

SEQUENTIAL SCREENING: A BAYESIAN DYNAMIC PROGRAMMING ANALYSIS OF OPTIMAL GROUP-SPLITTING

Peter I. Frazier

Cornell University
Ithaca, NY 14850, USA

Bruno Jedynak
Li Chen

Johns Hopkins University
3400 North Charles St.
Baltimore, MD 21218, USA

ABSTRACT

Sequential screening is the problem of allocating simulation effort to identify those input factors that have an important effect on a simulation's output. In this problem, sophisticated algorithms can be substantially more efficient than simulating one factor at a time. We consider this problem in a Bayesian framework, in which each factor is important independently and with a known probability. We use dynamic programming to compute the Bayes-optimal method for splitting factors among groups within a sequential bifurcation procedure (Bettonvil & Kleijnen 1997). We assume importance can be tested without error. Numerical experiments suggest that existing group-splitting rules are optimal, or close to optimal, when factors have homogeneous importance probability, but that substantial gains are possible when factors have heterogeneous probability of importance.

1 INTRODUCTION

When faced with complex stochastic simulations involving hundreds or thousands of input factors, it is common that only a small number of these input factors have a important effect on the distribution of the output, while other input factors have little or no effect. The task of identifying these important factors is called *screening*. Screening is important both for the insight it offers into the simulator's behavior, and because subsequent analysis is easier if only a small number of input factors need be considered.

One simple method for screening is to perform simulation experiments that vary one input factor at a time. For problems with many factors, however, this simple method is inefficient, and screening can be performed with less simulation effort using a sequential method that chooses which simulation experiment to perform next based on the results of previous experiments. One prominent sequential method, introduced by Bettonvil and Kleijnen (1997), is *sequential bifurcation*.

Sequential bifurcation operates by performing simulations on groups of factors, which allows quick elimination of groups that do not contain important factors, without simulating each unimportant factor individually. Groups of factors identified as containing at least one important factor are divided into two sub-groups for subsequent testing. Groups that do not contain an important factor, but that do contain several factors with small but non-zero effects that together mimic an important factor, may also be divided and subsequently tested.

One important decision in the design of any sequential bifurcation algorithm is exactly how each group should be divided. One choice is to split sub-groups in half (Wan, Ankenman, and Nelson 2006; Cheng 1997), while another is to choose the first subgroup to contain a number of factors that is a power of 2 (Bettonvil and Kleijnen 1997; Kleijnen 2009). Kleijnen (2009) and Bettonvil (1990) further recommend group-splitting rules that encourage clustering factors with similar probabilities of importance.

The relative quality of these group-splitting rules, and how they compare with the best possible group-splitting rule, has not been studied in a formal way. In this paper, we study the question of how groups

should be split into subgroups to obtain *optimal* performance, in terms of a measure of the total simulation effort required. We work within a Bayesian framework, and the notion of optimality we use is *average-case optimality* with respect to a prior distribution.

For tractability, we make two assumptions. First, we assume that the total expected simulation effort is proportional to the number of input points at which simulations are performed. Second, we assume that the hypothesis tests determining importance make no errors. This second assumption holds when simulation is deterministic, and for stochastic simulation is an approximation that is most reasonable when hypothesis tests are performed with a large amount of simulation effort and a high degree of accuracy. This assumption parallels the assumption of negligible error made in Bettonvil and Kleijnen (1997). While this second assumption may misrepresent the final number of incorrect determinations under stochastic simulation, our focus is on how group splitting affects the number of input points at which simulation is performed. The number of incorrect determinations may be controlled separately through the choice of hypothesis testing procedure, and the simulation effort used at each input point.

Sequential bifurcation was introduced by Bettonvil and Kleijnen (1997) and subsequently extended and modified by a number of authors. Much of the work on sequential bifurcation is surveyed in Kleijnen (2009) and an introduction to the sequential bifurcation algorithm may be found in the textbook Kleijnen (2007). We provide a brief overview here.

When sequential bifurcation was introduced in Bettonvil and Kleijnen (1997), it assumed negligible random errors in the simulation response. It was extended to allow for more substantial random errors in Cheng (1997), although no formal statistical guarantee on solution quality was provided. Later, Wan, Ankenman, and Nelson (2006) provided a modified version of sequential bifurcation with statistical guarantees on the number of incorrect determinations. Although much of the work on sequential bifurcation assumes a linear model with no interactions for the responses, the original work Bettonvil and Kleijnen (1997) considered interactions between pairs of factors, and Bettonvil, Kleijnen, and Persson (2006), Yaesoubi, Roberts, and Klein (2010) and Wan, Ankenman, and Nelson (2010) discuss interactions and other more general statistical models. Shi, Kleijnen, and Liu (2012) extends sequential bifurcation to multiple responses and utilizes an improved hypothesis testing procedure.

To find the average-case optimal group-splitting strategy, we use methods from dynamic programming and Markov decision processes (see, e.g., Dynkin and Yushkevich (1979), Powell (2007)). This use of dynamic programming to design algorithms with optimal or near-optimal average-case performance has been considered for other problems in simulation, especially Bayesian ranking and selection (Chick and Gans 2009; Chick, Branke, and Schmidt 2010; Frazier, Powell, and Dayanik 2008; Frazier and Powell 2008; Chick and Frazier 2012), but also optimization via simulation as studied within Bayesian global optimization (Brochu, Cora, and de Freitas 2009). In such problems, finding an average-case optimal algorithm generally requires solving a partially observable Markov decision process (POMDP). In most cases, the state space of this POMDP is very large, making it computationally infeasible to obtain exact solutions (Lovejoy 1991; Powell and Ryzhov 2012).

