Topics in Markov Chain Theory and Simulation Optimisation

Berkhout, J.

2016

document version
Publisher's PDF, also known as Version of record

2. SIMULATION OPTIMISATION: AN ACCELERATED STOPPING RULE FOR NPHA

This chapter is based on [28].

In this chapter we present an accelerated stopping rule for improving the performance of the nested partition hybrid algorithm (NPHA) from [169], which is a general purpose algorithm for simulation optimisation, also known as stochastic discrete optimisation. Numerical examples will illustrate the impact of the accelerated stopping rule on the overall performance of NPHA.

The outline of this chapter is as follows. In Section 2.1 the simulation optimisation problem setting is introduced, Section 2.2 explains the original version of NPHA, and Section 2.3 defines the accelerated stopping rule supported with numerical experiments which illustrates the improvement that can be achieved. Furthermore, Section 2.3 contains an elaboration about the possible theoretical improvement that can be obtained. Lastly, in Section 2.4 some conclusions are made about the accelerated stopping rule in the setting of NPHA.

2.1 Problem Setting Simulation Optimisation

2.1.1 The Model

In this chapter we consider the following optimisation problem:

$$\theta^* = \arg\min_{\theta \in \Theta} z(\theta)\,, \tag{2.1}$$

where $z(\theta)$ is a real-valued cost function representing the costs of operating a given complex system under design θ, and Θ is a (large) finite set containing all possible designs. Design θ^* is defined as the design with the lowest cost (i.e., the best design) and, for ease of presentation, we assume throughout the chapter that θ^* is unique. Typically, no explicit form of the objective function $z(\theta)$ is available and/or function evaluations are stochastic and computationally expensive.

Write $z(\theta) = E[L(\theta)]$, where $L(\theta)$ is a random variable based on θ. We assume that there is no analytical expression for $z(\theta)$ known and that it can only be estimated via discrete-event simulation. This simulation yields realizations for the random variable $L(\theta)$ which we denote by $L_1(\theta), L_2(\theta), \ldots, L_n(\theta)$ in case of n realizations. These realizations are independent and identically distributed (i.i.d.) by assumption. Also, we assume that simulations for designs θ and θ', with $\theta \neq \theta'$, are independent. The

expected costs $z(\theta) = E[L(\theta)]$ is estimated by means of the sample average $\hat{z}_n(\theta)$, i.e.,

$$\hat{z}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_i(\theta),\ \theta \in \Theta\,,\ n \geq 1,$$

and the law of large numbers yields $\lim_{n\to\infty} \hat{z}_n(\theta) = z(\theta)$ with probability one. In Example 2.1 a typical simulation optimisation problem can be found which is called the worker allocation problem.

Example 2.1 (**Worker Allocation Problem**, [52])**.** Suppose that we have a tandem queue which includes two stages of service, see also Figure 2.1. Customers arrive from the outside at station 1. After finishing service at this station they move on to station 2. Upon service completion at station 2, they leave the system. The system consists of two service stages, both with some different general service time distributions. The goal is to allocate a given number of workers to the two stages such that the average waiting time of the customers is as small as possible. Each worker can help one customer at the time and when there is no worker available the customer has to wait. Further, we assume that workers assigned to one stage cannot help at the other stage due to safety and training requirements. Suppose that W_1 and W_2 are the number of workers assigned to stage 1 and stage 2 respectively and W is the total number of workers available, i.e., it should hold that $W_1 + W_2 = W$. So the number of possible solutions is equal to $W + 1$. It's clear that $W_1 = 0$ or $W_2 = 0$ will not result in an optimal solution for this problem so that $W_1 \geq 1$ and $W_2 \geq 1$, in which case there are $W - 1$ possible solutions. In contrast to Jackson networks studied in Chapter 3, there is no closed-form analytical expression known to evaluate an allocation for this problem so that the optimizer has to rely on simulation analysis for evaluating allocations. ◁

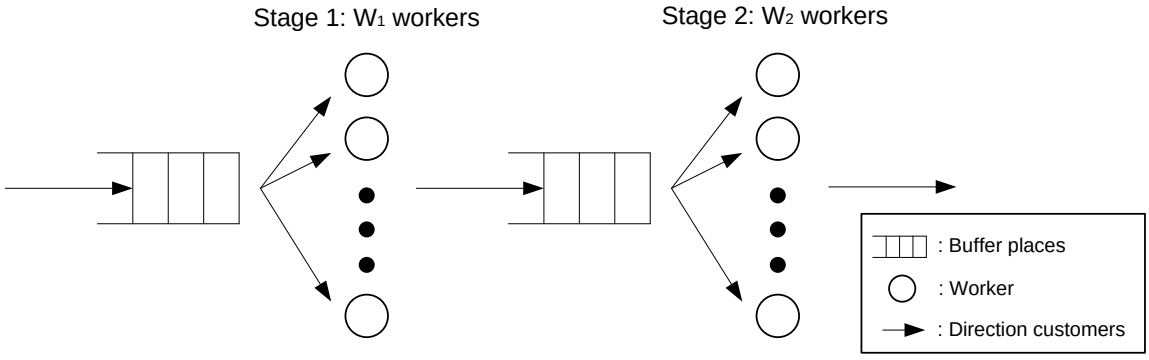

Fig. 2.1: Illustration of the worker allocation problem. W_1 and W_2 are the number of workers assigned at stage 1 and stage 2 respectively.

There are mainly two reasons why stochastic optimisation problems are hard to solve in practice:

1. the size of the design space Θ is for most practical problems enormous. As a result it is difficult to find good designs, not speaking about finding the best design, among all possible designs.

2. since no analytical expression is available to evaluate designs, simulation analysis is the only tool possible for estimating the performance of designs. In particular when estimating steady state measures the computational effort is substantial.

2.1.2 Literature Review

With the increase of computational power on standard computers, solving (2.1) by means of simulation based search algorithms has become possible, and this has initiated intense research on efficient algorithms for (approximately) solving (2.1), see, for example, [72, 74]. As observed in [98], random search algorithms are by far the most commonly used algorithms. The random search algorithms differ mostly in terms of their neighbourhood structure, sampling methods to obtain candidate designs, and ranking and selection of candidate designs. Examples are the stochastic ruler method [194, 10], Andradòttir's random search methods [12, 13], simulated annealing [9], the stochastic comparison algorithm [78], and the COMPASS method [97, 193, 100]. For more details on recent developments, we refer to [14, 72, 74, 98], the outstanding survey in [97] and the recent published Handbook of Simulation Optimization by M. C. Fu, [73].

2.2 Background on Nested Partition Hybrid Algorithm

In this section we will focus on the nested partition hybrid algorithm (NPHA) [169], which is an extension of the nested partitions (NP) method [170]. The reason for studying NPHA is that it is one of the standard methods for simulation optimisation and has good numerical performance, [169]. Furthermore, the algorithm has an attractive philosophy of combining different techniques that have good performances individually. In particular, NPHA has three main ingredients: NP [170], Ordinal Optimization (OO) [96, 95] and Optimal Computing Budget Allocation (OCBA) [53][1]. NPHA uses NP as a basic framework to steer the search process. More specifically, NP iteratively *partitions* the current most *promising region*, i.e., the region containing the design with the smallest average cost, of the design space into smaller *subregions*. All subregions are considered together with some surrounding region, which is defined by the *backtracking rule*, and NPHA tries to find the most promising region for the next iteration (in case that the surrounding region is most promising, the algorithm *backtracks* to a certain larger region which usually contains the previous most promising region). NPHA terminates when it satisfies a user-defined stopping criterion, e.g., when a new most promising region contains one design (i.e., when maximal 'depth' is reached).

