2}\right) \\
&= \mathbb{E}\left[1 - F_{\chi^2_{n_0-1}}\left(\frac{(n_0 - 1)Z_i^2}{\eta^2}\right)\middle| Z_i\right] \\
&\geq 1 - a.
\end{aligned}$$

Following the same proof as in Theorem 3, we get that the total number of rounds is at most $8\eta^2 H_\delta$. Next we compute the order of the parameter η. According to Lemma 1, letting $\varepsilon = 1$, we get

$$P(Z \leq x) \leq G(x) = 2\ln_2 N e^{-\frac{x^2}{8}}.$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\mathbb{P}\left(Z \geq \eta\sqrt{\frac{\chi^2}{n_0 - 1}}\right)\right] \leq & \mathbb{E}\left[G\left(\eta\sqrt{\frac{\chi^2}{n_0 - 1}}\right)\right] \\
= & 2\ln_2 N \int_0^\infty e^{-\frac{-\eta^2 x}{8(n_0-1)}} \frac{1}{2^{n_0/2}\Gamma(n_0/2)} x^{n_0/2-1} e^{-x/2} dx \\
= & \frac{\ln_2 N}{2^{n_0/2}\Gamma(n_0/2)} \int_0^\infty x^{n_0/2-1} e^{-\left(\frac{\eta^2}{8(n_0-1)} + \frac{1}{2}\right)} dx \\
= & \frac{\ln_2 N}{2^{n_0/2}\Gamma(n_0/2)} \left(\frac{\eta^2}{8(n_0 - 1)} + \frac{1}{2}\right)^{-n_0/2} \Gamma\left(\frac{n_0}{2}\right) \\
= & \frac{\ln_2 N}{2^{n_0/2}} \left(\frac{\eta^2}{8(n_0 - 1)} + \frac{1}{2}\right)^{-n_0/2}.
\end{aligned}$$

Letting the above formula be less than or equal to a yields

$$\eta^2 \leq \frac{1}{8(n_0 - 1)} \left[\left(\frac{\ln_2 N}{2^{n_0/2-1} a}^{2/n_0}\right) - \frac{1}{2}\right].$$

Hence we have the order of the parameter η, and the rest of the proof is as the same as the proof of Theorem 3.

□

3.3.3 On Computing the Parameter η

In order to compute η that satisfies (3.12) for large enough N in (3.13), we can approximate the expectation by Monte Carlo:

$$\mathbb{E}\left[F_{\chi^2_{n_0-1}}\left(\frac{(n_0-1)Z^2}{\eta^2}\right)\right] \leq a \approx \frac{1}{n}\sum_{i=1}^{n} F_{\chi^2_{n_0-1}}\left(\frac{(n_0-1)Z^2}{\eta^2}\right).$$

We can estimate the distribution of Z using Monte Carlo as discussed in §3.2.4. We give the value of η for both UEP and UEPu for some commonly used scenarios, (which are also used in our experiments in §3.5) where $k = 100, 1000$, and 10000, $n_0 = 20$ and 50, and $N = 10^5$ in Table 3.4.

Table 3.4: Approximated value of η for UEP and UEPu for $N = 10^5$

k	UEP		UEPu	
	$n_0 = 20$	$n_0 = 50$	$n_0 = 20$	$n_0 = 50$
100	5.62	4.40	4.80	4.19
1000	6.77	5.18	5.51	4.80
10000	8.00	6.11	6.20	5.37

3.4 Lower Bound on the Sample Complexity of PAC Procedures

In this section we develop a lower bound on the sample complexity of PAC procedures, and compare the lower bound with the sample complexity of EP, assuming known variances.

Informally speaking, a PAC procedure sequentially obtains a sample from one of the competing systems, and then decides either to stop and declare one of the systems the best, or to continue with further sampling. Stopping decisions and decisions on which system to sample may be randomized. To capture this setup more formally, we construct a probability space as follows. Let $\Omega = [0,1] \times \bigotimes_{i=1}^{\infty} (\mathcal{S} \times \mathbb{R} \times [0,1])$ denote the sample space, and equip Ω with the usual product sigma field. Each $\omega \in \Omega$ takes the form

$$\omega = (u(0), (i(1), x(1), u(1)), ((i(2), x(2), u(2)), \ldots).$$

For $t = 1, 2, \ldots$, define the random variables $I(t, \omega) = i(t)$ (denoting the index of the system to be sampled at time t), and $X(t, \omega) = x(t)$ (the value of the observed sample from System i at time t). For $t = 0, 1, \ldots$, define the random variable $U(t, \omega) = u(t) \in [0, 1]$ (a random variable that permits us to randomize decisions).

Take $\mathcal{F}_0$ to be the sigma field generated by $U(0)$, and for $n \geq 1$ let $\mathcal{F}_n$ be the sigma field generated by $U(0), ((I(t), X(t), U(t)) : t = 1, 2, \ldots, n)$. For $t \geq 1$, we take $I(t)$ to be $\mathcal{F}_{t-1}$ measurable since the procedure can only make decisions on which system to sample based on the history. The procedure takes only one sample from one system at each step; procedures taking multiple samples at a step fit

within this framework through a redefinition of "step." For $t \geq 1$, given $I(t)$, $X(t)$ is conditionally independent of $\mathcal{F}_{t-1}$ and normally distributed with mean $\mu_{I(t)}$ and variance $\sigma^2_{I(t)}$. For $t \geq 1$, $U(t)$ is independent of $\mathcal{F}_{t-1}$ and of $I(t), X(t)$, and for $t \geq 0$ is uniformly distributed on $[0, 1]$. Thus $(U(t) : t \geq 0)$ is an i.i.d. sequence of uniform $[0, 1]$ random variables that can be viewed as providing randomization, if needed. The procedure stops at the finite-valued stopping time T, so that in particular, the event $\{T = t\}$ is $\mathcal{F}_t$ measurable. (Notice that T can depend on $U(T)$ so can be randomized.) At time T the procedure reports an $\mathcal{F}_T$-measurable system I^* as the best.

To simplify notation, in what follows we index the systems in the following way. First index all of the systems i where $\mu_i < \max_j \mu_j - \delta$ by μ_i in increasing order, i.e., $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_\ell < \max_j \mu_j - \delta$, assuming ℓ such systems. Then index all of the other systems by σ_i in increasing order, so that $\mu_i \geq \max_j \mu_j - \delta$, $\forall i > \ell$, and $\sigma_{\ell+1} \leq \sigma_{\ell+2} \ldots \leq \sigma_k$. Break ties arbitrarily.

The following result and proof is inspired by a related result provided by [44] for Bernoulli MAB models. Let $\mathbb{1}$ denote the usual indicator function.

Theorem 5. *Consider any selection procedure as defined immediately above that delivers a PAC guarantee. Let $T_i = \sum_{t=1}^{T} \mathbb{1}(I(t) = i)$ be the sample size for System i, and $T = \sum_{i=1}^{k} T_i$ be the total sample size. Under assumptions A0, A1 and A3, there exist positive constants α_0, c and b, such that for any $\alpha \leq \alpha_0 < \frac{1}{2}$,*

$$\mathbb{E}[T] > c \sum_{i=1}^{k-1} \left(\frac{\sigma_i}{\max(\Delta_i, \delta)} \right)^2 \ln \frac{1}{b\alpha}. \tag{3.14}$$

[30] also provides a lower bound on the sample complexity for elimination procedures. Their result applies to a specific class of elimination procedures in the IZ setting, the error probability α is equally assigned to each system, and identical variances are assumed. Theorem 5 can be viewed as a generalization of their result to any PAC procedure without those assumptions.

[35] gives a lower bound for i.i.d. sampling without assuming normality as we do. Their result does not allow relaxation in the problem, i.e., they take $\delta = 0$. Assuming normal sampling distributions with identical variances, their bound simplifies to $c\sum_{i=1}^{k-1}(\sigma/\Delta_i)^2 \ln(1/(b\alpha))$. Though the constants c and b are different, this bound has the same order as our result in the special case where $\delta = 0$.

PROOF. Consider the hypotheses

$$H_0 : \hat{\mu}_\ell = \mu_\ell, \text{ for } \ell = 1, 2, \ldots, k,$$

and for $i = 1, 2, \ldots, k$,

$$H_i : \hat{\mu}_i = \mu_k + \delta, \text{ and } \hat{\mu}_\ell = \mu_\ell \text{ for } \ell \neq i.$$

System i is the only δ-optimal system under H_i, $i \geq 1$. Let $\mathbb{E}_i$ and $\mathbb{P}_i$ denote the expectation and probability under hypothesis H_i, respectively, for $i = 0, 1, 2, \ldots, k$. Suppose, with the intention of reaching a contradiction, that for some $i \geq 1$,

$$\mathbb{E}_0[T_i] \leq t_i^* = \frac{c\sigma_i^2}{\max(\Delta_i, \delta)^2} \ln \frac{1}{b\alpha}. \tag{3.15}$$

Define the event $S_i = \{I^* = i\}$, i.e., System i is selected ultimately. Since $\sum_{i=1}^{k} \mathbb{P}_0(S_i) = 1$, the number of systems i that satisfy $\mathbb{P}_0(S_i) \geq 1/2$ is at most one.

In other words, for at least $k-1$ systems, $\mathbb{P}_0(S_i^c) \geq 1/2$, where S_i^c is the complement of S_i. From now on we restrict i to index one of these $k-1$ systems.

Next, define the event $A_i = \{T_i \leq at_i^*\}$, where a is a constant that we will decide later. Since

$$t_i^* \geq \mathbb{E}_0[T_i] \geq at_i^* \mathbb{P}_0(T_i > at_i^*) = at_i^*(1 - \mathbb{P}_0(T_i \leq at_i^*)),$$

we have $\mathbb{P}_0(A_i) \geq 1 - 1/a$. Recall that we define, for $t \geq 0$, $W_{i,t} = \sum_{j=1}^{t}(X_{ij} - \mu_i)$, where X_{ij} is the jth sample taken from System i. Define the event

$$B_i = \left\{ \max_{1 \leq t \leq at_i^*} \frac{W_{i,t}}{\sigma_i} \geq -\sqrt{t_i^* \ln \frac{1}{b\alpha}} \right\}.$$

Since $(W_{i,t}/\sigma_i : t = 1, 2, \ldots)$ is a random walk with i.i.d. standardized sub-Gaussian increments, by Lemma 4,

$$\mathbb{P}_0(B_i) \geq 1 - \exp\left(-\frac{t_i^* \ln \frac{1}{b\alpha}}{2at_i^*} \right) = 1 - (b\alpha)^{\frac{1}{2a}}.$$

Let $D_i = S_i^c \bigcap A_i \bigcap B_i$. Then

$$\mathbb{P}_0(D_i) \geq 1 - \frac{1}{a} - (b\alpha)^{\frac{1}{2a}} - \frac{1}{2} = \frac{1}{2} - \frac{1}{a} - (b\alpha)^{\frac{1}{2a}}.$$

Denote the sequence $(i(1), i(2), \ldots, i(t), x(1), x(2), \ldots, x(t), u(0), u(1), \ldots, u(t))$ by $\xi_{0:t}$. Define the likelihood

$$L_i(\xi_{0:T}) = \mathbb{1}(u(0) \in [0,1]) \prod_{t=1}^{T} \mathbb{P}(I(t) = i(t) | \mathcal{F}_{t-1}) \phi(x(t); \mu_{i(t)}, \sigma_{i(t)}^2) \mathbb{1}(u(t) \in [0,1]),$$

where $\phi(\cdot; \mu, \sigma^2)$ is the density function of the normal distribution with mean μ and variance σ^2.

