In *New Developments and Applications in Experimental Design*,
N. Flournoy, W.F. Rosenberger, and W.K. Wong, eds., Institute of Math. Stat.
Lecture Notes – Monograph Series Vol. **34**, 1998, pp. 91–105.

FLEXIBLE ALGORITHMS FOR CREATING AND ANALYZING ADAPTIVE SAMPLING PROCEDURES

BY JANIS P. HARDWICK AND QUENTIN F. STOUT

University of Michigan, Ann Arbor, MI 48109

We describe a collection of algorithms and techniques that have been developed to aid in the design and analysis of adaptive allocation procedures. The emphasis is on providing flexibility to the investigator, so that appropriate statistical and practical concerns can be addressed directly. The techniques described allow for optimizations previously not attainable. They also permit exact evaluations for a wide range of criteria and are intended to encourage investigators to explore more alternatives. Optimizations investigated include 2- and 3-population fully sequential models, few-stage models, and models with constrained switching between options. One of our algorithmic approaches, path induction, speeds up the process of evaluating a procedure multiple times so that thorough robustness studies can be undertaken. Our approaches can be utilized with both Bayesian and frequentist analyses.

1 Introduction

In situations where data are collected over time, adaptive sampling methods often lead to more efficient results than do fixed sampling techniques. When sampling or "allocating" adaptively, sampling decisions are based on accruing data. In contrast, when using fixed sampling procedures, the sample sizes taken from different populations are specified in advance and are not subject to change. Using adaptive techniques can reduce costs, time and improve the precision of the results for a given sample size. Fully sequential adaptive procedures, in which one adjusts after each observation, are the most powerful. Such procedures are rarely used, however, due to difficulties related to generating and implementing good procedures as well as to complications associated with analyzing the resulting data.

Our goal is to help researchers utilize adaptive allocation by creating a collection of algorithms to optimize and analyze a variety of sequential procedures. The techniques are detailed in [12, 15, 16, 18, 20, 21, 13]. The intent here is to illustrate how this growing suite of algorithms allows researchers the flexibility to incorporate a variety of statistical objectives and operational considerations into the design and analysis process. Optimal procedures may be needlessly complex or difficult to employ and explain. An added value associated with their generation, however, is the ability to establish the efficiency of suboptimal procedures. If one can show that the relative efficiency of a procedure is high compared with the optimal one, then

---

Research supported in part by National Science Foundation under grants DMS-9157715 and DMS-9504980.
AMS 1991 subject classification. Primary 62L05; secondary 62K05, 62N05, 62A15, 62L10
Key words and phrases. Sequential allocation, Dynamic programming, Path induction, Bandit problems, Algorithms, High-performance computing, Stochastic optimization, Parallel computing, Exact analysis, Bernoulli response, Switching, Few-stage

investigators may be justified in implementing the typically simpler and more intuitive suboptimal option. Since this collection of algorithms also allows for the optimization of strategies that are constrained to have desirable operational characteristics, the likelihood that investigators can incorporate such goals and still achieve statistical efficiency is increased.

1.1 Motivation

Years ago it was not only analytically, but also computationally, infeasible to attain exact solutions to most adaptive allocation problems. As an example, Bradt and Karlin (1956) argue that if, for a specific problem, the optimal sequential procedure were "practically obtainable, the interest in any other design criteria which have some justification although not optimal is reduced to pure curiosity." They immediately add, however, that obtaining optimal procedures is not practicable. Then, as an illustration of the "intrinsically complicated structure" of optimal procedures, the authors detail the first step of the optimal solution to a simple sequential design problem involving only three Bernoulli observations. While during this same year, Bellman (1956) identified the fact that problems of this nature could, in principle, be solved via dynamic programming, such solutions are still typically viewed to be infeasible. For example, Wang (1991), addressing a variation of the problem in Bradt and Karlin (1956), notes that "In theory the optimal strategies can always be found by dynamic programming but the computation required is prohibitive".

Historically, then, the computational complexity of sequential sampling problems forced researchers to pursue approximations or bounds that could be obtained analytically. Usually the process has been to determine a lower bound for some statistical objective function and then to evaluate how close various ad hoc strategies came to the bound as the sample size approaches infinity. It has generally been felt that if a procedure achieved second order efficiency then it was probably a suitable rule. There do not, however, appear to be good ways to assess how accurate the approximations are for specific sample sizes. Often, for example, in order to employ a procedure it is necessary to specify constants that are only vaguely determined by the asymptotics.

This situation motivated us to work on greatly extending the range of problems that could be analyzed and optimized computationally. While some of the gains can be attributed to the ever increasing power of computers, much is due to algorithms and implementation. To state that a problem can be "solved with dynamic programming" is as vague as saying that one need only "do the math". Careful implementations of complex dynamic programming variations, along with new algorithmic techniques such as *path induction*, have been necessary to achieve the results reviewed here. While the models for which optimal solutions can be computed are often, albeit arguably, deemed to be "too simplistic", it is nevertheless the case that the insight one garners from evaluating these models are likely to lead to better heuristics that apply as well to more complex scenarios. Furthermore, as the research progresses, the complexity and functionality that can be evaluated increases.

