In *Journal of Statistical Planning and Inference* 104 (2002), pp. 121-145.

Optimal Few-Stage Designs

Janis Hardwick and Quentin F. Stout
University of Michigan, Ann Arbor, Michigan 48109 USA

Abstract

Optimal designs are presented for experiments in which sampling is carried out in stages. There are two Bernoulli populations and it is assumed that the outcomes of the previous stage are available before the sampling design for the next stage is determined. At each stage, the design specifies the number of observations to be taken and the relative proportion to be sampled from each population. Of particular interest are 2- and 3-stage designs.

To illustrate that the designs can be used for experiments of useful sample sizes, they are applied to estimation and optimization problems. Results indicate that, for problems of moderate size, published asymptotic analyses do not always represent the true behavior of the optimal stage sizes, and efficiency may be lost if the analytical results are used instead of the true optimal allocation.

The exactly optimal few stage designs discussed here are generated computationally, and the examples presented indicate the ease with which this approach can be used to solve problems that present analytical difficulties. The algorithms described are flexible and provide for the accurate representation of important characteristics of the problem.

Keywords: sequential analysis, dynamic programming, algorithms, clinical trials, two-stage, three-stage, experimental design, group allocation, adaptive, sampling, bandit, product of means
AMS Subject Classification: 62L05, 62K05, 62L04, 68W01

1 Introduction

It is well known that adaptive sampling or allocation, in which decisions are made based on accruing data, is more efficient than fixed sample allocation, in which all decisions are made in advance. Allocating adaptively can reduce costs or time, or improve the results for a given sample size. Fully sequential adaptive designs, in which one adjusts after each observation, are the most powerful. However, they are rarely used, due to concerns over their design, analysis, and implementation.

While advances in computing hardware and algorithms make it easier to optimize and analyze certain fully sequential designs, and while portable computers make them more accessible, there are still difficulties with implementation. Experimental responses may be too slow for a design requiring full updates prior to each allocation. Further there may be set-up costs that dissuade individual allocations.

One way to address such concerns is to incorporate a restricted form of sequential allocation, in which decisions are made in stages. The most common of these is a 2-stage experiment, in which an initial decision is made to observe specified numbers from the various populations; and then, once the results have been obtained, to make a second and final decision as to how to sample in the last stage. Within each stage, updates are not required so the impact of response delays is minimal.

Both 2- and 3-stage designs have received extensive analytical treatment, and the results typically indicate that the designs are first- and second-order asymptotically optimal respectively. In particular there is a considerable body of literature on 2- and 3-stage designs for obtaining fixed precision confidence intervals and for minimizing risk functions when observational costs are incurred. (See for example Ghosh (1975) and Ghurye and Robbins (1954).) A review of these types of sequential few-stage designs is provided in Ghosh, Mukhopadhyay and Sen (1997).

Despite the volume of this work, however, there doesn't appear to be work in which attempts have been made to fully *optimize* few-stage designs. In particular, two features that we allow to vary freely in the designs described here are the stage lengths and the proportions allocated from each population within each stage. Both sets of design parameters (lengths and proportions) can be critical to the efficiency of a design. With regard to selecting stage lengths as a function of total sample size or total expected sample size, we have found that published analyses are quite vague except in limiting cases which may not be relevant in practice. With regard to how to allocate within each stage, previously published few-stage designs are typically characterized by having equal allocation in the first stage. We make no such restriction and provide examples illustrating that this assumption has the potential to be arbitrarily damaging.

Figure 1 illustrates the manner is which a 3-stage design might flow. The two shades within the rectangles (the "stages") represent the different proportions sampled from each population within the stage. So, in the first stage of Figure 1 we see that approximately $\frac{1}{3}^{rd}$ of the observations are from $\mathcal{P}_1$ and the rest are from $\mathcal{P}_2$. In the second stage there are two rectangles which represent a couple of the many ways one could sample in the next stage. Note in particular that one rectangle is shorter than the other, representing a shorter stage size.

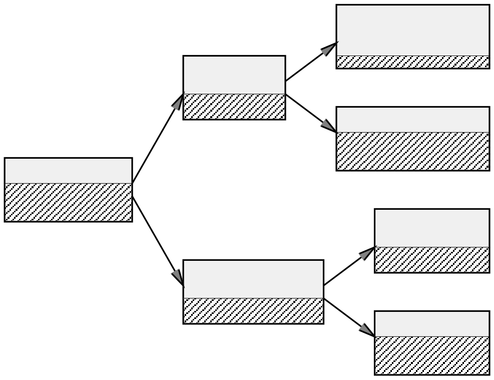

Figure 1: Schematic of a 3-Stage Design

The goals of this paper, which extend the work in Hardwick and Stout (1995), are

- to provide efficient algorithms for determining optimal few-stage designs,
- to motivate the use of such algorithms via a number of examples,
- to show how easily the base algorithm can be adjusted to handle new design variations, and
- to compare the exact computational results with the analytical results that have appeared in the literature.

Definitions used in describing the algorithms are presented in Section 2 and the base algorithms are introduced in Section 3 and Appendix A.

In Section 4 we illustrate the range of our algorithms by applying them to several sample problems. We give optimal solutions for four separate examples and compare our results with the previously best results that appearing in the literature.

Among results of interest, we found that, for some of the problems examined, asymptotic results do not appear to provide useful guidelines in practice. On the other hand, we found that some ad hoc approaches performed remarkably well compared to optimal designs. Perhaps of more interest is the ease with which one can optimize variations as long as the new problem fits roughly within the population model framework utilized here. Many problems that might be extremely difficult or impossible to address analytically may be simple to optimize fully with only minor alterations to the base algorithms described here. This benefit may encourage designers to utilize models which more accurately reflect the important factors in the experiment, rather than choosing models which are analytically tractable.

Many of our examples can be computed by programs more efficient than one would expect from the worst-case scenarios used in the base algorithms. We address this in Appendix B. We also show how to incorporate new constraints such as fixed stage sizes.

Finally, in Section 5, we discuss some extensions of this work and efforts to extrapolate exact optimizations for moderate sample sizes to predict nearly optimal allocations for sample sizes larger than can be fully optimized.

2 Definitions

With the exception of Sections 4.2 and Appendix B, we assume that the total sample size of the experiment, n, is fixed. This assumption is used merely to simplify descriptions and comparisons, and, as Section 4.2 shows, one can modify our algorithms to handle cases where the sample size is random but bounded. There are two independent Bernoulli populations, $\mathcal{P}_1$ and $\mathcal{P}_2$. We use a Bayesian approach, in which the success parameters of the two populations have independent distributions. (In all of our examples these distributions are beta, but our work applies to general distributions.) Thus, at any given point one can determine the probability that the next observation on a given population will be a success. Suppose that at some point in time we have observed s_i successes and f_i failures on $\mathcal{P}_i$. Then the vector (s_1, f_1, s_2, f_2) is a sufficient statistic, and forms a natural index for the state space describing the experiment. States, denoted as v, will be treated as vectors so that one can add observations in a natural manner.

We are interested in k-stage designs in which k is small. In a 1-stage design, the only decision required is the number of observations to sample from $\mathcal{P}_1$, as all remaining observations are sampled from $\mathcal{P}_2$. If $k > 1$, one determines how many observations to take from $\mathcal{P}_1$ and $\mathcal{P}_2$ in stage 1. These are denoted as L_{11} and L_{12} respectively, and the total number of observations in stage 1 is denoted by L_1, where $L_1 = L_{11} + L_{12}$. Once the initial observations have been obtained, one is left with a $(k-1)$-stage experiment of size $n - L_1$, where the priors have been updated to include the initial observations. Without loss of generality, we require that each stage have at least one observation, so $k \leq n$. Our algorithms are correct for all such k and n, but our analyses assume $k \ll n$ since that is the case of interest. If, for example, $k = n$, then the problem is fully sequential and simpler approaches could be used.

k : number of stages

n : sample size, $n \geq k$

s_i, f_i : successes and failures on $\mathcal{P}_i$, $i = 1, 2$

$\mathbf{s}_i, \mathbf{f}_i$: vectors denoting 1 success or failure on $\mathcal{P}_i$; hence $|\mathbf{s}_i| = |\mathbf{f}_i| = 1$

o_i : number of new observations assigned to $_i$

$p_i(s, o; v)$: probability of s successes among o observations on $\mathcal{P}_i$, starting at state v

$R^*_t(v)$: value of starting t-stage experiment, $1 \leq t \leq k$, at state v and proceeding optimally ($R^*_0(v)$ is the objective function)

$R_t(o_1, o_2; v)$: value of starting t-stage experiment at state v, assigning o_i observations to $\mathcal{P}_i$, and proceeding optimally.