NP can deal with large designs spaces and the efficiency of this algorithm is based on the implicit assumption that there is structure in the problem, i.e., good designs tend to cluster together and thus the partitioning results in subregions in which designs are relatively similar. In case the problem structure is poor and similar designs don't cluster in subregions the nested partition method cannot utilize the problem structure so that it follows by the "no free lunch" theorems[2] from [192] that NP will not be better

[1] In fact, NPHA in [169] uses the allocation rule from [55], but as argued in [53] this rule is inferior to the allocation rule from [53]. So here we use the OCBA rule from [53] in NPHA.

[2] These theorems establish that for any algorithm, any elevated performance over one class of problems is offset by performance over another class.

than purely random search. When applying NP, effort should be put in ensuring that the structure of the problem is exploited by choosing the right representation of the problem with appropriate partition scheme. To find the most promising region, NP samples designs from each region according to some *sampling scheme*. OO and OCBA are integrated to efficiently identify the region which contains the best design with some certainty. Here, 'certainty' means that the probability of correct selection should be larger or equal to some user-defined value. In the following, more formal NPHA background is introduced where we mainly focus on details of NPHA with respect to selection of the most promising region. For a more detailed exposition of NPHA we refer to [169].

The notation introduced in this section follows the same idea as in [169]. Suppose NPHA is running and in iteration $k-1$ the most promising region found is $\sigma(k)$, notice that this is a subset of Θ which follows by the user-defined partitioning rule which, for efficiency, elaborates the problem structure. In iteration k the current most promising region $\sigma(k)$ is partitioned in $M_{\sigma(k)}$ subregions (we use this notation to indicate that the number of partitions may depend on $\sigma(k)$) which are denoted as follows:

$$\sigma_1(k), \sigma_2(k), \ldots, \sigma_{M_{\sigma(k)}}(k).$$

Note that it holds that $\sigma_i(k) \cap \sigma_j(k) = \emptyset$ for $i \neq j$ and

$$\bigcup_{l=1}^{M_{\sigma(k)}} \sigma_l(k) = \sigma(k)\,.$$

Also the surrounding region of the current most promising region $\sigma(k)$ (which depends on the user-defined backtracking rule) is considered which we denote as $\sigma_{M_{\sigma(k)}+1}(k)$. From each region $\sigma_j(k)$, $j = 1, 2, \ldots, M_{\sigma(k)} + 1$, we randomly draw $N_{\sigma_j(k)}$ designs according to an OO sampling scheme. The set containing the sampled designs in $\sigma_j(k)$ is denoted with $D_{\sigma_j(k)}$ for which it holds that

$$D_{\sigma_j(k)} = \{\theta^{j1}, \theta^{j2}, \ldots, \theta^{jN_{\sigma_j(k)}}\}, \quad j = 1, 2, \ldots, M_{\sigma(k)} + 1,$$

which means that θ^{jl} is the l-th sampled design from subregion $\sigma_j(k)$. Let set $H(k)$ contain all sampled design in iteration k, it then follows that

$$H(k) = \bigcup_{j=1}^{M_{\sigma(k)}+1} D_{\sigma_j(k)}.$$

Note that the cardinality of set $H(k)$ is $N_{\sigma_j(k)}(M_{\sigma(k)}+1)$. Based on the sampled designs, each region $\sigma_j(k)$, $j = 1, 2, \ldots, M_{\sigma(k)} + 1$, is evaluated.

The design in $H(k)$ that has the lowest estimated performance value is defined as $\theta_a(k)$. The most promising region in the k-th iteration is obtained as the region that contains the most promising design, i.e.,

$$\theta_a(k) = \arg\min_{\theta \in H(k)} \hat{z}_{n_\theta}(\theta) \quad \text{and} \quad j_a^k = j \ \text{ iff } \ \theta_a(k) \in \sigma_j(k), \tag{2.2}$$

where n_θ is the number of realizations used for the sample average for design θ, and j_a^k denotes the index of the most promising region. Unfortunately, it is not possible

to choose the best design from $H(k)$ with certainty unless we simulate each design infinitely often.

OCBA [53] will be incorporated in order to (approximately) maximize simulation efficiency for a given finite simulation budget (i.e., the number of simulations available). OCBA [53] is an efficient simulation budget allocation algorithm that takes both (sample) means and (sample) variances into account in order to efficiently choose the number of simulation samples for all designs. It ensures that more simulation budget is allocated to designs that are critical in identifying good designs. More specifically, the fraction of the additional simulation budget assigned by OCBA to $\theta \in H(k)$ is determined through the estimated difference in performances between $\theta_a(k)$ and $\theta \in H(k) \setminus \{\theta_a(k)\}$, weighted by the sampled variances.

In order to present the formal OCBA rule [53] we need to introduce some notation with respect to selecting the most promising region, i.e., finding the best design out of $H(k)$. We denote by $CS(k)$ the event that the design with the lowest estimated performance value chosen in iteration k of the NPHA iteration has indeed the best performance value of $H(k)$, i.e., the event of *Correct Selection*. Put differently, $CS(k)$ is equivalent to the event $\{z(\theta_a(k)) \leq z(\theta),\ \theta \in H(k) \setminus \{\theta_a(k)\}\}$. In order to guarantee the selection of the most promising region with some minimal probability, the simulation budget for the designs out of $H(k)$ is increased until the probability of $CS(k)$ is at least P^*, where P^* is a predefined confidence level. This defines a *stopping rule* for the optimisation problem in (2.2). Since it is impossible to exactly calculate $\Pr(CS(k))$ efficiently in the case of normal random variables, NPHA uses an approximation for $\Pr(CS(k))$ called the *Approximate Correct Selection* probability, denoted by $\Pr(ACS(k))$, where the event $ACS(k)$ is similar to the event $CS(k)$ but with the difference that the pairwise comparisons between $\theta_a(k)$ and the designs in $H(k)$ are assumed to be independent (which is clearly an unrealistic assumption). This probability $\Pr(ACS(k))$ is a a lower bound for the confidence level of the ordinal comparison, i.e., $\Pr(CS(k))$, and experiments in [51] show that it is a suitable approximation in many cases. The precise statement is stated in the theorem below.

Theorem 2.1 ([51])**.** *The probability of Approximate Correct Selection in iteration k of NPHA is given by:*

$$\Pr(ACS(k)) = \prod_{\theta \in H(k) \setminus \{\theta_a(k)\}} \Phi\left(\frac{\hat{z}_{n_\theta}(\theta) - \hat{z}_{n_{\theta_a(k)}}(\theta_a(k))}{\sqrt{\frac{\sigma_\theta^2}{n_\theta} + \frac{\sigma_{\theta_a(k)}^2}{n_{\theta_a(k)}}}} \right), \tag{2.3}$$

where σ_θ^2 is the variance of the random variable $L(\theta)$ and $\Phi(\cdot)$ is the cumulative normal distribution function. $\Pr(ACS(k))$ *is a lower-bound for the Correct Selection probability, denoted as* $\Pr(CS(k))$*. In mathematical terms this means that* $\Pr(CS(k)) \geq \Pr(ACS(k))$*.*

Note that in practice variance $\sigma_{n_\theta}^2$ is typically unknown in which case the standard normal distribution in Theorem 2.1 is replaced by the t-distribution and the variance is estimated by an appropriate estimator.