To illustrate the flexibility of this computational methodology, we demonstrate the manner in which

1. dynamic programming can be used to *generate* optimal procedures,
2. path induction can be used to *evaluate* both optimal and suboptimal procedures for their statistical properties and operating characteristics, and
3. a variety of criteria and constraints can be added to a base problem to incorporate important aspects that might otherwise be ignored or – due to analytic intractability – be treated in a seriously suboptimal manner.

Another important characteristic of the algorithms is that all results are exact.

In the next section, Section 2, we define the model and two sample problems that are used to illustrate the results obtainable. In Section 3 we address optimizing fully sequentially procedures, and illustrate the progress that has been made in solving large problems. In Section 4, we address scenarios in which model assumptions have been violated, and illustrate the valuable role of path induction in enabling sensitivity analyses. In Section 5 we consider the practical restriction that sampling be specified only by stages rather than fully sequentially; and in Section 6 we consider the related problem of sampling when repeated switching among populations is undesirable. Finally, in Section 7, we outline ongoing work to extend these techniques to new problems.

2 Problem Formulation

Throughout, we restrict attention to problems for which the population outcomes are independent Bernoulli random variables. To *design* optimal sampling strategies, we assume a Bayesian perspective and model the success parameters of the populations as independent beta random variables. These assumptions can be relaxed, and we have evaluated procedures where the populations are not independent, and ones where the success parameters are mixtures of betas. While we utilize Bayesian assumptions for optimal design, we can *analyze* arbitrary procedures using either Bayesian or frequentist criteria. These criteria need not be tied to the assumptions used to generate the procedure.

To simplify the discussion, the total number of observations, n, is fixed. It is important to note that this is merely for convenience in describing the complexity of the algorithms, and is *not* a requirement of the procedures. For example, in [15], n represents the maximum possible number of observations, where optional stopping is available and the goal is to minimize the expected number of observations needed.

A number of different types of goals can be treated using the techniques discussed here, and we have selected two problems to serve as examples. The first is the classic *multi-armed bandit* problem with finite horizon n, [5], where the objective is to sample from among a Bernoulli populations (the "arms") in such a way that the sum of the n observations is maximized. The second, the *product of means* problem, is to minimize the mean squared error of the estimate of the product of the success probabilities for the different populations. It arises in reliability and other settings and has been studied by several authors in a variety of forms (see [19, 25, 32] and the references therein).

Note that, while the aim in the bandit problem is to quickly identify and then to sample from the best population, for the product of means problem one needs a sufficient, although usually disproportional, number of observations from each population because each contributes to the function estimator. The nonlinear nature of the function further complicates the interactions.

3 Computational Approaches

Before discussing algorithms for specific class of procedures, it is important to understand how they would be used. To optimize the design for a specific problem, one would decide the class of procedure desired, such as fully sequential, 2-stage, fully sequential with constrained switching, etc. The algorithm for that class of procedure would be used, and the only adaptation needed for the specific problem would be to determine the value of the objective function at the terminal states, i.e., at the end of the experiment. For example, for the bandit problem this is merely the number of successes observed. Except for these few lines of code, the algorithm is unchanged, even though the procedures it produces may vary dramatically for different objective functions.

The primary computational challenge of adaptive allocation is the fact that the state spaces involved are quite large. To describe the time and space requirements of algorithms, we use "generalized O-notation" from computer science, in which we say a function $f(n) = \Theta(g(n))$ if there exist positive constants C, D, N such that $Cg(n) \leq f(n) \leq Dg(n)$ for all $n \geq N$.

n	Arms	Equiv. Evals	State RAM	Who and When	Machine
200	2	6.7×10^7	1.3×10^6	Berry & Eick, 1987	Cray 2
400	2	1×10^{11} †	1×10^7	Hardwick & Stout, 1993	Sun 3 work.
100	3	1.4×10^{11} †	8×10^7	Oehmke, Hardwick, Stout, 1997	IBM SP2
200	3	8.9×10^{12} †	2.7×10^9	Oehmke, Hardwick, Stout, 1998	IBM SP2

† State eval's if backward induction, versus path induction, used for 100 procedure eval's.

Table 1: Computational Solutions for Multi-Armed Bandits

3.1 2-Population Fully Sequential Procedures

Multi-population fully sequential allocation presents a state space that has impeded analysis since its earliest embodiments. While such problems can be solved with dynamic programming, for a Bernoulli populations and a sample size of n, the number of states is $\binom{n+2a}{2a} \approx n^{2a}/(2a)!$. Even when $a = 2$, the growth rate is daunting and has been a hindrance to exact evaluation and optimization. Most of the work on optimizing fully sequential models has come from interest in bandit problems, although, as noted above, with a change of terminal objective function many bandit algorithms could optimize any problem. Armitage (1985) cited the 2-armed bandit (2-AB) as being a problem for which "the computation involved is prohibitive except for trivially small horizons". More recently, however, refined algorithms, careful implementations, and ongoing advances in computer speed and memory, have greatly extended the range of sample sizes that can be exactly evaluated. Some of the techniques for careful implementation are detailed in [16].