L_{ji} : number of observations on $\mathcal{P}_i$ in stage j.

L_j : number of observations in stage j, i.e., $L_j = L_{j1} + L_{j2}$.

Figure 2: Notation

There is an objective function $R^*_0(v)$ that is the value of each final state v (i.e., states for which $|v| := s_1 + f_1 + s_2 + f_2 = n$), and the goal is to minimize the expected value of R^*_0. The *value of allocation* $\mathcal{A}$ is the sum, over all final states v, of $R^*_0(v)$ times the probability of $\mathcal{A}$ reaching v. An *optimal* k*-stage allocation* is a k-stage allocation that achieves the minimum value among all k-stage allocations. The only restriction on the objective function is the requirement that it can be determined by knowing only the final state reached and the prior distributions.

To describe the time and space requirements of algorithms, we use "generalized O-notation" from computer science, in which O and o have the same meanings as in statistical use; and 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 $n \geq N$. Notation used in the remainder of the paper are displayed in Figure 2.

3 Optimal Few-Stage Allocation

The starting point for our algorithms is the simple version given in Figure 3. It proceeds in a typical dynamic programming fashion, from the end of the experiment towards its beginning. In a fully sequential allocation, dynamic programming usually proceeds by analyzing all states with $|v| = n$, then all states with $|v| = n - 1$, and so on until one reaches state $(0, 0, 0, 0)$.

A similar scheme is used here, but there is an additional implicit part of the state space, namely, the number of stages so far. The number of stages, as opposed to the number of observations, is not part of the sufficient statistics, but is a crucial part of the dynamic programming. It controls the outermost loop level, ranging from the last stage towards the first.

The equations in the loops determine the best continuation at any stage and state by taking the minimum over all possible options. In other words,

$$R^*_t(v) = \min\{R_t(o_1, o_2; v) : o_1, o_2 \text{ legal}\}.$$

```
{Evaluate last (kth) stage}
For all states $v$ with $k-1 \leq |v| \leq n-1$,
    $R^*_1(v) = \min_{o_1+o_2=n-|v|} R_1(o_1, o_2; v)$
{Evaluate middle stages}
For $t = 2$ to $k-1$
    For all states $v$ with $k-t \leq |v| \leq n-t$,
        $R^*_t(v) = \min_{1 \leq o_1+o_2 \leq n-|v|-t+1} R_t(o_1, o_2; v)$
{Evaluate initial stage}
$R^*_k(0) = \min_{1 \leq o_1+o_2 \leq n-k+1} R_k(o_1, o_2; 0)$
```

Figure 3: Simple Few-stage Algorithm

"Legal" values are determined by the constraints that there are t stages remaining, each of which must have at least one observation, that $|v|$ observations have already occurred, and that there will be a total of n observations. Thus, the legal values of o_1 and o_2 are those such that

$$1 \leq o_1 + o_2 \leq n - |v| - t + 1 \quad \text{if } t > 1 \quad \text{and} \quad o_1 + o_2 = n - |v| \quad \text{if } t = 1$$

For each stage, one proceeds through the entire range of states. However, the evaluation at each state is more complex than in the fully sequential case. In fully sequential designs, there are only two options that need to be evaluated (sample either from $\mathcal{P}_1$ or $\mathcal{P}_2$), and each of these involves only two successor states. Thus, one can evaluate each state in $\Theta(1)$ time, and complete the design in $\Theta(n^4)$ time (since there are $\Theta(n^4)$ states).

For the few-stage problem, however, there are many options at each stage. In the general case, one must decide the number of observations allocated to $\mathcal{P}_1$ amd $\mathcal{P}_2$, creating $O(n^2)$ options. Further, to evaluate $R_t(o_1, o_2; v)$ one must consider $O(n^2)$ outcomes:

$$R_t(o_1, o_2; v) = \sum_{s'_1=0}^{o_1} \sum_{s'_2=0}^{o_2} p_1(s'_1, o_1; v)\, p_2(s'_2, o_2; v) \cdot\ R^*_{t-1}(v + (s'_1, o_1 - s'_1, s'_2, o_2 - s'_2)),$$

where $p_i(s, o; v)$ is the probability of observing s successes among o observations on $\mathcal{P}_i$, if one started at state v. Thus, if straightforward implementations are used, it takes $O(n^2)$ time to evaluate $R_t(o_1, o_2; v)$; $O(n^4)$ time to evaluate $R_t(v)$ for each state v; and $\Theta(n^8)$ time to evaluate the entire stage over all $\Theta(n^4)$ states. Thus the total time for all stages, using a straightforward implementation of dynamic programming as in Figure 3, would be $\Theta(kn^8)$. The space required would be $\Theta(n^4)$, since all of the results of each stage are needed to compute the preceding one.

In these analyses, and throughout the paper, there is an implicit assumption that one can compute all of the values $p_i(s'_i, o_i; v)$ in time no more than the number of states involved, using space no more than the number of states involved. Similar assumptions are made concerning the terminal cost function $R^*_0(v)$.

In Appendix A it is shown that the number of calculations can be dramatically reduced. Utilizing that work gives the following:

Theorem 3.1 *The optimal k-stage allocation for an experiment of n observations from 2 Bernoulli populations can be determined in*

- $\Theta(n^3)$ *time and* $\Theta(1)$ *space, if* $k = 1$,
- $\Theta(n^5)$ *time and* $\Theta(n^3)$ *space, if* $k = 2$,
- $\Theta(kn^6)$ *time and* $\Theta(n^4)$ *space, if* $k \geq 3$. □

The above is the worst case scenario that will work for any few-stage problem. Often, there are features of the problem that allow us to design a more efficient algorithm. In fact, all of the examples that we consider in Section 4 have characteristics that allowed for faster algorithms. In Appendix B, we discuss ways to improve upon this base result when special design constraints are encountered.

4 Examples and Applications

The few-stage optimization algorithm is applicable to a wide range of problems. We have chosen the particular examples in this section because they are ones for which prior asymptotic or approximate analyses provide a framework for comparison.

4.1 2-Stage Bandit

We begin with a 2-stage *two-armed bandit* example, a problem with a large legacy of associated literature. In a bandit problem the goal is to maximize the total reward obtained when sampling sequentially from among the different available populations or "arms". A Bernoulli bandit is one in which the outcomes from the populations are distributed as Bernoulli random variables which can be thought of as having outcomes "success" or "failure". In this case, one seeks to determine how to sample from the different arms so as to maximize the total number of successes. (Thus the "$\min$" of Section 3 should be a "$\max$".) A two-armed Bernoulli bandit can be a model for a two therapy clinical trial in which one is strongly motivated to cure as many of the subjects in the experiment as possible.