Consider the pair of hypotheses H_0 and H_i, which differ only in the specification of the mean $\hat{\mu}_i$ of System i. How does the likelihood of $\xi_{0:T}$ differ under these two hypotheses? The sampling decisions $I(t)$ are $\mathcal{F}_{t-1}$ measurable ($t \geq 1$), and hence are the same. The likelihoods of the observed samples are thus different only for those t where $I(t) = i$, when the means are different. Therefore the likelihood ratio of these two hypothesis only depends on the samples from System i. Let X_{ij} be the j-th sample from System i, which is observed at the jth time that $I(t) = i$. Then, after cancelling common terms,

$$\begin{aligned}
\frac{L_i(\xi_{1:T})}{L_0(\xi_{1:T})} &= \prod_{j=1}^{T_i} \frac{\exp\left(-\frac{(X_{ij}-\mu_k-\delta)^2}{2\sigma_i^2}\right)}{\exp\left(-\frac{(X_{ij}-\mu_i)^2}{2\sigma_i^2}\right)} \\
&= \prod_{j=1}^{T_i} \exp\left(-\frac{(2X_{ij}-\mu_k-\delta-\mu_i)(\mu_i-\mu_k-\delta)}{2\sigma_i^2}\right) \\
&= \prod_{j=1}^{T_i} \exp\left(-\frac{(2(X_{ij}-\mu_i)+\mu_i-\mu_k-\delta)(\mu_i-\mu_k-\delta)}{2\sigma_i^2}\right) \\
&= \prod_{j=1}^{T_i} \exp\left(-\frac{(\mu_k+\delta-\mu_i)^2-2(X_{ij}-\mu_i)(\mu_k+\delta-\mu_i)}{2\sigma_i^2}\right) \\
&= \exp\left(\frac{2W_{i,T_i}(\Delta_i+\delta)-T_i(\Delta_i+\delta)^2}{2\sigma_i^2}\right),
\end{aligned}$$

recalling that $W_{i,t} = \sum_{j=1}^{t}(X_{ij}-\mu_i)$.

On the event D_i,

$$T_i \leq \frac{ac\sigma^2}{\max(\Delta_i,\delta)^2}\ln\frac{1}{b\alpha} \quad \text{and} \quad W_{i,T_i} \geq -\frac{\sqrt{c}\sigma^2\ln\frac{1}{b\alpha}}{\max(\Delta_i,\delta)},$$

so that

$$\begin{aligned}
\frac{L_i(\xi_{1:T}))}{L_0(\xi_{1:T}))} &= \exp\left(\frac{2W_{i,T_i}(\Delta_i+\delta) - T_i(\Delta_i+\delta)^2}{2\sigma_i^2}\right) \\
&\geq \exp\left(\frac{-\frac{2\sqrt{c}\sigma^2(\Delta_i+\delta)\ln\frac{1}{b\alpha}}{\max(\Delta_i,\delta)} - \frac{ac\sigma_i^2(\Delta_i+\delta)^2\ln\frac{1}{b\alpha}}{\max(\Delta_i,\delta)^2}}{2\sigma_i^2}\right) \\
&\geq \exp\left(-2\sqrt{c}\ln\frac{1}{b\alpha} - 2ac\ln\frac{1}{b\alpha}\right) \\
&= (b\alpha)^{2\sqrt{c}+2ac},
\end{aligned}$$

where the second inequality is due to the fact that $(\Delta_i + \delta)/\max(\Delta_i, \delta) \leq 2$. By changing the probability measure,

$$\begin{aligned}
\mathbb{P}_i(S_i^c) &\geq \mathbb{P}_i(D_i) \\
&= \mathbb{E}_i[\mathbb{1}(D_i)] \\
&= \mathbb{E}_0\left[\frac{L_i(\xi_{1:T}))}{L_0(\xi_{1:T}))}\mathbb{1}(D_i)\right] \\
&\geq (b\alpha)^{2\sqrt{c}+2ac}\mathbb{E}_0[\mathbb{1}(D_i)] \\
&\geq (b\alpha)^{2\sqrt{c}+2ac}\left(\frac{1}{2} - \frac{1}{a} - (b\alpha)^{\frac{1}{2a}}\right).
\end{aligned}$$

For any given α, one can set the constants a and b large enough, and c small enough so that the lower bound on $P_i(S_i^c)$ is arbitrarily close to $1/2 > \alpha_0 \geq \alpha$, which contradicts the PAC assumption that $P_i(S_i^c) \leq \alpha$. Specifically, one can set the constants as follows.

1. Let $\epsilon > 0$ be arbitrary. Choose $a > 0$ large enough and $b > 0$ small enough so that

$$\frac{1}{a} + (b\alpha)^{\frac{1}{2a}} < \epsilon.$$

2. Choose $c > 0$ small enough so that

$$(b\alpha)^{2\sqrt{c}+2ac} > 1 - \epsilon.$$

3. Hence we get a lower bound on $P_i(S_i^c)$,

$$P_i(S_i^c) > (1/2 - \epsilon)(1 - \epsilon).$$

4. Since ϵ was arbitrary, we can make the lower bound on $P_i(S_i^c)$ arbitrarily close to $1/2 > \alpha_0 \geq \alpha$. For example, when $\alpha = 0.05$, one can let $\epsilon = 0.4$ and set $a = 5$, $b = 10^{-6}$ and $c = 10^{-4}$ to satisfy the above inequalities.

Therefore, $\mathbb{P}_i(S_i^c) > \alpha$, which contradicts the PAC assumption. Hence we must have

$$\mathbb{E}_0[T_i] > \frac{c\sigma_i^2}{\max(\Delta_i, \delta)^2} \ln \frac{1}{b\alpha}.$$

This is true for each i such that $\mathbb{P}_0(S_i^c) \geq 1/2$. Summing over these systems and noting that our indexing of systems yields the $k-1$ lowest values of the summands concludes the proof. □

Comparing the lower bound in (3.14) and the result in Theorem 3, we see that the upper bound of EP differs from the lower bound by a factor that is

$$\mathcal{O}\left(\ln k + \ln \ln \frac{\sigma}{\delta}\right).$$

Hence, EP is optimal up to a logarithmic factor.

3.5 Numerical Results

In this section, we summarize the results of numerical experiments intended to demonstrate the performance of the EP with the multi-samples strategy on standard test problems, and to compare it to two leading R&S procedures: the KN procedure [36] and the BIZ procedure [18]. We consider both known-variances and unknown-variances cases. The experiments are conducted in sequential computing environment with the Gap-Minimization sampling rule that is discussed in §4.2.1.

The KN procedure is a leading frequentist R&S procedure. Some KN-based procedures, e.g., [24], have better performance in some settings with heterogeneous variance, but the KN procedure is still regarded as a yardstick for selection procedures [37, 6, 18]. Since the original KN procedure deals with unknown-variances cases, in the known-variances case we modify it so that it exploits known variances in the same way as in [18]. The BIZ procedure is a Bayes-inspired procedure, but it delivers a pre-specified frequentist PCS and has a tight lower bound on worst-case PCS under the IZ setting for the known-variances case. The BIZ procedure is more efficient than KN on a variety of problems and is regarded as a state-of-the-art procedure. However, for the unknown-variances case it is a heuristic procedure that does not offer a statistical guarantee. For both KN and BIZ, only PCS guarantees, and not PAC guarantees, have been proved.

In the experiments we consider 3 classes of test problem configurations.

1. Slippage Configuration (SC), where $\mu_1 = \mu_2 = \ldots = \mu_{k-1} = 0$ and $\mu_k = \delta$. This is typically the most difficult configuration.
2. Monotone Increasing Means (MIM), where $\mu_i = (i-1)\delta$. In this configuration, it is easier than in SC to correctly select System k.
3. Random problem instances (RPI-C), where $\mu_i \sim \mathcal{N}(0, C^2\delta^2)$, and are i.i.d.. For example, in RPI-2, $\mu_i \sim \mathcal{N}(0, 4\delta^2)$. The constant C here determines how spread-out the means of the systems are. We regard this class of configurations as representative of practical problems, at least compared to the SC, which is synthetic and usually used for worst-case analysis.

The variances for all configurations are Chi-squared distributed: $\sigma_i^2 \sim \chi^2(4)$, and are i.i.d. We set $\delta = 0.1$ and $\alpha = 0.05$ for all configurations. The samples X_{ij} are normally distributed with mean μ_i and variance σ_i^2 and are independent, as assumed in our problem formulation. For the size of the problems, we choose $k = 100, 1000, 10000$ for small, medium and large-scale problems, respectively.

3.5.1 Statistical Validity and Efficiency

To check the validity and efficiency of the EP and compare with the benchmark procedures, we run these procedures on a set of standard test problem configurations. For each configuration, we run 1000 independent experiments to get the estimated PAC and average sample sizes. The results for both known-variances and unknown-variances cases are shown in Tables 3.5 and 3.6, respectively.

Table 3.5: Estimated PAC and average sample size of the procedures under different problem configurations for known-variances case.

Config.	k	Avg. Size Unit	KN_known		BIZ_known		KEP	
			Est. PAC	Avg. Size	Est. PAC	Avg. Size	Est. PAC	Avg. Size
SC	100	10^5	0.981	3.18 ± 0.04	0.961	2.18 ± 0.03	1.000	3.40 ± 0.05
	1000	10^6	0.976	4.08 ± 0.05	0.955	2.41 ± 0.02	1.000	3.34 ± 0.04
	10000	10^7	0.985	5.02 ± 0.05	0.959	2.55 ± 0.01	1.000	3.85 ± 0.08
MIM	100	10^4	1.000	2.39 ± 0.04	0.998	1.52 ± 0.03	1.000	1.75 ± 0.05
	1000	10^4	1.000	4.27 ± 0.05	0.987	2.41 ± 0.03	1.000	2.39 ± 0.06
	10000	10^4	1.000	6.94 ± 0.07	0.976	3.84 ± 0.03	1.000	3.79 ± 0.08
RPI-2	100	10^4	1.000	9.98 ± 0.15	1.000	7.04 ± 0.13	1.000	7.38 ± 0.21
	1000	10^5	1.000	8.53 ± 0.08	0.999	5.49 ± 0.04	1.000	3.42 ± 0.06
	10000	10^6	1.000	8.17 ± 0.05	0.999	5.14 ± 0.02	1.000	2.36 ± 0.03
RPI-5	100	10^4	1.000	4.84 ± 0.09	0.998	3.19 ± 0.07	1.000	2.68 ± 0.13
	1000	10^5	1.000	3.74 ± 0.04	1.000	2.36 ± 0.02	1.000	0.86 ± 0.02
	10000	10^6	1.000	3.53 ± 0.02	1.000	2.19 ± 0.003	1.000	0.51 ± 0.006
RPI-10	100	10^4	1.000	2.67 ± 0.09	1.000	1.69 ± 0.07	1.000	1.34 ± 0.02
	1000	10^5	1.000	1.98 ± 0.03	1.000	1.22 ± 0.02	1.000	0.37 ± 0.02
	10000	10^6	1.000	1.81 ± 0.02	1.000	1.13 ± 0.006	1.000	0.20 ± 0.004

Table 3.6: Estimated PAC and average sample size of the procedures under different problem configurations for unknown-variances case, $n_0 = 50$. The estimated PAC of UEP and UEPu are all 1.000 and hence omitted.