To be specific as to the benefits claimed, it is our understanding that, prior to our work, the largest 2-population fully sequential problem that had been solved appeared in Berry and Eick (1995). Using a Cray 2 supercomputer in 1987, they solved a 2-AB with a sample size of $n = 200$. At the other end of the resource spectrum, Jones (1992), using a (presumed) personal computer, solved a variation of the 2-AB problem, but could only handle size $n = 25$. Concurrently, in 1991 we addressed virtually the same variation with samples of size $n = 150$ using a modest desktop workstation [14], and by 1993 were solving problems of size $n = 400$. Further, we needed to evaluate the procedures 100 times to determine the probability of correct selection. These evaluations were performed using path induction (Section 3.3), saving an order of magnitude in time over the usual backward induction.

3.2 Multiple Populations and Parallel Algorithms

More recently, we have been pursuing problems involving more than 2 populations. Such problems are of great interest, particularly in the design of clinical trials to sequentially select or test among several populations (e.g., see [6, 8, 9, 10, 26, 30]). We know of no prior work in which exact, optimal solutions for fully sequential problems with 3 or more populations are determined.

In [12], we describe initial work on the Bernoulli 3-population fully sequential problem with samples up to $n = 100$. Since the state space of the 3-population problem is roughly three orders of magnitude greater than that of the 2-population problem, it was necessary to move onto a modest parallel computer which offers greater compute power and memory than a standard workstation. We used an IBM SP2, although the program is written in a fashion that allows it to be ported to a variety of parallel and distributed systems. The program is being revised to scale to larger problems and parallel systems, and we can presently run problems with sample sizes of $n = 200$. Table 1 summarizes the sizes of bandit problems that have been solved to date.

3.3 Path Induction

The progress in addressing fully sequential allocation was not limited to mere size increases. A new technique we refer to as path induction has allowed for *multiple evaluations* of arbitrary procedures [20]. The multiple evaluations are useful because they provide insight into the behavior of the procedures. For example, whereas the "solution" to a simple 2-AB problem furnishes the number of successes to be expected, path induction can give information such as the distribution of the same quantity or similar information relating to other procedure attributes.

Path induction is a two phase process. In the "initialization" phase, the algorithm works from the front of the experiment towards its conclusion, going through the state space to determine the number of paths that reach each terminal state. In the "evaluation" phase, path counts are used to determine the probability of reaching each terminal state, which can be used to determine whatever average quantities one desires. In general, since there are far fewer terminal states than total states, use of path induction results in significant savings when multiple evaluations of a given procedures are needed. This method is exact and quite general, and is applicable to a wide range of procedures, analyses and criteria. For more details and algorithms, see [20].

In Section 4, path induction is applied to the problem of assessing the robustness of prior specifications in a Bayesian setting. Path induction can be also be used to assess frequentist characteristics of arbitrary procedures — due to space limitations we refer the reader to [20] for examples. Multiple evaluation is often called for in frequentist analyses because some criteria, such as the probability of correctly selecting the best populations, are defined in terms of extremal values over a parameter range. If the parameter values that yield the extremal value are unknown, then one must evaluate at many values to locate the extreme. Path induction can provide significant speedups in such settings, compared to the standard use of backward induction which works through all of the states. For example, for a-population fully sequential allocation there are $\binom{n+2a-1}{2a-1} \approx n^{2a-1}/(2a-1)!$ terminal states, versus $\approx n^{2a}/(2a)!$ total states, so each evaluation via path induction is roughly a factor of $n/(2a)$ faster than an evaluation via backward induction.

4 Sensitivity Analysis: An Application of Path Induction

A common argument used to dismiss procedures built on Bayesian foundations is that the prior distribution may be misspecified. This could mean that the prior assumes too much knowledge and overpowers the data or it may simply mean that the prior is not a reasonable representation of reality. In studying adaptive procedures of the sort proposed here, it is the Bayesian framework that allows us to specify and locate optimal procedures. Since an investigator may not have prior information or may not wish to be using Bayesian design, it is important to have an idea as to how prior assumptions affect various characteristics of the procedure. In other words, it is valuable not only to examine how efficient a Bayesian procedure is when the prior parameters have been "misspecified", but also to evaluate how well it performs according to frequentist measures.

Here we apply path induction (Section 3.3) to the study of robustness in the product of means example. This example is from [19], in which sequentially estimating polynomial functions of means is considered. In particular, we examine the impact of departing from the prior parameter configurations used in the design of a procedure. The goal is to answer questions such as:

> Suppose we determine the optimal allocation procedure associated with a certain prior distribution on the problem parameters. Will the performance be degraded if one assesses the procedure with a significantly different prior distribution?