Although the state space of the dynamic program that we consider is high-dimensional in its original form, and thus apparently difficult to solve because of the curse of dimensionality (Powell 2007), we are able to exploit structure (in the form of conditional independence) to transform to a low-dimensional state space, resulting in an efficient computational scheme for solving the dynamic program. Thus, group splitting for sequential bifurcation joins a small number of other problems in simulation for which the Bayes-optimal fully sequential algorithm can be computed explicitly. Other examples include ranking and selection with one known and one unknown alternative (Chick and Frazier 2012), multiple comparisons with a known standard (Xie and Frazier 2011), entropy-based search (Jedynak, Frazier, and Sznitman 2012), and stochastic root-finding (Waeber, Frazier, and Henderson 2011).

In Section 2 we define the screening problem, and the sequential bifurcation algorithm. In Section 3 we define the group splitting problem as a Markov decision process. In Section 4 we present our optimality

results: Section 4.1 considers factors with homogeneous probability of importance; and Section 4.2 considers factors with two different probabilities of importance. Section 5 concludes.

2 SCREENING AND SEQUENTIAL BIFURCATION

In this section, we provide an overview of the problem of sequential screening for stochastic simulation, and the class of sequential bifurcation (SB) methods.

We have a stochastic simulator that takes an input vector $\vec{x} \in \mathbb{R}^K$, where $K > 1$, and produces stochastic output $y(\vec{x})$. As is common in the literature on screening, we assume a linear model with main effects and no interactions. In this model, close to some central point that we take to be 0 without loss of generality, the distribution of $y(\vec{x})$ is modeled as

$$y(\vec{x}) = \beta_0 + \sum_{k=1}^K \beta_k x_k + \varepsilon(\vec{x}),$$

where $\beta_0, \beta_1, \dots, \beta_K$ are unknown factor coefficients, and the distribution of $\varepsilon(\vec{x})$ is normal with mean 0 and an unknown variance $\sigma^2(\vec{x})$. We assume that the $\varepsilon(\vec{x})$ are independent, which can be assured by driving the simulation with independent pseudo random number streams, and which precludes the use of common random numbers considered by some work on screening. We also assume, as in other literature on screening (Bettonvil and Kleijnen 1997; Wan, Ankenman, and Nelson 2006), that the sign of each β_k is known. With this assumption, we have without loss of generality that $\beta_k \geq 0$ for each $k > 0$. If this is not met, and β_k is instead non-positive, we may transform the input vector by replacing x_k by $-x_k$ to obtain an equivalent problem in which $\beta_k \geq 0$.

The component directions $k = 1, \dots, K$ are called *factors*, and our goal in screening is to determine which of these factors have a non-zero value for β_k . Those factors are called *important*, and the remaining factors with $\beta_k = 0$ are called *unimportant*. Although for simplicity we take all factors with strictly positive β_k to be important, some other work on screening (e.g., Cheng (1997)) assumes another formulation in which there are parameters $\delta \geq 0$ and $a > 0$, and factors with $\beta_k < \delta$ are unimportant; factors with $\beta_k > \delta + a$ are important, and factors with $\beta_k \in [\delta, \delta + a]$ are of ambiguous importance, and may be correctly declared to either be important or unimportant.

In screening, the experimenter first chooses low and high levels for each factor. The low level is set to 0, while the high level is set to a strictly positive number w_k . Often, w_k is equal to 1, but other values are possible (Wan, Ankenman, and Nelson 2006). The experimenter then performs a sequence of experiments where each component of \vec{x} is either low or high. SB assumes further that the input vector \vec{x} for each simulation experiment is equal to $\vec{x}(k) = [w_1, \dots, w_k, 0, \dots, 0]$ for some $k \in \{0, \dots, K\}$, where $\vec{x}(0) = [0, \dots, 0]$.

Simulation output from experiments run at $\vec{x}(k)$ have a mean value of $E[y(\vec{x}(k))] = s(k) = \beta_0 + \sum_{k'=1}^k w_{k'} \beta_{k'}$, and averaging this simulation output provides an estimate of this quantity. Furthermore, given two factors $1 \leq k_1 < k_2$, one can estimate $s(k_2) - s(k_1 - 1) = \sum_{k'=k_1}^{k_2} w_{k'} \beta_{k'}$ with the difference between simulation output at $\vec{x}(k_1 - 1)$ and $\vec{x}(k_2)$. SB algorithms use these estimates to find the important factors.

The original SB method (Bettonvil and Kleijnen 1997), as well as a number of extensions and modifications (Cheng 1997; Wan, Ankenman, and Nelson 2006), follow the algorithmic structure given below in Figure 1. This algorithmic structure stores and evaluates *groups* of numerically contiguous factors, $\{k_1, \dots, k_2\}$. Each group can be tested to see whether it contains important factors by performing a hypothesis test for whether $s(k_2) - s(k_1 - 1)$ is strictly positive. If $s(k_2) - s(k_1 - 1) > 0$, then the group contains an important factor. Conversely, if $s(k_2) - s(k_1 - 1) = 0$, then it does not. Figure 1 does not specify a hypothesis testing procedure for performing these individual hypothesis tests, as versions of SB appearing in Bettonvil and Kleijnen (1997), Cheng (1997), Wan, Ankenman, and Nelson (2006) use different hypothesis testing procedures, all within the same basic structure.