Config.	k	Avg. Size Unit	KN_unknown		BIZ_unknown		UEP	UEPu
			Est. PAC	Avg. Size	Est. PAC	Avg. Size	Avg. Size	Avg. Size
SC	100	10^5	0.984	3.72 ± 0.03	0.944	2.14 ± 0.03	3.91 ± 0.04	3.54 ± 0.04
	1000	10^6	0.977	4.76 ± 0.02	0.958	2.34 ± 0.02	3.88 ± 0.04	3.34 ± 0.03
	10000	10^7	0.986	5.56 ± 0.01	0.959	2.48 ± 0.01	4.64 ± 0.08	3.95 ± 0.08
MIM	100	10^4	1.000	3.08 ± 0.03	0.993	1.43 ± 0.01	2.35 ± 0.03	2.21 ± 0.03
	1000	10^4	1.000	8.81 ± 0.03	0.972	6.21 ± 0.02	7.63 ± 0.04	7.26 ± 0.03
	10000	10^5	1.000	5.48 ± 0.004	0.965	5.14 ± 0.002	5.37 ± 0.005	5.28 ± 0.004
RPI-2	100	10^4	1.000	12.9 ± 0.15	1.000	7.06 ± 0.12	8.34 ± 0.20	7.35 ± 0.19
	1000	10^5	1.000	11.9 ± 0.07	0.999	5.39 ± 0.04	4.08 ± 0.07	3.43 ± 0.06
	10000	10^6	1.000	11.05 ± 0.03	0.998	5.02 ± 0.02	3.11 ± 0.03	2.39 ± 0.02
RPI-5	100	10^4	1.000	6.32 ± 0.09	0.998	3.24 ± 0.06	3.17 ± 0.12	2.93 ± 0.11
	1000	10^5	1.000	5.32 ± 0.04	1.000	2.36 ± 0.02	1.19 ± 0.03	1.07 ± 0.02
	10000	10^6	1.000	5.16 ± 0.007	1.000	2.13 ± 0.006	0.76 ± 0.007	0.67 ± 0.004
RPI-10	100	10^4	1.000	3.34 ± 0.06	1.000	1.58 ± 0.04	1.59 ± 0.08	1.54 ± 0.08
	1000	10^5	1.000	2.80 ± 0.02	1.000	1.18 ± 0.01	0.76 ± 0.02	0.70 ± 0.01
	10000	10^6	1.000	2.65 ± 0.003	1.000	1.09 ± 0.004	0.55 ± 0.002	0.53 ± 0.002

BIZ does not provide a statistical guarantee for heterogeneous or unknown-variances cases. Therefore, though efficient, BIZ is regarded as a heuristic procedure in our experiments, while KN has provable PCS.

We first discuss the known-variances results. All procedures pass the practical validity check since their estimated PACs are all above 0.95 for all problem configurations. For the SC, BIZ has the smallest average sample size and lowest estimated PAC. This is not surprising, since BIZ has nearly tight PCS (when the variances are common) in this configuration. Comparing to KN, KEP is about 20% more efficient than KN in medium and large-scale problems.

For the MIM configurations, the estimated PAC of KN and KEP are 100% in all scales, while BIZ makes mistakes sometimes but still retains a high confidence. Regarding efficiency, KEP outperforms KN in all scales, especially for medium and large-scale problems. The performances of KEP and BIZ are comparable in all scales. The sample size does not increase much as the problem scale increases. The reason is that the systems added in medium and large scale problems are far away from the best system, so it is easy to distinguish them from the best one.

We have three kinds of RPIs with increasing deviations of the system means: RPI-2, RPI-5 and RPI-10. KN and KEP do not make mistakes in these configurations and the estimated PAC of BIZ is also very close to 100%. Regarding efficiency, KEP outperforms both KN and BIZ in most cases, except for the small-scale problem in RPI-2, where KEP takes a few more samples than does BIZ. KEP shows the most significant advantage in the large scale problem in RPI-10, where it only

uses 11% of the samples used by KN and 18% of the samples used by BIZ. Overall, KEP saves between 20% and 80% of the samples of KN and BIZ in most cases.

The results for the unknown-variances case are similar to those for the known-variances case. The performances of UEP and UEPu are close but UEPu is always slightly better than the UEP. Here we set $n_0 = 50$ since if n_0 is too small then η will be too large and the efficiency of the procedure will decrease. In practice, however, we can use the empirical EP as discussed in the later section, where the impact of n_0 on η is weaker. For the SC, BIZ has the best performance in all scales, and the EP variants are usually more efficient than KN. For the MIM, the order of efficiency of the procedures is still BIZ, UEPu, UEP and then KN, but the differences between them are not as large as in the SC. For the RPI configurations, UEP and UEPu outperform BIZ and KN in most cases, but the improvement is not as large as in the known-variances case. The reason is that when the variances are unknown we have an initial stage to take n_0 samples from each system for variance estimation, and that offsets the EPs' advantage in requiring fewer samples from the inferior systems.

To summarize, EP is less efficient than BIZ in the SC, but is more efficient in more practical problems. In general, we can see a tendency that KEP performs better when the scale of the problem is larger and when the system means come from the random instances.

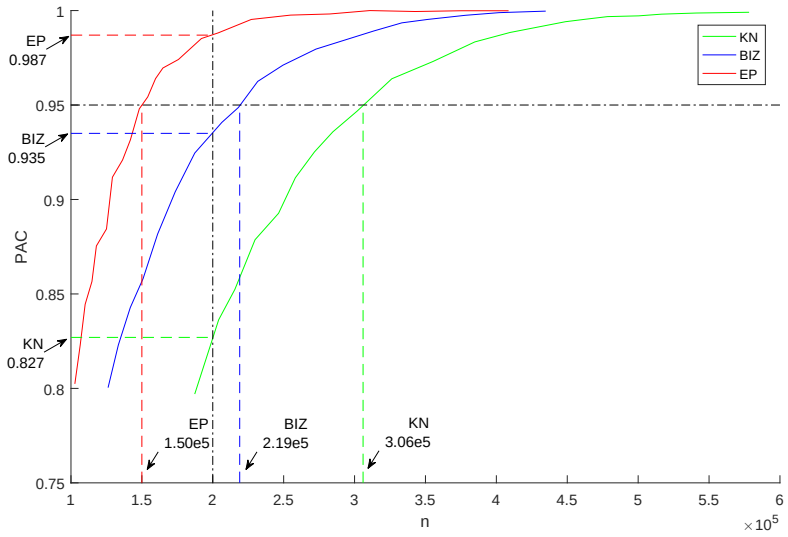

Figure 3.2: Empirical PAC vs. total sample size n for different procedures in SC. Variances are known.

3.5.2 Empirical PAC and Practical Efficiency

As seen in the previous section, EP never errs in selecting an approximately correct system. These results indicate that EP is reliable, but this over-delivery of the PAC guarantee suggests that EP takes more samples than are needed. In other words, the gap between the practical PAC and the lower bound of the theoretical PAC is large and there is room for further improvement.

In the proof of the theoretical results, the target PAC, $1 - \alpha$, that we provide as the input parameter is actually a lower bound on the true PAC, and such a lower bound can be loose as a result of the inequalities employed in the proofs. When we compare the accuracy of the procedures, we may prefer to compare

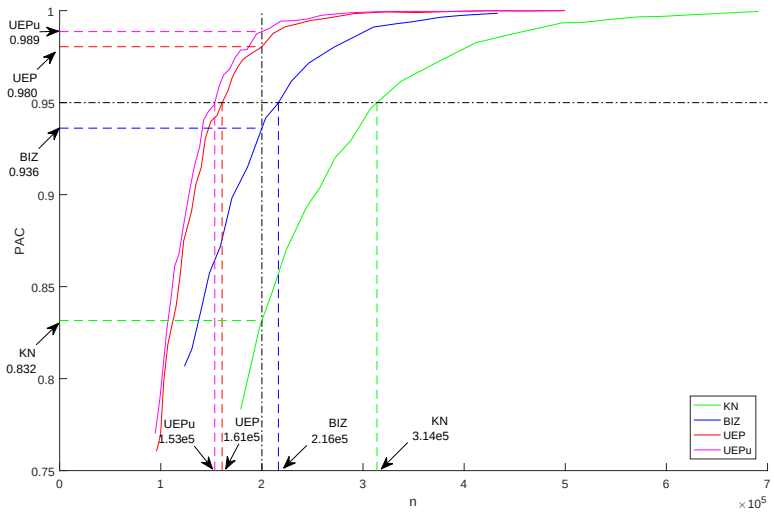

Figure 3.3: Empirical PAC vs. total sample size n for different procedures in SC. Variances are unknown.

their true PAC instead of the lower bounds, especially in situations where the computational budget is limited. To that end, we conduct some experiments to compare the efficiency of EP with other procedures based on their true PAC.

To vary the empirical PAC of the procedures, we change the values of some parameters in the procedures, e.g., η in KN as defined in [36] and η as defined here, which is essentially changing the value of the input parameter α. These two η's have different meanings. In order to differentiate them, we call them η_{KN} and η, respectively. For each value of the parameter, we repeat the experiment and report the empirical PAC and average sample size.

Intuitively and empirically, the SC is the most difficult configuration, so if a

procedure satisfies the statistical guarantee in this case, it should also be valid for other configurations, and this result with stronger assumptions is partially proved in [18]. Therefore, we consider the SC to illustrate the practical efficiency of the procedures. We consider a small-scale problem in the SC where $k = 100$ and $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2 = 4$. To achieve the desired range of PAC, for KN we set $\eta_{\text{KN}} \in [4.12, 10.81]$, for BIZ we set $1-\alpha \in [0.8, 0.999]$, for KEP we set $\eta \in [2.22, 4.6]$, and for UEP and UEPu we set $\eta \in [2.1, 5]$. The empirical PAC and corresponding average sample size is obtained using 10,000 independent replications of each experiment.

The results for known-variances are shown in Figure 3.2, which originally appeared in [43]. From the figure we can see that EP is the most efficient procedure, BIZ is in second place, and KN is in third place. To achieve PAC = 0.95, on average KEP uses 1.50×10^5 samples, while BIZ and KN use 2.19×10^5 and 3.06×10^5 samples, which are 46% and 104% more than KEP, respectively. On the other hand, if the total number of samples is restricted to 2×10^5, EP achieves 98.7% accuracy, while BIZ and KN only get 93.5% and 82.7%, respectively. For the unknown-variances case, the results are similar, as shown in Figure 3.3. UEPu is slightly more efficient than UEP, and they both outperform BIZ and KN. To achieve PAC = 0.95, on average UEPu, UEP, BIZ and KN take 1.53×10^5, 1.61×10^5, 2.16×10^5 and 3.14×10^5 samples respectively, and using a sample size of 2×10^5, the empirical PAC of these 4 procedures are $98.9\%, 98.0\%, 93.6\%$ and 83.2%, respectively. In summary, the EPs show a strong advantage over BIZ and KN in efficiency in this set of experiments.

Based on these result, it seems we can set the parameter η to the value that

achieves $1 - \alpha$ PAC in SC, so that a practical PAC is still secured (albeit without theoretical guarantee), while the efficiency can be improved. Hence, we get a very efficient heuristic R&S procedure. We give the efficiency of the heuristic procedure in Table 3.7, cf. Table 3.5, 3.6. From the table we can see the clear improvement. For the known-variances case, the heuristic EP only takes 30% to 50% of the samples of the original one, and hence is also much more efficient than KN and BIZ. For the unknown-variances case, the improvement is similar, except for the medium and large scale case in MIM and some spread-out RPI configurations, where the samples taken in the initial stage play a key role and cannot be improved.

In summary, using the same number of samples, the EPs achieve higher empirical PAC than KN and BIZ. Based on this observation, we developed heuristic procedures, and the experimental results show that the efficiency is clearly improved and is more efficient than KN and BIZ in all scenarios.

3.6 Conclusions and Future Work

In this chapter, we present EP, a new fully sequential selection procedure that delivers a PAC guarantee. Unlike elimination-based procedures, EP keeps all of the systems in contention, though the frequency of being sampled is based on their performance, depending on the sampling rule. The EP accommodates a variety of sampling rules, and two specific sampling rules are discussed.