The evaluation uses two sets of prior parameters, the $\mathcal{A}$ parameters, $[(\mathcal{A}_1, \mathcal{A}_2); (\mathcal{A}_3, \mathcal{A}_4)]$, and the $\mathcal{B}$ parameters, $[(\mathcal{B}_1, \mathcal{B}_2); (\mathcal{B}_3, \mathcal{B}_4)]$, where the subscripts 1 and 2 refer to the beta parameters for arm one and 3

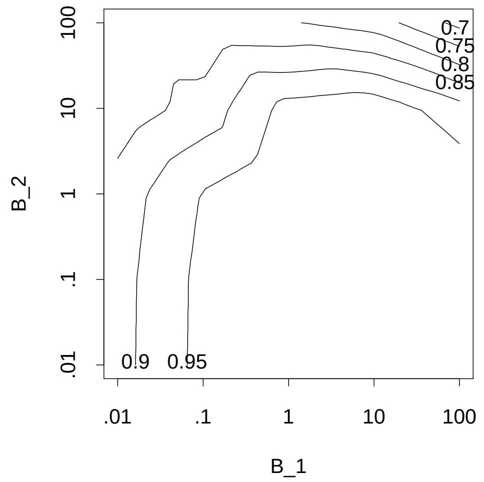

Figure 1: Relative efficiency, product of means, $n = 100$. Design prior $\mathcal{A} = [(1,1);(1,1)]$.

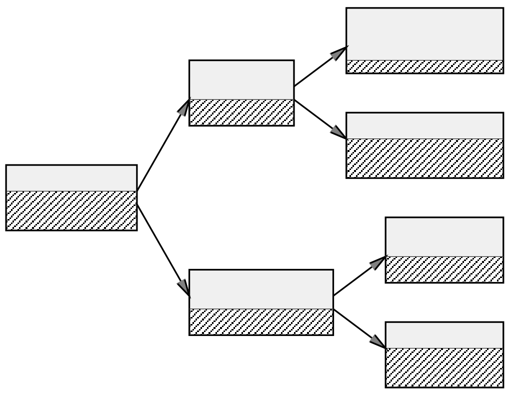

Figure 2: A 3-stage Procedure. Width represents stage size, shading indicates proportions.

and 4 refer to the beta parameters for arm 2. The analysis goes as follows:

(1) Generate the optimal procedure for the $\mathcal{A}$ parameters.

(2) For each choice of $\mathcal{B}$ parameters,

- (i) Using the $\mathcal{B}$ parameters, compute the operating characteristic, which in this case is the Bayes risk, of the procedure from (1).
- (ii) Compute the Bayes risk of the optimal procedure for the $\mathcal{B}$ parameters.
- (ii) The *relative efficiency* is the ratio of risk obtained from (ii) with that obtained from (i).

Figure 1 is an interpolated surface plot of the relative efficiencies, where the $\mathcal{A}$ parameters were fixed at $[(1,1);(1,1)]$, and 25 configurations of the $\mathcal{B}$ parameters were used: $\mathcal{B}_1 = 0.01,\ 0.1,\ 1,\ 10,\ 100$, $\mathcal{B}_2 = 0.01,\ 0.1,\ 1,\ 10,\ 100$, $\mathcal{B}_3 = 1$, and $\mathcal{B}_4 = 1$. For example, the value at B_1 = 0.01, B_2 = 0.01 is .865, which is the relative efficiency of the procedure generated using the uniform distribution but evaluated using $\mathcal{B} = [(0.01, 0.01);(1,1)]$ versus the optimal procedure generated and evaluated using this $\mathcal{B}$ distribution.

Overall, the data in Figure 1 suggest that if a conservative prior (uniform) is used to design the allocation procedure, then even if the values of the prior parameters differ by an order of magnitude from those used in analyzing the results, the procedure will still be quite efficient. The efficiency under such departures is typically above 90%, although when the deviation from the design parameters is closer to two orders of magnitude the efficiency reduction may be unacceptable.

The optimal sampling procedure for this problem was generated via dynamic programming, using the $\mathcal{A}$ design parameters. Then it had was re-evaluated for each set of $\mathcal{B}$ parameters. Using path induction, instead of backward induction, for the 25 re-evaluations saved an order of magnitude in the time. This improvement grows as the sample size grows.

5 Few-Stage Procedures

In some settings, fully sequential procedures may be neither possible nor desirable. For example, responses may not be immediate and new allocations may have to be made before earlier results have been observed.

An important special case of this is when several allocations need to be made simultaneously. In such settings, one may perform *staged* or *grouped* allocation, in which sampling is specified for groups of experimental units at one time. At the end of a stage, information is updated and allocations are specified for the next stage. Staged allocation also permits randomization within each stage, another attribute often desired in practice.

There are many ways to formalize an s-stage allocation with fixed sample size, and here we take the most general approach. At the start of a stage, based on all of the observations accrued to date, the number of observations to sample from each population is specified. Thus, not only may the proportions allocated to each population depend on outcomes of the previous stages; but also the *length* of a stage can adapt to observed data. Because such procedures are highly efficient for small s, most researchers are interested in the cases for which $s = 1$, 2, or 3, which we refer to as being *few-stage*. Figure 2 depicts flow options of a 3-stage procedure. The two shades within the rectangles (the stages) represent the different proportions sampled from each population within the stage. So, in the first stage approximately $\frac{1}{3}^{rd}$ of the observations are from population 1 and the rest are from population 2. In the second stage there are two rectangles which represent just a couple of the many ways one could sample in the next stage. While the third stage must bring the total sample size to n, there is still flexibility in the proportions sampled from the populations.