1. **Determine whether there are important factors in $\{1, \dots, K\}$**
 Test the hypothesis $H_0 : s(K) = s(0)$ versus the hypothesis $H_1 : s(K) > s(0)$. This requires obtaining simulation replications of $y(\vec{x})$ at $\vec{x} = \vec{x}(0)$ and $\vec{x} = \vec{x}(K)$.
 - (a) If we accept H_0 (there are no important factors in $1, \dots, K$) then stop the screening procedure, and output that there are no important factors.
 - (b) If we accept H_1 (there is at least one important factor in $1, \dots, K$) then create an empty stack, and push the group $\{1, \dots, K\}$ onto the stack.
2. Until the stack is empty, do the following:
 - (a) **Pop a group off the stack**
 Pop a group off the top of the stack. This group is of the form $\{k_1, \dots, k_2\}$ for some k_1, k_2 with $1 \leq k_1 \leq k_2 \leq K$. This group has been determined to contain an important factor by a previous hypothesis test, and simulation samples have already been obtained at $y(\vec{x}(k_1 - 1))$ and $y(\vec{x}(k_2))$.
 - (b) **Split the group**
 Using the group-splitting rule, choose a value k satisfying $k_1 \leq k < k_2$. We will split the group $\{k_1, \dots, k_2\}$ into two groups of factors, $\{k_1, \dots, k\}$ and $\{k + 1, \dots, k_2\}$, and test each of them individually to determine whether they contain important factors. This pair of tests, performed below in Steps 2c and 2d requires obtaining simulation replications of $y(\vec{x})$ at $\vec{x} = \vec{x}(k)$. It also reuses simulation replications previously obtained at $\vec{x} = \vec{x}(k_1)$ and $\vec{x} = \vec{x}(k_2)$, and may run additional replications at these two values of \vec{x} .
 - (c) **Determine whether there are important factors in the first sub-group**
 Test the hypothesis $H_0 : s(k_1 - 1) = s(k)$ versus the hypothesis $H_1 : s(k_1 - 1) > s(k)$.
 If we accept H_0 (none of the factors k_1, \dots, k are important) then continue.
 If we accept H_1 and $k_1 = k$ then output that factor k_1 is important and continue.
 If we accept H_1 and $k_1 < k$ (there is at least one important factor in $\{k_1, \dots, k\}$) then push the group $\{k_1, \dots, k\}$ onto the stack and continue.
 - (d) **Determine whether there are important factors in the second sub-group**
 Test the hypothesis $H_0 : s(k) = s(k_2)$ versus the hypothesis $H_1 : s(k) > s(k_2)$.
 If we accept H_0 (none of the factors $k + 1, \dots, k_2$ are important) then continue.
 If we accept H_1 and $k + 1 = k_2$ then output that factor k_2 is important and continue.
 If we accept H_1 and $k + 1 < k_2$ (at least one of the factors in $\{k + 1, \dots, k_2\}$ is important) then push the group $\{k + 1, \dots, k_2\}$ onto the stack and continue.

Figure 1: The algorithmic structure of sequential bifurcation algorithms. In this paper, we consider the optimal choice of the group-splitting rule used in Step 2b.

Figure 1 leaves unspecified the group-splitting rule, used in Step 2b to split the original group into sub-groups. Some existing SB algorithms (Cheng 1997; Wan, Ankenman, and Nelson 2006) split in half if the number of elements n in the original group is even, or into one sub-group with $(n + 1)/2$ elements and another with $(n - 1)/2$ if n is odd. Other algorithms, including the one originally proposed in (Bettonvil and Kleijnen 1997), suggest choosing one sub-group to contain a number of elements that is a power of 2.

In this paper, we analyze which group-splitting rule minimizes the number of input points at which simulation must be performed. Although simulation effort varies in practice across the values of \vec{x} tested, we use the number of input points at which simulations are performed as a proxy for the overall amount of simulation effort. If the total simulation effort were proportional to the number of input points simulated, then a group-splitting policy that minimized the expected value of one measure would also minimize the expected value of the other measure.

In the specific context of Figure 1 the number of distinct input points (values of \vec{x}) at which SB performs simulations is exactly the two points tested in Step 1 plus one additional point for each iteration through Step 2b.

3 FORMULATION OF THE OPTIMAL GROUP SPLITTING PROBLEM

To support our analysis of the optimal group-splitting rule, we abstract the operations performed by SB in terms of a collection of random variables. Let $\theta = (\beta_0, \beta_1, \dots, \beta_K, \sigma^2(\cdot))$. θ specifies the distribution of $y(\vec{x})$ for each \vec{x} . We work within a Bayesian framework, in which we place a prior distribution P on θ . We consider prior distributions in which the random variables $(\beta_1, \dots, \beta_K)$ are independent of each other, and of both β_0 and $\sigma^2(\cdot)$. The random variables $1\{\beta_k > 0\}$, $k = 1, \dots, K$ that indicate whether each factor k is important are independent.

We specify $Z(k_1, k_2)$ to be 1 if $\beta_k > 0$ for any $k \in \{k_1, \dots, k_2\}$, and 0 otherwise. That is, $Z(k_1, k_2)$ indicates whether $\{k_1, \dots, k_2\}$ contains an important factor. $Z(k_1, k_2)$ depends implicitly on θ , and is entirely determined by it. The prior probability that factor k is important can be written as $P(Z(k, k) = 1) = P(\beta_k > 0)$.

We assume that the hypothesis tests performed in Step 1, 2c and 2d in Figure 1 are free from errors. That is, the decision of the hypothesis test $H_0 : s(k_1 - 1) = s(k_2)$ vs. $H_1 : s(k_1 - 1) > s(k_2)$ is equal to $Z(k_1, k_2)$. When simulation is deterministic, this assumption holds, while for stochastic simulation it does not, and is an approximation made for tractability as noted in Section 1. With this assumption, the number of distinct points \vec{x} that an SB algorithm simulates is completely determined by the collection of random variables $\{Z(k_1, k_2) : 1 \leq k_1 \leq k_2 \leq K\}$.

In Figure 2, we write the general algorithmic framework from Figure 1 as a controlled Markov process using the additional notation defined above. Rather than pushing sets of the form $\{k_1, \dots, k_2\}$ onto the stack, it pushes pairs (k_1, k_2) . The group splitting rule is specified through a function π , which maps the current set of pairs on the stack, and the pair of factors just popped off the stack, to a choice of factor at which to split the current group.