The OCBA rule [53] asymptotically maximizes a lower bound of the probability of selecting the truly best design by dividing a simulation budget Δ, which tends to

infinity, among the designs according to

$$\begin{aligned} &1. \quad \frac{n_\theta}{n_{\theta'}} = \left(\frac{\sigma_\theta(\hat{z}_{n_{\theta_a(k)}}(\theta_a(k)) - \hat{z}_{n_{\theta'}}(\theta'))}{\sigma_{\theta'}(\hat{z}_{n_{\theta_a(k)}}(\theta_a(k)) - \hat{z}_{n_\theta}(\theta))} \right), \quad \theta, \theta' \in H(k) \setminus \{\theta_a(k)\} \text{ and } \theta \neq \theta', \\ &2. \quad n_{\theta_a(k)} = \sigma_{\theta_a(k)} \sqrt{\sum_{\theta \in H(k) \setminus \{\theta_a(k)\}} \frac{n_\theta^2}{\sigma_\theta^2}}, \end{aligned} \tag{2.4}$$

The OCBA allocation given above is derived by taking $\Delta \to \infty$. While it is impossible to have an infinite computing budget in real life, OCBA provides a simple means for allocating simulation budget in a way that the efficiency can be significantly improved, for a more detailed discussion see [52]. In particular, (2.4) gives $N_{\sigma_j(k)}(M_{\sigma(k)} + 1) - 1$ equations so that together with budget constraint $\sum_{\theta \in H(k)} n_\theta = \Delta$ we can solve for n_θ, $\theta \in H(k)$.

There are different extensions of the OCBA approach, for an extensive overview of these developments of OCBA we refer to [52]. The OCBA developed in [53] has the advantage that it avoids solving an additional budget allocation problem which would lead to extra computation cost. Moreover, the computational tests in [42] showed that OCBA is one of the top performers of different selection procedures.

In order to solve problem (2.2), NPHA uses pseudo code I below for finding the best design $\theta_a(k)$ with P^* certainty out of set $H(k)$ (so that it is guaranteed that we also select the most promising region with P^* certainty). In pseudo code I there are two user-defined variables, n_0 and Δ, where n_0 is the number of initial simulation runs for each design and Δ gives the extra amount of simulation runs that is available in the next iteration of the pseudo code.

Pseudo code I for finding the most promising region with P^* certainty

Start: *Perform n_0 initial simulation runs for each design.*

While $\Pr(ACS(k)) < P^*$**:** *increase the computational budget with Δ and divide this extra computational budget among the different designs according to the optimal computational budget allocation rule from* (2.4) *until the stopping rule holds.*

Return: *The most promising region containing the best found design.*

The idea behind the pseudo code is to first get some insight in the designs using n_0 start simulation. Afterwards, the number of simulations that can be used is increased and efficiently allocated among the different designs using OCBA. So the value for n_0 should not be too small else it could result in poor estimates for the performance value and its variance, resulting in premature termination of the comparison. Furthermore, Δ should not be too small since it will result in unnecessarily many iterations. On the other hand, choosing n_0 and Δ too large can result in efficiency loss because of unnecessarily many simulations. So a careful trade-off should be made in choosing these values. For more details see, e.g., [169].

2.3 Accelerated Stopping Rule

In this section the accelerated stopping rule is introduced. In particular, Subsection 2.3.1 gives the idea and the mathematical details. It also contains a discussion about using

the OCBA rule for the accelerated stopping rule. Subsection 2.3.2 contains an informal explanation why the accelerated stopping rule is 'accelerating' in the NPHA setting and it also contains a formal proof of the theoretical improvement that can be achieved in a specific case. Lastly, to put the accelerated stopping rule to the test the successive two subsections are devoted to two numerical experiments, namely Subsections 2.3.3 and 2.3.4, respectively. The first numerical experiment considers an academic problem of which the optimal solution is known in advance and the second numerical experiments considers the Buffer Allocation Problem for a comparison study.

2.3.1 General Idea

In this subsection the accelerated stopping rule for NPHA is introduced. To begin with we define the event of the *Correct Selection of the most promising Region*, denoted by $CSR(k)$, as the event that we correctly choose the best region according to the drawn samples in an iteration of NPHA. The difference between the events $CS(k)$ and $CSR(k)$ is that the event $CSR(k)$ also includes the possibility that another design in the most promising region is the truly best design with P^* certainty. This prevents the situation that NPHA puts much effort in identifying the truly best design while it is already clear that the best design is from a specific region. For example, if we have two almost identical (in terms of the performance values) designs from one region, say σ^*, and these designs are clearly the best designs out of the drawn sample of designs. Then we don't have to identify the truly best design because we already know that region σ^* is the most promising region. More specially, when the subregions become smaller this will have a significant effect on the number of simulations used. To see this, note that the probability that designs have similar performance values will become larger when the subregions become smaller.

Denote the event that θ is indeed the truly best design in $H(k)$ based on n_θ ($\forall\theta \in H(k)$) simulations, by $CS(\theta, k)$. In [51] the simulation information n_θ ($\forall\theta \in H(k)$) is used to obtain the posterior distribution of $z(\theta)$, defined as $\tilde{L}(\theta)$ for design θ, which can be used to calculate $\Pr(CS(\theta, k))$. In case of a non-informative prior[3], it holds for the posterior distribution that

$$\tilde{L}(\theta) \sim \mathcal{N}\left(\hat{z}_{n_\theta}(\theta),\ \frac{\sigma^2_{n_\theta}}{n_\theta}\right), \tag{2.5}$$

where $\sigma^2_{n_\theta}$ is the variance of $L(\theta)$ (which again can be approximated by replacing the normal distribution with the t-distribution instead). Using the posterior, the probability $\Pr(CS(\theta, k))$ can be expressed as follows:

$$\Pr(CS(\theta, k)) = \Pr\left(\tilde{L}(\theta) < \min_{\theta' \in H(k)\setminus\{\theta\}} \tilde{L}(\theta')\right). \tag{2.6}$$

This holds since $CS(\theta, k)$ describes the event that θ is indeed the truly best design in $H(k)$, which is equal to the event that $\tilde{L}(\theta)$, $\theta \in H(k)$, is smaller than all other $\tilde{L}(\theta')$, $\theta' \in H(k) \setminus \{\theta\}$, in terms of the posterior distribution. Note that $\Pr(CS(k)) = \Pr(CS(\theta_a(k), k))$, where $\theta_a(k)$ is the selected design in iteration k ($\theta_a(k) \in H(k)$).

[3] In [51] the prior distribution for $z(\theta)$ is $N(0, v^2)$ where v is an extremely large value. This means that we do not have any idea about the value of $z(\theta)$ before doing simulations.

For the accelerated stopping rule we define the event *Approximate Correct Selection of the most promising Region* ($ACSR(k)$), which is similarly defined as the event $CSR(k)$ but again with the (unrealistic) assumption that all pairwise comparisons between any two designs in $H(k)$ are independent. In the theorem below we will show that using our above results on $\Pr(CS(\theta,k))$, we obtain a feasible expression for $\Pr(ACSR(k))$, which provides a lower bound for $\Pr(CSR(k))$.

Theorem 2.2. *The probability of the Approximate Correct Selection of the most promising Region in iteration k of NPHA, given by*

$$\Pr(ACSR(k)) = \sum_{\theta \in D_{\sigma_{j_a^k}(k)}} \prod_{\theta' \in H(k)\setminus\{\theta\}} \Phi\left(\frac{\hat{z}_{n_{\theta'}}(\theta') - \hat{z}_{n_\theta}(\theta)}{\sqrt{\frac{\sigma^2_{\theta'}}{n_{\theta'}} + \frac{\sigma^2_{\theta}}{n_{\theta}}}}\right), \tag{2.7}$$

is a lower-bound for the Correct Selection of the most promising Region. In mathematical terms this means that $\Pr(CSR(k)) \geq \Pr(ACSR(k))$.