Despite their practical importance, with the exception of a few very special cases, we know of no previous exact optimizations or evaluations of few-stage procedures. Part of the reason for this may be the counter-intuitive fact that they are more complex to optimize than are fully sequential procedures. While dynamic programming can be used, the most straightforward approaches are impractical. In [17, 21], we provide efficient, although more complicated, dynamic programming algorithms for optimizing s-stage procedures with 2 Bernoulli populations and arbitrary objective function. The algorithms take $\Theta(n^3)$ time for $s = 1$, $\Theta(n^5)$ time for $s = 2$, and $\Theta(sn^6)$ time if $s \geq 3$. Note that the calculations for the first and last stages are significantly easier than for intermediate stages, in that either their start or end is predetermined. Situations in which the total sample size, itself, is random are also addressed in [21], as are situations where the stage sizes are fixed.

There are a number of questions one might ask about few-stage procedures. For any given problem, and for any given number of stages, s, naturally arising questions include:

1. How well a given s-stage procedure compares to the optimal s-stage or to the optimal fully sequential procedures;

2. How well the optimal s-stage procedure compares to the optimal $(s + 1)$-stage procedure; and

3. How do the stage lengths of the optimal s-stage procedure grow with the total sample size?

There has been a tendency to use asymptotic analyses to address the first of these questions, but the answers came in weak forms indicating, for example, that a given 2-stage procedure is first-order optimal or a given 3-stage procedure is second-order optimal, [11]. For any specific sample size, however, these results do not indicate whether the given procedure is 95% of optimal or merely 50% of optimal. The type of asymptotic results available for the last two questions are even weaker.

In [17, 21], we apply our general algorithms to various problems to determine the efficiency of optimal 1-, 2-, and 3-stage procedures, as compared to the optimal fully sequential procedure. In this section, we use the product of means example to illustrate the results we have observed. When the prior distributions are uniform, efficiencies are presented in Figure 3. Here one can see that, by the time n reaches about 20, the optimal 2-stage procedure achieves approximately 97% efficiency and the optimal 3-stage procedure is virtually fully efficient. Similar results were observed over a wide variety of parameter configurations and for the bandit problem. These results are consistent with the asymptotic analyses that suggest that little efficiency is lost when well-chosen 2- and 3-stage procedures are used.

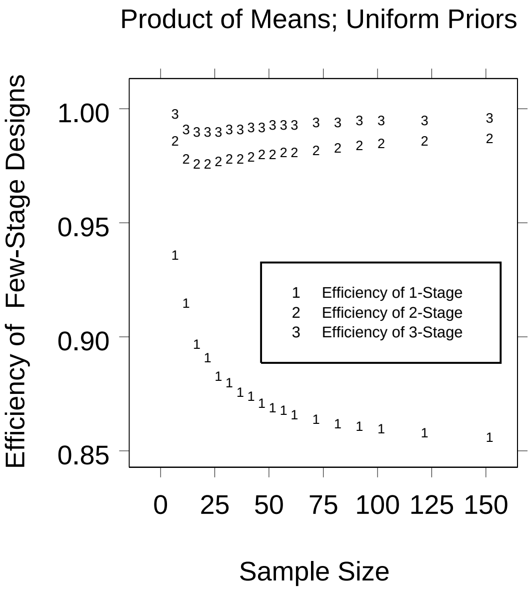

Figure 3: Efficiency of 1-, 2- and 3-Stage Procedures for Product of Means with Uniform Priors

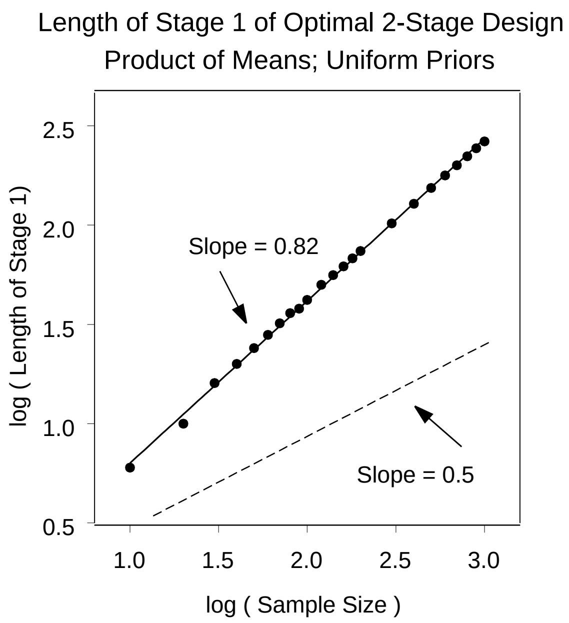

Figure 4: Growth of Stage 1 of 2-Stage Procedure for Product of Means with Uniform Priors

The work in [17, 21] also revealed unexpected results, namely that the initial stages of optimal few-stage procedures are much larger than those suggested in the literature. In general, for example, being told that a stage size grows like the square root of n is not useful when $n = 100$. At this point the constant associated with the growth rate becomes critical to the selection of a good procedure. We have used the product of means problem here because it is one of the rare scenarios for which we have been able to locate growth rate guidelines in the literature.