A heuristic for optimizing such a problem is to sample at least some from each population, but to identify the better of the two as quickly as possible and then to sample from it exclusively. This brings us to early conceptualizations of 2-stage designs for clinical trials such as those proposed by Colton (1965). Colton suggested that observations be taken in pairs during the first stage and that all observations in the second stage be sampled from the population that was apparently superior after the first stage. At issue was the length of the first stage size, L_1, which depends not only on the total sample size, n, but also, for Bayesian designs, on the prior distributions. Canner (1970) addresses optimal first stage lengths for the Bayesian case. He analytically determined that, when uniform priors are assumed, the optimal first stage size is approximately $\sqrt{2n+4}-2$. Computationally, he ascertained that, for arbitrary beta priors, the optimal first stage asymptotically grows as the square root of n. Note that the scenarios considered in both Colton (1965) and Canner (1970) are not true bandit set-ups since equal allocation in the first stage is mandatory. The restriction greatly simplifies the analysis and computation, but causes a loss of efficiency.

A staged *1-armed* bandit version of this problem was later approached by Clayton and Witmer (1988). For the 1-armed bandit, the success rate for one of the two populations is assumed known. While this assumption greatly restricts the applicability of the result, the solution is significantly simpler. In 1996, Cheng reported on a 2-stage two-armed Bayesian Bernoulli bandit in which the apparently better population is sampled from exclusively in the second stage (unless there are ties). She provides an upper bound for the number to sample from $\mathcal{P}_1$ and $\mathcal{P}_2$ during the first stage when a total of n observations are to be taken and the prior distributions are beta.

Here, we fully optimize the problem examined by Cheng (1996). Note that, due to the simplified second stage, the time required to compute this optimal 2-stage allocation and its value is reduced from $\Theta(n^5)$ in Theorem 3.1 to $\Theta(n^4)$. Figure 4 provides a comparison of first stage sizes determined by

- the upper bound given in Cheng (1996);
- the optimal 2-stage design constrained to using equal allocation in the first stage, i.e., the case solved in Canner (1970);
- the fully optimized version computed using the algorithms described in Section 3.

The data in the figure arise from using beta prior parameters $p_1 \sim Be(2,1)$ and $p_2 \sim Be(1.5,1.5)$, which is the configuration used in Section 3 of Cheng (1996). Figure 4 illustrates that the stage sizes obtained from the bounds

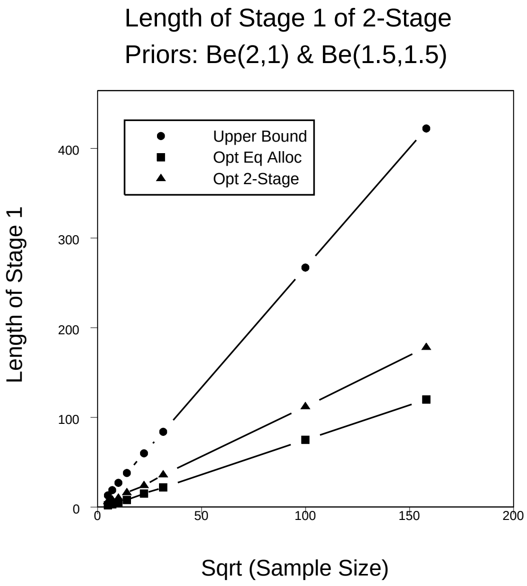

Figure 4: Stage 1 Length for 2-Stage Bandit

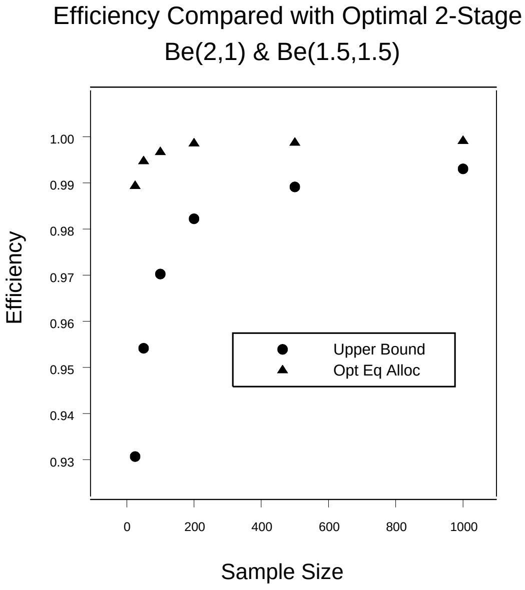

Figure 5: Relative Efficiency of Sub-Optimal Rules

in Cheng (1996) suggest a first stage size that is considerably larger than is needed. In fact, it is interesting to note that L_1, the number used in stage one when sampling optimally, is quite a bit smaller than L_{11}, the number assigned merely to $\mathcal{P}_1$ of stage one by the upper bound rule. The optimal equal allocation procedure, however, selects first stage sizes that are very similar to those called for by the optimal rule. The results in Figure 4 are typical of those obtained using a variety of different prior parameter configurations. In Figure 5, the relative efficiencies of the values of the upper bound and optimal equal allocation rules are shown. The efficiencies are taken relative to the the optimal 2-stage strategy for this problem. Note that the results are better when using optimal equal allocation than when using the upper bound rule. Note also how well the optimal equal allocation procedure performs compared to the fully optimal strategy for this problem. The best equal allocation strategy was computationally determined by Canner in 1970. Thus, for all practical purposes, an excellent solution to versions of this problem were given nearly 30 years ago. Even so, one couldn't ascertain this without having the fully optimal procedure.

Another point to consider is that, while the calculations to obtain optimal stage sizes for the present problem is quite trivial, good analytic solutions for the problem are still being sought a generation later. For example, Cheng, Su and Berry (1998) give a bound that improves on the one in Cheng (1996).

4.2 2-Stage Bandit with Cost and Random Total Sample Size

In this section, we address extensions of the example in Section 4.1 in which we allow the total sample size, n, to be a random variable, and we add a cost c per observation for the first stage. This illustrates the fact that variations such as optional stopping are not difficult to incorporate into the basic few-stage algorithms in Section 3.

When a basic two-stage procedure is applied in a clinical setting, the first stage is thought to represent a controlled clinical trial. The end result the first stage affects the decision as to which of the two treatments is superior and is to be used in the second stage. Whether the second stage is of fixed or random length, it affects the total length of the trial, and there are important questions that involve trial length. In particular, there has been considerable discussion in the literature addressing the tradeoff relationship between the length of a clinical trial and the patient horizon (i.e., all patients who will need treatment). (See Anscombe (1963), Armitage (1985), Bather (1985), Colton (1965), and Simon (1977).) For example, it has been pointed out that the total sample size of a trial could/should incorporate rough estimates of the patient horizon, the rate of introduction of new therapies, the magnitude of the anticipated improvement in the ongoing trial and so forth.

Clearly, there are numerous ways to model the such concerns and the model chosen here is simply an illustration. First suppose that an investigator believes that if the best therapy has a low success rate, then new therapies are likely to be proposed at a faster rate than they would be if the present therapy is already quite good. One can model this simply by assuming that the patient horizon is an increasing function of the success probability of the superior treatment.

Let the total length of the trial, n, range between two specified values $n_1 \leq n_2$, and define the patient horizon to be $n(p^*) = n_1 + p^*(n_2 - n_1)$, where p^* denotes the posterior estimate of the treatment declared superior. (Note that this imposes a debatable upper bound on the horizon for this problem.) Then, the reward function to be optimized is

$$\mathbf{E}\left[p_1 \cdot L_{11} + p_2 \cdot L_{12} - c \cdot L_1 + p^* \cdot (n(p^*) - L_1)\right], \quad (1)$$

where $p^* = p^*(L_1) = \max\{\widehat{p_1}(L_{11}), \widehat{p_2}(L_{12})\}$, and $\widehat{p}_i(L_{1i})$ denotes the posterior estimate of p_i, $i = 1, 2$ based on L_{1i} observations. Note that the reward function is quadratic in p^*. The expectation in (1) is taken with respect to the Bayesian model in which p_1 and p_2 are independent random variables and, conditional on this, the experimental observations are Bernoulli random variables with success rates p_1 or p_2. Although even this simple model may pose interesting analytical challenges, it is quite straightforward to optimize computationally.

