In *Journal of Sequential Planning and Inference* 137 (2007), 2654–2665

Response Adaptive Designs that Incorporate Switching Costs and Constraints

Janis P. Hardwick Quentin F. Stout
University of Michigan, Ann Arbor, MI 48109-2121 USA

Abstract

This paper examines the design and performance of sequential experiments where extensive switching is undesirable. Given an objective function to optimize by sampling between Bernoulli populations, two different models are considered. The *constraint* model restricts the maximum number of switches possible, while the *cost* model introduces a charge for each switch. Optimal allocation procedures and a new "hyperopic" procedure are discussed and their behavior examined. For the cost model, if one views the costs as control variables then the optimal allocation procedures yield the optimal tradeoff of expected switches vs. expected value of the objective function.

Our approach is quite general, applying to any objective function, and gives users flexibility in incorporating practical considerations in the design of experiments. To illustrate the effects of the switching restrictions, they are applied to the problems of minimizing failures and of minimizing the Bayes risk in a nonlinear estimation problem. It is observed that when there are no restrictions the expected number of switches in the optimal allocation grows approximately as the square root of the sample size, for sample sizes up to a few hundred. It is also observed that one can dramatically reduce the number of switches without substantially affecting the expected value of the objective function. The adaptive hyperopic procedure is introduced and it is seen to perform nearly as well as the optimal procedure. Thus one need sacrifice only a small amount of statistical objective in order to achieve significant gains in practicality. We also examine scenarios in which switching is desirable, even beyond that which would occur in the optimal design, and show that similar computational approaches can be applied.

Keywords: response adaptive sampling, switching, bandit problem, hyperopic design, optimal tradeoffs, sequential allocation, dynamic programming, experimental design

1 Introduction

In situations where data are collected over time, response adaptive sampling or allocation, in which decisions are made based on accruing data, is more efficient than fixed sampling, where all decisions are made in advance. Response adaptive sampling can reduce costs or time, or improve the results for a given sample size. However, fully adaptive designs are rarely used, due to various concerns over their design, analysis, and implementation.

In some settings, one concern is that many designs, such as randomized allocation, switch repeatedly among the alternatives, a design attribute that may be costly or impossible [13]. Banks and Sundaram note [3] "it is difficult to imagine a relevant economic decision problem in which the decision-maker may costlessly move between alternatives". For example, in an industrial setting, one may need to reconfigure fixtures each time a switch occurs. In a clinical setting, similar setup or training costs may be required to switch among treatment alternatives. More generally, important scenarios that involve switching include:

i. You have an initial cost α_i to change to population i, and incur incremental cost β_i for each observation as long as you stay on that population, where $\alpha_i \gg \beta_i$.

ii. You have to make up a treatment in batches, or need to set up fixtures to conduct several identical tests at the same time, or sign a contract committing to purchasing a specific number of units, so when you decide that the next m observations are on population i you incur a cost $\alpha_i + \beta_i m$. Here m is specified in advance.

iii. You can switch at most S times. For example, you may need to use a special apparatus to which you have only S accesses.

Note that these are not mutually exclusive. For example, both i) and iii) may be true in the same situation.

Such cost and constraint concerns can be quite important, although they are rarely directly incorporated in designs. One exception is in bandit theory where, in trying to minimize expected failures, costs, or time, a few authors have investigated a cost structure as in i) [1, 3, 5]. However, these results apply to the case in which an infinite number of geometrically discounted observations are assumed. Another analysis of switching costs appears in [4], where the setting is quite specialized with one of two populations having a known probability of success. Here again the objective is to minimize the expected number of failures. Perhaps the most relevant work appears in [6], in which the authors introduce an asymptotically optimal heuristic for a finite sample size bandit problem with switching costs.

There appears to be no previous work that applies to more general sequential problems that may involve arbitrary objective functions. In particular, none applies to nonlinear objective functions. Further, earlier work addresses only concern i). Basic questions one might ask in any setting include:

- How many switches occur in the optimal design which does not consider the relevant switching concerns?
- How much do we lose by reducing the number of switches (perhaps dramatically)?
- How do we model this and can we optimize the problem?
- Are there other (simpler) designs that come close to the optimized switching designs?

In Section 2, basic definitions are given. In Section 3.1 a constraint model is defined, corresponding to concern iii) above. In Section 3.2 a cost model is defined, generalizing concern i). For both models, algorithms are given that generate the optimal design for arbitrary objective functions, and which can evaluate arbitrary designs with respect to these objectives. In Section 4 we introduce a new intuitive procedure based on an adaptive *hyperopic* approach. In Section 5 we focus on the specific objective functions of (a) minimizing expected failures and (b) minimizing the Bayes risk in a nonlinear estimation problem. We determine the optimal designs with and without switching restrictions, and compare these to hyperopic designs. These comparisons include both Bayesian and frequentist evaluations. In Section 6 we briefly examine quite different, although related, situations in which switching is desirable. In the final section, we conclude with discussion of generalizations and observations concerning the results. While no work will be done here on cost model ii), note that it can be viewed as a staged allocation problem and hence can be optimized by the techniques developed in [9].