In a Bayesian analysis of the two-stage product of means problem, Rekab (1992) states that one should take $\lim_{n\to\infty} \frac{L_{1:2}}{n} = 0$ and $\lim_{n\to\infty} L_{1:2} = \infty$ where $L_{1:2}$ is the length of stage one of a 2-stage procedure. These suggestions are clearly too vague to implement, but a more in-depth examination of the frequentist version of the same problem is provided by Noble (1990). The author suggests that $L_{1:2}$ should grow like $\Theta(n^{0.5})$, and he provides some direction for choosing appropriate constants. Noble's guidelines, while frequentist, suggest that for uniform priors and $n = 100$, $L_{1:2}$ should be about 14. In [17], however, we found that the optimal value is 42; and that, in the range $n = 10$ to 1,000, the optimal $L_{1:2}$ grows at a rate which is closer to linear than to the square root. Figure 4 is a log-log plot of the optimal first stage length versus the sample size for a 2-stage procedure with uniform priors. The line fit to the points represents growth at a rate of $n^{0.82}$. The other line on the plot represents growth of $n^{0.5}$. It should be noted that Noble's guidelines are for a procedure in which the terminal estimator does not utilize the observations in the first stage, an unusual design which may help explain why his suggested first stage sizes are so much smaller than those optimal for a Bayesian estimator which utilizes all observations.

More recently, the two-stage product of means problem was analyzed in Zheng et al. (1998) in a frequentist setting. While their analytic results hint that an optimal growth rate for $L_{1:2}$ is roughly $n^{0.5}$, their simulation studies confirm our observations that the growth rate is actually higher than this. We do not know the asymptotic growth rate of the optimal $L_{1:2}$, and finding it is a challenging problem, though it may well be irrelevant for practical sample sizes.

6 Constrained Switching

Investigators are occasionally concerned that fully sequential procedures may switch sampling among the different populations with an unguarded frequency, [28]. Unconstrained, sequential procedures optimize the objective function but ignore practical considerations such as cost, timing and convenience. In some instances, switching among the different populations may not even be possible due to the need to reserve equipment or to finish up a batch of some product needed for the experiment. In an industrial setting, for example, it is often necessary to reconfigure fixtures each time a switch among populations occurs. In a clinical setting there may be similar setup or training costs inherent in each switch among treatment alternatives. There are a number of ways to model the ill effects of switching and here we consider just a few possibilities:

1. There is a setup cost α_i to change to population i, and an incremental cost β_i for each observation as long as you continue sampling from population i. Here we assume that $\alpha_i \gg \beta_i$.
2. If batches of a treatment with a short life span must be prepared or if one can conduct several identical tests concurrently, then you decide that the next m observations are to be sampled from population i and a cost of $\alpha_i + \beta_i m$ is incurred. Here, m is specified in advance.
3. There is an upper bound, S, on the number of times you can switch during the experiment.

While cost structures of the sort listed can be important to investigators, they are seldom incorporated into experimental procedures. One exception comes up in certain control theory problems [1, 22] in which cost structures such as (1) are utilized. However, the results in [1, 22] are applicable only to the special case in which there is geometric discounting of an infinite horizon and no terminal objective. We have been unable to locate research that applies to more general sequential problems such as those allowing arbitrary objective functions, finite horizons and flexible mechanisms for modeling switching considerations.

For the 2-population Bernoulli response fully sequential setting, we developed dynamic programming algorithms that produce procedures which optimize objective functions under constraints on either the maximum or expected number of switches [18]. To establish the impact that switching costs had on sample problems, we use trade-off curves that allow us to examine the relative efficiency of procedures that switch less often than the unconstrained optimal ones do.

We use the 2-AB problem to illustrate the type of results one might observe. First, as a function of n, we calculated the expected number of switches made by the optimal 2-AB procedure that ignores switching considerations. For a variety of different prior parameter configurations, it was found that the expected number of switches appears to grow somewhat faster than logarithmically, but slower than n^α for any α. Note that, despite this, a power growth may still provide a good approximation for sample sizes up to a few hundred. It's also of interest to note that the maximum number of switches grows linearly in n.

Next, we imposed fixed costs for switching between populations and obtained the optimal 2-AB strategies under these conditions. In Figure 5, the optimal tradeoffs between the expected number of failures and the expected number of switches are plotted for the case in which $n = 200$ and both prior distributions are uniform. These tradeoffs are expressed in terms of the efficiency of the objective function. Note that one expects to switch about seven times if the experiment follows the optimal sampling procedure. If however, by imposing costs, the expected number of switches is limited to, say, only one, then the efficiency of the experiment is diminished by only about 2%. This suggests that imposing constraints can significantly reduce switching without correspondingly reducing the efficiency of the procedure.