Table 3.7: Estimated PAC and average sample size of the heuristic EPs under different problem configurations for known and unknown-variances cases.

Config.	k	Avg. Size Unit	KEP_heuristic		UEP_heuristic		UEPu_heuristic	
			Est. PAC	Avg. Size	Est. PAC	Avg. Size	Est. PAC	Avg. Size
SC	100	10^5	0.954	1.53 ± 0.03	0.955	1.58 ± 0.02	0.956	1.53 ± 0.02
	1000	10^6	0.956	1.48 ± 0.03	0.962	1.61 ± 0.02	0.957	1.57 ± 0.04
	10000	10^7	0.963	1.58 ± 0.06	0.965	2.21 ± 0.04	0.961	2.15 ± 0.03
MIM	100	10^4	0.984	0.72 ± 0.03	0.991	0.85 ± 0.01	0.992	0.82 ± 0.01
	1000	10^4	0.986	0.88 ± 0.03	0.991	5.68 ± 0.01	0.989	5.68 ± 0.01
	10000	10^4	0.987	1.85 ± 0.03	0.992	50.8 ± 0.01	0.994	50.8 ± 0.01
RPI-2	100	10^4	0.997	2.83 ± 0.08	0.997	2.99 ± 0.03	0.996	2.99 ± 0.04
	1000	10^5	0.997	1.26 ± 0.02	1.000	1.38 ± 0.01	0.999	1.36 ± 0.01
	10000	10^6	0.998	1.01 ± 0.01	0.999	1.02 ± 0.004	0.999	1.00 ± 0.004
RPI-5	100	10^4	0.996	1.02 ± 0.02	1.000	1.34 ± 0.02	0.999	1.33 ± 0.02
	1000	10^5	0.997	0.30 ± 0.01	0.999	0.64 ± 0.003	0.998	0.64 ± 0.003
	10000	10^6	0.996	0.17 ± 0.002	1.000	0.54 ± 0.0006	1.000	0.54 ± 0.0006
RPI-10	100	10^4	0.998	0.48 ± 0.03	1.000	0.91 ± 0.02	0.999	0.90 ± 0.02
	1000	10^5	0.995	0.12 ± 0.004	0.998	0.56 ± 0.002	0.998	0.56 ± 0.002
	10000	10^6	0.997	0.06 ± 0.0009	1.000	0.51 ± 0.0003	1.000	0.51 ± 0.0003

Besides the statistical guarantee and a general lower bound on the sample complexity of PAC procedures, we also provide an analysis of the sample complexity of the EP, relating it to the problem complexity. Experimental results show that EP outperforms the state-of-the-art procedures KN and BIZ in efficiency on a variety of problems, especially for large-scale problems with randomly sampled expectations. If the comparison is done based on empirical PAC, the performance of EP is even more impressive, and, inspired by this observation, we offered a heuristic version of the EP.

Several research directions suggest themselves. First, the empirical PAC of the EP is much larger, from a practical perspective, than the guaranteed value of PAC we derived; reducing this gap would be desirable. Second, it may be worthwhile to examine the sample complexity and empirical performance of EP with different shaped envelopes (the confidence boundary). Third, the upper bound on the sample complexity of EP and the lower bound of any R&S procedure with a PAC guarantee differ by a factor of $\ln k$. We believe this $\ln k$ term is necessary but have not been able to prove it, so perhaps the lower bound can be strengthened. Fourth, the EP performs well in practical large-scale problems, and it is natural to consider applying it in a parallel computing environment.

CHAPTER 4

SAMPLING RULES FOR THE ENVELOPE PROCEDURE IN A PARALLEL COMPUTING ENVIRONMENT

4.1 The Envelope Procedure in a Parallel Computing Environment

In this section we discuss some specific aspects and features of the EP with unknown variances (i.e., UEP in §3.3, but we will use the term EP instead in this chapter since no confusion will be caused) in parallel computing environment, some of which apply to general R&S procedures while others are specifically designed for the EP.

4.1.1 Master and Worker Scheme

We consider the master-worker framework (also known as master-slave framework) in a parallel computing environment, where a single core acts as a ”master” process, which stores all the data information and coordinate the parallel procedure, and all other cores act as ”worker” cores, which execute the instructions from the master. This parallel scheme is also adopted in [41], [48] and [47], though the details in implementation are different.

For the EP, the workers are responsible for the simulation tasks. Their working cycle is simple: A worker sends a message to the master saying it is ready, and receives a simulation job assigned by the master. It then executes the job and send the simulation result back to the master. This cycle is repeated until it receives a stop message from the master. The master has four tasks: it (1) determines what simulations to run next according to the sampling rule based on the information collected so far, (2) sends simulation jobs to an available worker, (3) receives simulation results from the worker and update the statistics, and (4) check the stopping criterion and send stop message to workers when it is met.

To reduce the amount of communication work between master and workers, the simulation jobs assigned to workers are in batches rather than single replications. Since the only information EP uses for screening are sample means and sample variances, the workers only need to report the sums and the sample variances of the simulation results to the master, and the master can maintain these summary statistics instead of storing all of the simulation results.

The batch size should be chosen carefully. If the batch size is too small, the master might be overwhelmed by messages. If the batch size is too large, on the other hand, some samples might be redundant. The ideal batch size depends on the average simulation time of one replication T_{sim}, the number of workers c, and the average screening time T_{scr}. To avoid the master being overwhelmed by messages, the batch size b should not be less than $cT_{\text{scr}}/T_{\text{sim}}$. In practice, one can estimate T_{sim} and T_{scr} by the samples collected in the initial stage to determine b.

4.1.2 Random Completion Times and Vector Filling

As discussed in [48], [41] and [47], the randomness of completion times of simulations in a parallel computing environment can cause nontrivial issue in statistical validity. In the procedure, estimators are constructed from collections of simulation results. In a parallel computing environment, multiple simulation replications can be carried on different cores simultaneously, and the order of the completion of simulation replications is random. If we construct those estimators by collecting simulation results in the order in which they are received, then the distribution of the estimator might be changed due to the possible statistical dependencies between completion time and simulation results.

In our case, for example, we maintain the mean value of system outputs $\bar{X}_i$'s during the main stage (Step 3 - 5 in §3.3) of the procedure, and the statistical validity of the procedure is based on the fact that the samples are i.i.d.. However, if we collect simulation replications in the order of their completions then we lose this property and hence the statistical validity of the procedure. To avoid the loss of statistical guarantee, we use the "vector-filling" method [41] to keep the order of simulation outputs as same as the order in which the simulations are initiated. A similar approache is discussed in [47].

Specifically, we maintain a result list throughout the main stage of EP. When the master assigns a simulation job to a worker, the index of the job is also sent to the worker. When the worker completes the job and sends the simulation result

along with the index back to the master, the master stores the result in the position of the result list corresponding to the index. A result can be used by the master (we call it *absorbed* by the master) for making decisions only when it and all of its precedents are received. We only need to maintain the cumulative sum of the *absorbed* results. In this way, simulation results are used for screening in the exact order in which they were assigned. The statistical guarantee of the EP is based on the fact that the sample means of the systems always stay within their envelopes throughout the whole procedure, whenever observed, so the randomness in the number of absorbed result by a fixed time does not hurt its validity. Therefore, the statistical validity of the parallel procedure is guaranteed as in the serialized procedure.

An advantage of the EP is that it does not require synchronization between different systems. The statistical validity of the EP is based on the fact that the sample means always stay within the envelopes around their true means throughout the whole procedure with high probability, so the sample sizes of the systems can be arbitrary (as long as they do not exceed N_i's) when the decisions (choice of systems from which to take samples and whether to terminate or not) are made. In other words, we can make decisions at any time and there is no need to wait for any simulation job to be completed, provided vector-filling is maintained.

4.1.3 Sequential Assignment and Information Update

In a parallel computing environment, cores send messages of availability to the master one by one, along with new information. Therefore, the sampling rule does not need to determine the numbers of replications for all systems or all cores in one round. Instead, we make the sampling decisions sequentially. In this way, we can make sampling decisions based on the most recent information to try to avoid redundant samples.

Usually the statistics get updated only when the master receives simulation results from the workers. However, the action of sending simulation jobs to workers also contains information. For example, consider top-two sampling strategy of the EP. When the master assigns a simulation job of System j^* to a worker, the sample size n_{j^*} will increase by a batch size. Therefore the identity of j^* may change, since the upper confidence bound of this system will change. This suggests that, if the current j^* will not be the one with the highest confidence upper bound, then the it should not be picked when the next available core arrives, even though the simulation result for it has not been received yet.

For this reason, we redefine the upper and lower confidence bound as

$$L_i = \bar{X}_i - \frac{\eta S_i}{\sqrt{n_i + m_i}},$$
$$U_i = \bar{X}_i + \frac{\eta S_i}{\sqrt{n_i + m_i}},$$

where n_i is the number of samples that have been received, and m_i is the number of samples that have been sent to workers but have not yet been absorbed. When

a job of simulating system i is sent to worker, we should update m_i and j^* accordingly, while when the result is received by the master and is *absorbed* (see §4.1.2), we should update m_i, n_i, $\bar{X}_i$, i^* and j^* accordingly.

System i^* can have the same issue. When there are many workers, it is possible that the master assigns many jobs of simulating i^* to workers before any of the jobs is completed, since the sample means and the identity of i^* do not change when sending jobs. There is a chance that the current i^* could be a system with low mean after the first few samples are completed and all of the jobs after those early samples are in vain. This scenario can happen especially in the early stage of the procedure, where the uncertainty on the means of the systems is still very high. To resolve this problem, we propose a Baysian approach to estimate the output of the systems that are being simulated.

We assume that the unknown mean of System i has the non-informative conjugate normal prior distribution. As shown in [14], the posterior distribution of μ_i is

$$\tilde{\mu}_i \sim \mathcal{N}\left(\bar{X}_i, \frac{\sigma_i^2}{n_i}\right).$$

In practice we approximate σ_i^2 by sample variance S_i^2. Therefore, the posterior mean of m_i replications of System i follows normal distribution with mean $\bar{X}_i$ and variance $\sigma_i^2/n_i+\sigma_i^2/m_i$. When the master assign simulation jobs to workers, we randomly generate outputs according to this posterior distribution before the real results are sent back. In this way, the action of sending jobs also creates (random) information as to the identity of i^*, and hence the problem of over-sampling can

be relieved. When the real results are received, the randomly generated results are replaced by the real ones.

4.1.4 Inferior System Elimination

In a parallel computing environment, the speed of the screening or making sampling decision is much more important than in a sequential environment since the master makes such decisions more frequently as discussed in §4.1.3, and it affects the utilization of the workers. For the EP, such decisions are usually based on the identity of i^* and j^* among all systems. For practical problems, there are usually some systems whose means are much lower than the best. Including these inferior systems as candidates for i^* and j^* is unnecessary and slows the process of making sampling decisions.

Therefore, we present a method to eliminate those systems from the candidates for i^* and j^* without harming the statistical validity of EP. Remember that the PAC guarantee of EP is based on the fact that, with high probability, the true means are within the confidence band throughout the procedure. Conditioning on that desired event, we have

$$\mu_k - \delta \geq \max_i \bar{X}_i - \frac{\eta\sigma_i}{\sqrt{n_i}} - \delta = \textit{thresh}$$

and

$$\mu_i \leq \bar{X}_i + \frac{\eta\sigma_i}{\sqrt{n_i}}, \forall i.$$

If for System i, $\bar{X}_i + \frac{\eta\sigma_i}{\sqrt{n_i}} < thresh$, then

$$\mu_i < \mu_k - \delta,$$

and hence System i is not a δ-optimal system, and it can be eliminated.