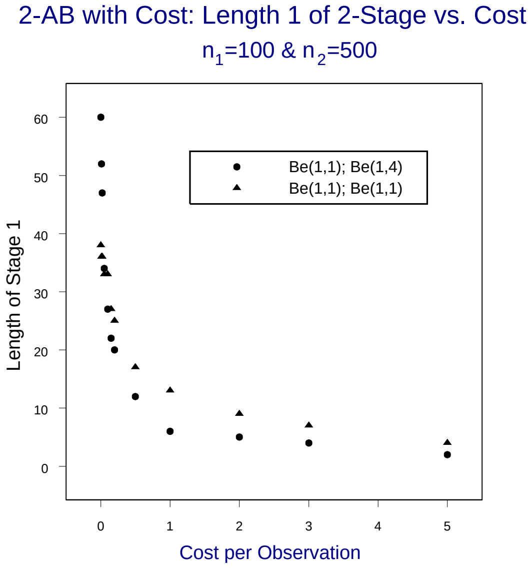

Figure 6: 2-Stage 2-Armed Bandit with Cost per Observation and Variable Sample Size

We tested this model for several parameter configurations. In Figure 6, one sees the exponential-like decay in L_1, the length of the first stage, as the cost per observation in the first stage increases from 0 to 5. Figure 6 provides a comparison of this behavior for two prior parameter configurations given $n_1 = 100$ and $n_2 = 500$ in each case. Using uniform priors, we find that L_1 varies from 4 to 38. There is a corresponding (unshown) variation in $\mathbf{E}[n]$, the expected total sample size, which varies from 349 to 363. In the second case, we take the priors to be $Be(1,1)$ and $Be(1,4)$. Here, the range of L_1 is larger, $2 \leq L_1 \leq 60$, and the range of the sample size varies from 300 to 310. In both cases, most of the drop in sample size occurs when the cost per observation is between 0 and 1.

These results were for a specific model, but many other models can be optimized with similar ease. In particular, the program used to determine the above results is suitable for any model in which the expected patient horizon is determined by the endpoint of the initial stage. One can also easily add costs and variable patient horizons to designs with more stages.

4.3 2- and 3-Stage Nonlinear Estimation

In Sections 4.1 and 4.2, we discussed a 2-stage model in which the allocations within the second stage have a trivial form, i.e., all observations in the second stage are taken from a single population. Here we evaluate a 2-stage design that takes better advantage of the few-stage algorithm in Figure 3 to adjust the allocation proportions in the second stage as well as in the first.

The problem we consider is that of minimizing the mean squared error when estimating the product of the success probabilities of two independent Bernoulli populations. Here again, the success probabilities are modeled as independent beta random variables. This problem has applications in systems reliability and also in estimating area. Note that in nonlinear problems of this nature, it is typically the case that when sampling equally from the two populations one loses considerable efficiency. If, however, one samples differentially, using merely an optimal 1-stage design, then the efficiency can be substantially increased and will continue to increase as the number of stages grows. Several good allocation strategies for versions of the product of means problem have been discussed by Page (1985, 1990, 1995). Also, for the more general problem of estimating any polynomial function of two means, a variety of allocation rules, including the best fixed allocation rule, are compared with the optimal fully sequential strategy in Hardwick and Stout (1996).

While 2-stage sampling rules are not evaluated in Hardwick and Stout (1996), asymptotic solutions for the 2-stage product of means problem have been proposed in Noble (1990), Rekab (1992) and Zheng, Seila and Sriram (1995). In this section, we discuss how the optimal 2-stage procedure relates to the asymptotics suggested by these authors.

Rekab (1992) proposes that the length of the first stage, L_1, be such that

$$\lim_{n\to\infty} \frac{L_1}{n} = 0 \text{ and } \lim_{n\to\infty} L_1 = \infty.$$

This suggestion concurs with the literature on 2-stage designs but is of scant use in determining an optimal or necessarily good L_1 for any specific n. Further, it does not predict the order of growth.

More specific asymptotic guidelines were suggested in Noble (1990) and Zheng, Seila and Sriram (1995). In both articles, the authors take a frequentist approach. Our own formulation is Bayesian, so exact comparisons are not appropriate. However, for moderate n, the design in which both prior distributions are taken to be uniform provides an acceptable basis for comparison.

Noble indicates that the rate of growth of the first stage in a 2-stage design for this problem should be $\Theta(\sqrt{n})$ with upper and lower bounds given by

$$\sqrt{\frac{n}{4\sigma_1\sigma_2}} \leq L_1 \leq \sqrt{\frac{n}{2\sigma_1\sigma_2}} \tag{2}$$

for $\sigma_i = p_i(1-p_i)$, $i = 1, 2$. Applying these bounds, we find, for example, that when $n = 100$,

$$\text{if } \; p_1 = p_2 = \begin{cases} .25, & \text{then } (12 \leq L_1 \leq 16) \\ .5, & \text{then } (10 \leq L_1 \leq 14) \\ .9, & \text{then } (16 \leq L_1 \leq 24). \end{cases}$$

Later, Zheng, Seila and Sriram (1995) provided an approximation that suggests that $L_1 = 2n^{\alpha}$ for α in the range $(.5, (1 - \ln(2)/\ln(n))$, i.e., $2\sqrt{n} < L_1 < n$.

In Figure 7, the optimal size first stage length for the uniform case is plotted for sample sizes ranging from 10 through 1000. These stage lengths closely follow the line $\log_{10}(L_1) = -0.016 + 0.817\log_{10}(n)$. For this range of sample sizes, then, the optimal stage size grows like $\Theta(n^{0.817})$.

These optimal solutions were first reported in Hardwick and Stout (1995). However, independently, Zheng, Seila and Sriram (1995), used simulation to search for a value of α that minimized their approximated mean-squared error. They concluded that a good value for α is $0.8 - (\ln 2/(2\ln n)$, i.e., that $L_1 = \sqrt{2}n^{0.8}$. For $n = 100$, this gives a first stage length of $L_1 = 56$.

In the uniform case, we find that the optimal first stage size is $L_1 = 42$, a value significantly larger than Noble's approximations and somewhat less than that of Zheng, Seila and Sriram. Note, however, that Noble's final estimator does not use information from the first stage, which partially accounts for his suggestion of shorter first stages.

In Figure 8 we have plotted the efficiency of the optimal 1-, 2- and 3-stage designs for this problem as a function of the total sample size, n. We know of no published guidelines for the selection of stage sizes for three stage allocation. In Figure 8, efficiency is measured relative to the optimal fully sequential design. One can see that even the optimal 1-stage design (also known as the *best fixed design*) is not terribly inefficient. It's also of interest to note how extremely efficient both the 2- and 3-stage designs are. The difference in efficiency between the two designs

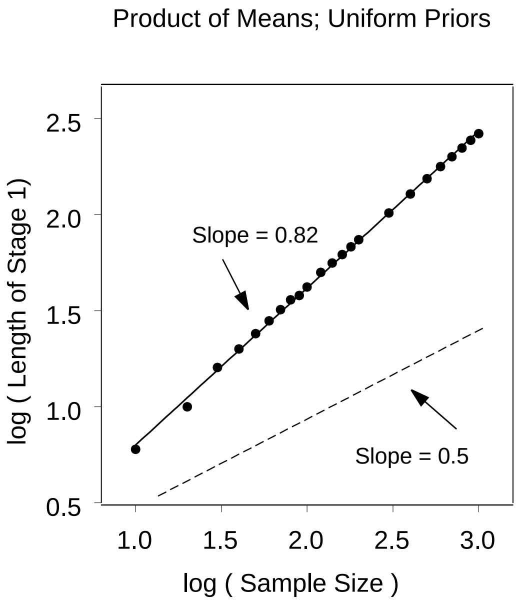

Figure 7: Comparison of Optimal vs. Predicted Growth of L_1 in 2-Stage Experiment