1. Let $t = 0$, $M_0 = 2$, and $Z^{(0)} = Z(1, K)$.
 If $Z^{(0)} = 0$ then stop and output that there are no important factors.
 If $Z^{(0)} = 1$ create an empty stack, and push the group $(1, K)$ onto the stack.
2. Let S_t be the current vector of pairs of entries on the stack. Until the stack is empty, do the following:
 - (a) Increment t and let $M_t = M_{t-1} + 1$. Pop a group off the top of the stack, call it $(k_1^{(t)}, k_2^{(t)})$.
 - (b) Choose a value $k^{(t)} = \pi^{(t)}(k_1^{(t)}, k_2^{(t)}, k_1^{(s)}, k_2^{(s)}, z_1^{(s)}, z_2^{(s)} : s < t)$ satisfying $k_1^{(t)} \leq k^{(t)} < k_2^{(t)}$.
 Observe $Z_1^{(t)} = Z(k_1^{(t)}, k^{(t)})$ and $Z_2^{(t)} = Z(k^{(t)} + 1, k_2^{(t)})$.
 - (c) If $Z_1^{(t)} = 1$ and $k_1^{(t)} = k^{(t)}$ then output that factor $k^{(t)}$ is important.
 If $Z_1^{(t)} = 1$ and $k_1^{(t)} < k^{(t)}$ then push the group $(k_1^{(t)}, k^{(t)})$ onto the stack.
 If $Z_1^{(t)} = 0$ then do not change the stack.
 - (d) If $Z_2^{(t)} = 1$ and $k^{(t)} + 1 = k_2^{(t)}$ then output that factor $k^{(t)} + 1$ is important.
 If $Z_2^{(t)} = 1$ and $k^{(t)} + 1 < k_2^{(t)}$ then push the group $(k^{(t)} + 1, k_2^{(t)})$ onto the stack.
 If $Z_2^{(t)} = 0$ then do not change the stack.
3. Let T be the current value of t and M_T be the current value of M_t .

Figure 2: The algorithmic structure of sequential bifurcation algorithms from Figure 1, where $Z(k_1, k_2)$ indicates whether $\{k_1, \dots, k_2\}$ contains an important factor, and π is the group splitting rule.

This algorithm tracks the number of distinct $\vec{x}(k)$ at which simulation replications are obtained, our surrogate for the overall simulation effort, in the random variable M_T . This random variable is initialized to 2, for the pair of points at which simulations are performed in Step 1, and then incremented each time we pass through the loop in Step 2. Because we assume hypothesis tests make no errors, M_T is fully determined by θ and π . The measure of performance that we consider is the expected value of M_T , integrating over

our prior distribution on θ . The dependence of this expectation on π is indicated by the notation E^π .

$$E^\pi[M_T] = \int E^\pi[M_T|\theta]P(d\theta). \quad (1)$$

Note that the Bayesian framework *does not* resample θ after every test is performed. Instead, θ is modeled as fixed, and as having been drawn from the prior distribution P before the screening procedure begins.

The objective (1) does not consider the *accuracy* of the SB method, in terms of the number of factors correctly selected as important or unimportant. Instead, we assume that accuracy has been satisfactorily controlled through design of the hypothesis tests, and our goal is instead to minimize the sampling effort required through selection of the group-splitting method. Although our assumption of error-free hypothesis testing does not allow us to quantitatively model the dependence of accuracy on the group-splitting rule, intuition suggests that reducing the number of hypothesis tests performed also reduces the number of opportunities at which errors can be made, which should tend to improve overall accuracy.

4 OPTIMAL GROUP SPLITTING

Our goal in our analysis is to choose a group-splitting rule to use in Step 2b so as to minimize (1). In Section 4.1 we first pursue this goal under an assumption that the probability of a factor’s importance is the same for all factors. In Section 4.2 we relax the homogeneity hypothesis and allow factors with two different probabilities of importance. In each section, we provide an explicit computational method for finding the optimal group splitting rule and present a numerical investigation comparing splitting in half to the optimal rule.

4.1 Homogeneous Factors

We first perform our analysis for the case that the probability $P(Z(k,k) = 1)$ that factor k is important is constant across k . Let $p = P(Z(k,k) = 1)$ and let $q = 1 - p$. This case is the simplest Bayesian prior. (This case also applies in the frequentist setting if we permute the factors uniformly at random before beginning the screening process, although performing such a permutation would violate the advice of (Bettonvil and Kleijnen 1997) to sort the factors in increasing order of importance, and would likely degrade performance.) We use dynamic programming to calculate the optimal number of factors to include into the first and second sub-groups when performing group splitting in Step 2b.

To support our analysis, we define a function $r(u,n)$, which is the probability that a sub-group of u factors contains an important factor, given that it is a subset of a parent group of n factors that contains an important factor. For $u \leq n$, $r(u,n)$ is defined formally and then computed as

$$\begin{aligned} r(u,n) &:= P(Z(1,u) = 1|Z(1,n) = 1) \\ &= \frac{P(Z(1,u) = 1, Z(1,n) = 1)}{P(Z(1,n) = 1)} = \frac{P(Z(1,u) = 1)}{P(Z(1,n) = 1)} = \frac{1 - q^u}{1 - q^n}. \end{aligned} \quad (2)$$

In this computation of $r(u,n)$, we have used that

$$P(Z(1,u) = 1) = 1 - P(Z(1,u) = 0) = 1 - \prod_{k=1}^u P(Z(k,k) = 0) = 1 - q^u \quad (3)$$

by the independence of the importance of the factors.

We now proceed with our dynamic programming analysis. Let $W^*(K)$ be the value of $E^{\pi^*}[M_T]$ under the optimal group splitting rule π^* , as a function of the overall number of factors K . To compute $W^*(K)$, we first define another function V^* . Let $V^*(n)$ be the expected number of *additional* points $\vec{x}(k)$ at which simulations must be run, under the optimal group splitting rule, to completely process all of the factors within a group of n factors previously classified as containing an important factor. That is,

$$V^*(n) = E^{\pi^*}[M_T - M_t | \mathcal{S}_t = \{(k, k+n)\}]. \quad (4)$$

$V^*(n)$ does not depend on the value of t or k in (4), as long as k satisfies $1 \leq k \leq K - n$, and as long as there is some strictly positive probability of reaching $S_t = (k, k + n)$ for some k under policy π^* . Note that reaching $S_t = (k, k + n)$ implies that $Z(k, k + n) = 1$.