This elimination scheme usually does not affect the sampling decisions and hence the total number of samples of the procedure, since the eliminated systems are those that should not be sampled anyway. The purpose of introducing elimination to the EP is to make the selection of i^* and j^* more efficiently. Therefore, we recommend doing the elimination only once after the initial stage, or only a few times during the main stage, because the elimination itself also takes time.

4.1.5 The Parallel EP

We summarize the parallel EP here. The master starts with Algorithm 1 and the workers run Algorithm 3. And then the master runs Algorithm 2 and the workers run Algorithm 3 again.

4.2 Sampling Rules

We introduced the top-two strategy in §3.2.2, which is designed for sample complexity analysis. Now we introduce two more sampling rules for practical use: the Gap-Minimization strategy and the No-Waste strategy. The former one is

Algorithm 1: Parallel EP: Master Initialization

```
1:  while Not every worker is closed do
2:      Receive a message from an available worker w.
3:      if Tag == READY then
4:          if Not every system is assigned then
5:              Send a message with Tag=SIM of simulating next system to w.
6:          else
7:              Send message with Tag=STOP to w.
8:      else if Tag == DONE then
9:          Update ∑X and ∑X^2 of this system according to the result.
10:         if Not every system is assigned then
11:             Send a message with Tag=SIM of simulating next system to w.
12:         else
13:             Send message with Tag=STOP to w.
14:     else if Tag == CLOSED then
15:         Mark w as closed.
16: Calculate the statistics.
```

Algorithm 2: Parallel EP: Master Main Stage

while Not every worker is closed **do**
    Check the stopping rule only based on the *absorbed* results.
    Receive a message from an available worker w.
    if *Tag* == **READY then**
        if The stopping criterion is not met **then**
            Send a message with *Tag*=**SIM** of simulating a system according to the sampling rule to w.
            Update m of the system and j^*.
        else
            Send message with *Tag*=**STOP** to w.
    else if *Tag* == **DONE then**
        if The stopping criterion is not met **then**
            Send a message with *Tag*=**SIM** of simulating next system to w.
            Update m (the number of **assigned** but not **completed** jobs) of the system and j^*.
        else
            Send message with *Tag*=**STOP** to w.
        Save the result to the corresponding position of the result list.
        while The result in the head of the result list has been received **do**
            Absorb the result in the head by updating $\sum X$, n and m of the corresponding system according to the result.
            Move the head of the result list forward.
        Update i^* and j^*.
    else if *Tag* == **CLOSED then**
        Mark w as closed.
Return the selected system and other statistics.

Algorithm 3: Parallel EP: Worker

1: Send message with *Tag*=**READY** to the master.
2: **while** True **do**
3: Receive a message from the master.
4: **if** *Tag* == **SIM then**
5: Do the simulation job according to the message.
6: Send a message with *Tag*=**DONE** of returning the result to the master.
7: **else if** *Tag* == **STOP then** Break.
8: Send message with *Tag*=**CLOSED** to the master.

designed for a serial computing environment, while the latter is designed for a parallel computing environment.

4.2.1 Gap Minimization

Here we consider a sampling strategy that allocates multiple samples in a greedy fashion. Denoting the number of samples we can draw in each round by M, in each round, we try to shrink the absolute value of the (negative) gap between the the right-hand side and left-hand side of (3.3) by allocating those samples. We ignore the change in the sample means since the change is unknown at the point of making allocation decisions, and to ensure tractability. In other words, the $\bar{X}_i$'s are constants and the only variables here are the number of samples m_i. Formally,

in each round t we want to solve the optimization problem

$$\begin{aligned}
&\underset{m_1,m_2,\cdots,m_k}{\operatorname{argmin}} \left\{ \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}+m_{i^*}}} + \max_{j\neq i^*} \left\{ \bar{X}_j(n_j) + \frac{\eta\sigma_j}{\sqrt{n_j+m_j}} \right\} \right\}, \\
&\quad \text{s.t. } \sum_i m_i = M, \\
&\quad\quad m_i \geq 0,
\end{aligned}$$

where the constants ($\bar{X}_{i^*}(n_{i^*})$ and δ) in (3.3) are dropped and we omit the iterator t for notational convenience. The integrality of the number of samples is also relaxed for simplicity.

We start with a simple case where we only consider systems i^* and j^* as in the 2-sample strategy. In this case, the optimization problem becomes

$$\begin{aligned}
&\underset{m_{i^*},m_{j^*}}{\operatorname{argmin}} \left\{ \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}+m_{i^*}}} + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*}+m_{j^*}}} \right\}, \\
&\quad \text{s.t. } m_{i^*} + m_{j^*} = M, \\
&\quad\quad m_{i^*} \geq 0, m_{j^*} \geq 0.
\end{aligned}$$

Plugging $m_{j^*} = M - m_{i^*}$ into the objective function and setting the derivative with respect to m_{i^*} to zero, we get

$$-\frac{\eta\sigma_{i^*}}{2\left(n_{i^*}+m_{i^*}\right)^{\frac{3}{2}}} + \frac{\eta\sigma_{j^*}}{2\left(n_{j^*}+M-m_{i^*}\right)^{\frac{3}{2}}} = 0,$$

which is equivalent to

$$\frac{n_{j^*}+M-m_{i^*}}{n_{i^*}+m_{i^*}} = \left(\frac{\sigma_{j^*}}{\sigma_{i^*}}\right)^{\frac{2}{3}}.$$

Solving for m_{i^*} yields

$$m_{i^*} = \frac{n_{i^*} + n_{j^*} + M}{\left(\frac{\sigma_{j^*}}{\sigma_{i^*}}\right)^{\frac{2}{3}} + 1} - n_{i^*}.$$

The feasible range of m_{i^*} is $[0, M]$. From the derivative of the objective function, it is not hard to see that the objective function is convex, so the optimal solution is

$$m_{i^*} = \min\left(\max\left(\frac{n_{i^*} + n_{j^*} + M}{\left(\frac{\sigma_{j^*}}{\sigma_{i^*}}\right)^{\frac{2}{3}} + 1} - n_{i^*}, 0\right), M\right), \tag{4.1}$$

and correspondingly

$$m_{j^*} = M - m_{i^*}. \tag{4.2}$$

A complication is that the identity of j^* may change due to the change of the width of the confidence interval. If there exists a system j that is not i^* or j^*, s.t.

$$\bar{X}_j(n_j) + \frac{\eta\sigma_j}{\sqrt{n_j}} > \bar{X}_{j^*}(n_{j^*}) + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*} + m_{j^*}}}, \tag{4.3}$$

where m_{j^*} is given by (4.2), then only considering Systems i^* and j^* is not equivalent to the original optimization problem. When more systems need to be considered, the closed-form solution of the optimization problem is not tractable anymore. One can exactly solve the problem by iterating over the number of samples to draw from i^*, with the rest of the samples being carefully allocated to other systems.

The key idea is to iterate over the number of samples to draw from i^*. Before we start the iteration, we sort all systems but i^* by their upper confidence bounds, say $j_1, j_2, \ldots, j_{k-1}$, i.e.,

$$U_{j_1}(n_{j_1}) \geq U_{j_2}(n_{j_2}) \geq \ldots \geq U_{j_{k-1}}(n_{j_{k-1}}).$$

Initially we assign all samples to i^*. In the next iteration, we move a batch of samples from i^* to j_1, calculate the resulting value of the objective function and save it. Next, we move a batch of samples from i^* to the current system with the highest upper confidence bound, compute the objective values and save. We keep iterating like that until all samples are taken from i^*. In this process, we maintain the minimum value of the objective function and the corresponding allocation $(m_i, i = 1, \ldots, k)$. Finally we take the allocation that minimizes the outer objective function as our solution; see Algorithm 4, where we drop the iterator t for notational convenience.

4.2.2 No-Waste Sampling

In a parallel computing environment, as discussed in §4.1.3, the master makes sampling decisions sequentially instead of determining the sample sizes of all systems in one round. The question we need to answer is that, which system should we sample so that the new batch of replications will not be a waste. The first step to answer this question is find out the number of samples needed for each system so that the total number of samples is minimized. We still ignore the change in the sample means to ensure tractability. By allocating samples to different systems, we can decrease the upper confidence bound of i^* and increase the lower confidence bound of the others, so that (3.3) is satisfied.

Algorithm 4: Gap-Minimization strategy

Compute m_{i^*} and m_{j^*} according to (4.1) and (4.2).
if (4.3) does not hold **then**
    $m_j \leftarrow 0$ for all $j \in \mathcal{S} \setminus \{i^*, j^*\}$.
    return
Sort systems $\mathcal{S} \setminus i^*$ by $U_j(n_j)$ and denote the sorted indices by $j_1, j_2, \ldots, j_{k-1}$.
Initialization: Let $U_i \leftarrow \bar{X}_i + \eta\sigma_i/\sqrt{n_i}$ for each i, $i^* \leftarrow \operatorname{argmax}_i \bar{X}_i$, $j^* \leftarrow \operatorname{argmax}_{j \neq i^*} U_j$.
Let $m_j \leftarrow 0$ for $j \neq i^*$, $m_{i^*} \leftarrow B$, $b \leftarrow 1$.
Let $minV \leftarrow \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}+B}} + U_{j^*}$, $minM_i = m_i$ for each i.
while $m_{i^*} > 0$ **do**
    Let $m_{i^*} \leftarrow m_{i^*} - b$, $m_{j^*} \leftarrow m_{j^*} + b$, $U_{j^*} \leftarrow \bar{X}_{j^*} + \eta\sigma_{j^*}/\sqrt{n_{j^*}+m_{j^*}}$.
    Let $j^* \leftarrow \operatorname{argmax}_{j \neq i^*} U_j$, $V \leftarrow \eta\sigma_{i^*}/\sqrt{n_{i^*}+m_{i^*}} + U_{j^*}$.
    if $V < minV$ **then**
        Let $minV \leftarrow V$, $minM_i = m_i$ for each i.
Let $m_i \leftarrow minM_i$ for each i.

Formally, we consider the following optimization problem:

$$\begin{aligned} \min_{m_1,\ldots,m_k} & \sum_{i=1}^{k} m_i \\ \text{s.t.} & \bar{X}_{i^*} - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}+m_{i^*}}} \geq \max_{j \neq i^*} \bar{X}_j + \frac{\eta\sigma_j}{\sqrt{n_j+m_j}} - \delta \\ & m_i \geq 0, i = 1, 2, \ldots, k. \end{aligned}$$

Let $\ell = \bar{X}_{i^*} - \eta\sigma_{i^*}/\sqrt{n_{i^*} + m_{i^*}}$, then

$$n_{i^*} + m_{i^*} = \left(\frac{\eta\sigma_{i^*}}{\bar{X}_{i^*} - \ell}\right)^2, \text{ and}$$
$$n_j + m_j \geq \left(\frac{\eta\sigma_j}{\ell + \delta - \bar{X}_j}\right)^2, \forall j \neq i^*.$$

It is implied that

$$\bar{X}_{i^*} - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}}} \leq \ell < \bar{X}_{i^*}, \text{ and}$$
$$\ell > \bar{X}_j - \delta, \forall j \neq i^*.$$

Therefore, the original problem is equivalent to the following problem,

$$\min_{\ell} f(\ell) = \left(\frac{\eta\sigma_{i^*}}{\bar{X}_{i^*} - \ell}\right)^2 + \sum_{j \neq i^*} \max\left(\left(\frac{\eta\sigma_j}{\ell + \delta - \bar{X}_j}\right)^2, n_j\right)$$
$$\text{s.t.} \max_{j \neq i^*}\left(\bar{X}_j - \delta\right) < \ell < \bar{X}_{i^*}, \text{ and}$$
$$l \geq \bar{X}_{i^*} - \frac{\eta\sigma_{i^*}}{\sqrt{n_{i^*}}}.$$

which is a convex optimization problem that can be solved efficiently. Once the solution is achieved, one can calculate m_i's to determine the number of replications to simulate for each system.