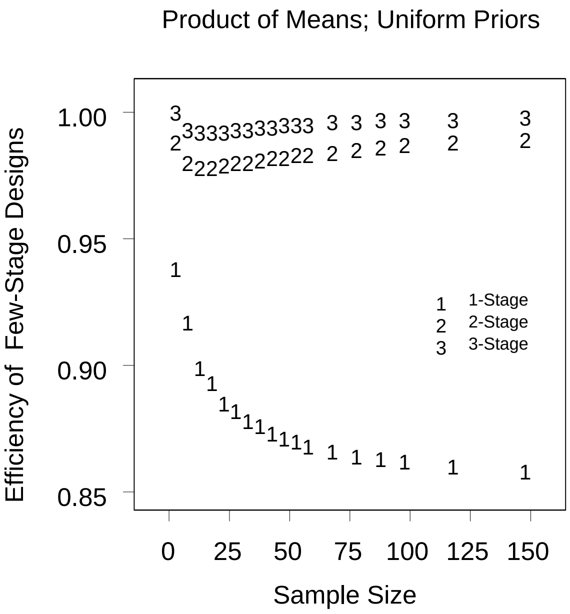

Figure 8: Efficiency of Optimal 1-, 2-, & 3-Stage Procedures Relative to Fully Sequential

is minimal provided that one uses reasonable prior parameter specifications for p_1 and p_2. As we see in the next section, there do, however, exist situations in which optimal 2-stage designs fare much worse than optimal 3-stage designs.

Finally, Figure 9 shows the length of the optimal second stage as a result of the outcome of the first stage, where the total sample size is 100 and uniform priors are used. Note that the optimal second stage does not have a constant length, and has non-monotonic behavior.

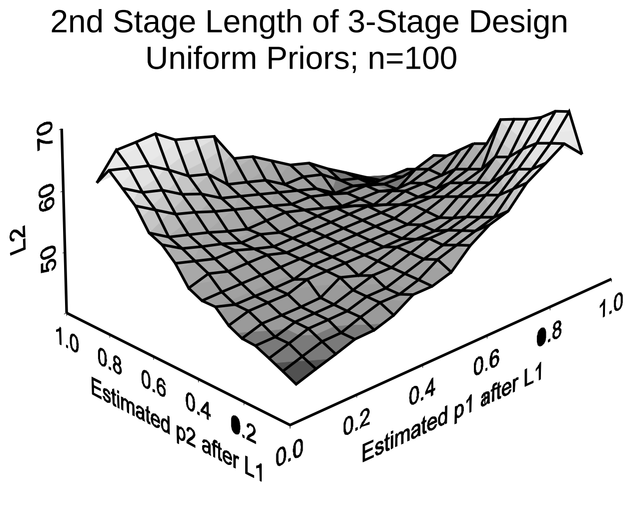

Figure 9: Product of Means, 1st Stage Allocation is 15 for each population

4.4 3-Stage Estimation with Ethical Cost

As noted, the literature on multi-stage designs is not rife with detailed examples of 3-stage allocation procedures. Typically there are critical design features left unspecified. With the present example, we review the issues most often addressed in the classical literature on 3-stage designs and relate them to the capabilities of the algorithms described in Section 3.

The focus here is on an estimation problem from Woodroofe and Hardwick (1991), in which one seeks to minimize the risk of estimating the difference in two population means using a weighted squared error loss plus a cost per failure. The specific function to be minimized is

$$n^2(p_1 - p_2 - (\widetilde{p_1} - \widetilde{p_2}))^2 + n_1(1 - p_1) + n_2(1 - p_2),$$

where $\widetilde{p_i}$ is a consistent estimator of p_i, $i = 1, 2$, and n_i is the number of observations on $\mathcal{P}_i$. Problems of this nature arise, for example, in clinical trials with ethical costs or in destructive industrial testing. In Woodroofe and Hardwick (1991) the observations in the two populations are normally distributed. However, the analytic arguments are essentially the same for the binomial case. This article is of interest because it is among the very few that provide guidelines for the stage sizes of an asymptotically optimal 3-stage design. Thus we have a basis for comparison. Note also that the approach to the estimation problem in Woodroofe and Hardwick (1991) is quasi-Bayesian. While the sequential designs are generated via a Bayesian decision theoretic approach, the allocation rules themselves, as well as the estimators, are independent of the prior distributions used in the analysis of the integrated risk function as long as the priors fall within a fairly broad general class.

We next describe the allocation procedures used in Woodroofe and Hardwick (1991). Let $n_1^*(n; p_1, p_2)$ be the sample size from $\mathcal{P}_1$ that minimizes the risk function when n is to be the total sample size and p_1 and p_2 are known. That is, if p_1 and p_2 were known, then the optimal 1-stage design would be to allocate $n_1^*(n; p_1, p_2)$ to $\mathcal{P}_1$, and $n - n_1^*(n; p_1, p_2)$ to $\mathcal{P}_2$. Next, let $\widehat{p_i}(m)$ be the maximum likelihood estimator for p_i when m observations have been taken from $\mathcal{P}_i$, $i = 1, 2$.

These allocation procedures make use of two positive integer sequences, $L_1(n)$ and $L_3(n)$, $n \geq 5$, which specify the lengths of stage 1 and 3, respectively, for a sample size of n. Letting $L_2(n) = n - L_1(n) - L_3(n)$ be the length of the second stage, these sequences can be arbitrary as long as they satisfy

$$L_1(n) + L_3(n) < n; \quad L_1(n), L_3(n) \text{ are even}; \quad \lim_{n\to\infty} \frac{L_2(n)}{n} = 1; \quad \lim_{n\to\infty} \frac{\sqrt{L_1(n)}L_3(n)}{n \log n} = \infty \qquad (3)$$

To simplify notation, we write $L_i = L_i(n)$, $i = 1, 2, 3$. Recall that L_{ij} is the number sampled during Stage i from $\mathcal{P}_j$.

The sampling goes as follows:

Stage 1: Sample $\frac{L_1}{2}$ from each population (so $L_{11} = L_{12} = \frac{L_1}{2}$).

Stage 2: Sample L_{21} more from $\mathcal{P}_1$ and L_{22} more from $\mathcal{P}_2$, where

$$L_{11} + L_{21} = \min\left\{n - L_{12} - L_3, \max\{L_{11},\ n_1^*(L_1 + L_2;\ \widehat{p_1}(L_{11}), \widehat{p_2}(L_{12})\}\right\} \qquad (4)$$

and $L_{22} = n - L_1 - L_3 - L_{21}$.

Stage 3: Sample L_{31} more from $\mathcal{P}_1$ and L_{32} more from $\mathcal{P}_2$, where

$$L_{11} + L_{21} + L_{31} = \min\left\{n - L_{12} + L_{22},\ \max\{L_{11} + L_{21}, n_1^*(n; \widehat{p_1}(L_{11} + L_{21}), \widehat{p_2}(L_{12} + L_{22})\}\right\} \qquad (5)$$

and $L_{32} = n - L_1 - L_2 - L_{31}$.

We refer to the Bernoulli version of the 3-stage procedures described in Woodroofe and Hardwick (1991) as WH procedures. The WH procedures use a fairly standard technique for determining allocations. The concept applies to both few-stage and fully sequential designs in which optimal 1-stage allocations can be derived as long as certain parameters are specified. (See for example Melfi and Page (1998) and Robbins, Simons and Starr (1967).) The idea is simply to determine the optimal 1-stage allocation using estimators of the unknown parameters, updating the estimators before each new allocation decision is made. Given the decision about the best allocation to be used for the entire experiment, one subtracts the allocations that have already occurred to determine the allocation(s) now to be made. If the estimators are consistent, this sequential process is generally good enough to guarantee asymptotic optimality.

Of interest here are the performance differences between WH procedures and fully optimized 3-stage procedures. Note first of all that WH procedures require equal allocation in the first stage. This can substantially reduce the efficiency of a design when there is a large discrepancy between the population success rates. Next, note that WH procedures are actually a class of procedures, and that one must choose a member of the class by selecting the stage sizes for the experiment in advance. The only information we have to help us do this is provided in (3), and this means that the efficiency of these procedures can vary from being highly efficient to being not very good. To gain a better understanding of how these designs vary in practice, we developed an algorithm to optimize the WH procedures and to evaluate them for arbitrary stage sizes to be input by the user.