$W^*(K)$ can be computed from $V^*(K)$ because setting $t = 0$ shows $V^*(K) = E[M_T - 2 | Z(1, K) = 1]$. From this we have,

$$W^*(K) = 2 + P(Z(1, K) = 1)V^*(K) = 2 + (1 - q^K)V^*(K)$$

Here, we have used the expression (3). Both W^* and V^* implicitly depend on q .

We now give a recursive algorithm for computing V^* . First,

$$V^*(1) = 0,$$

because a single factor previously classified as important requires no further simulation.

Now, consider some $n > 1$. When first splitting this group of n factors, if we split it into one group of $u \in \{1, \dots, n - 1\}$ factors and another of $n - u$, then the expected number of additional points to be simulated will be the sum of 1 (from this first split), and the expected number of additional points required to process each of the first and second subgroups.

If the first subgroup is found to contain no important factors, it will require no additional simulation. If it is found to be important (which occurs with probability $r(u, n)$), it will require $V^*(u)$ additional points in expectation. Thus, the first subgroup requires $r(u, n)V^*(u)$ additional points in expectation. Similarly, the second subgroup requires $r(n - u, n)V^*(n - u)$. Optimizing over u provides the following recursive expression,

$$V^*(n) = \min_{u \in \{1, \dots, n-1\}} 1 + r(u, n)V^*(u) + r(n - u, n)V^*(n - u).$$

The set of optimal values for u (which is not necessarily unique) is the set of values attaining this minimum.

$$u^*(n) \in \operatorname{argmin}_{u \in \{1, \dots, n-1\}} 1 + r(u, n)V^*(u) + r(n - u, n)V^*(n - u). \quad (5)$$

The expected number of factors at which we must run simulations under the group splitting rule of dividing each group in half can be computed with a similar recursive algorithm, where we replace the maximization over u with the decision $u^{\text{SB}}(n) = \lfloor n/2 \rfloor$, where $\lfloor x \rfloor$ is the largest integer less than or equal to x .

$$\begin{aligned} W^{\text{SB}}(K) &= 2 + (1 - q^K)V^{\text{SB}}(K), \\ V^{\text{SB}}(1) &= 0, \\ V^{\text{SB}}(n) &= 1 + r(u, n)V^{\text{SB}}(u^{\text{SB}}(n)) + r(n - u^{\text{SB}}(n), n)V^*(n - u^{\text{SB}}(n)). \end{aligned}$$

Although we choose to round down in the expression $\lfloor n/2 \rfloor$, the value of $W^{\text{SB}}(K)$ remains unchanged if we instead use $\lceil n/2 \rceil$.

Figure 3 compares splitting in half (labeled SB) to the optimal policy, assuming that each factor is important with a constant probability of $p = 0.1$. The left panel shows $W^*(K)$ and $W^{\text{SB}}(K)$ as a function of K . The center panel shows $u^*(n)$ and $u^{\text{SB}}(n)$ as a function of n , where we take $u^*(n)$ to be the minimal value in the argmax set (5). The right panel shows the ratio $W^*(K)/W^{\text{SB}}(K)$ of the number of factors tested as a function of K . When this ratio is 1, SB is optimal, and when it is strictly less than 1, SB is strictly suboptimal. For other numbers of factors, SB falls within 98% of optimal, at least for this range of the number of factors, and this value of p .

The central panel of Figure 3 shows an interesting behavior. In the plot, the pair $(u^*(n), n - u^*(n))$ always contains the largest integer power of 2 less than or equal to $n/2$. We found this behavior also held for other values of p and larger values of n . This is consistent with the proposed group-splitting rule of (Bettonvil and Kleijnen 1997). This leads us to make the following conjecture.

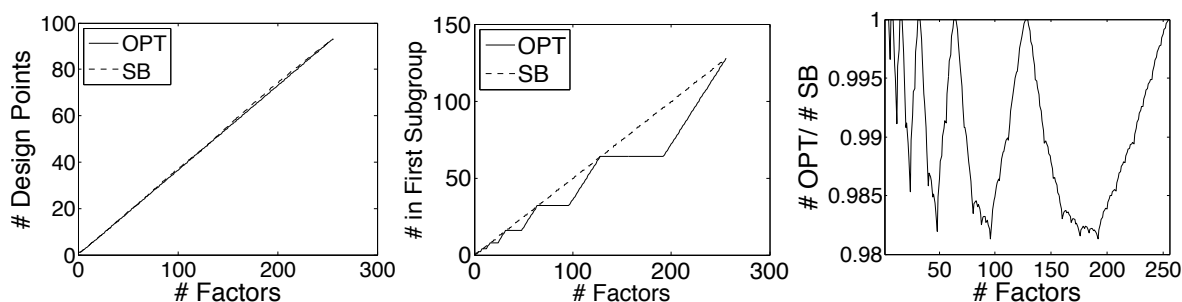


Figure 3: Comparison of the optimal policy (OPT) with the rule of splitting groups in half (SB), where factors have a homogeneous probability of importance of $p = 0.1$. **Left:** Expected number of input points simulated under the two policies, $W^*(K)$ and $W^{SB}(K)$. **Center:** Number of factors to include into the first sub-group under the two policies, $u^*(n)$ and $u^{SB}(n)$, as a function of the number of factors n in the parent group. **Right:** Ratio $W^*(K)/W^{SB}(K)$ of the expected number of input points simulated between OPT and SB. SB is between 98% and 100% of optimal.

Conjecture 1 When factors have a homogeneous probability of importance $P(Z(k, k) = 1) = p$, the optimal group-splitting policy splits groups of size n into one sub-group of size $2^{\lfloor \log_2(n/2) \rfloor}$ and another of size $n - 2^{\lfloor \log_2(n/2) \rfloor}$, consistent with the rule proposed by (Bettonvil and Kleijnen 1997).