2 Definitions

While a table of notation is presented in Figure 6, we also review notation here. Throughout, we assume that the sample size N is fixed. This assumption merely simplifies our analyses and examples, and the algorithms can easily be adapted to include stopping rules. If optional stopping is desired (which may be quite natural in cost models), then N should be interpreted as the maximum possible sample size.

There are K Bernoulli populations, and at any point the only decision required is to choose which of these to observe. We use a Bayesian approach, where the success parameters of the populations have independent prior distributions. However, while this approach is used to create the allocation design, the design may have desirable frequentist properties as well; and for all designs we can easily evaluate operating characteristics from either perspective. Suppose that at some point we have observed s_i successes and f_i failures on Population i. Then the vector $(s_1, f_1, \ldots, s_K, f_K)$ is a sufficient statistic and forms a natural index for the state space describing the experiment. Note, however, that it does not provide sufficient information to determine the expected or worst-case number of switches.

There is an objective function $R^*(v)$ which is the value of each final state v (i.e., states for which $|v| = N$), and the goal is to minimize the expected value of R^*. The *expected value of allocation* $\mathcal{A}$, denoted $\mathcal{R}_\mathcal{A}$, is the sum, over all final states v, of $R^*(v)$ times the probability that $\mathcal{A}$ reaches v. For a given restriction, as in Sections 3.1 and 3.2, let *opt* denote the optimal allocation. Within the restriction, the *efficiency of allocation procedure* $\mathcal{A}$ is $\mathcal{R}_{opt}/\mathcal{R}_\mathcal{A}$. When there are no switching restrictions then *opt* can be determined exactly using straightforward dynamic programming. However, when there are restrictions the situation becomes more complicated.

3 Models

3.1 Constraint Model

In the *constraint model* there is an upper bound S on the number of times one can switch among the populations to be sampled (the initial observation is not counted as a switch). The goal is to optimize the expected value of the objective function subject to this constraint. When $S \geq N-1$, the problem is equivalent to the standard optimization problem without switching considerations.

A dynamic programming algorithm for determining the optimal allocation procedure under the constraint model is described in detail in Figure 7 in the Appendix. It proceeds from the end of the experiment towards the beginning, where at each state it evaluates the value of continuing with no switching vs. the value of switching and then continuing with 1 fewer switch allowed. Note that the algorithm actually finds the value of the optimal experiments corresponding to all constraints less than or equal to S, not just the optimal experiment for S. This is quite useful, since it allows one to examine the range of optimal tradeoffs of maximum switches vs. expected objective function all from a single run. With straightforward changes, the algorithm can also be used to determine the value and efficiency of an arbitrary allocation procedure.

3.2 Cost Model

In the *cost model*, if the last population sampled was i, and we now sample j, then we pay a cost $c(i,j)$ (if the first observation is on i then we pay $c(i,i)$). The goal is to minimize the expected value of the terminal objective plus costs. This is a flexible model that includes cost concern i) of Section 1 as a special case. As with the constraint model, a dynamic programming algorithm for determining the optimal procedure for a given cost function is described in Figure 8 in the Appendix, and here too the algorithm can easily be modified to evaluate arbitrary or suboptimal allocation procedures.

A particularly important special case occurs when $c(i,i) = 0$ and $c(i,j) = \alpha$, $i \neq j$, for then the cost component is proportional to the number of switches. Let R^{α}_{opt} denote the optimal value obtained using this cost function (where *value* is now the terminal objective plus cost), and let C^{α} denote the expected cost of the allocation procedure achieving R^{α}_{opt}. Then $R^{\alpha}_{opt} - C^{\alpha}$ is the optimal expected value of the objective function under the constraint that the expected cost is no more than C^{α}, i.e., under the constraint that the average number of switches is no more than C^{α}/α. Thus this model achieves optimal tradeoffs of the objective function vs. expected number of switches, but does so through the indirect control parameter α. To investigate a specific expected number of switches, one must search through α. One can use the monotonic decreasing behavior of C^{α}/α to greatly speed this search.

4 Adaptive Hyperopic Allocation

While the algorithms in the Appendix can be used to determine the optimal designs for the constraint and cost models, they are often viewed as being too difficult to compute for non-Bernoulli distribution functions, or resulting in designs that are too complex and opaque. Hyperopic procedures are often efficient ad hoc alternatives to optimal adaptive designs. A hyperopic procedure is one that makes sampling decisions based only on the remaining trial length and the prior/posterior distributions. For the cost model we use a simple adaptive hyperopic procedure based on 1-stage allocation procedures. For simplicity, let us assume that we have only 2 populations. Then, for an experiment with m observations remaining and given posteriors, let $R_{\mathit{fix}}(i,j)$ be the value of the fixed allocation rule that puts i observations on Population 1 and j observations on Population 2, for $i + j = m$ and $0 \leq i,j \leq m$. If $R_{\mathit{fix}}(m_1, m_2) = \min\{\, R_{\mathit{fix}}(i,j) \,|\, 0 \leq i,j \leq m,\ i+j = m \,\}$, then the pair (m_1, m_2) is the *best fixed* allocation rule.