As in Top-two strategy, the system i^* and j^* are the key systems in stopping rule. However, if we have already taken enough samples from a system, we do not need to take more sample from it. Therefore the question is, whether the number of sample from i^* or j^* is enough.

We can find the answer from solving the previous optimization problem. We should take samples from System i^* or j^* if and only if $m_{i^*} > 0$ or $m_{j^*} > 0$ in the optimal solution of the problem.

System i^*: Since $m_{i^*} = 0$ corresponds to $\ell_1 = \bar{X}_{i^*} - \eta\sigma_{i^*}/\sqrt{n_{i^*}}$, it suffices to check (1) if ℓ_1 is not a feasible solution, i.e.,

$$\ell_1 \leq \max_{j \neq i} \bar{X}_j - \delta, \text{ and}$$

(2) if ℓ_1 is not the optimal solution, i.e.,

$$f'(\ell_1) = \frac{(\eta\sigma_{i^*})^2}{\left(\bar{X}_{i^*} - \ell_1\right)^3} - \sum_{j \in J_{\ell_1}} \frac{(\eta\sigma_j)^2}{\left(\ell_1 + \delta - \bar{X}_j\right)^3} < 0,$$

where

$$J_\ell = \left\{ j : \left(\frac{\eta\sigma_j}{\ell + \delta - \bar{X}_j} \right)^2 > n_j \right\}.$$

If (1) and/or (2) is satisfied, then $m_{i^*} > 0$.

System j^*: Since $m_{j^*} = 0$ is equivalent to $m_j = 0$, $\forall j \neq i$, which corresponds to $\ell_2 = \bar{X}_{j^*} + \frac{\eta\sigma_{j^*}}{\sqrt{n_{j^*}}} - \delta$, it suffices to check (1) if ℓ_2 is not a feasible solution, i.e.,

$$\ell_2 \geq \bar{X}_{i^*}, \text{ and}$$

(2) if ℓ_2 is not the optimal solution, i.e.,

$$f'(\ell_2) = \frac{(\eta\sigma_{i^*})^2}{\left(\bar{X}_{i^*} - \ell_2\right)^3} - \frac{(\eta\sigma_{j^*})^2}{\left(\ell_2 + \delta - \bar{X}_{j^*}\right)^3} > 0.$$

If (1) and/or (2) is satisfied, then $m_{j^*} > 0$.

Remark: If the sample size of System i^* has reached the limit N_{i^*}, then j^* should always be sampled, and vice versa.

4.3 Computational Study

In this section, we demonstrate the performance of the EP with the no-waste sampling rule in a parallel computing environment, and compare it with the vector-filling KN (VKN) procedure [41]. Since the variances are unknown in our experiments, we use UEP and replace σ_i by S_i.

The VKN procedure extends the KN procedure [36] to a parallel computing environment with the vector-filling method, and it also adopts the master-worker scheme. It has the same statistical validity as the KN procedure. The original vector-filling method in [41] requires a large amount of memory since it needs to store all of the simulation results in the exact order in which the simulation jobs are sent from the master. To save the memory so that we can test large-scale problem, we improved this strategy by storing the cumulative sum of the *absorbed* results instead of the individuals, as described in §4.1.2.

Both the EP and the VKN are implemented using the Message-Passing Interface (MPI), which is a standardized and portable message-passing system to function on parallel computing architectures. Our algorithms are implemented in python and we use the MPI for Python package (mpi4py). The MPI allows us to fully customize the work-flow and the message-passing scheme between master and workers, and hence provides high flexibility in designing the algorithms.

We test the performance on the "throughput-maximization" problem taken from SimOpt.org [23]. The setup of the problem is discussed in §2.4.2, and the

details can be found in [23]. For our experiments, we set the warm-up period as 2000 jobs and the average throughput is based on the following 50 jobs. We consider two configurations: 1) $R = B = 20$, in which $k = 3249$, and 2) $R = B = 50$, in which $k = 57624$. The parameters are $\alpha = 0.05$, $\delta = 0.1$ throughout the experiments and the batch size is set to be the same as n_0.

The performance of the procedures is demonstrated with three measurements: number of samples, wall-clock time and utilization. The number of samples is a standard measurement for the efficiency of R&S procedures. In parallel computing environment, considering the screening time, synchronization issue and idling time, the wall-clock time is a more important factor. In addition, we define the utilization as

$$\text{utilization} = \frac{\text{total time spent on simulation}}{\text{wall-clock time} \times \text{number of workers}},$$

which measures how efficiently the workers are used on simulation effort.

The experimental results are shown in Table 4.1. From the experimental results we can see that, the EP outperforms the VKN in all configurations in all of the measurement. The EP uses many fewer samples and much less wall-clock time to complete, and the utilization of the workers of the EP is higher than the VKN as well, especially for the large-scale problem. For the EP, there is no tendency that the utilization will change a lot as the number of cores or the number of systems increases. In comparison, the utilization of VKN drops dramatically for the large-scale problem. The reason is that screening involves pair-wise comparison, which is very time-consuming, and many workers are idling waiting for the instructions

Table 4.1: A Comparison of EP and VKN in Parallel Computing Environment.

k	n_0	Number of cores	Procedure	Number of samples	Wall-clock time (sec)	Utilization %
3,249	20	8	EP	1.2×10^5	2.3×10^2	99.7
		8	VKN	1.8×10^5	3.3×10^2	98.4
		16	EP	1.3×10^5	1.2×10^2	99.6
		16	VKN	1.8×10^5	1.8×10^2	97.0
		24	EP	1.2×10^5	0.7×10^2	99.5
		24	VKN	1.8×10^5	1.2×10^2	95.7
57,624	50	16	EP	4.2×10^6	3.8×10^3	99.9
		16	VKN	7.8×10^6	10.2×10^3	68.9
		24	EP	4.1×10^6	$2.5 * 10^3$	99.9
		24	VKN	7.2×10^6	7.5×10^3	65.2

from the master when it is doing the screening job.

4.4 Conclusions

In this chapter, we discussed the EP in a parallel computing environment and two sampling rules. For the parallel EP, we use the master-worker parallel framework. The vector-filling technique is used for resolving the issue brought by random completion times. We also designed information update and system elimination schemes to adapt to the sequential assignment of simulation jobs in a parallel environment. Furthermore, we designed Gap-Maximization sampling rule and No-Waste sampling rule. They both can work in a parallel environment, but the latter is specifically designed for it since it assigns job sequentially, so we recommend it and used it for our experiments. The experimental results show that the parallel EP outperforms the VKN in all of the measurements for the parallel computing environment across all test configurations.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we focused on three topics of the R&S problem. First, we explored the problem of predicting the simulation budget of R&S procedures, which is important when dealing with large-scale problems. We presented two approaches for estimating the total number of samples needed for a R&S procedure to terminate, both of which rely on the estimation of the problem configuration, which is not trivial. We developed a linear combination estimator that exhibits excellent performance in a realistic setting and reasonable performance even in the slippage configuration. Experimental results for both synthetic test problems and a realistic problem together suggested that our approach is effective and sufficient for application.

Second, we presented EP, a new fully sequential selection procedure that delivers a PAC guarantee. Unlike elimination-based procedures, EP keeps all of the systems in contention, though the frequency of being sampled is based on their performance, depending on the sampling rule. The EP accommodates a variety of sampling rules, and two specific sampling rules are discussed. Besides the statistical guarantee and a general lower bound on the sample complexity of PAC procedures, we also provide an analysis of the sample complexity of the EP, relating it to the problem complexity. Experimental results show that EP outperforms the state-of-the-art procedures KN and BIZ in efficiency on a variety of problems, especially for large-scale problems with randomly sampled expectations. If the

comparison is done based on empirical PAC, the performance of EP is even more impressive, and, inspired by this observation, we offered a heuristic version of the EP.

Finally, we developed several features for the EP specifically designed for a parallel computing environment, which improves the efficiency of EP while preserving its statistical validity. In addition, two sampling rules for the EP are presented, one of which is for a serial computing environment, while the other is for a parallel computing environment. Computational experiments are done to compare the performance of the parallel EP with the VKN procedure, and the results show that the EP outperforms the VKN in all of the measurements for the parallel computing environment across all test configurations.

In this thesis, we only discussed the method for predicting the sample complexity of procedures, instead of the more desirable goal of predicting running times. These two goals are closely connected, but the latter one is more difficult to analyze. At the end of the day, the prediction of the running time is the ultimate goal. The factors that affect the running time include synchronization issues, the detail of the parallelization scheme, and the implementation of the procedure. A possible way to obtain a reasonable prediction of the running time is to use our methods to get a prediction on the sample complexity, and then consider all of these issues to predict the running time based on the sample complexity.

For the EP, there is a clear gap between the empirical PAC and the guaranteed value of PAC we derived. By reducing this gap, theoretically or practically, the

efficiency can be further improved. Another gap that inspires a future direction is the one between the upper bound on the sample complexity of the EP and the lower bound of any R&S procedure with a PAC guarantee. Intuitively, the $\ln k$ term in the bound is necessary, but we did not find a way to prove it. Furthermore, the current shape of the envelope might not be the optimal one. Further study on the impact of these factors on the sample complexity is an intriguing topic.

From the numerical results we can see that the current parallel scheme for the EP works well. However, due to the limit of the computational resources, we did not try very large experiments, say, 100 thousand systems or 1 million systems, on 64 cores, 128 cores or hundreds of cores. For this level of configurations, the current scheme may or may not be able to maintain a high utilization. If it does not, the natural question to ask is how can we further improve the parallel scheme so that it scales with the number of systems and number of cores. For example, if the screening becomes the bottleneck, a possible solution is using another core as an assistant for the master to help with calculating the statistics and screening. In addition, besides vector-filling, is there a more efficient way to keep the statistical validity in parallel computing environments? We leave this as a future research direction.

APPENDIX A

APPENDIX FOR CHAPTER 2

A.1 Relaxed Condition in Proposition 2

Proposition 2, below, complements Proposition 1, and relaxes the assumption that the means are unique. In fact, an inspection of the proof of Proposition 1 reveals that the proof holds under the weaker assumption that only μ_i is unique, rather than assuming that *all* the means are unique. (One need only note that the event $B_i(n)$, that $\bar{X}_i(n)$ is the ith smallest of the sample means, holds eventually under this weaker assumption.) Accordingly, here we assume that μ_i is not unique. To that end, we again assume, without loss of generality, that the μ vector is ordered from smallest to largest, and let $l \leq i \leq u$ be such that $u - l > 1$ and $\mu_{l-1} < \mu_l = \mu_{l+1} = \cdots = \mu_i = \cdots = \mu_u < \mu_{u+1}$. We explicitly allow $l = 1$ or $u = k$, in which case the corresponding strict inequality is vacuous. For a vector v, let $f(v)$ return the $(i - l + 1)$th smallest component of v. We need the following assumption.

A5 The joint central limit theorem

$$\sqrt{n}\begin{pmatrix} \bar{X}_l(n) - \mu_i \\ \bar{X}_{l+1}(n) - \mu_i \\ \vdots \\ \bar{X}_u(n) - \mu_i \end{pmatrix} \Rightarrow N(0, \Sigma)$$

holds as $n \to \infty$, for some multivariate normal random vector $N(0, \Sigma)$.

Notice that A5 holds under the independence assumption A4, in which case Σ is the diagonal matrix with diagonal entries equal to the variance constants $\sigma_l^2, \sigma_{l+1}^2, \ldots, \sigma_u^2$, so that the situation we focus upon in the chapter is covered by A5. But the conclusion of Proposition 2 applies more generally than the setting of Assumption A4, e.g., in certain steady-state simulation settings, or in quantile-estimation settings, which is why we adopt A5 as the basic assumption.