This conjectured optimal group-splitting rule does not depend on the probability of importance. This is practically beneficial because, if the conjecture is true, implementations of SB using the optimal group-splitting rule would not need to elicit prior information from the user. If the above conjecture is true, then it also implies that dividing in half gives the optimal decision when n is a power of 2. If K is an integer power of 2, then dividing it in half will cause every sub-group to have a size that is a power of 2, and the “in half” rule will make every group splitting decision optimally. This is supported by the behavior observed in the right panel of Figure 3, which shows that OPT/SB is 1 when K is a power of 2.

4.2 Non-homogeneous Factors

We consider in this section a non-homogeneous prior that we believe is common in applications. It is often the case that some factors are a priori more likely to be important than others. We assume that there are two distinct types of factors. Factors of type I are more likely to be important while the other factors, of type II, are less likely to be important. Keeping the same Bayesian model, $P(Z(k, k) = 1) = p^{(1)}$ when k is a factor of type I, and $P(Z(k, k) = 1) = p^{(2)}$ when k is a factor of type II. $p^{(1)} \geq p^{(2)}$ and the equality case is the case treated in the previous section. We notate $q^{(1)} = 1 - p^{(1)}$ and $q^{(2)} = 1 - p^{(2)}$. We also notate K_1 , $0 \leq K_1 \leq K$, to be the number of factors of type I and $K_2 = K - K_1$ to be the number of factors of type II. In SB, it is standard to order the factors by believed importance (Kleijnen 2009). The symmetry of the problem allows this order to be either increasing or decreasing. Most authors, e.g., Bettonvil and Kleijnen (1997) and Bettonvil (1990) have adopted an increasing order, but we adopt a decreasing order. Thus we assume that factors of type I appear before (i.e., have a lower factor index k) those of type II. Thus, we may assume that $k_1 \leq k_2$ implies $P(Z(k_1, k_1) = 1) \geq P(Z(k_2, k_2) = 1)$.

The analysis below generalizes the analysis performed in the previous section. When splitting a group, one can choose the relative size (i.e., number of factors) of each subgroup but also the number of factors of type I within each subgroup. To support our analysis, we define the function $r(v, u, m, n)$ which is the probability that a sub-group of u factors, v of which are of type I, contains an important factor, given that it is a subset of a parent group of n factors, m of which are of type I, that contains an important factor. For

$u \leq n$ and $\max(0, m - n + u) \leq v \leq \min(u, m)$,

$$r(v, u, m, n) = \frac{1 - q^{(1)v} q^{(2)u-v}}{1 - q^{(1)m} q^{(2)n-m}}$$

which is the ratio of the probability that the subgroup contains an important factor to the probability that its parent group contains an important factor.

Now we compute the optimal number of factors using dynamic programming. We follow the same logical process as in the previous section. We define $W^*(K_1, K)$ as the value of $E^{\pi^*}[M_T]$ under the optimal group splitting rule π^* , as a function of the number K_1 of factors of type I and as a function of K , the total number of factors, $K_1 \leq K$. We then define $V^*(m, n)$, the expected number of additional points $\vec{x}(k)$ at which simulations must be run, under the optimal splitting rule, to completely process all the factors within a group of n factors, m of which are of type I, previously classified as containing an important factor, $m \leq n$. V^* and W^* satisfy the following relation:

$$W^*(K_1, K) = 1 + (1 - q^{(1)K_1} q^{(2)K-K_1})V^*(K_1, K)$$

We now discuss a recursive algorithm for computing V^* . First,

$$V^*(0, 1) = 0,$$

$$V^*(1, 1) = 0,$$

since a single factor previously classified as important needs no further simulation. Secondly, optimizing over the number of factors u to include in the first group provides the following expression. Recall that the factors are ordered, with factors of type I appearing first and factors of type II appearing second. In the following expression, the first inner minimization considers those values of u causing only type I factors to be in the first group. The second inner minimization considers those values of u from including all of the factors of type I in the first group, leaving only factors of type II for the second group.

$$V^*(m, n) = \min \left\{ \min_{1 \leq u \leq \min(m, n-1)} 1 + r(u, u, m, n)V^*(u, u) + r(m-u, n-u, m, n)V^*(m-u, n-u), \right. \\ \left. \min_{m+1 \leq u \leq n-1} 1 + r(m, u, m, n)V^*(m, u) + r(0, n-u, m, n)V^*(0, n-u) \right\},$$

where a minimum over an empty set is understood to be infinity. When $m = 0$, the inner minimum over $1 \leq u \leq \min(m, n-1)$ is infinity, and the overall minimum is achieved by the inner minimum over $m+1 \leq u \leq n-1$. When $m = n$, the inner minimum over $m \leq u \leq n-1$ is infinity, and the overall minimum is achieved by the inner minimum over $1 \leq u \leq \min(m, n-1)$.

Without loss of generality, we assume that the factors of type I are the first K_1 factors with indexes 1 to K_1 . The “in-half” group splitting rule consists then in choosing $u^{SB} = \lfloor n/2 \rfloor$ and $v^{SB} = \min(u^{SB}, m)$ when the parent group has n factors, m of which are of type I. Replacing the optimization over u and v by these values we obtain,

$$W^{SB}(K_1, K) = 1 + (1 - q^{(1)K_1} q^{(2)K-K_1})V^{SB}(K_1, K), V^{SB}(0, 1) = 0, V^{SB}(1, 1) = 0 \\ V^{SB}(m, n) = 1 + r(v^{SB}, u^{SB}, m, n)V^{SB}(v^{SB}, u^{SB}) + r(m - v^{SB}, n - u^{SB}, m, n)V^{SB}(m - v^{SB}, n - u^{SB})$$