Using these simple allocation rules, one can create families of adaptive hyperopic rules as follows:

1. At each state v, having made $N - m$ observations, find the optimal best fixed allocation (m_1, m_2) based on the posteriors and the fact that there are m remaining observations.
2. Use (m_1, m_2) to decide which population to sample at v.
3. Update the posteriors and sample size remaining and repeat.

While the decision made at each state is based on a fixed sampling procedure, these hyperopic rules are adaptive because they update the decision process after each observation.

Note that in step 2 there are many ways to utilize the latest best fixed allocation values (m_1, m_2). For example, suppose we last sampled from Population 1.

Design ignoring costs:

a. Sample next from the population corresponding to $\max\{m_1, m_2\}$, or
b. Use a biased coin with P(allocate to Population 1) $= m_1/m$

Design incorporating switching and sampling costs:

Assuming that population 1 was just observed. For $i+j=m$, let $S(i,j)$ be defined by

$$S(i,j) = \begin{cases} R_{\mathit{fix}}(m,0) + m \cdot c(1,1) & \text{if } i=m \\ R_{\mathit{fix}}(i,j) + i \cdot c(1,1) + c(1,2) + (j-1) \cdot c(2,2) & \text{if } i<m \end{cases}$$

That is, $S(i,j)$ is the cost of staying on population 1 for i more observations and then switching to population 2. Redefine (m_1, m_2) to be $\text{argmin}\{S(i,j) \mid 0 \le i,j \le m,\, i+j=m\}$, and switch only if $(m_1, m_2) = (0, m)$.

The specific hyperopic procedure we use, $\mathcal{H}$, is the latter variation, incorporating both switching and sampling costs.

The advantage of $\mathcal{H}$ over the optimal allocation is that one does not need dynamic programming to determine what to do at each state, but can instead perform a simple calculation to make the decision. The disadvantage, of course, is that it is not optimal. The important question is how much efficiency is lost by using this simple procedure. These issues will be examined in Section 5.

While $\mathcal{H}$ is defined in terms of the cost model, it can also be used to produce designs for the constraint model. The switching costs can be used as tuning variables, where a higher cost results in a lower maximum number of switches. One could easily vary this approach by having the cost of a switch increase as the number of switches increases. However, we have not evaluated such rules.

5 Examples

To show that the models of the previous sections are practical, they are applied to two objective functions. For both we assume that $K=2$ and that the priors are independent beta distributions.

The first is the problem of minimizing expected losses or failures during the experiment. This can be modeled by the classical *2-armed bandit* problem with finite horizon and uniform discounting. We choose this example because it is well-known and has been widely studied in a variety of scenarios, although we could find no prior work on the expected number of switches. As noted in the introduction, there is interest in switching aspects of bandit models, but analytical results seem to focus on models with infinite horizon and geometric discounting [1, 3, 4, 5]. Our work can be considered as a compliment to such results, giving exact optimizations and evaluations for experiments of finite duration.

The second example is a nonlinear estimation problem. Here the objective is to minimize the Bayes risk of an estimate of the *product* of two population means using squared error loss. This problem has also been studied by several authors [7, 11, 12], but we could find no prior work that discussed switching aspects. Note that for the first example one would like all of the observations to be from the better population, while for the second, extensive observations are needed from both populations.

We have evaluated the optimal and hyperopic procedures for a variety of priors and sample sizes, but due to space considerations we only show results for a couple of scenarios. Other cases examined exhibit similar behavior. Throughout we assume that there are no sampling costs, only switching costs, i.e., that $c(1,1) = c(2,2) = 0$. Recall that we use a fixed sample size N to simplify comparisons, but can easily modify the algorithms to use nonzero sampling costs and optional stopping.