Proof. Recall that we defined the set of all drawn designs from region $\sigma_{j_a^k}(k)$ in iteration k as $D_{\sigma_{j_a^k}(k)}$. Then it holds for $\Pr(CSR(k))$ that:

$$\Pr(CSR(k)) = \Pr\left(\text{“a drawn design from } \sigma_{j_a^k}(k) \text{ is indeed the best design”}\right)$$

using the definition of $CS(\theta,\, k)$ gives

$$= \Pr\left(\bigcup_{\theta \in D_{\sigma_{j_a^k}(k)}} CS(\theta,\, k)\right)$$

observing that the intersection between two events $CS(\theta,k)$ and $CS(\theta',k)$ is empty for $\theta \neq \theta'$, and expressing the probability in terms of the posterior distribution as shown in (2.6) leads to

$$= \sum_{\theta \in D_{\sigma_{j_a^k}(k)}} \Pr\left(CS(\theta,\, k)\right) = \sum_{\theta \in D_{\sigma_{j_a^k}(k)}} \Pr\left(\tilde{L}(\theta) < \min_{\theta' \in H(k)\setminus\{\theta\}} \tilde{L}(\theta')\right)$$

elaborating on Lemma 2.1 below yields

$$\geq \sum_{\theta \in D_{\sigma_{j_a^k}(k)}} \prod_{\theta' \in H(k)\setminus\{\theta\}} \Pr\left(\tilde{L}(\theta) < \tilde{L}(\theta')\right) \tag{2.8}$$

and using the distribution of the posterior in (2.5)

$$= \sum_{\theta \in D_{\sigma_{j_a^k}(k)}} \prod_{\theta' \in H(k)\setminus\{\theta\}} \Phi\left(\frac{\hat{z}_{n_{\theta'}}(\theta') - \hat{z}_{n_\theta}(\theta)}{\sqrt{\frac{\sigma^2_{\theta'}}{n_{\theta'}} + \frac{\sigma^2_{\theta}}{n_{\theta}}}}\right) = \Pr(ACSR(k))$$

which proves that $\Pr(CSR(k)) \geq \Pr(ACSR(k))$. Furthermore, note that from the definition of $\Pr(ACS(k))$ it follows that $\Pr(ACSR(k))$ can be rewritten as:

$$\Pr(ACSR(k)) = \Pr(ACS(k)) + \sum_{\theta \in D_{\sigma_{j_a^k}(k)}\setminus\{\theta_a(k)\}} \prod_{\theta' \in H(k)\setminus\{\theta\}} \Phi\left(\frac{\hat{z}_{n_{\theta'}}(\theta') - \hat{z}_{n_\theta}(\theta)}{\sqrt{\frac{\sigma^2_{\theta'}}{n_{\theta'}} + \frac{\sigma^2_{\theta}}{n_{\theta}}}}\right),$$

which shows that $\Pr(ACSR(k)) \geq \Pr(ACS(k))$ because the summation of a product of probabilities is greater than or equal to zero. □ □

Remark 2.1. *Instead of filling in the probability of approximate correct selection from Lemma 2.1 at* (2.8) *one can also insert an alternative probability of approximate correct selection based on the Bonferroni Inequalities, see also the development of the so-called APCS-B in [52]. This leads to the following alternative lower-bound for* $\Pr(CSR(k))$ *in iteration* k *of NPHA*

$$\sum_{\theta \in D(k)} \left(1 - \sum_{\theta' \in H(k) \setminus \{\theta\}} \Phi \left(\frac{\hat{z}_{n_\theta}(\theta) - \hat{z}_{n_{\theta'}}(\theta')}{\sqrt{\frac{\sigma^2_{\theta'}}{n_{\theta'}} + \frac{\sigma^2_{\theta}}{n_{\theta}}}} \right) \right),$$

which is also used in the analysis of Appendix A.

Again, in practice the possible unknown variance σ^2_θ can be estimated by an appropriate estimator and using the t-distribution in (2.7) instead of the normal distribution. The result of the following lemma is used in the proof of Theorem 2.2.

Lemma 2.1. *Suppose that* $X_1, X_2, \ldots, \ldots, X_n$ *are* n *mutually independent random variables. Then it holds that:*

$$\Pr(X_i < \min_{j \neq i} X_j) \geq \prod_{j \neq i} \Pr\left(X_i < X_j\right) . \tag{2.9}$$

Proof. From Lemma 4 in [51] it directly follows that for n mutually independent random variables, $Y_1, Y_2, \ldots, Y_n$, it holds that $\Pr(Y_i > \max_{j \neq i} Y_j) \geq \prod_{j \neq i} \Pr\left(Y_i > Y_j\right)$. By multiplying each each side with a minus this can be written as (note that this does not affect the independence of the random variables): $\Pr(-Y_i < \min_{j \neq i} -Y_j) \geq \prod_{j \neq i} \Pr\left(-Y_i < -Y_j\right)$. Now substitute in this last inequality $-Y_i$ with $X_i, \forall i \in \{1, 2, \ldots, n\}$. This shows relation (2.9) and ends the proof. □ □

Utilizing the philosophy of selecting the *most promising region* instead of finding the *best design* using $\Pr(ACSR(k))$ instead of $\Pr(ACS(k))$ gives us the following pseudo code for NPHA:

Pseudo code II for finding the most promising region with P^* certainty

Start: *Perform* n_0 *start simulations for each design.*

While $\Pr(ACSR(k)) < P^*$**:** *increase the computational budget with* Δ *and divide this extra computational budget among the different designs according to the optimal computational budget allocation rule from* (2.4) *until the stopping rule holds.*

Return: *The best found design.*

From Theorem 2.2 it follows that $\Pr(ACSR(k)) \geq \Pr(ACS(k))$ and thus the stopping criteria will be reached earlier. Consequently, the number of simulation runs needed will drop while we have still P^* certainty that we select the most promising

region. In the following subsection we will explore the theoretical improvements of using the accelerated stopping rule in the NPHA setting.

It is worth noting that the OCBA rule is established for optimizing the lower bound for the probability of correct selection, i.e., $\Pr(ACS(k))$, whereas we want to optimize $\Pr(ACSR(k))$. That OCBA is also appropriate in combination with the accelerated stopping rule can be seen as follows. For the OCBA rule in combination with the accelerating stopping rule to become inefficient, OCBA has to assign the entire additional simulation budget to one subregion. However, if all promising designs belong to a single subregion, the accelerating stopping rule will most likely already have detected this region as most promising, and thereby prevented the algorithm from making a call to OCBA. In any other case, due to the lack of pooled characteristic for a subregion, we are forced to consider the designs indiscriminately meaning that OCBA will ensure the highest value for $\Pr(ACSR(k))$ given a fixed amount of simulation budget since it will allocate the simulation budget to those designs that are critical in the process of identifying good designs. Identifying the good designs is also essential for finding the most promising region. We do not claim that there doesn't exists a smarter allocation rule than OCBA for the proposed stopping rule. But research in the direction of such an allocation rule faces a lot of difficulties, mostly because of the extra summation in the criteria of the new stopping rule. For more details we would like to refer to Appendix A. The analysis in Appendix A strengthens our belief that extra research effort and possible extra assumptions needed in finding such a smarter allocation rule are not worth the small amount of efficiency that it would possible realize. In light of our given arguments above, we believe that the combination of OCBA and the accelerated stopping rule is already highly efficient.