In general, we found that optimal WH procedures perform very well compared with optimal 3-stage procedures. Still, without the algorithm that provides an optimal WH procedure, one may have difficulty determining appropriate stage sizes using only the information in (3). Furthermore, one would not be able to assess the efficiency of any WH procedure if one could not determine the fully optimal 3-stage procedure.

As an example, consider a case in which there is significant disparity between the prior estimates of the population success probabilities p_1 and p_2. Figure 10 displays the stage sizes of four different designs for the case in which $n = 50$ and the beta prior parameters are $p_1 \sim Be(1, 10)$ and $p_2 \sim Be(10, 1)$. The last column of Figure 10 gives the efficiency, e, of each design taken relative to an optimal fully sequential design. As expected, the fully optimal 3-stage procedure is virtually fully efficient with $e = 0.9994$. The optimal WH procedure, which specifies very different stage sizes, is also extremely good with $e = 0.9990$. Note, however that if we use a WH procedure guided by, say, the stage sizes used in the optimal 3-stage procedure, we obtain an efficiency of only $e = 0.790$. As it happens, even a good 2-stage procedure outperforms a WH procedure based on guessing the stage sizes using only (3). In particular, the optimal 2-stage procedure for this problem is 0.997 efficient.

Design Type	L_1	E(L_2)	E(L_3)	Efficiency
Optimal 3-Stage	33	4	13	0.9994
Optimal WH	6	40	4	0.9990
Optimal 2-Stage	38	12	–	0.997
WH using guess	34	4	12	0.790

Figure 10: Efficiency of Designs Compared to Fully Sequential Design. $p_1 \sim Be(1, 10)$; $p_2 \sim Be(10, 1)$ and $n = 50$

Despite the results from the previous example, WH procedures generally seem to be quite robust with respect to departures from the optimal WH stage sizes. This suggests that the way the allocations are adapted within the second and third stages may be more important than the actual stage lengths themselves. Another point of interest is that WH procedures don't depend on information in the prior distributions. In one sense this is positive because it allows for the intended frequentist interpretations of the data. It also suggests robustness if one takes a Bayesian interpretation. Recall, however, that this type of built-in robustness leads only to asymptotic optimality. As we saw in the previous example, where the priors were discrepant, the lack of inclusion of prior information in the determination of the sampling strategy had the potential to seriously reduce design efficiency. On the other hand, if the priors for p_1 and p_2 are approximately the same, then most WH procedures will be highly efficient since equal allocation itself is nearly optimal.

To provide some insight as to how big optimal stage sizes for this problem are, Figure 11 gives $L_1(n)$, $\mathbf{E}\left[L_2(n)\right]$ and $\mathbf{E}\left[L_3(n)\right]$ for sample sizes ranging between 10 and 100. Uniform priors are used. Note that, as was seen in

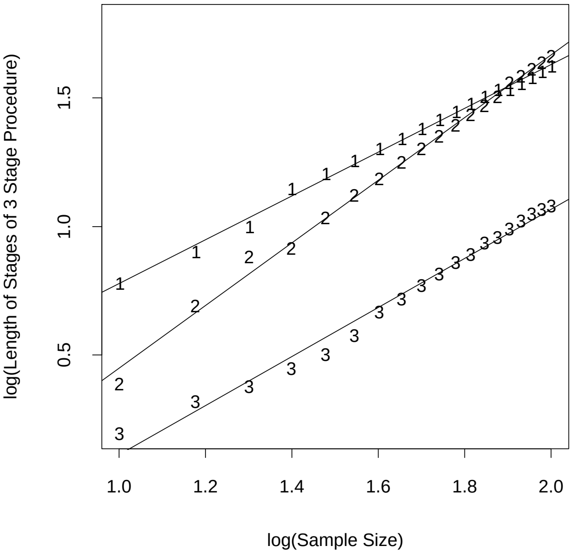

Figure 11: Optimal Stage Sizes in 3-Stage Experiment, Uniform Priors

Section 4.3, the optimal Stage 1 sizes are generally larger than those suggested in the literature, whereas for the problem in Section 4.1, we found that optimal Stage 1 sizes were far smaller than suggested in the literature. There doesn't appear to be a general rule, and this emphasizes the need to be able to calculate optimal stage sizes for various different procedures.

5 Final Remarks

We have shown that it is possible to fully optimize few-stage allocation designs for useful sample sizes. Further, the results of these optimizations indicate that asymptotic guidelines may be quite misleading for reasonable sample sizes, and may not even predict true growth rates. These results, which we found unexpectedly, are not likely to have been uncovered without the ability to perform exact calculations for sample sizes of interest.

The few-stage algorithms developed here can be applied to a wide variety of problems, with flexible optimization goals, stopping rules, etc. Additional points being pursued include sensitivity analysis of few-stage rules, handling multiple populations, modeling censoring, allowing multiple endpoints, allowing additional constraints, and incorporating covariates.

As part of an ongoing project, we are using the algorithms given here, combined with graphical approaches, to visualize aspects of the optimal rules. We hope to achieve a better understanding of the structure of few-stage optimal rules; and, more generally, to gain insight into the structure of good adaptive rules. There are both practical and statistical reasons for this effort. Investigators are often uncomfortable utilizing adaptive allocation schemes for which they have no intuition, such as those described here that are optimized by a computer. Users have a better understanding of, and greater affinity for, simple fixed allocation schemes. However, if a user could explore an adaptive design and gain a better understanding of the decisions it makes, then they might gain enough confidence to utilize the design.

As for the statistical aspects, we believe that exploring adaptive rules for moderate sample sizes can help suggest analyses and designs for much larger sizes. Thus, we hope for a synergistic interplay between analysis, computation, and visualization. For example, for the product of means problem, plots of the efficiency of a 2-stage rule as a function of L_{11} and L_{21} show that this is usually, although not always, a unimodal surface (Beta priors with parameters less than 1, for example, can cause it to be multimodal). In cases where one could prove a priori that it is unimodal, one could drastically reduce the number of calculations needed for the optimal first stage, and hence could optimize far larger problems.

As another example, the data in Figures 7 and 11 suggest explicit growth rates that are consistent through a wide range of sample sizes. This leads one to consider approaching problems with large sample sizes by extrapolating the optimal allocations computed for moderate sample sizes, perhaps coupled with hill-climbing approaches, as in preceding paragraph, to improve the initial extrapolation. Thus explicit constructions, rather than vague guidelines, are obtained for producing near-optimal allocation schemes. We are presently pursuing this approach. Extrapolation techniques can compliment analytical approaches to give better insight and guidance for large problems.

Acknowledgments

Research supported in part by National Science Foundation grants DMS–9504980 and DMS–0072910.

References

[1] Anscombe, F. (1963), “Sequential medical trials”, *J. Amer. Stat. Assoc.* **58**: 365–383.

[2] Armitage, P. (1985), “The search for optimality in clinical trials”, *Int’l. Stat. Rev.* **53**: 15–24.

[3] Bather, J.A. (1985), “On the allocation of treatments in sequential medical trials”, *Int’l. Stat. Rev.* **53**: 1–13.

[4] Canner, P.L. (1970), “Selecting one of two treatments when the responses are dichotomous”, *J. Amer. Stat. Assoc.* **65**: 293–306.

[5] Cheng, Y. (1996), “Multistage bandit problems”, *J. Stat. Plan. and Inf.* **53**: 153–170.

[6] Cheng, Y., Su, F., and Berry, D.A. (1998), “Group sequential strategies in two-stage bandit problems: Implications for clinical trial sample sizes”, Tech. Rep. 98-28, Inst. Stat. Dec. Sci., Duke Univ.