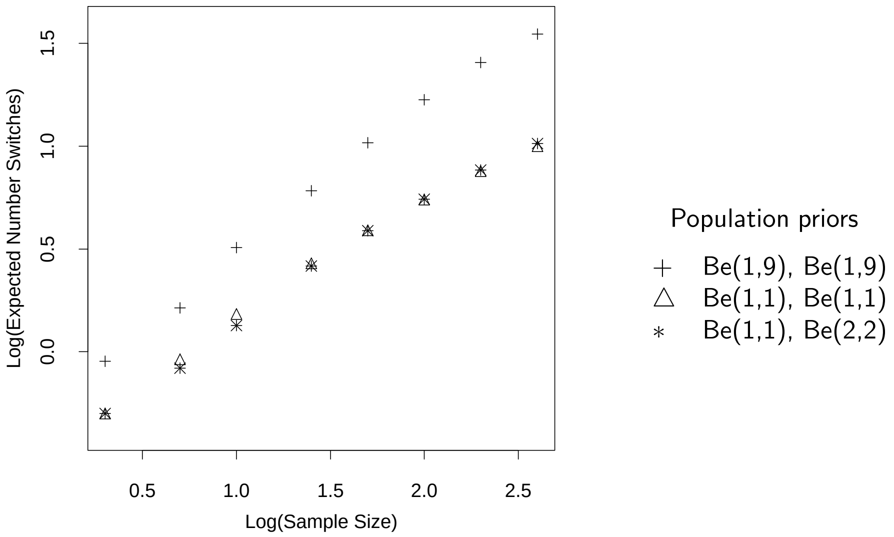


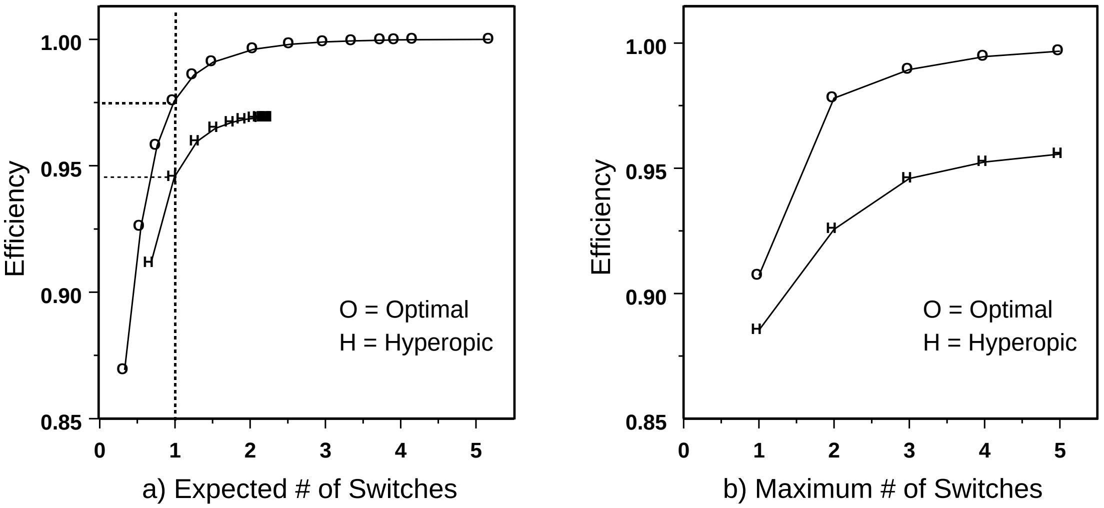
Figure 1: Minimizing Failures, No Switching Considerations

5.1 Minimizing Failures

To begin, in Figure 1, the expected number of switches when using the optimal sequential solution with no switching restrictions is plotted as a function of the sample size N. Three different prior combinations are included to illustrate model variation. Regardless of prior, the number of switches grows roughly as the square root of N for the range of N considered, although the rate appears to slow as N increases. Note that for classical randomized designs this rate is even greater, growing linearly with the sample size. The maximum number of switches is nearly N in all cases and is not shown.

The data in Figure 2 arise from a sample size of 100 when uniform priors are used for each population. In Figure 2 a), the optimal tradeoffs between the expected value of the objective function and the expected number of switches are plotted, as are the tradeoffs obtained by using the hyperopic procedure. The efficiency measure here is simply the ratio of the optimal rule with no switching costs to the objective with varying switching costs. The reference lines highlight the relative efficiencies when the expected number of switches is one. Figure 2 b) shows the corresponding tradeoffs involving the maximum number of switches. The extremal points have been omitted in order to present the regions of most interest. When the number of switches is unlimited then the optimal design has 35.08 expected failures and an average of 5.49 switches. The maximum number depends upon the method used to break ties. For example, suppose the observations are always failures. While this has an extremely small probability, it will nonetheless affect the maximum number of switches. If Population 1 is tried first then the optimal decision is to next try Population 2. With a failure on it, the optimal rule could choose either. If it, say, always breaks ties by trying Population 1, then, in the presence of repeated failures, 99 switches will occur. On the other hand, if ties are broken by staying on the current population, then only 50 will occur.

With unrestricted switching the hyperopic procedure averages 36.20 failures with an average of 2.26 switches, and the worst case analysis is as for the optimal procedure. Thus, the hyperopic rule



$N = 100$, Uniform priors on both populations

Figure 2: Optimal Tradeoffs: Expected number of failures

is 97% efficient compared to the optimal when switching is not considered in the design. When the number of switches is 0 then both populations have 50 expected failures, giving an efficiency of 0.70. However, with merely 1 switch on average, the efficiency jumps to 94.5% with the optimal rule being 97.5% efficient. When the maximum number of switches is constrained then the performance is somewhat worse, as Figure 2 makes clear, but it is still true that with only 1 switch the optimal design is 90.7% efficient, and the hyperopic procedure is 88.5% efficient. In Figure 3 we give more detailed information about these designs.