2.3.2 A Possible Theoretical Improvement

The expression of $\Pr(ACSR(k))$ efficiently handles the situation when designs from one region look the same in terms of their performance value. To illustrate this, suppose that we try to identify the most promising region by finding the best design with at least P^* certainty out of set $H(k) = \{\theta_1, \theta_2, \ldots, \theta_{N_k}\}$ using $\Pr(ACS(k))$ in iteration k. Here N_k stands for the number of drawn design in iteration k and the designs are drawn from the following subregions: $\sigma_1(k), \sigma_2(k), \ldots, \sigma_{M(\sigma)}(k)$ (possibly the designs are also drawn from the surrounding region $\sigma_{M_{\sigma(k)}+1}(k)$). Each design $\theta \in H(k)$ is simulated for n_θ number of times in iteration k. Furthermore, we suppose, without loss of generality, that

$$z(\theta_1) < z(\theta_2) < \cdots < z(\theta_{N_k})\,. \tag{2.10}$$

For illustration purposes we assume that $z(\theta_1) = z(\theta_2) - \epsilon,\ \epsilon \downarrow 0$, which means that θ_1 and θ_2 are nearly the same in terms of their performance value. It is thus also reasonable to assume that $\hat{z}_{n_{\theta_1}}(\theta_1) \approx \hat{z}_{n_{\theta_2}}(\theta_2)$ after n_{θ_1} and n_{θ_2} simulation runs respectively. From the ordering in (2.10) it is plausible that

$$\hat{z}_{n_{\theta_1}}(\theta_1) < \hat{z}_{n_{\theta_2}}(\theta_2) < \cdots < \hat{z}_{n_{\theta_{N_k}}}(\theta_{N_k})\,.$$

Lastly, for illustration purposes we assume that the designs θ_1 and θ_2 are from the same subregion $\sigma_1(k)$ (without loss of generality) so that it holds $\theta_1, \theta_2 \in \sigma_1(k)$.

Now consider the expression for $\Pr(ACS(k))$ in (2.3) of Theorem 2.1. The expression is a product form of, in our case, (N_k-1) cumulative distribution functions of the standard normal distribution. One of these product form terms is equal to:

$$\Phi\left(\frac{\hat{z}_{n_{\theta_2}}(\theta_2) - \hat{z}_{n_{\theta_1}}(\theta_1)}{\sqrt{\frac{\sigma^2_{\theta_2}}{n_{\theta_2}} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1}}}}\right)$$

and because $\hat{z}_{n_{\theta_1}}(\theta_1) \approx \hat{z}_{n_{\theta_2}}(\theta_2)$ this term is approximately equal to $\frac{1}{2}$. This means that for each design that is approximately equal to the best found design, the lower bound $\Pr(ACS(k))$ in (2.3) is halved. It thus will take a large amount of simulations until $\Pr(ACS(k)) \geq P^*$, when P^* is relatively close to 1.

In case that $\theta_1, \theta_2 \in \sigma_1(k)$ this deficiency can be circumvented using $\Pr(ACSR(k))$ instead of $\Pr(ACS(k))$. In our illustrating example this means that we also add the lower bound for the probability that θ_2 is the best design which we calculate as

$$\prod_{\theta \in H(k) \setminus \{\theta_2\}} \Pr\left(\tilde{L}(\theta) < \tilde{L}(\theta_2)\right).$$

From $\hat{z}_{n_{\theta_1}}(\theta_1) \approx \hat{z}_{n_{\theta_2}}(\theta_2)$ it follows that

$$\prod_{\theta \in H(k) \setminus \{\theta_2\}} \Pr\left(\tilde{L}(\theta) < \tilde{L}(\theta_2)\right) \approx \prod_{\theta \in H(k) \setminus \{\theta_1\}} \Pr\left(\tilde{L}(\theta) < \tilde{L}(\theta_1)\right),$$

which basically means that the halving of $\Pr(ACS(k))$ is compensated by adding about the same value to $\Pr(ACSR(k))$. Practically, this means that we don't put much effort in identifying the best design while it is already clear in an early stage which region to choose as the next most promising region.

The Nested Partitions method works most efficiently when the considered subregions contain designs that are similar in terms of their performance value ([169]). To achieve this, the chosen partitioning must result in subregions which tend to include designs with similar performance value. It is thus preferable that two designs from the same subregion look the same in terms of their expected costs. In the illustration above this means that the assumption that $z(\theta_1) = z(\theta_2) - \epsilon,\ \epsilon \downarrow 0$ and thus $\hat{z}_{n_{\theta_1}}(\theta_1) \approx \hat{z}_{n_{\theta_2}}(\theta_2)$ is reasonable when θ_1 and θ_2 are from the same subregion. As a result, when using NP efficiently, the probability that two or more designs from one region look the same in terms of their performance value is significant and thus the result of using $\Pr(ACSR(k))$ instead of $\Pr(ACS(k))$ will be significant.

The theoretical improvement can be made more explicit in the specific case for which it holds that

$$z(\theta_1) < z(\theta_2) \ll z(\theta_3) < \cdots < z(\theta_{N_k}), \tag{2.11}$$

where θ_1 and θ_2 are from the same subregion. Observe that it is the same case as the illustration above with the extension that $z(\theta_2) \ll z(\theta_3)$. Assume that after putting E amount of effort all means and variances are estimated correctly. Since at this stage it is clear that θ_1 and θ_2 are far better in performance than the rest of the designs it is reasonable to assume that $\Pr(ACSR(k)) \approx 1$. While for the value of $\Pr(ACS(k))$ it holds

$$\Pr(ACS(k)) = \prod_{\theta \neq \theta_1} \Pr(\bar{J}_{\theta_1} < \bar{J}_{\theta}) \approx \Pr(\bar{J}_{\theta_1} < \bar{J}_{\theta_2}), \tag{2.12}$$

where the approximation follows from the observation that it is most likely that $\theta \neq \theta_2 :\ \Pr(\bar{J}_{\theta_1} < \bar{J}_{\theta}) \approx 1$. From (2.12) it thus follows that

$$\Pr(ACS(k)) \approx \Phi\left(\frac{\epsilon}{\sqrt{\frac{\sigma^2_{\theta_2}}{n_{\theta_2}} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1}}}}\right). \tag{2.13}$$

Take a q for which it holds that $\Phi(q) \approx 1$, then it follows from (2.13) that if we want $\Pr(ACS(k)) \approx 1$, then it should hold that, after performing some extra simulation x and y for θ_1 and θ_2, respectively,

$$\frac{\epsilon}{\sqrt{\frac{\sigma^2_{\theta_2}}{n_{\theta_2}+y} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1}+x}}} \geq q \ \Leftrightarrow\ \frac{\epsilon^2}{\frac{\sigma^2_{\theta_2}}{n_{\theta_2}+y} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1}+x}} \geq q^2 \ \Leftrightarrow\ \frac{\sigma^2_{\theta_2}}{n_{\theta_2}+y} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1}+x} \leq \frac{\epsilon^2}{q^2}. \tag{2.14}$$

Define $x + y = \delta$, where δ is the total amount of extra simulations needed so that $\Pr(ACS(k)) \approx 1$, then it follows from [53] in the case of 2 designs that $x = \frac{\sigma_{\theta_1}}{\sigma_{\theta_2}} y$ is optimal. Considering the sketched scenario after E amount of effort as a 2 design case of the budget allocation problem is realistic since at the considered stage it is most likely that the relation (2.11) is identified. In other words, in identifying the best design, θ_1 and θ_2 will be the only serious candidates left to whom OCBA will allocate simulation budget to. So it follows that

$$x = \frac{\delta}{1 + \frac{\sigma_{\theta_2}}{\sigma_{\theta_1}}} \quad \text{and} \quad y = \frac{\delta}{1 + \frac{\sigma_{\theta_1}}{\sigma_{\theta_2}}}. \tag{2.15}$$

Inserting (2.15) into the RHS of (2.14) we obtain that

$$(2.14) \ \Leftrightarrow\ \frac{\sigma^2_{\theta_2}}{n_{\theta_2} + \frac{\delta}{1+\frac{\sigma_{\theta_1}}{\sigma_{\theta_2}}}} + \frac{\sigma^2_{\theta_1}}{n_{\theta_1} + \frac{\delta}{1+\frac{\sigma_{\theta_2}}{\sigma_{\theta_1}}}} \leq \frac{\epsilon^2}{q^2}.$$