A similar argument holds when the switching consideration is to constrain the maximum number of switches allowed. In Figure 6, information comparable to that in Figure 5 is plotted, but this time the focus is on the worst-case number of switches. While the unconstrained optimal procedure may switch at nearly

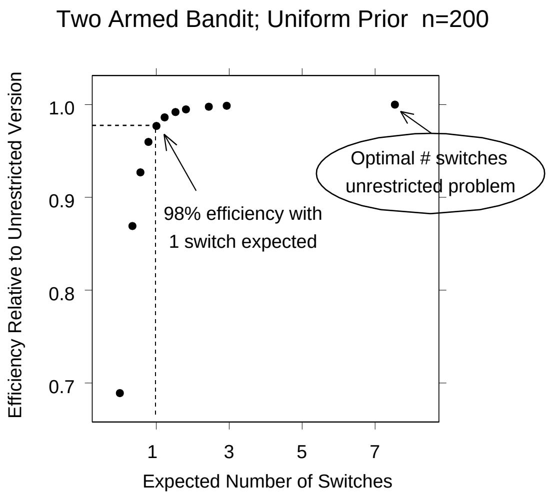

Figure 5: Efficiency vs. E(# Switches)

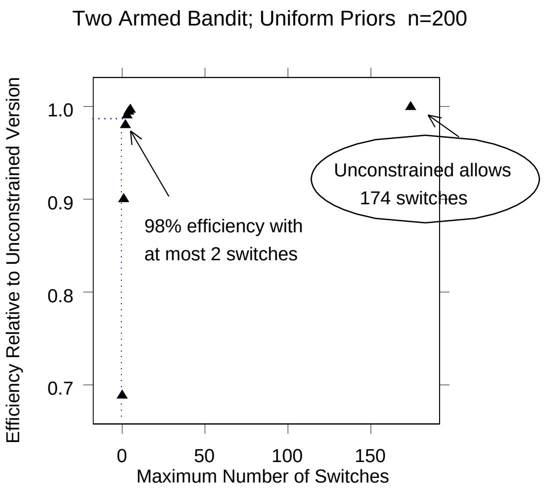

Figure 6: Efficiency vs. Max(# Switches)

every step, the efficiency is reduced by only a couple of percent if the maximum number of switches is held to only two. If at most one switch were allowed, then the efficiency would drop to about 90%.

Similar results hold for the product of means estimation problem which has a completely different form of objective function (see [18]). A critical point to note, however, is that to obtain efficiencies of the level presented here, the timing of the switches must be as dictated by the optimal rule for this constraint. While it might be preferable to enable the investigator to *choose* when to allow the switches, this could seriously impact the efficiency of the procedure.

A direction for future research is to study the characteristics of the optimal, but constrained, procedures. In doing so, one would hope to develop ad hoc procedures that retain the spirit, and hence the efficiency, of the optimal procedures but which also allow the investigator more freedom in calling the shots, i.e., switches. A quite different variation we are examining is to optimize procedures in which there are concerns about time trends. In such settings, occasional switching may be an aid in separating time effects from population effects.

7 Further Remarks

As was stated, the emphasis here is not to proselytize about specific solutions to specific applications, but rather to illustrate that a wide range of approaches and criteria may be relevant. We hope to allow investigators significant flexibility in choosing the relevant criteria and approaches used. Further, by carrying out evaluations of the sort needed with sufficient speed, we wish to encourage investigators to consider procedures that incorporate criteria that typically have been ignored or merely approximated. For example, robustness evaluations such as those shown in Section 4 are very helpful, as are the multiple re-evaluations that allow us to study distributions of important design characteristics. Path induction enables such analyses.

This research program is an ongoing effort, with several new avenues currently being explored. One of these avenues is the extension of few-stage procedures to sample sizes larger than can be handled exactly. In this situation, we are looking at taking exact solutions to smaller problems and extrapolating them to obtain good solutions to larger ones. This approach is producing better solutions than were previously obtainable through asymptotics, especially since most asymptotic analyses give so little specific guidance. Other areas of investigation include the development of algorithms for adaptive procedures for situations in which

- the responses are delayed and hence not available before allocations must be decided,
- covariate information needs to be incorporated (e.g., dose response settings),
- equal allocation approaches are optimized for criteria such as stopping time [15], and
- censoring of observations occurs [13].

Progress has been made on all of these, although much work (and more variations) remains to be done.

Finally, we remind the reader that the different algorithms correspond to different classes of procedures or analyses, and not to different objective functions. Thus an algorithm for optimizing, say, 3-stage allocation is significantly different than one for optimizing fully sequential allocation. However, once the algorithm for a class of procedure has been developed, adapting to different objective functions, such as bandits versus product of means, is a minor change. Occasionally one can exploit special properties of the objective function to simplify calculations (this is discussed in [21]), but all of our descriptions herein are for general objectives.

Acknowledgments.

We greatly appreciate the comments and recommendations by our referees. The parallel program for the 3-population fully sequential model was developed by Robert Oehmke, using facilities provided by the University of Michigan's Center for Parallel Computing.

References

[1] AGRAWAL, R., HEGDE, M.V., AND TENEKETZIS, D. (1988), "Asymptotically efficient adaptive allocation rules for the multiarmed bandit problem with switching costs", *IEEE Trans. Auto. Cont.* **33** 899–906.

[2] ARMITAGE, P. (1985), "The search for optimality in clinical trials", *Int. Statist. Rev.* **53** 15–24.

[3] BELLMAN, R. (1956), "A problem in the sequential design of experiments", *Sankhya A* **16** 221–229.