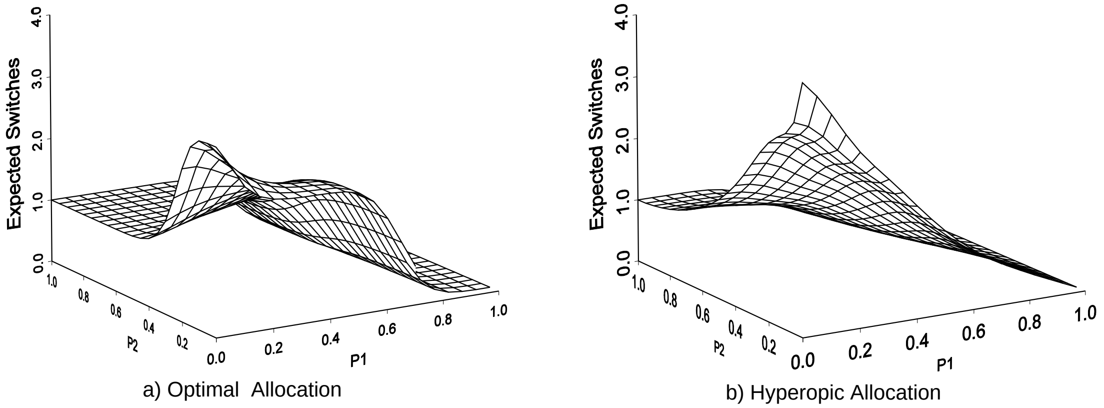
We determined the pointwise behavior (as opposed to integrated) for a number of operating characteristics for the present objective and procedures. For example, for each procedure we generated statistics such as the variance and the maximum number of switches. We also have the expected number of observations allocated to each population as well as the standard deviation of this number. Such pointwise behavior is illustrated in Figure 4 where the expected number of switches are plotted as a function of the success parameters. In this figure, the priors were Be(1,1) and Be(1,1.25) and, on average, the procedure allowed only 1 switch. Note that while the procedures were Bayesian, they nevertheless have reasonable behavior throughout the entire parameter range, i.e., from a frequentist perspective.

When minimizing failures, the hyperopic rule $\mathcal{H}$ simplifies to merely comparing staying on the current arm for the rest of the trial vs. immediately switching to the other arm and staying with it ($S(m, 0)$ vs. $S(0, m)$ in the notation of Section 4). This is because the linearity of the objective function implies that if some observations were planned on each arm, then it would be better to just make all of them on the arm perceived to have the higher success rate. Note that if there are no switching costs then the hyperopic rule is the same as the standard myopic rule, always choosing the arm with the highest posterior probability of success. In general, of course, the hyperopic and myopic rules are not the same.

With regard to the general results for the hyperopic rule, it may be surprising to note that, for this objective, the asymptotic efficiency of $\mathcal{H}$ is less than 1. For example, for uniform priors, if the

	Optimal				Hyperopic			
	E(fail)	SD(fail)	E(swit)	SD(swit)	E(fail)	SD(fail)	E(swit)	SD(swit)
Unrestricted	35.08	24.40	5.49	6.72	36.12	24.26	2.70	4.87
E(swit) = 1	35.96	24.15	1.00	0.82	37.10	24.93	1.00	0.94
Max(swit) = 1	38.67	25.96	0.55	0.50	39.63	26.02	0.44	0.50

Figure 3: $N = 100$, Uniform priors on both populations, Minimizing Failures



a) Optimal Allocation

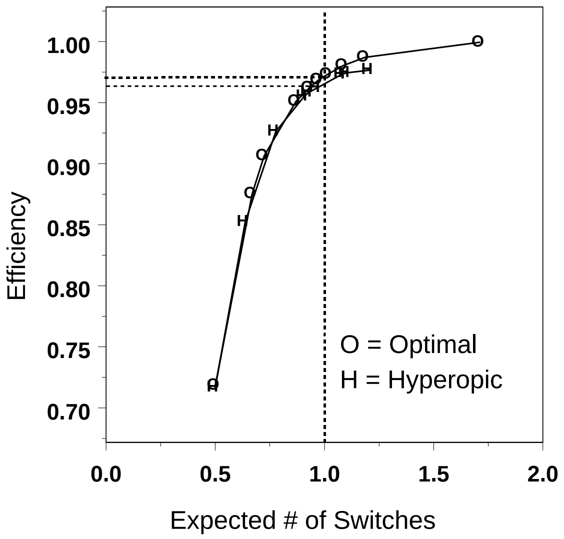
b) Hyperopic Allocation

Figure 4: E(Switches) = 1, $N = 100$, $P_1 \sim Be(1,1)$; $P_2 \sim Be(1,1.25)$

first observation is from Population 1 and results in a failure then $\mathcal{H}$, and the optimal design, will switch to Population 2. There is a nonzero probability that the posterior estimate for Population 2 is always greater than 1/3 no matter how many observations occur, in which case $\mathcal{H}$ will never decide it is worthwhile to switch back to the first population, even if the switching costs are zero, despite the fact that there is a nonzero probability that Population 1 is actually the better one. The optimal procedure, however, will continue to occasionally sample Population 1. One can guarantee asymptotic efficiency of 1 by more sophisticated hyperopic designs where at each step one uses the best 2-stage design, rather than the best 1-stage design, to guide the sampling decision. However, these analytical results say little about the behavior for useful sample sizes, and the examples make it clear that this extra complication is unnecessary.

5.2 Bayes Risk

The problem of estimating the product of two success probabilities arises in reliability settings, and, as noted, it has been studied several times. The objective function is the Bayes risk of the squared error loss of the terminal estimate of $P_1 \cdot P_2$, where P_i is the success probability of population i. In Figure 5 the optimal tradeoffs of objective function vs. expected number of switches are shown. Notice that, for this nonlinear objective, the hyperopic strategy performs significantly better than it did for the linear objective. If we restrict the procedure to having only one switch on average, the optimal and hyperopic rules are respectively 97% and 96% efficient. With an average of only 1.7 switches, the optimal rule is 99.7% efficient. That is, it is essentially fully efficient at 1.7 expected switches. With no switching costs at all, the optimal and hyperopic rules max out at 6.6 and 1.3



$N = 100$, priors are Be(1,2) and Be(2,1)

Figure 5: Optimal tradeoffs: Bayes risk for product of means

average switches respectively. Note how few switches are used by the no-cost hyperopic rule as compared to the no-cost optimal rule.