If we assume that it also holds that $\frac{n_{\theta_1}}{n_{\theta_2}} = \frac{\sigma_{\theta_1}}{\sigma_{\theta_2}}$ (i.e., the relation that follows from [53] in case of 2 designs also holds for the first simulations) it follows from the last result that (by multiplying the first term under and above with $1 + \frac{\sigma_{\theta_1}}{\sigma_{\theta_2}}$ and the second term under and above with $1 + \frac{\sigma_{\theta_2}}{\sigma_{\theta_1}}$)

$$(2.14) \ \Leftrightarrow\ \frac{\sigma^2_{\theta_2}(1 + \frac{\sigma_{\theta_1}}{\sigma_{\theta_2}})}{n_{\theta_2} + n_{\theta_1} + \delta} + \frac{\sigma^2_{\theta_1}(1 + \frac{\sigma_{\theta_2}}{\sigma_{\theta_1}})}{n_{\theta_1} + n_{\theta_2} + \delta} \leq \frac{\epsilon^2}{q^2}$$

which can be rewritten as

$$(2.14) \ \Leftrightarrow\ \delta \geq q^2(\sigma_{\theta_1} + \sigma_{\theta_2})^2/\epsilon^2 - n_{\theta_1} - n_{\theta_2},$$

which shows that the total extra amount of simulation budget assigned to designs θ_1 and θ_2, i.e., $x + y = \delta$, in order to ensure that $\Pr(ACS(k))$ approximately equals $\Pr(ACSR(k)) \approx 1$ is of order $O(1/\epsilon^2)$. This illustrates that the total extra amount of simulation budget needed to ensure that $\Pr(ACS(k))$ approximately equals $\Pr(ACSR(k)) \approx 1$ can be significant for $\epsilon > 0$ small. In other words, $\Pr(ACSR(k))$ prevents putting much effort in identifying the best design while it is already clear in an early stage which region to choose as the next most promising region. In the following subsections we will explore the numerical improvements of using the accelerated stopping rule in the NPHA setting.

2.3.3 Numerical Experiments 1: an Academic Problem

In this subsection numerical experiments can be found that test the performance of both stopping rule versions in the setting of NPHA. To this end an academic problem is considered to be described in the following. Two types of numerical experiments are performed in the following, i.e., numerical experiments 1.a and 1.b, respectively. The first experiment performs macro replications to test both stopping rules in a scenario which is likely to occur when using the Nested Partition method, while the second test emphasis on testing the proposed stopping rule in a disadvantageous scenario. Throughout this subsection NPHA 1 will be used to indicate NPHA with the original stopping rule, while NPHA 2 indicates the NPHA version with the proposed new stopping rule.

Problem Setting

To ensure a fair, controllable and executable comparison we consider the following academic problem that represents the essence of the problem faced in each iteration of the NPHA. Suppose that the considered region in an iteration of NPHA consist of 100 designs numbered from 1 to 100, where the i-th design has an expected cost of $z(\{i\}) = E[L(\{i\})] = i$, with $L(\{i\}) \sim \mathcal{N}(i, \sigma^2)$. In our case we have a minimization objective so it is clear that design 1 is optimal. Next, the considered region is logically partitioned into M partitions (or subregions) where partition 1 contains the designs from 1 to $100/M$, partition 2 contains the designs from $100/M+1$ to $2(100/M)$, etcetera. So in case $M = 4$, the partitions are $\{1, \ldots, 25\}$, $\{26, \ldots, 50\}$, $\{51, \ldots, 75\}$ and $\{76, \ldots, 100\}$. Once the partitions are obtained, we randomly draw without replacement h designs from every partition and the goal is to find with P^* certainty the partition which contains the design with best performance while using the least amount of simulation budget possible (note that for this experiment the best partition is known to be the first partition).

Numerical Experiments 1.a

To assess the performance of both stopping rules in the setting of NPHA, we ran 100 independent macro replications where in each replication we randomly draw a set of designs. For each drawn set of designs, both versions of the stopping rules together with OCBA are used to determine which partition is most promising, i.e., contains the best design. Because of the randomness when evaluating the designs we repeat this last step 100 times and average over the performance measures to obtain one replication result. Before evaluating both stopping rules for OCBA for a drawn set of design we first reset the random number generator in MATLAB for a fair comparison. In this subsection we consider two main performance measures which are obtained by averaging over the replication results, namely the difference in the realized correct decision percentage between NPHA 1 and NPHA 2 (which can be measured since we know the true most promising region of this academic problem in advance) and the percentage drop with respect to the mean number of simulations used to determine the most promising region with P^* certainty.

For the numerical experiments we take $\sigma = 100$, $n_0 = 25$ and $\Delta = 5$ (unless stated otherwise). Observe that n_0 is quite large but this is to prevent premature termination

h	$P^* = 60\%$ (in %)	$P^* = 65\%$ (in %)	$P^* = 70\%$ (in %)
2	-0.38 [-0.54 -0.22]; 16.16	-0.10 [-0.21 0.01]; 11.67	-0.07 [-0.13 -0.01]; 18.49
3	-0.18 [-0.27 -0.09]; 26.58	-0.03 [-0.07 0.01]; 21.11	0 [0 0]; 27.81
4	-0.02 [-0.08 0.04]; 31.51	-0.02 [-0.06 0.02]; 29.31	0 [0 0]; 35.05
5	0.00 [-0.03 0.03]; 32.32	0 [0 0]; 29.95	0 [0 0]; 38.35

Tab. 2.1: The results for $M = 2$. Entry 'x [LCI RCI]; y' reads as x is the average difference in the real correct decision percentage (the real correct decision percentage of NPHA 2 minus that of NPHA 1) together with its 95% confidence interval bounds and y is the percentage drop of the mean number of simulations used for NPHA 2 instead of NPHA 1.

of the comparison because of poor initial estimates of the mean and variances. In the Tables 2.1, 2.2 and 2.3 the results can be found for M equal to 2, 3 and 4 respectively for different values of h and P^*. The values of P^* are chosen relatively small in order to make the analysis executable and furthermore to make the difference in percentage of true correct selection more clear, because a large P^* results in large real percentage of correct selection (near 1) meaning that there is almost no difference to observe between both stopping rules. The tables give information about the two values we are interested in when using NPHA 2 instead of NPHA 1: the average difference in realized percentage that we choose the correct partition as most promising (i.e., partition 1 in advance) and the percentage drop of the mean number of simulations needed for P^* certainty. Recall that the stopping rule in NPHA 1 guarantees to pick *the best design* with P^* certainty while NPHA 2 guarantees to pick *the region* that contains the best design with P^* certainty, so both methods have a realized percentage of choosing the correct partition of at least P^*. But the values of these realized percentages may differ, note that the realized percentage of NPHA 1 will in general not be worse than that of NPHA 2 (which can be observed from the numerical results because most mean differences are negative).

From the numerical results of 1.a it follows that the difference of the true correct selection percentages for both stopping rules are significant smaller than 0.5% (i.e., half percent). This means that the true correct selection percentages of NPHA 1 and NPHA 2 lie close to each other. Furthermore, the results show that NPHA 2 requires in general at least around 20% less simulations compared to NPHA 1 and in some cases even more than 40% less simulations (i.e., when the values of h, M and P^* increase). In conclusion, the experiments showed that NPHA 2 has approximately the same percentage of correct decision as NPHA 1 has (in worst case the difference is still smaller than a half percent), while it needs relatively less simulations compared to NPHA 1.