Figure 4 compares the optimal policy with the rule of splitting in half (labeled SB in the figure). Paralleling Figure 3, the left panels show $W^\pi(K_1, K)$ as a function of K for the two policies; the center panels show the number of factors included into the first sub-group as a function of the number of factors

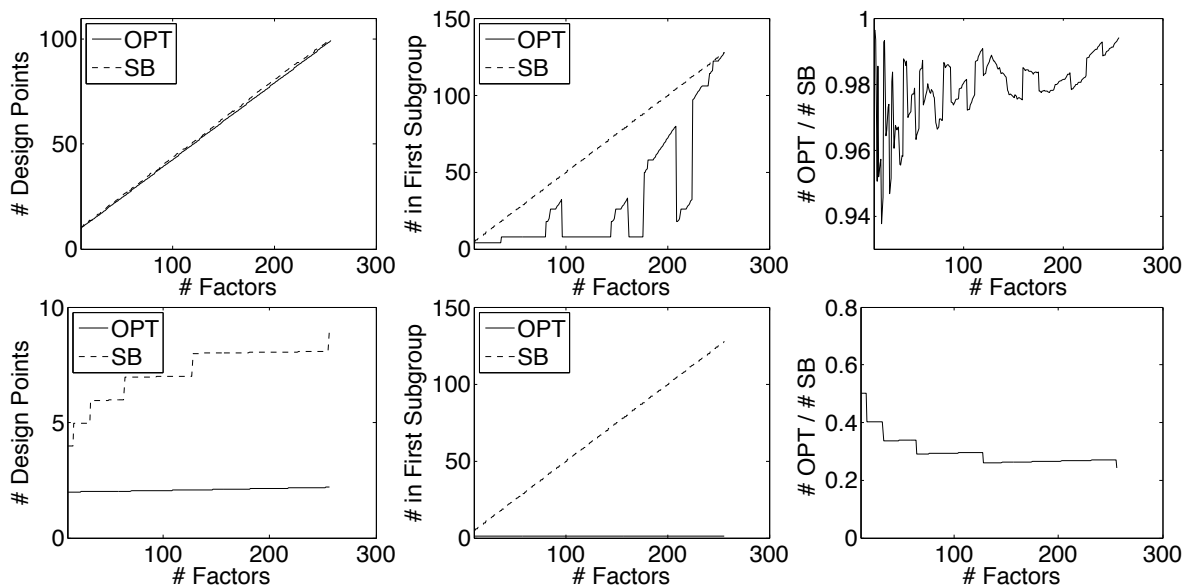


Figure 4: Comparison of the optimal policy (OPT) and the rule of splitting groups in half (SB), where we have K_1 factors with probability of importance $p^{(1)}$, and the remaining factors have probability $p^{(2)}$. In the **Top Row**, $p^{(1)} = 0.75$, $p^{(2)} = 0.1$, and $K_1 = 10$. In the **Bottom Row**, $p^{(1)} = 0.99$, $p^{(2)} = 0.0001$, and $K_1 = 1$. **Left Panels:** Expected number of input points simulated under the two policies, $W^*(K_1, K)$ and $W^{SB}(K_1, K)$. **Center Panels:** Number of factors to include into the first sub-group under the two policies, as a function of the number of factors n in a parent group containing K_1 factors with probability of importance $p^{(1)}$. **Right Panels:** Ratio $W^*(K_1, K)/W^{SB}(K_1, K)$ of the expected number of input points simulated between OPT and SB. Values strictly less than 1 indicate the suboptimality of SB.

n in the parent group, where the number of type I factors in this parent group is held fixed at K_1 ; and the right panels show the ratio $W^*(K_1, K)/W^{SB}(K_1, K)$. The top row shows a collection of problems in which $p^{(1)} = 0.75$, $p^{(2)} = 0.1$, and $K_1 = 10$, while the bottom row shows a more extreme collection of problems in which the $p^{(1)} = 0.99$, $p^{(2)} = 0.0001$, and $K_1 = 1$.

In the top row, which we feel is more typical of situations many encountered in practice, SB is within 94% of optimal. In the bottom row, which is less representative of typical situations, but which serves to illustrate the spectrum of behaviors that can be observed, SB is much further away from optimal, achieving between 25% and 50% of optimal.

In the problems considered in the bottom row, the optimal policy places the one type I factor alone in the first sub-group, and then leaves the type II factors in a second sub-group. Because the type II factors are unlikely to be important, this large second sub-group is often immediately eliminated, providing $W^*(K_1, K)$ close to 3, as two points are tested in the initialization phase, and testing one additional point is sufficient to establish that the first factor is important, and the rest are not. In contrast, the rule of splitting in half places unimportant type II factors together with the important type I factor, requiring roughly $2 + \log_2(K)$ input points be tested overall before isolating the single type I factor.

5 CONCLUSION

We have considered group-splitting policies for the sequential bifurcation algorithm within a Bayesian framework. Using a dynamic programming analysis, we have provided recursions for computing the Bayes-optimal group-splitting policy for a version of the problem with homogeneous factors, and for another version with two types of factors: one with a relatively high probability of importance, and the other with a relatively low probability of importance.

We have shown using numerical experiments that, when factors are homogeneous, splitting groups in half is close to optimal, and the group splitting rule recommended by (Bettonvil and Kleijnen 1997) appears to be optimal. When there are two types of factors, the optimal policy offers a larger percentage improvement over splitting in half, with the largest benefit being present when we have a small number of factors that are very likely to be important, and a large number of factors that are very unlikely to be important.

The analysis that we have pursued is stylized in two ways: it assumes that hypothesis tests do not make errors, and that the simulation effort is constant across hypothesis tests. In future work, we plan to relax these assumptions, and to use dynamic programming to analyze the optimal policy, or bounds on the performance of the optimal policy, in the presence of more realistic conditions.

In future work, we also plan to relax the restrictions imposed by restricting to algorithms within the sequential bifurcation framework. We plan to consider a broader class of policies that do not necessarily test groups according to the nesting structure imposed by sequential bifurcation, to determine whether greater efficiency can be obtained with other sequential policies.