With a nonlinear objective function such as is encountered here, the hyperopic rule does not simplify as it did for the problem of minimizing failures. That is, the optimal fixed allocation rule may allocate some observations to each population, reducing the uncertainty about P_i for each. One could use the simpler hyperopic rule which only compares staying on the current arm vs. switching immediately and staying with the other arm, but our experimental results indicated that this is markedly inferior to the hyperopic rule $\mathcal{H}$ described in Section 4. One can view each step of $\mathcal{H}$ as asking if the best fixed allocation would take a few more observations on the current arm before switching to the other arm, and switching only if the answer is no.

We examined many other behaviors for this estimation problem. However, for space considerations, we don't include more results here. The basic operating characteristics observed for the bandit problem are quite similar to those seen to occur for this problem.

6 Desirable Switching

In some situations there are experiments for which the switching concerns are quite the opposite of those just analyzed. For example, it may be that the success probabilities of populations are likely to change over time, and thus each needs to be periodically reexamined. Continuing the list of potential design requirements from the Introduction, it may be the case that

iv. No population can remain unobserved for more than m observations.

v. Each population must be observed at least r times.

vi. At least S switches must occur.

Note that these design goals do not necessarily preclude the simultaneous incorporation of the earlier conditions. Also, the present concerns can also apply in scenarios in which the success probabilities do not change. For example, stipulation v) can arise when a design has good Bayesian properties but the frequentist behavior is poor at some regions of the parameter space. Note that restriction v) only weakly implies switching, in that it can be achieved with a single switch.

Switching requirements may also arise when a heuristic design is being used to minimize failures and one wishes to avoid premature convergence to an inferior population. Premature convergence is an extremely general problem in the area of optimization, and so "exploration rules" are introduced to prevent it. For example, hill-climbing approaches which find a local maximum are occasionally restarted at different regions in that hope that eventually one will get closer to a global maximum. In simulated annealing the temperature parameter controls the exploration, while in evolutionary algorithms it is the mutation rate or introduction of new random individuals that is used to achieve this. Some of the failure minimization designs examined in [14] have such exploration characteristics.

Optimal designs that address the issues in v) and vi) have the property that they satisfy the constraints as soon as possible. That is, there is no advantage in postponing the required observations since they might in fact help improve later sampling decisions. However, using large values of r or S can also degrade the expected value of the objective function since they force observations of inferior populations.

It is concern iv) that most appropriately addresses situations where the population success probabilities might change over time. One can satisfy iv) by specifying that at every multiple of m one samples once from each population, and then returns to the standard design rules. When the goal is to minimize failures the condition might be relaxed somewhat and be modified to do such switching at logarithmically increasing intervals. It is known that for static success probabilities the optimal design samples each population at least a logarithmic number of times [10], and so such prespecified switching need not significantly increase the number of failures even when the population means are unchanging. There are more sophisticated heuristics that sample each population at least a logarithmic number of times and achieve slightly better performance [2, 10].

One way to address a more sophisticated version of iv) is to introduce a cost function $r(i, \ell)$ which is the cost of not having sampled from population i for the last ℓ observations. If one sets $r(i, \ell) = \infty$ for $\ell \geq m$ then iv) will be satisfied. However, by using a gradually increasing function, instead of a step function, one can model more appropriate tradeoffs between insuring repeated sampling on all populations vs. the cost of doing so. That is, if switching causes little change in the expected value of the objective function then it would occur more frequently than if it caused a large change.

To determine the optimal design and evaluate an arbitrary design for such a cost function, the notion of state has to be extended to indicate how long it has been since each population has been sampled, and then straightforward changes to the algorithm in Figure 8 can optimize the design. One complication is that the number of states increases by a factor of N^{K-1}, making it difficult to compute when the sample size and number of populations are large.

7 Final Remarks

Practical considerations are important in the conduct of experiments, so it is useful to offer investigators ways to directly address such considerations in the design of their experiments. One such consideration is the extensive switching that commonly occurs with many experimental designs. This paper has addressed this concern by examining allocation methods that examine objective function vs. switching consideration tradeoffs. Given an arbitrary objective function, and given either switching costs or switching constraints, the algorithms provided determine the optimal sequential experiment for the resulting model. In some cases an investigator may not utilize the sequential allocation procedure that optimizes a tradeoff, but may want to use it as a benchmark against which suboptimal designs are evaluated. This was illustrated through the introduction of the adaptive hyperopic design, which achieved nearly optimal performance for both linear and non-linear objective functions. For a nonlinear objective the hyperopic design achieves its performance by being more subtle then just comparing switching now vs. never switching.