Numerical Experiments 1.b

The goal of this second experiment is to test both stopping rules on a scenario that looks disadvantageous in advance for the proposed stopping rule. The considered scenario occurs when the best design is (without losing on generality (w.l.o.g.)) from partition 1 which furthermore consists of designs which are clearly relatively less in performance than the other designs. Suppose there is another partition, say (w.l.o.g.) the second partition, which consists of designs which are only a bit less in performance

h	$P^* = 60\%$ (in %)	$P^* = 65\%$ (in %)	$P^* = 70\%$ (in %)
2	-0.26 [-0.42 -0.10]; 11.15	-0.16 [-0.29 -0.03]; 12.86	-0.29 [-0.42 -0.16]; 20.66
3	-0.16 [-0.26 -0.06]; 20.78	-0.09 [-0.17 -0.01]; 26.12	-0.1 [-0.17 -0.03]; 37.04
4	-0.06 [-0.12 -0.00]; 26.30	-0.01 [-0.03 0.01]; 31.30	-0.01 [-0.03 0.01]; 40.63
5	-0.01 [-0.04 0.02]; 31.18	0 [-0.03 0.03]; 34.11	-0.01 [-0.03 0.01]; 41.33

Tab. 2.2: The results for $M = 3$. Entry 'x [LCI RCI]; y' reads as x is the average difference in the real correct decision percentage (the real correct decision percentage of NPHA 2 minus that of NPHA 1) together with its 95% confidence interval bounds and y is the percentage drop of the mean number of simulations used for NPHA 2 instead of NPHA 1.

h	$P^* = 60\%$ (in %)	$P^* = 65\%$ (in %)	$P^* = 70\%$ (in %)
2	-0.4 [-0.57 -0.23]; 11.03	-0.33 [-0.47 -0.19]; 14.48	-0.31 [-0.45 -0.17]; 22.64
3	-0.06 [-0.15 0.03]; 19.41	-0.14 [-0.23 -0.05]; 25.47	-0.08 [-0.14 -0.02]; 36.54
4	-0.04 [-0.12 0.04]; 24.97	-0.06 [-0.11 -0.01]; 31.91	0.01 [-0.01 0.03]; 41.22
5	-0.01 [-0.03 0.01]; 26.85	-0.01 [-0.03 0.01]; 34.41	0 [0 0]; 37.08

Tab. 2.3: The results for $M = 4$. Entry 'x [LCI RCI]; y' reads as x is the average difference in the real correct decision percentage (the real correct decision percentage of NPHA 2 minus that of NPHA 1) together with its 95% confidence interval bounds and y is the percentage drop of the mean number of simulations used for NPHA 2 instead of NPHA 1.

than the overall best design from partition 1. Observe that this scenario is not likely to occur often when using the Nested Partition method since the partitions should cover design which look the same in terms of performance. The considered instance for this numerical experiment consists of $M = 4$ partitions each consisting of $h = 4$ chosen (i.e., 'drawn' in the above setting) designs, in particular: $\{0.9, 2.5, 2.6, 2.7\}$, $\{1, 1.1, 1.2, 1.3\}$, $\{1.4, 1.5, 1.6, 1.7\}$ and $\{2.1, 2.2, 2.3, 2.4\}$ respectively. Furthermore, we have $P^* = 80\%$, $\sigma = 0.5$, $n_0 = 7$ and $\Delta = 5$. For the chosen designs we repeat both OCBA procedures 500 times (before each OCBA procedure we reset the seed for a fair comparison) and then average over the results to obtain the true correct selection percentage of both methods and the mean number of simulation used for both methods.

The true correct selection percentage of NPHA 1 is 86.8% and for NPHA 2 it is 84.6%, while the mean number of simulations used are 209.92 and 205.04 respectively. From these results it can be concluded that the real percentages (observe that both lie above $P^* = 80\%$) of the correct decision using NPHA 1 is 1.2% larger than for NPHA 2. This is mostly because NPHA 2 concludes in some cases earlier than NPHA 1 that partition 2 must contain the best design and misses the fact that design with performance 0.9 from partition 1 is actually the best design. Also the percentage drop of the mean number of simulations used, which is 2.32%, is not large since the scenario that was sketched in the main article for which the proposed stopping rule is most beneficial doesn't arise. It can thus be argued that the performance of NPHA 2 is less in this kind of worst case scenarios in terms of realized correct decision percentage, although choosing partition 2 is not too bad realizing that the performances of designs in partition

2 do not differ that much from the best design. This can be further illustrated by making the performances of the best design in partition 2 larger, e.g., 1.8 so that partition 2 becomes $\{1.8, 1.1, 1.2, 1.3\}$. In this case the true correct selection percentage of NPHA 1 is 95.0% and for NPHA 2 it is 94.2%, while the mean number of simulations used are 150.26 and 147.52, respectively. It shows that the real correct decision percentage difference between NPHA 1 and NPHA 2 (which is equal to 0.8%) already decreases compared to the previous case, since NPHA 2 is less likely to overlook the best design since its performance is more distinguishing. Furthermore, when choosing a larger P^*, say $P^* = 90\%$ (starting from the first scenario in this subsection) the change that NPHA 2 misses the best design also becomes smaller. For $P^* = 90\%$ the true correct selection percentage of NPHA 1 is 93.2% and for NPHA 2 it is 93.0%, while the mean number of simulations used are 305.51 and 304.33, respectively. In this case the percentage difference is 0.2% which is significant smaller than the 1.2% that followed from the first case in this subsection.

The results of the numerical experiments 1.b show that the NPHA 2 loses a small amount of efficiency compared to NPHA 1 when choosing a scenario where NPHA 2 is in disadvantage. But this minor inefficiency only occurs in scenarios where the consequences are small as the numerical experiments showed. Also, the disadvantageous case for NPHA 2 is unlikely to happen when using the Nested Partitions method as prescribed.

2.3.4 Numerical Experiments 2: Buffer Allocation Problem

This subsection performs a comparison study for the Buffer Allocation Problem (BAP), which may be considered as a more realistic problem than the purely academic problem from the previous subsection. In this numerical experiment the two NPHA settings will be compared, i.e., with the original stopping rule and with the accelerated stopping rule. In the following, the Buffer Allocation Problem is first introduced and afterwards the numerical experiment is described and results are to be presented.

Problem Description

Consider a tandem system of N single server first come first served (FCFS) queues numbered from 1 to N as depicted in Figure 2.2. Products arrive from the outside at

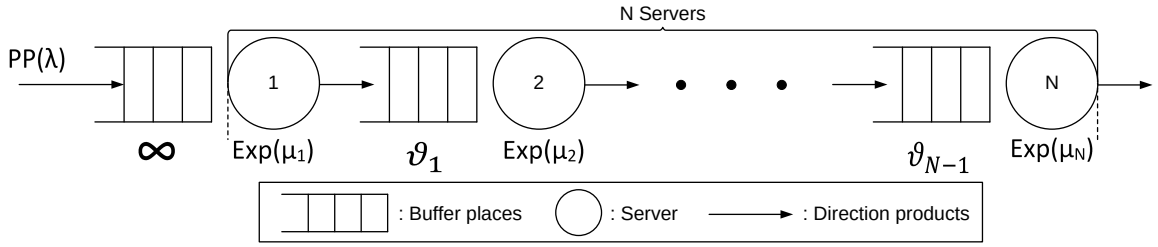

Fig. 2.2: The queuing system for the Buffer Allocation Problem

server 1 according to a Poisson stream with intensity λ (in short PP(λ)). After the product received service at server 1 it moves to server 2 and so forth. Service times at the servers are exponentially distributed (Exp($\cdot$)). After the service of the product

is completed at server N, the product leaves the system. The buffer capacity of the first server is infinite and the allocated number of buffers in front of the other $(N-1)$ servers must sum to some given constant $B \in \mathbb{N}$. We assume manufacturing blocking, i.e., if a product finds upon service completion at server i no available free buffer place at sever $i+1$, the product waits at server i until a buffer place becomes available at server $i+1$. During this time, no service at server i is possible and we say that server i is blocked.