[4] BERRY, D.A. AND EICK, S.G. (1995), "Adaptive assignment versus balanced randomization in clinical trials — a decision-analysis", *Statist. in Med.* **14** 231–246.

[5] BERRY, D.A. AND FRISTEDT, B. (1985), *Bandit Problems: Sequential Allocation of Experiments*, Chapman and Hall.

[6] Betensky, R. A. (1996), "An O'Brien-Fleming sequential trial for comparing three treatments", *Annals of Statistics* **24** 1765–1791.

[7] BRADT, R. AND KARLIN, S. (1956), "On the design and comparison of certain dichotomous experiments", *Ann. Math. Statist.* **27** 390–409.

[8] BURINGER, H., MARTIN, H. AND SCHRIEVER, K.H. (1980), *Nonparametric Sequential Selection Procedures*, Birkhauser.

[9] Coad, D. S. (1993), "Sequential allocation involving several treatments", *Adaptive Designs* (N. Flournoy & W. F. Rosenberger, ed.'s), Institute Math. Stat. Lec. Notes **25**, pp. 95–109.

[10] CHENG, Y. (1994) "Multistage decision procedures", *Seq. Anal.* **13** 329–349.

[11] HALL, P. (1981), “Asymptotic theory of triple sampling for sequential estimation of a mean”, *Ann. Statist.* **9** 1229–1238.

[12] HARDWICK, J., OEHMKE, R. AND STOUT, Q.F. (1997), “A parallel program for 3-arm bandits”, *Comp. Sci. and Statist.* **29** 390–395.

[13] HARDWICK, J., OEHMKE, R. AND STOUT, Q.F. (1998), “Adaptive allocation in the presence of censoring” *Computing Science and Statistics* **30**, to appear.

[14] HARDWICK, J. AND STOUT, Q.F. (1991), “Bandit strategies for ethical sequential allocation”, *Comp. Sci. and Statist.* **23** 421–424.

[15] HARDWICK, J. AND STOUT, Q.F. (1993), “Optimal adaptive equal allocation rules”, *Comp. Sci. and Statist.* **24** 597–601.

[16] HARDWICK, J. AND STOUT, Q.F. (1993), “Exact computational analyses for adaptive designs”, *Adaptive Designs* (N. Flournoy & W. F. Rosenberger, ed.’s), Inst. Math. Stat. Lec. Notes **25** 223–237.

[17] HARDWICK, J. AND STOUT, Q.F. (1995), “Determining optimal few stage allocation rules”, *Comp. Sci. and Statist.* **27** 342–346.

[18] HARDWICK, J. AND STOUT, Q.F. (1996), “Sequential allocation with minimal switching”, *Comp. Sci. and Statist.* **28** 567–572.

[19] HARDWICK, J. AND STOUT, Q.F. (1996), “Optimal allocation for estimating the mean of a bivariate polynomial”, *Seq. Anal.* **15** 71–90.

[20] HARDWICK, J. AND STOUT, Q.F. (1998), “Path induction for evaluating sequential allocation procedures”, *SIAM J. Sci. Comp.*, to appear.

[21] HARDWICK, J. AND STOUT, Q.F. (1998), “Optimal few-stage allocation procedures”, submitted.

[22] HOFRI, M. AND ROSS, K.W. (1987), “On the optimal control of two queues with server setup times and its analysis”, *SIAM J. Computing* **16** 399–420.

[23] JONES, P. (1992), “Multiobjective Bayesian Bandits”, *Bayesian Statistics 4: Proc.* 4^{th} *Valencia Int’l Meeting*, 689–695.

[24] NOBLE, W. (1990), *First Order Allocation*, Ph.D. Thesis, Michigan State Univ.

[25] PAGE SHAPIRO, C. (1985), “Allocation schemes for estimating the product of positive parameters”, *J. Amer. Statist. Assoc.* **80** 449–454.

[26] PALMER, C. (1993), “Selecting the best of k treatments”, *Adaptive Designs* (N. Flournoy & W. F. Rosenberger, ed.’s), Inst. Math. Stat. Lec. Notes **25** 110–123.

[27] REKAB, K. (1992), “A nearly optimal 2-stage procedure”, *Comm. Stat. – Theory Meth.* **21** 197-201.

[28] SCHMITZ, N. (1993), *Optimal Sequentially Planned Decision Procedures,* Springer-Verlag Lecture Notes.

[29] SIMON, R. (1977), “Adaptive treatment assignment methods and clinical trials”, *Biometrics* **33** 743–744.

[30] THALL, P.F., SIMON, R., AND ELLENBERG, S.S. (1989), "A two-stage design for choosing among several experimental treatments and a control in clinical trials", *Biometrics* **45** 537–547.

[31] WANG, Y.-G. (1991), "Sequential allocation in clinical trials", *Comm. Stat. – Theo. Meth.* **20** 791–805.

[32] ZHENG, S., SEILA, A.F., AND SRIRAM, T.N. (1998), "Asymptotically risk eficient two stage procedure for estimating the product of $k(\geq 2)$ means", *Statistics and Decisions*, to appear.