The cost model can easily be extended to depend on the number of observations so far, and on the outcome of the observation. This would allow one to optimize interesting cases such as bandit problems with non-uniform weights. One can also merge the algorithms to optimize the expected objective function plus switching costs, under a constraint on the maximal number of switches allowed. We also examined scenarios for which switching was *desirable*, showing that they fit into the same optimization and evaluation framework with relatively simple changes.

As illustrative examples we examined the problem of minimizing failures and a nonlinear estimation problem with the goal of minimizing the Bayes risk. It was observed that sequential designs optimized without regard for switching considerations tend to have extensive switches, but that the number of switches can be dramatically reduced with only minor loss of efficiency in the objective function.

For both problems it was observed that the expected number of switches for unrestricted optimal sequential designs grows fairly rapidly, roughly on the order of the square root of the sample size, for sample sizes of a few hundred. Near the upper end of this region the growth seems to be slowing, which leads us to believe that the asymptotic rate is far slower. Thus it is expected that purely asymptotic results would poorly predict the observed behavior. Because asymptotics often give weak guidance for the design of specific experiments, we believe that computational insight and optimization fills an important role. However, to better fill that role, it helps to have computational approaches that model all of the factors that are relevant to the investigator. This work is just a small piece in a larger project to develop such models and programs.

Acknowledgements This research was supported by in part by National Science Foundation grant DMS–0072910.

References

[1] Assawa, M. and Teneketzis, D. (1996), “Multi-armed bandits with switching penalties”, *IEEE Trans. Auto. Control* **41**: 328–348.

[2] Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002), “Finite-time analysis of the multi-armed bandit problem”, *Machine Learning* **47**: 235–256.

[3] Banks, J.S. and Sundaram, R.K. (1994), “Switching costs and the Gittins index”, *Econometrica* **62**: 687–694.

[4] Benzing, H., Kalin, D., and Theodorescu, R. (1987), "Optimal policies for sequential Bernoulli experiments with switching costs", *J. Inform. Process. Cybernet.* **23**: 599–607.

[5] Bergemann, D. and Välimäki, J. (2001), "Stationary multi-choice bandit problems", *J. Economic Dynamics & Control* **25**: 1585–1594.

[6] Brezzi, M. and Lai, T.Z. (2002), "Optimal learning and experimentation in bandit problems", *J. Economic Dynamics & Control 27*: 87–108.

[7] Hardwick, J. and Stout, Q.F. (1993), "Optimal allocation for estimating the product of two means", *Computing Science and Stat.* **24**: 592–596.

[8] Hardwick, J. and Stout, Q.F. (1996), "Sequential allocation with minimal switching", *Computing Science and Stat.* **28**: 567–572.

[9] Hardwick, J. and Stout, Q.F. (2002), "Optimal few-stage designs", *J. Stat. Plan. and Infer.* **104**: 121–145.

[10] Lai, T.L. and Robbins, H. (1985), "Asymptotically efficient adaptive allocation rules", *Advances in Applied Math.* **6**: 4–22.

[11] Page, C. (1987), "Sequential designs for estimating products of parameters", *Seq. Anal.* **6**: 351–371.

[12] Rekab, K. (1993), "A sampling scheme for estimating the reliability of a series system", *IEEE Trans. Reliability* **42**: 287-290.

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

[14] Wang, Y.-G. (1991), "Gittins indices and constrained allocation in clinical trials", *Biometrika* **78**: 101–111.

Appendix: Algorithms for Determining Optimal Allocation

The following algorithms first appeared in [8]. They find the optimal design given the options available. However, they can also be used to evaluate an arbitrary design. This is done by replacing the minimum operations, which choose the best population to sample from, by the choice that the design would make.

In Figure 7, a dynamic programming algorithm is given for determining the optimal allocation design when there can be no more than S switches. As is usual, it proceeds from the end of the experiment towards the beginning. The critical observation is embedded in the equations at the innermost level of the loops. At each state v and number of switches remaining σ, one evaluates the value of staying on the current population vs. switching to a different population, where there is one fewer switch available for the remainder of the experiment. The option giving the minimum value is then chosen.

In Figure 8 a dynamic programming algorithm is given for minimizing the sum of objective function and sampling costs. Similarly to the previous algorithm, the critical recursive equations are at the innermost level, where the choice is made by taking the minimum of the cost of staying on the current population and continuing optimally vs. switching and paying a switching cost and then continuing optimally. As the calculations proceed one keeps track of both the sum of objective plus switching costs, which is the quantity being minimized, and the switching costs. At the very end the switching costs are subtracted to determine the expected value of the objective function for each initial population to be sampled, and the minimum of these is chosen.

The algorithms in Figures 7 and 8 can be modified to evaluate an arbitrary design $\mathcal{B}$. To do so, one merely needs to replace the minimization operation with the choice that $\mathcal{B}$ would make.

As for the computational resource requirements: for creating the optimal allocation design with no switching constraints, using standard dynamic programming, $\mathcal{R}_{opt}$ can be determined in $\Theta(KN^{2K}/(2K)!)$ time since there are $\Theta(N^{2K}/(2K)!)$ states, each requiring the evaluation of K alternatives. Here the Θ notation from computer science is used to indicate that it is simultaneously an upper and lower bound, to within a multiplicative constant. We assume that R^* can be computed for all terminal states in time proportional to the number of terminal states, and since it is only computed $\Theta(KN^{2K-1}/(2K-1)!)$ times it is not the dominant factor. Similarly, to calculate the posterior probabilities of success for every population one needs only determine $\Theta(KN^2)$ values which can be stored in a table, and hence they too are not the dominant factor. For our examples only beta distributions were used, and hence the posterior calculations were trivial. When more complex distributions are used the calculations may be far more complicated, using techniques such as particle filtering, but the number of such calculations is small compared to the number of states.