ACKNOWLEDGMENTS

Peter I. Frazier was supported by AFOSR YIP FA9550-11-1-0083 and NSF EAGER 1142251.

REFERENCES

- Bettonvil, B. 1990. *Detection of important factors by sequential bifurcation*. Ph. D. thesis, Tilburg University.
- Bettonvil, B. W. M., and J. P. C. Kleijnen. 1997. "Searching for important factors in simulation models with many factors: sequential bifurcation". *European Journal of Operational Research* 96 (1): 180–194.
- Bettonvil, B. W. M., J. P. C. Kleijnen, and F. Persson. 2006. "Screening for the important factors in large discrete-event simulation: Sequential bifurcation and its applications". In *Screening: Methods for Experimentation in Industry, Drug Discovery and Genetics*, edited by A. M. Dean and S. M. Lewis, 287–307. New York: Springer Verlag.
- Brochu, E., M. Cora, and N. de Freitas. 2009, November. "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning". Technical Report TR-2009-023, Department of Computer Science, University of British Columbia.
- Cheng, R. C. H. 1997, December. "Searching for important factors: sequential bifurcation under uncertainty". In *Proceedings of the 1997 Winter Simulation Conference*, edited by S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 275–280. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Chick, S. E., J. Branke, and C. Schmidt. 2010. "Sequential Sampling to Myopically Maximize the Expected Value of Information". *INFORMS J. on Computing* 22 (1): 71–80. to appear.
- Chick, S. E., and P. I. Frazier. 2012. "Sequential Sampling for Selection with Economics of Selection Procedures". *Management Science* 58:550–569.
- Chick, S. E., and N. Gans. 2009. "Economic Analysis of Simulation Selection Problems". *Management Sci.* 55 (3): 421–437.
- Dynkin, E. B., and A. A. Yushkevich. 1979. *Controlled Markov Processes*. New York: Springer.
- Frazier, P. I., and W. B. Powell. 2008, December. "The Knowledge-Gradient Stopping Rule for Ranking and Selection". In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 305–312. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Frazier, P. I., W. B. Powell, and S. Dayanik. 2008. "A Knowledge Gradient Policy for Sequential Information Collection". *SIAM Journal on Control and Optimization* 47 (5): 2410–2439.
- Jedynak, B., P. I. Frazier, and R. Sznitman. 2012. "Twenty questions with noise: Bayes optimal policies for entropy loss". *Journal of Applied Probability* 49 (1): 114–136.

- Kleijnen, J. P. C. 2007. *Design and analysis of simulation experiments*. Springer Verlag.
- Kleijnen, J. P. C. 2009. “Factor screening in simulation experiments: review of sequential bifurcation”. In *Advancing the Frontiers of Simulation*, edited by C. Alexopoulos, D. Goldsman, J. R. Wilson, and F. S. Hillier, 153–167. New York: Springer.
- Lovejoy, W. S. 1991. “A survey of algorithmic methods for partially observed Markov decision processes”. *Annals of Operations Research* 28 (1): 47–65.
- Powell, W. B. 2007. *Approximate Dynamic Programming: Solving the curses of dimensionality*. New York: John Wiley and Sons.
- Powell, W. B., and I. O. Ryzhov. 2012. *Optimal Learning*. Wiley.
- Shi, W., J. P. C. Kleijnen, and Z. Liu. 2012. “Factor Screening for Simulation with Multiple Responses : Sequential Bifurcation”. Technical Report CentER Discussion Paper No. 2012-032, CentER, Tilburg University.
- Waeber, R., P. I. Frazier, and S. G. Henderson. 2011, December. “A Bayesian Approach to Stochastic Root Finding”. In *Proceedings of the 2011 Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 4038–4050. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Wan, H., B. E. Ankenman, and B. L. Nelson. 2006. “Controlled Sequential Bifurcation: A New Factor-Screening Method for Discrete-Event Simulation”. *Operations Research* 54 (4): 743–755.
- Wan, H., B. E. Ankenman, and B. L. Nelson. 2010. “Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening”. *INFORMS Journal on Computing* 22 (3): 482–492.
- Xie, J., and P. I. Frazier. 2011. “Sequential Bayes-Optimal Policies for Multiple Comparisons with a Known Standard”. in review.
- Yaesoubi, R., S. D. Roberts, and R. W. Klein. 2010, December. “A modification of Cheng’s method: An alternative Factor Screening method for stochastic simulation models”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hagan, and E. Yücesan, 1034–1047. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

PETER I. FRAZIER is an assistant professor in the School of Operations Research and Information Engineering at Cornell University. He received a Ph.D. in Operations Research and Financial Engineering from Princeton University in 2009. He is the recipient of an AFOSR Young Investigator Award in 2010. His research interest is in dynamic programming and Bayesian statistics, focusing on the optimal acquisition of information. He works on applications in simulation, global optimization, operations management, and medicine. His web page can be found via <http://www.orie.cornell.edu>, and his email address is pf98@cornell.edu.

BRUNO JEDYNAK received his Ph.D. in Applied Mathematics from the Université Paris Sud. He spent a year as a post-doc in the department of Statistics at the University of Chicago. He was then appointed professor at the Université des Sciences et Technologies de Lille, France. He is currently a faculty member of the Department of Applied Mathematics and Statistics at The Johns Hopkins University (JHU) and faculty member of the Center for Imaging Science at JHU. His webpage is <http://cis.jhu.edu/~bruno/> and his email address is bruno.jedynek@jhu.edu.

LI CHEN is a second year Ph.D student in the Department of Applied Mathematics and Statistics at the Johns Hopkins University. She received her Bachelor of Science Degree in Mathematics from the University of Oregon and Master of Science Degree from Oregon State University. She is supported fully by a departmental assistantship. Her research interests lie in the fields of applied probability, data mining, machine learning and pattern recognition. Her email address is lchen87@jhu.edu.