Proposition 2. *Fix $i \in \{1, 2, \ldots, k\}$. If A1 holds, then the naive estimator $M_i(n)$ is a strongly consistent estimator of μ_i, for each $i = 1, 2, \ldots, k$. If A1, A2 and A5 hold, then $M_i(n)$ satisfies the central limit theorem*

$$\sqrt{n}(M_i(n) - \mu_i) \Rightarrow f(N(0, \Sigma))$$

as $n \to \infty$. If A1–A3 hold, then the mean squared error of $M_i(n)$ is $\mathcal{O}(n^{-1})$.

PROOF. Define $C_i(n)$ to be the event that the sample means $\bar{X}_l(n), \bar{X}_{l+1}(n), \ldots, \bar{X}_u(n)$ correctly take positions $l, l+1, \ldots, u$ (in any order within these positions) in the ranking of the sample means. On the event $C_i(n)$, it follows that $M_i(n) \in \{\bar{X}_l(n), \bar{X}_{l+1}(n), \ldots, \bar{X}_u(n)\}$. Then, as in the proof of Proposition 1, for some (random) N, $C_i(n)$ happens for all $n \geq N$ a.s. Consistency then follows as before.

For the central limit theorem, we first observe that $\mathbb{1}(C_i(n)) = 1$ eventually a.s., and on the event $C_i(n)$, $M_i(n) = f(\bar{X}_{[l,u]}(n))$, where the vector $\bar{X}_{[l,u](n)} =$

$(\bar{X}_l(n), \bar{X}_{l+1}(n), \ldots, \bar{X}_u(n))$. We then obtain that

$$\begin{aligned}\sqrt{n}(M_i(n) - \mu_i) &= \sqrt{n}(M_i(n) - \mu_i)\mathbb{1}(C_i(n)) + \sqrt{n}(M_i(n) - \mu_i)\mathbb{1}(C_i^c(n)) \\ &= \sqrt{n}[f(\bar{X}_{[l,u]}(n)) - \mu_i]\mathbb{1}(C_i(n)) + \sqrt{n}(M_i(n) - \mu_i)\mathbb{1}(C_i^c(n)) \\ &= f(\sqrt{n}(\bar{X}_{[l,u]}(n) - \mu_i)\mathbb{1}(C_i(n)) + \sqrt{n}(M_i(n) - \mu_i)\mathbb{1}(C_i^c(n)).\end{aligned}$$

The result now follows from Slutsky's theorem, the continuous mapping theorem, and the fact that f is continuous.

To establish the mean-squared error result, we proceed almost exactly as before. First,

$$\sqrt{n}(M_i(n) - \mu_i) = \sqrt{n}(f(\bar{X}_i[l,u](n)) - \mu_i) + \sqrt{n}(M_i(n) - f(\bar{X}_i[l,u](n)))\mathbb{1}(C_i^c(n)),$$

so that

$$n\mathbb{E}((M_i(n) - \mu_i)^2) \leq 2n\mathbb{E}(f(\bar{X}_i[l,u](n)) - \mu_i)^2 + 2n\mathbb{E}\left[(M_i(n) - \bar{X}_i(n))^2\mathbb{1}(C_i^c(n))\right]. \tag{A.1}$$

Since f selects a single term from a vector, the first term in (A.1) is bounded by $2\sum_{q=l}^{u}\sigma_q^2$. The second term is handled almost exactly as before; we omit the details. □

A.2 Supplementary Tables

The supplementary tables for the experimental results in §2.4 are given below.

Table A.1: Comparison of naive and linear combination estimators for simulation approach for KN. The values given are the averages over 100 replications of the ratio of the estimator to true value

KN

	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.26	0.71	0.20	0.72	0.18	0.73
SC-INC	0.37	1.05	0.27	1.03	0.23	1.06
SC-DEC	0.24	0.60	0.19	0.67	0.16	0.71
MDM	0.89	0.94	0.93	0.96	0.96	0.98
MDM-INC	1.01	1.03	0.99	1.03	1.00	1.01
MDM-DEC	0.79	0.82	0.88	0.90	0.91	0.92
RPI1	0.65	1.07	0.65	1.00	0.63	0.99
RPI2	0.94	1.05	0.90	1.00	0.87	0.99
RPI1-HET	0.95	1.07	0.96	1.08	0.97	1.10

Table A.2: Comparison of naive and linear combination estimators for simulation approach for BIZ. The values given are the averages over 100 replications of the ratio of the estimator to true value

BIZ						
	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.24	0.73	0.19	0.72	0.17	0.73
SC-INC	0.22	0.77	0.17	0.76	0.15	0.81
SC-DEC	0.22	0.63	0.17	0.67	0.15	0.71
MDM	0.91	0.94	0.95	0.96	0.98	0.99
MDM-INC	0.99	1.01	0.99	1.02	1.00	1.01
MDM-DEC	0.77	0.82	0.86	0.87	0.96	0.96
RPI1	0.64	1.07	0.64	1.00	0.65	0.99
RPI2	0.93	1.04	0.90	0.99	0.89	0.99
RPI1-HET	0.88	1.07	0.89	1.05	0.86	1.06

Table A.3: Comparison of naive and linear combination estimators for simulation approach for GSP. The values given are the averages over 100 replications of the ratio of the estimator to true value

GSP						
	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.13	0.68	0.08	0.59	0.06	0.68
SC-INC	0.14	0.67	0.08	0.64	0.05	0.82
SC-DEC	0.13	0.63	0.07	0.62	0.05	0.66
MDM	0.75	0.80	0.82	0.85	0.89	0.90
MDM-INC	0.94	0.98	0.96	0.99	0.99	1.00
MDM-DEC	0.58	0.58	0.68	0.71	0.79	0.79
RPI1	0.55	1.27	0.65	1.08	0.57	0.96
RPI2	1.03	1.10	0.89	1.00	0.88	0.98
RPI1-HET	1.07	1.20	0.92	1.16	0.93	1.17

Table A.4: Comparison of naive and linear combination estimators for first-order approach for KN. The values given are the averages over 100 replications of the ratio of the estimator to true value

KN

	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.29	0.67	0.22	0.72	0.19	0.72
SC-INC	0.45	1.01	0.30	1.04	0.24	1.11
SC-DEC	0.29	0.60	0.21	0.68	0.18	0.69
MDM	1.27	0.92	1.35	0.96	1.23	0.97
MDM-INC	1.91	1.04	2.03	1.04	1.43	1.01
MDM-DEC	1.19	0.80	1.24	0.85	1.22	0.91
RPI1	0.75	1.08	0.67	1.02	0.64	0.98
RPI2	1.17	1.06	0.96	1.00	0.90	0.97
RPI1-HET	1.29	1.17	1.07	1.17	0.99	1.18

Table A.5: Comparison of naive and linear combination estimators for first-order approach for BIZ. The values given are the averages over 100 replications of the ratio of the estimator to true value

BIZ						
	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.22	0.76	0.20	0.71	0.18	0.71
SC-INC	0.22	0.71	0.19	0.74	0.18	0.84
SC-DEC	0.19	0.61	0.16	0.68	0.15	0.70
MDM	0.89	0.90	0.97	0.96	0.99	0.99
MDM-INC	1.03	1.03	1.04	1.02	1.02	1.01
MDM-DEC	0.73	0.78	0.84	0.87	0.93	0.94
RPI1	0.64	1.09	0.66	0.99	0.67	0.96
RPI2	0.93	1.04	0.97	0.99	0.98	0.99
RPI1-HET	0.85	1.10	0.86	1.05	0.90	1.07

Table A.6: Comparison of naive and linear combination estimators for first-order approach for GSP. The values given are the averages over 100 replications of the ratio of the estimator to true value

GSP

	$k = 100$		$k = 500$		$k = 2000$	
Configuration	Naive	Linear Combination	Naive	Linear Combination	Naive	Linear Combination
SC	0.13	0.68	0.08	0.59	0.06	0.68
SC-INC	0.14	0.67	0.08	0.64	0.05	0.82
SC-DEC	0.13	0.63	0.08	0.62	0.05	0.66
MDM	0.76	0.80	0.83	0.86	0.88	0.90
MDM-INC	0.96	0.99	0.96	0.99	0.98	1.00
MDM-DEC	0.58	0.57	0.70	0.70	0.78	0.80
RPI1	0.62	1.25	0.61	1.03	0.58	0.98
RPI2	0.99	1.10	0.91	1.02	0.90	1.00
RPI1-HET	0.99	1.23	0.97	1.17	0.94	1.18

APPENDIX B

APPENDIX FOR CHAPTER 3

B.1 Supporting Results for Theorem 3

Lemma 3 (Chernoff Bound for sub-Gaussian Random Walk)**.** *Define* $(W_n, n = 1, 2, \ldots)$ *as a random walk with i.i.d. standardized (mean 0, scale factor 1) sub-Gaussian increments. For any function* $f(n)$*,*

$$\mathbb{P}\left(W_n \geq \sqrt{2xnf(n)}\right) \leq \exp\left(-xf(n)\right), \forall n, \forall x.$$

PROOF.

$$\begin{aligned}\mathbb{P}\left(W_n \geq \sqrt{2xnf(n)}\right) &\leq \exp\left(-t\sqrt{2xnf(n)}\right)\left(\exp\left(\frac{t^2}{2}\right)\right)^n \\ &= \exp\left(-t\sqrt{2xnf(n)} + \frac{nt^2}{2}\right)\end{aligned}$$

Choosing $t = \sqrt{\frac{2xf(n)}{n}}$ gives the desired inequality. □

Lemma 4 (Hoeffding's Maximal Inequality)**.** *If* $X_1, X_2, \ldots, X_N$ *are i.i.d. standard (with scale parameter 1) sub-Gaussian random variables, then*

$$\mathbb{P}\left(\max_{n=1,2,\ldots,m} W_n \geq x\right) \leq \exp\left(-\frac{x^2}{2m}\right).$$

PROOF. First, for any non-negative t, $\exp\left(tW_n\right)$ is a submartingale with respect

to the natural filtration $\{\mathcal{F}_n\}$, since

$$\begin{aligned}\mathbb{E}\left[\exp\left(tW_n\right)|\mathcal{F}_{n-1}\right] &= \mathbb{E}\left[\exp\left(tX_n\right)\exp\left(tW_{n-1}\right)|\mathcal{F}_{n-1}\right] \\ &= \mathbb{E}\left[\exp\left(tX_n\right)\right]\exp\left(tW_{n-1}\right) \\ &\geq \exp\left(tW_{n-1}\right),\end{aligned}$$

where the last inequality holds since

$$\mathbb{E}\left[\exp\left(tX_n\right)\right] \geq \exp\left(t\mathbb{E}X_n\right) = 1$$

by Jensen's inequality and the fact that $\mathbb{E}X_n = 0$. Hence, by Doob's martingale inequality,

$$\begin{aligned}\mathbb{P}\left(\max_{n=1,2,\ldots,m} W_n \geq x\right) &= \mathbb{P}\left(\max_{n=1,2,\ldots,m} \exp\left(tW_n\right) \geq \exp\left(tx\right)\right) \\ &\leq \frac{\mathbb{E}\left[\exp\left(tW_m\right)\right]}{\exp\left(tx\right)} \\ &= \frac{\left(\mathbb{E}\left[\exp\left(tX_n\right)\right]\right)^m}{\exp\left(tx\right)} \\ &\leq \exp\left(\frac{1}{2}t^2m - tx\right).\end{aligned}$$