For the cost model the number of states, and time per state, is the same as for the unconstrained optimization. When constraining switches there are $\Theta(S \cdot N^{2K}/(2K)!)$ states, corresponding to the usual states coupled with the various number of switches that have occurred to reach that state. Each requires the evaluation of K alternatives, so the total time is $\Theta(SKN^{2K}/(2K)!)$.

The algorithms, as given, only determine the value of the optimal design. If one also records the option taken at each state, then the design itself can be recovered, starting at the initial state and following the chosen options. The storage to retain this information is proportional to the number of states, as is the storage of the algorithms as presented. However, if one only wants to determine the value, then one can use the well-known technique of overwriting the arrays as the value of m decreases. This saves a factor of N in the space required.

N: sample size

K: number of populations

S: maximum number of switches possible (constraint model)

R^*: terminal objective function

$\mathcal{R}_{\mathcal{A}}$: expected value of R for allocation $\mathcal{A}$

$R(v)$: expected value of R^*, starting at state v and proceeding optimally (no switching considerations)

$\mathcal{R}_{opt}$: expected value of R for optimal sequential allocation (i.e., $R(\mathbf{0})$)

$R_i^c(v)$: expected value of objective + switching costs obtained by starting at state v, sampling from population i, and proceeding optimally (cost model)

$R_i^\sigma(v)$: expected objective obtained by starting at state v, sampling from pop. i, and proceeding optimally using no more than σ switches (constraint model)

$c(i, j)$: the cost of switching from pop. i to pop. j

$C_i(v)$: expected value of total switching costs obtained by starting at state v, sampling from population i, and proceeding optimally (cost model)

v: a state, that is, a vector denoting number of successes and failures observed on each population

$|v|$: the total number of observations at state v

$\mathbf{s}_i, \mathbf{f}_i$: vectors denoting 1 success or failure on pop. i

$p_i(v)$: probability of success on next observation of population i, when experiment is in state v

Figure 6: Notation

```
for all terminal states $v$ (i.e., states where $|v| = N$)
for all populations $i \in \{1, \ldots, K\}$
for all switches $\sigma \in \{0, \ldots, S\}$
    initialize $R_i^\sigma(v) = R^*(v)$
for $m = n - 1$ downto 0
    for all states $v$ with $|v| = m$
    for all switches $\sigma \in \{0, \ldots, S\}$
    for all populations $i \in \{1, \ldots, K\}$
        $r_{suc} = R_i^\sigma(v + \mathbf{s}_i)$
        $r_{fail} = R_i^\sigma(v + \mathbf{f}_i)$
        if $\sigma > 0$ then
            $r_{suc} = \min\{r_{suc}, \min\{R_j^{\sigma-1}(v + \mathbf{s}_i) : j \neq i\}\}$
            $r_{fail} = \min\{r_{fail}, \min\{R_j^{\sigma-1}(v + \mathbf{f}_i) : j \neq i\}\}$
        $R_i^\sigma(v) = p_i(v) \cdot r_{suc} + (1 - p_i(v)) \cdot r_{fail}$
$\mathcal{R}_{opt} = \min\{R_i^S(\mathbf{0}) : i \in \{1, \ldots, K\}\}$
```

Figure 7: Optimal Experimental Design for Constraint Model

```
for all terminal states $v$ (i.e., states where $|v| = N$)
for all populations $i \in \{1, \ldots, K\}$
    initialize $R_i^c(v) = R^*(v)$
    initialize $C_i(v) = 0$
for all $m = n - 1$ downto 0
    for all states $v$ with $|v| = m$
    for all populations $i \in \{1, \ldots, K\}$
        $jsuc = \mathrm{argmin}\{c(i, j) + R_j^c(v + \mathbf{s}_i) : j \in \{1, \ldots, K\}\}$
        $jfail = \mathrm{argmin}\{c(i, j) + R_j^c(v + \mathbf{f}_i) : j \in \{1, \ldots, K\}\}$
        $R_i^c(v) = p_i(v) \cdot [c(i, jsuc) + R_{jsuc}^c(v + \mathbf{s}_i)] +$
                $(1 - p_i(v)) \cdot [c(i, jfail) + R_{jfail}^c(v + \mathbf{f}_i)]$
        $C_i(v) = p_i(v) \cdot [c(i, jsuc) + C_{jsuc}(v + \mathbf{s}_i)] +$
                $(1 - p_i(v)) \cdot [c(i, jfail) + C_{jfail}(v + \mathbf{f}_i)]$
$\mathcal{R}_{opt} = \min\{R_i^c(\mathbf{0}) - C_i(\mathbf{0}) : i \in \{1, \ldots, K\}\}$
```

Figure 8: Optimal Experimental Design for Cost Model