[7] Clayton, M.K., and Witmer, J.A. (1988), “Two-stage bandits”, *Ann. Stat.* **16**: 887–894.

[8] Colton T. (1965), “A two-stage model for selecting one of two treatments”, *Biometrics* **21:** 169–180.

[9] Ghosh, B.K. (1975), “A two-stage procedure for the Behrens-Fisher problem”, *J. Amer. Stat. Assoc.* **70**: 457–462.

[10] Ghosh, M., Mukhopadhyay, N., Sen, P.K. (1997), *Sequential Estimation*, Wiley.

[11] Ghurye, S.G., Robbins, H. (1954), “Two-stage procedures for estimating the difference between means”, *Biometrika* **41:** 146–152.

[12] Hardwick, J. and Stout, Q.F. (1995), “Determining optimal few stage allocation rules”, *Computing Science and Stat.* **27:** 342–346.

[13] Hardwick, J. and Stout, Q.F. (1996), “Optimal allocation for estimating the mean of a bivariate polynomial”, *Seq. Anal.* **15**: 71-90.

[14] Melfi, V. and Page, C. (1998), “Adaptive designs for estimation of success probabilities: motivations and simulations”, *New Develop. and App. in Experimental Design*, N. Flournoy, W.F. Rosenberger, and W.K. Wong, eds., IMS Lec. Notes–Mono. Series **34**: 106–114.

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

[16] Page Shapiro, C. (1985), "Allocation schemes for estimating the product of positive parameters", *J. Amer. Stat. Assoc.* **80**: 449–454.

[17] Page, C. (1990), "Allocation proportional to coefficients of variation when estimating the product of parameters", *J. Amer. Stat. Assoc.* **85**: 1134–1139.

[18] Page, C. (1995), "Adaptive allocation for estimation", in *Adaptive Designs*, N. Flournoy and W. F. Rosenberger, eds., IMS Lec. Notes–Mono. Series **25**: 213–222.

[19] Rekab, K. (1992), "A nearly optimal 2-stage procedure", *Comm. Stat. – Theory Meth.* **21**: 197–201.

[20] Robbins, H., Simons, G. and Starr, N. (1967), "A sequential analogue of the Behrens-Fisher problem", *Ann. Math. Stat.* **38**: 1384–1391.

[21] Simon, R. (1977), "Adaptive treatment assignment methods and clinical trials", *Biometrics* **33**: 743–744.

[22] Zheng, S., Seila, A. and Sriram, T.N. (1995), "Optimal two stage procedures for estimating the product of two means", Univ. Georgia Tech. Report #95-15.

[23] Woodroofe, M. and Hardwick, J. (1991), "Sequential allocation for an estimation problem with ethical cost", *Ann. Stat.* **18**: 1358–1367.

A Basic Algorithm

The algorithm shown in 3 is relatively straightforward, but requires excessive time and space. Its requirements make it impractical for useful sample sizes, so to achieve exact optimizations and evaluations one needs to rearrange the calculations to eliminate redundancy and maximize reuse of space.

A.1 Time/Space Reductions

To reduce the time per stage, one reuses calculations among the states. To do so, note that at any state v,

$$\text{for } o_1 \geq 1: \qquad R_t(o_1,o_2;v) = p_1(1,1;v)\cdot R_t(o_1{-}1,o_2;v{+}\mathbf{s}_1) + p_1(0,1;v)\cdot R_t(o_1{-}1,o_2;v{+}\mathbf{f}_1),$$

$$\text{for } o_2 \geq 1: \qquad R_t(o_1,o_2;v) = p_2(1,1;v)\cdot R_t(o_1,o_2{-}1;v{+}\mathbf{s}_2) + p_2(0,1;v)\cdot R_t(o_1,o_2{-}1;v{+}\mathbf{f}_2).$$

Thus, if one computes and stores $R_t(o_1,o_2;v)$ for all o_1, o_2, and v, there is a natural way to reduce the calculation time to $\Theta(n^6)$ per stage. First, compute the values for all states v with $|v|=n$, then compute them for all states with $|v|=n-1$, and so on. Since there are $\Theta(n^6)$ options to be evaluated, this time is optimal unless one can determine that not all options need be evaluated.

However, if one proceeds in this way, the space requirements would also be $\Theta(n^6)$, and even the common trick of writing values for $|v|=m$ on top of the values originally stored for $|v|=m+1$ would only reduce the space to $\Theta(n^5)$. To reduce space to $\Theta(n^4)$, the calculation order can be rearranged to that given in Figure 12. Using this order, one need only store arrays corresponding to $R^*_t(\cdot)$, $R_t(o_1,0;\cdot)$ for a fixed value of o_1, and $R_t(o_1,o_2;\cdot)$ for fixed values of o_1 and o_2. $R_t(o_1,0;\cdot)$ is written on top of $R_t(o_1{-}1,0;\cdot)$, and $R_t(o_1,o_2;\cdot)$ is written on top of $R_t(o_1,o_2{-}1;\cdot)$.

Note that one must also keep track of the values of o_1 and o_2 for which the minimum $R^*_t(v)$ is obtained. This requires a constant amount of storage per state, and hence an extra $\Theta(n^4)$ space per stage.

A.2 Final and Initial Stages

The final stage is simpler than the general case, since the stage length is fixed and the problem of determining the optimal final allocation from a given state is the well-known optimal fixed-allocation problem with a fixed sample size. This can often be algebraically simplified to take only $\Theta(1)$ time per state, or $\Theta(n^4)$ overall, a point which is pursued in Appendix B. For those cases where no algebraic simplification is possible, the ordering in Figure 12 can be used to keep the time at $\Theta(n^5)$.

The initial stage is also simpler than the mid-stages since evaluation is required only at state $(0, 0, 0, 0)$. Thus the straightforward implementation takes only $\Theta(n^4)$ time. If there is only a single stage, then there are only $\Theta(n)$ options, needing $\Theta(n^3)$ total time.

```
{determine optimal decisions for stage $t$}
for all states $v$ with $k - 1 + 1 \leq |v| \leq n - t - 1$,
  initialize $R^*_t(v) = \infty$
for all $t\_end$ in $k - t + 1, \ldots, n - t + 1$
   for $o_1 = 0$ to $t\_end - k + t$
      for all states $v$ with $|v| = st\_end - o_1$
         if $o_1 \neq 0$ then
            compute $R_t(o_1, 0; v)$ using $R_t(o_1 - 1, 0; \cdot)$
            $R^*_t(v) = \min\{R^*_t(v), R_t(o_1, 0; v)\}$
         else $\{o_1 = 0\}$
            $R_t(0, 0, v) = R^*_{t-1}(v)$
      for $o_2 = 1$ to $t\_end - k + t - o_1$
         for all states with $|v| = t\_end - o_1 - o_2$
            compute $R_t(o_1, o_2; v)$ using $R_t(o_1, o_2 - 1; \cdot)$
            $R^*_t(v) = \min\{R^*_t(v), R_t(o_1, o_2; v)\}$
```

Figure 12: Improved Mid-stage Evaluation Order

Using these reductions gives the results in Theorem 3.1. All of the results follow directly from the above algorithms and observations, with the possible exception of the space analysis for $k \geq 3$. In the preceding comments, it may have seemed that an array of size $\Theta(n^4)$ was needed for each intermediate stage. However, since each stage is evaluated using only the results from the preceding stage, one never needs more than 2 such arrays at any one time. Hence one can alternate back and forth between two arrays, so that the space does not increase with k.

While the space required to determine the optimal design does not continue to grow for $k \geq 3$, additional space may be needed to store the decisions of the optimal design, so that it can be implemented or some post analysis can be performed. It is a common occurrence in dynamic programming that storing the decisions increases the space, because they cannot be written on top of each other. One may assume that one needs to store only one decision per state, which would imply that only $\Theta(n^4)$ space is needed, but for $k \geq 4$, there is the possibility that a given state could be reached at the end of more than one stage, and hence one would need to know how to optimally proceed for each different stage. While this does not normally occur, we have not been able to rule out the possibility, and hence the space may increase to $\Theta(kn^4)$. One can easily utilize disk storage for the decisions, since they are not referenced in the algorithm.