Choosing $t = x/m$ gives the desired inequality. □

Proposition 3. *Let ε^* be the solution of the following optimization problem*

$$\min_{\varepsilon}\ (1+\sqrt{\varepsilon})^2\left(\frac{3}{2}\ln\left(\frac{16}{\ln(1+\varepsilon)}\right) + \frac{3}{2}\ln\left(\frac{1}{a}\right) + \ln\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right)\right). \quad \text{(B.1)}$$

Then

$$\varepsilon^* = \mathcal{O}\left(\left(\ln\frac{k}{\alpha} + \ln\ln\frac{\sigma}{\delta}\right)^{-1}\right).$$

PROOF. We have the relationships

$$(1+\sqrt{\varepsilon})^2 = 1 + 2\sqrt{\varepsilon} + \mathcal{O}(\varepsilon),$$

$$\ln\left(\frac{1}{\ln(1+\varepsilon)}\right) = \ln\left(\frac{1}{\varepsilon}\right) + \mathcal{O}(\varepsilon),$$

and

$$\ln\ln\left(\frac{4\sqrt{2}(1+\sqrt{\varepsilon})(1+\varepsilon)\sigma}{\delta}\right) = \ln\ln\left(\frac{4\sqrt{2}\sigma}{\delta}\right) + \mathcal{O}(\sqrt{\varepsilon}).$$

Therefore, (B.1) equals

$$\left(1 + 2\sqrt{\varepsilon} + \mathcal{O}(\varepsilon)\right)\left[\ln\left(\frac{16}{a}\right) + \frac{1}{3}\ln\ln\left(\frac{4\sqrt{2}\sigma}{\delta}\right) + \ln\left(\frac{1}{\varepsilon}\right) + \mathcal{O}(\sqrt{\varepsilon})\right].$$

Let $C = \ln(16/a) + \frac{1}{3}\ln\ln\left(4\sqrt{2}\sigma/\delta\right)$. Then (B.1) is asymptotically equivalent to

$$f(\varepsilon) = (1 + 2\sqrt{\varepsilon})\left(C + \ln\left(\frac{1}{\varepsilon}\right)\right).$$

Setting the derivative to zero, the solution ε^* satisfies

$$C + \ln\left(\frac{1}{\varepsilon^*}\right) = 2 + \frac{1}{\sqrt{\varepsilon^*}}. \tag{B.2}$$

Let $x = \ln\left(\frac{1}{\varepsilon^*}\right)$. Then $\varepsilon^* = e^{-x}$, and so (B.2) becomes

$$C + x = 2 + e^{\frac{x}{2}}.$$

Therefore $C = \mathcal{O}\left(e^{\frac{x}{2}}\right)$ and $x = \mathcal{O}(\ln C)$, hence

$$\varepsilon^* = \mathcal{O}\left(\frac{1}{C}\right) = \mathcal{O}\left(\left(\ln\frac{k}{\alpha} + \ln\ln\frac{\sigma}{\delta}\right)^{-1}\right).$$

□

BIBLIOGRAPHY

[1] Jerry Banks. *Handbook of simulation: principles, methodology, advances, applications, and practice*. John Wiley & Sons, 1998.

[2] Robert E. Bechhofer. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *The Annals of Mathematical Statistics*, pages 16–39, 1954.

[3] Robert E. Bechhofer. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3):408–429, 1958.

[4] Robert E. Bechhofer, Thomas J. Santner, and David Goldsman. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. Wiley, New York, 1995.

[5] Patrick Billingsley. *Probability and Measure*. Wiley, New York, 3rd edition, 1995.

[6] Jürgen Branke, Stephen E. Chick, and Christian Schmidt. Selecting a selection procedure. *Management Science*, 53(12):1916–1932, 2007.

[7] Leo Breiman. Probability. *Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA*, 1992.

[8] Valerii V. Buldygin and Yu V. Kozachenko. Sub-Gaussian random variables. *Ukrainian Mathematical Journal*, 32(6):483–489, 1980.

[9] Chun-Hung Chen, Stephen E. Chick, Loo Hay Lee, and Nugroho A. Pujowidianto. Ranking and selection: efficient simulation budget allocation. In *Handbook of Simulation Optimization*, pages 45–80. Springer, 2015.

[10] Chun-Hung Chen, Jianwu Lin, Enver Yücesan, and Stephen E. Chick. Simulation budget allocation for further enhancing the efficiency of ordinal optimization. *Discrete Event Dynamic Systems*, 10(3):251–270, 2000.

[11] Stephen E Chick, Jürgen Branke, and Christian Schmidt. Sequential sampling to myopically maximize the expected value of information. *INFORMS Journal on Computing*, 22(1):71–80, 2010.

[12] Stephen E. Chick and Koichiro Inoue. New procedures to select the best simulated system using common random numbers. *Management Science*, 47(8):1133–1149, 2001.

[13] Stephen E. Chick and Koichiro Inoue. New two-stage and sequential procedures for selecting the best simulated system. *Operations Research*, 49(5):732–743, 2001.

[14] Morris H DeGroot. *Optimal statistical decisions*. McGraw-Hill, Inc., 1970.

[15] David J. Eckman and Shane G. Henderson. Fixed-confidence, fixed-tolerance guarantees for selection-of-the-best procedures. Working paper, 2018.

[16] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[17] Weiwei Fan, L Jeff Hong, and Barry L Nelson. Indifference-zone-free selection of the best. *Operations Research*, 64(6):1499–1514, 2016.

[18] Peter I. Frazier. A fully sequential elimination procedure for indifference-zone ranking and selection with tight bounds on probability of correct selection. *Operations Research*, 62(4):926–942, 2014.

[19] Peter Glynn and Sandeep Juneja. Ordinal optimization-empirical large deviations rate estimators, and stochastic multi-armed bandits. *arXiv preprint arXiv: 1507.04564*, 2015.

[20] Peter W Glynn and Philip Heidelberger. Bias properties of budget constrained simulations. *Operations Research*, 38:801–814, 1990.

[21] S. S. Gupta and S. Panchapakesan. *Multiple decision procedures: Theory and methodology of selecting and ranking populations*. Wiley, 1979.

[22] Shanti Swarup Gupta. *On a decision rule for a problem in ranking means*. PhD thesis, University of North Carolina at Chapel Hill, 1956.

[23] Shane G. Henderson and Raghu Pasupathy. Simulation optimization library. `http://www.simopt.org.`, 2014.

[24] L. Jeff Hong. Fully sequential indifference-zone selection procedures with variance-dependent sampling. *Naval Research Logistics (NRL)*, 53(5):464–476, 2006.

[25] L. Jeff Hong, Jun Luo, and Ying Zhong. Speeding up pairwise comparisons for large scale ranking and selection. In *Proceedings of the 2016 Winter Simulation Conference*, pages 749–757. IEEE Press, 2016.

[26] L. Jeff Hong, Barry L Nelson, and Jie Xu. Discrete optimization via simulation. In Michael C Fu, editor, *Handbook of Simulation Optimization*, volume 216 of *International Series in Operations Research & Management Science*, pages 9–44. Springer New York, 2015.

[27] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sebastien Bubeck. On finding the largest mean among many. *arXiv preprint arXiv:1306.3917*, 2013.

[28] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439, 2014.

[29] Kevin Jamieson and Robert Nowak. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. In *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, pages 1–6. IEEE, 2014.

[30] Christopher Jennison, Iain M. Johnstone, and Bruce W. Turnbull. Asymptotically optimal procedures for sequential adaptive selection of the best of several normal means. In *Statistical decision theory and related topics III*, pages 55–86. Elsevier, 1982.

[31] June Young Jung, Gary Blau, Joseph F. Pekny, Gintaras V. Reklaitis, and David Eversdyk. A simulation based optimization approach to supply chain

management under demand uncertainty. *Computers & chemical engineering*, 28(10):2087–2106, 2004.

[32] Nabil Kahale et al. Analytic crossing probabilities for certain barriers by brownian motion. *The Annals of Applied Probability*, 18(4):1424–1440, 2008.

[33] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 655–662, 2012.

[34] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, pages 1238–1246, 2013.

[35] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.

[36] Seong-Hee Kim and Barry L. Nelson. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 11(3):251–273, 2001.

[37] Seong-Hee Kim and Barry .L Nelson. On the asymptotic validity of fully sequential selection procedures for steady-state simulation. *Operations Research*, 54(3):475–488, 2006.

[38] Seong-Hee Kim and Barry L. Nelson. Selecting the best system. *Handbooks in operations research and management science*, 13:501–534, 2006.

[39] Soonhui Lee and Barry L. Nelson. Bootstrap ranking & selection revisited. In Andreas Tolk, Saikou Y. Diallo, Ilya O. Ryzhov, Levent Yilmaz, Stephen Buckley, and John A. Miller, editors, *Proceedings of the 2014 Winter Simulation Conference*, pages 3857–3868, Piscataway, New Jersey, 2014. Institute of Electrical and Electronics Engineers, Inc.

[40] Soonhui Lee and Barry L. Nelson. Computational improvements in bootstrap ranking & selection procedures via multiple comparison with the best. In Levent Yilmaz, Wai Kin (Victor) Chan, IlChul Moon, Theresa M. K. Roeder, Charles Macal, and Manuel D. Rossetti, editors, *Proceedings of the 2015 Winter Simulation Conference*, pages 3758–3767, Piscataway, New Jersey, 2015. Institute of Electrical and Electronics Engineers, Inc.

[41] Jun Luo, L Jeff Hong, Barry L Nelson, and Yang Wu. Fully sequential procedures for large-scale ranking-and-selection problems in parallel computing environments. *Operations Research*, 63(5):1177–1194, 2015.

[42] Sijia Ma. Rns running time. 2016.

[43] Sijia Ma and Shane G. Henderson. An efficient fully sequential selection procedure guaranteeing probably approximately correct selection. In Wai Kin (Victor) Chan, Andrea D'Ambrogio, Gregory Zacharewicz, Navonil Mustafee, Gabriel Wainer, and Ernest H. Page, editors, *Proceedings of the 2017 Winter Simulation Conference*, Piscataway, New Jersey, 2017. Institute of Electrical and Electronics Engineers, Inc.

[44] Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

[45] Matthew S. Maxwell, Shane G. Henderson, Huseyin Topaloglu, et al. Tuning approximate dynamic programming policies for ambulance redeployment via direct search. *Stochastic Systems*, 3(2):322–361, 2013.

[46] Barry L. Nelson and Souvik Banerjee. Selecting a good system: Procedures and inference. *IIE Transactions*, 33(3):149–166, 2001.

[47] Eric C. Ni, Dragos F. Ciocan, Shane G. Henderson, and Susan R. Hunter. Efficient ranking and selection in parallel computing environments. *Operations Research*, 65(3):821–836, 2017.

[48] Eric C Ni, Susan R Hunter, and Shane G Henderson. Ranking and selection in a high performance computing environment. In *Simulation Conference (WSC), 2013 Winter*, pages 833–845. IEEE, 2013.

[49] Edward Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, pages 174–180, 1964.

[50] Yijie Peng, Edwin KP Chong, Chun-Hung Chen, and Michael C Fu. Ranking and selection as stochastic control. *arXiv preprint arXiv:1710.02619*, 2017.

[51] Yijie Peng, Michael C Fu, Peter W Glynn, and Jianqiang Hu. On the asymp-

totic analysis of quantile sensitivity estimation by monte carlo simulation. In *Simulation Conference (WSC), 2017 Winter*, pages 2336–2347. IEEE, 2017.

[52] Yosef Rinott. On two-stage selection procedures and related probability-inequalities. *Communications in Statistics-Theory and methods*, 7(8):799–811, 1978.

[53] Moshe Shaked and George Shanthikumar. *Stochastic orders*. Springer Science & Business Media, 2007.

[54] SR Srinivasa Varadhan. *Stochastic processes*, volume 16. American Mathematical Soc., 2007.