The objective of the problem is to allocate the available B buffers among the $(N-1)$ servers so that the long-term mean sojourn time of products, denoted by $z(\theta)$, is minimized. Because there is no cost attached to using the available buffers it is optimal to use all the B available buffers. The design space of the Buffer Allocation Problem can be defined as

$$\Theta = \left\{ \theta = (\vartheta_1, \vartheta_2, \ldots, \vartheta_{N-1}) \in \mathbb{N}^{N-1} \ : \ \sum_{i=1}^{N-1} \vartheta_i = B \right\},$$

where $\vartheta_i,\ i \in \{1, 2, \ldots, N-1\}$, denotes the number of buffers places in front of server $i+1$. Note that the number of designs (i.e., buffer allocations) under this statement is equal to $(B + (N-2))!B!/(N-2)!$. Since there is no analytical expression known for $z(\theta)$ we have to resort to discrete-event simulation for evaluating $z(\theta)$.

Experimental Description and Results

For illustration purposes we choose $N = 10$ and $B = 17$ in the numerical experiments. The arrival rate $\lambda = 0.52$ and service rates $\mu_i = 1$, for $1 \leq i \leq N$. This results in 1.081.575 different buffer allocations.

Again, NPHA 1 represents NPHA with the standard stopping rule, and NPHA 2 denotes NPHA with the accelerated stopping rule. We solve the Buffer Allocation Problem using NPHA 1. At the same time that NPHA 1 is running the accelerated stopping rule measure, i.e., $\Pr(ACSR(k))$, from NPHA 2 is tracked. When $\Pr(ACSR(k)) \geq P^*$ and $\Pr(ACS(k)) < P^*$ we save the number of simulations used until that time and we also save the most promising index at that moment. After that we continue NPHA 1 until also $\Pr(ACS(k)) \geq P^*$. Note that NPHA 2 will always use less simulations in an iteration compared to NPHA 1 since $\Pr(ACSR(k)) \geq \Pr(ACS(k))$. At this moment we calculate the difference in the number of simulations used between both algorithms and we verify if both algorithms result in the same most promising region. The algorithm stops when a design at maximal depth is found (i.e., the point at which the considered region can't be partitioned any further). At this point we average over the found numerical results. Using this specific experiment setting the aim is to achieve a fair comparison. Because NPHA is of a stochastic character we are also going to repeat the described experiment (solving the Buffer Allocation Problem with NPHA 1 and NPHA 2 until maximal depth is reached) above for 41 times. Each time we use a different seed in the random number generator of MATLAB (we use s = RandStream('mcg16807','SEED',i); where i is the specific seed used and activated via RandStream.setDefaultStream(s);).

NPHA 1 and NPHA 2 use the same partitioning scheme to be introduced presently. Servers are considered from left to right. The number of available buffers left are partitioned into M disjoint subsets. For example, suppose that a server is considered

for which $\hat{B}$ buffers are still available after assigning buffers to the servers in front of this specific server. Then the number of available buffers can be partitioned into the following M subsets $\{0,\ldots,B_1\}$, $\{B_1+1,\ldots,B_2\}$, $\ldots$, $\{B_{M-1}+1,\ldots,\hat{B}\}$, where $B_i = \lfloor\delta\rfloor i$ with $\delta := (\hat{B}+1)/M$. From each subset buffer allocations are drawn. When the considered most promising buffer set becomes too small to partition it in M subregions (i.e., when $\delta \leq 1$), all the possible numbers of buffers from the buffer subset are considered as subregions. This process is logically repeated until all servers are considered. For the sampling from the subregions a weighted sampling scheme is used which puts more weight on the designs that divide the number of buffers evenly among the servers. Furthermore, we ensure that no same designs are sampled. The surrounding region is defined as the most promising region of the previous iteration (in case the current most promising region is the entire design space Θ it is empty).

The NPHA specific values are chosen as follows. In each step, we partition the most promising set further into $M = 2$ partitions. Per partition, there are $h = 4$ designs drawn. The confidence level of correct selections is set to $P^* = 70\%$ and the additional simulation budget for OCBA is $\Delta = 100$ and $n_0 = 10$.

In order to obtain an estimate for $z(\theta)$, $\theta \in \Theta$, a warm-up phase of 10^4 sojourn times are simulated. Once the system has reached steady state, a batch means strategy is used for the simulations. One simulation realization is the average over one the batch size containing 50 sojourn time realizations. To synchronize the simulation for the selected designs we use the standard clock method [180].

The results of the 41 experiments can be found in Figure 2.3. The figure plots two percentages against the seeds used in MATLAB. Each seed on the x-axis corresponds to an experiment as described above. The first plot shows the percentage drop in the number of simulations needed when the accelerated stopping rule is used. The second plot gives the percentage of the number of times that NPHA 1 resulted in the same decision for the most promising region as NPHA 2.

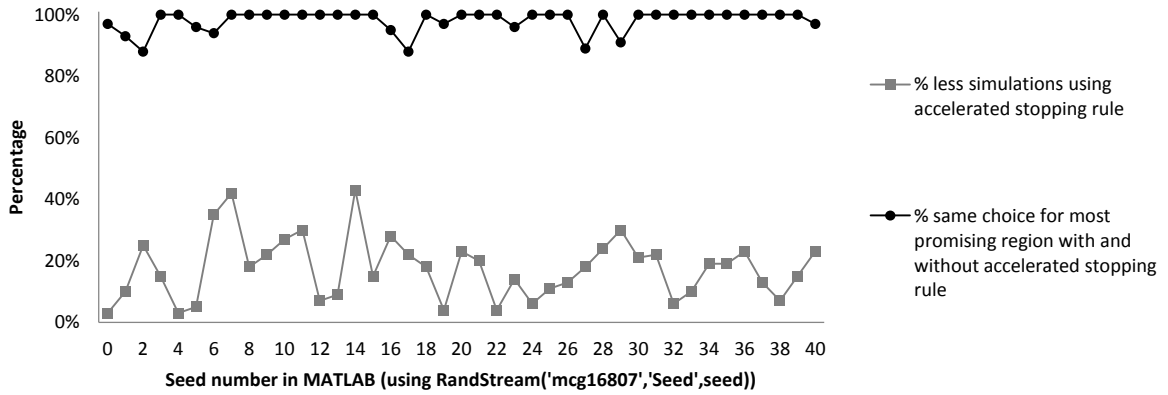

Fig. 2.3: Results of solving BAP using NPHA with (i.e., NPHA 2) and without (i.e., NPHA 1) the accelerated stopping rule for different seeds.

From the experimental results it can be concluded that NPHA 2 on average over the 41 experiments chooses 98% of the time the same most promising region. This shows that the accelerated stopping rule does not have a large impact on the choice of the most promising region during the progress of NPHA. Furthermore, from the results it follows that on average NPHA 2 needs 18% less simulations compared to NPHA 1,

which is considerable. In this instance, M (i.e., the number of partitions in NP) and the number of designs drawn from each region is rather small. When these numbers become larger, NPHA with the alternative stopping rule can be expected to use an even smaller number of simulations.

2.4 Conclusions

In this chapter we introduced an accelerated stopping rule for NPHA, [169]. The main idea of the accelerated stopping rule is that the additional terms to be calculated for the accelerated stopping rule are computationally less expensive than performing additional simulation runs. Numerical experiments showed that the improvement is considerable and that the accelerated stopping rule has little effect on the choice for the most promising region compared to the original stopping rule in NPHA. Furthermore, this chapter showed that the theoretical improvement that can be made in particular cases is significant. Overall, using the accelerated stopping rule in the NPHA setting is an elegant way to incorporate the philosophy of searching for the most promising *region* instead of finding the best *design* in NPHA. A topic of further research is to address the case when two designs from two different regions are close in terms of their performance value. In addition, it is interesting to investigate how NPHA together with the accelerated stopping performs compared to the NPHA version that also incorporates past information in the iteration-process.