B Algorithmic Refinements

The base algorithm in Section 3 is quite general, and assumes no special properties of the objective function nor of the prior distributions. The algorithm also allows for arbitrary allocation within the k-stage constraint. However, in many situations significant simplifications are possible, and these may dramatically reduce the time or space required. Some of these are explored in this section. Then, in Appendix B.3, we give an example of a design restriction that complicates rather than simplifies the algorithm's complexity.

B.1 Analytical 1-stage Determination

One common simplification arises when extra information is available about the optimal 1-stage allocation. For example, for a specific allocation, it may be possible to analytically determine its value, instead of explicitly summing over all of the possible outcomes. Even better, one may be able to analytically determine the value and form of the optimal 1-stage design, instead of trying all possible allocations. Another special scenario is when it is known that the last stage must sample from a single arm: if the value can be determined analytically then it is a special case of the above, while otherwise there are further improvements possible. All of these simplify the optimization of 1- and 2-stage designs, although there is no improvement for designs of 3 or more stages.

We assume that each of these analytic calculations can be computed in a fixed amount of time, independent of n. Incorporating such simplifications into the final stage calculations of the base algorithm yields the following reductions. For three or more stages no reductions occur.

Theorem B.1 *The optimal k-stage allocation for an experiment of n observations of two Bernoulli populations can be determined in*

$$\Theta(1) \textit{ time and } \Theta(1) \textit{ space, if } k = 1 \quad \textit{and} \quad \Theta(n^4) \textit{ time and } \Theta(1) \textit{ space, if } k = 2.$$

if the optimal final stage allocation is given by an analytic expression.

If the value of any specific final allocation is given by an analytic expression, but it is not known how to analytically select the best, then the optimal allocation can be determined in

$$\Theta(n) \textit{ time and } \Theta(1) \textit{ space, if } k = 1 \quad \textit{and} \quad \Theta(n^5) \textit{ time and } \Theta(1) \textit{ space, if } k = 2.$$

If it is known that the final stage must sample from a single population, but it is not known how to analytically evaluate the final stage, then the optimal allocation can be determined in

$$\Theta(n) \textit{ time and } \Theta(1) \textit{ space, if } k = 1 \quad \textit{and} \quad \Theta(n^4) \textit{ time and } \Theta(n^3) \textit{ space, if } k = 2.$$

Proof: The time and space changes from Theorem 3.1 are quite straightforward for $k = 1$. For $k = 2$, in the first case one merely needs to evaluate all $\Theta(n^2)$ possible allocations for the first stage, taking $O(n^2)$ time per allocation to evaluate all $O(n^2)$ possible outcomes. In the second case it will take an extra $O(n)$ time per outcome. In both cases, no special efforts are needed to reuse calculations, so no extra space is needed to store intermediate results.

For the third case, with $k = 2$, for each level m one can determine the value of the final stage allocating all to population 1, and the value of allocating all to population 2, from these values for level $m + 1$. These can then be used to determine all first stage allocations of size m. □

Note that such simplifications were utilized in the calculations for the examples in Sections 4.1, 4.2 and 4.3. In Sections 4.1 and 4.2, the necessary calculations are quite straightforward. For the problem in Section 4.3, the relevant algebraic manipulations are in Hardwick and Stout (1996).

B.2 Stages of Bounded Length

Another significant improvement is possible when it can be proven that fewer alternatives need to be evaluated because the length of a stage is bounded. For example, if it is known that the last stage must start by the m^{th} observation, with $m \ll n$, then the earlier stages need only be investigated up through m, rather than up through n.

An example of an a priori bound on stage length appears in Section 4.1. The explicit bounds provided in Cheng (1996) and Cheng, Su and Berry (1998) show that the first stage of a 2-stage design is of length $\Theta(\sqrt{n})$. Thus there are only $\Theta(n)$ options for the first stage, each having only $O(n)$ outcomes. Since the final stage can be determined analytically, the time to optimize a 2-stage design is reduced to only $\Theta(n^2)$.

The program used for Section 4.1 also incorporated a second technique for reducing the first stage size, based on curtailment. An upper bound B_2 on the per-observation outcome of the second stage was obtained by integrating the maximum of p_1 and p_2 over the joint distribution of the two populations, and an upper bound B_1 on the per-observation outcome of the first stage is the larger of the two prior means. Note that $B_2 \geq B_1$. First stage options were evaluated for $L_1 = 1, 2, \ldots$ Each time a better total value V was found, it was used to create a smaller upper bound m for the first stage. This is based on the observation that m must satisfy $m \cdot B_1 + (n-m) \cdot B_2 \geq V$. The larger V is, the smaller m can be.

Curtailment techniques were also used in Section 4.2. For these problems, the bounds on the first stage dramatically increased the range of n for which the problem can be fully optimized.

B.3 Fixed Stage Sizes

There are two different ways in which designs with fixed stage sizes can arise. In one case the stage sizes are specified in advance, while in the other the sizes are not specified.

If the stage lengths have been fixed, then the optimization problem is simplified. For example, if a 2-stage allocation has a specified first stage length L_1, then there are only $\Theta(L_1)$ first-stage options to be evaluated, and only $\Theta(L_1^3)$ possible starts for the second stage. If the last stage can be optimized analytically, then the total time can be reduced to $\Theta(L_1^3)$. Similar reductions occur for k-stage allocations with specified stage sizes.

For a general k-stage allocation with specified stage sizes, using the techniques of Section 3, each state can be involved in the computation of at most n different options. For $k \geq 3$ this reduces the total time to $\Theta(n^5)$. Comparing this to the values in Theorem 3.1, one sees that not only has a factor of n been eliminated, but so has the factor of k.

```
{Initialize $R'$ array}
For all states $v$ with $|v| = n$,
  determine $R'_0(v)$

{Evaluate stages}
call eval$(1, n)$

{determine $R'_t(v)$ for $k-t \leq |v| < t\_last$}
procedure eval$(t, t\_last)$
  if $t = k$
    $R'_k(0)$ = weighted sum of $R'_{k-1}(v)$,
      for $|v| = t\_last$
  else $\{t < k\}$
    for all $v$ with $k-t \leq |v| < t\_last$
      determine $R^*_t(v)$ as in Figure 12,
      using $R^*_{t-1}(\cdot)$ for states of size $t\_last$.
    for $s\_last = k-t$ to $n-k+t+1$
      call eval$(t+1, s\_last)$
```

Figure 13: Optimal Fixed Stage Allocation

When the stage sizes are fixed, but their size is not specified in advance, the local optimality principles that underlie the dynamic programming algorithms in Section 3 do not apply. It appears that the only way to obtain the optimal stage sizes is to try all possible sizes, and for each choice determine the optimal allocation within each stage.

There are $\binom{n}{k-1}$ choices of stage sizes, which would seem to imply that

$$\Theta\left(\binom{n}{k-1} \cdot n^5\right) = \Theta\left(n^{k+4}/(k-1)!\right)$$

time is required (using the previous result about fixed stage sizes). This can be reduced to

$$\Theta\left(\binom{n}{k-2} \cdot n^5\right) = \Theta\left(n^{k+3}/(k-2)!\right)$$

as follows. The main induction step, as in Section 3, is over the stages, and is again done in reverse order. Suppose that specific stage lengths have been chosen for stages $t+1 \ldots k$, and that $R'_{k-t+1}(v)$ denotes the value of starting stage t at state v and proceeding optimally, given these stage lengths. Then one loops through each possible stage length for stage t. For each value, one computes $R'_{k-t+1}(v)$ and recursively repeats the process for stage $t-1$. When the process reaches the second stage, this is now the standard 2-stage problem, solvable in $\Theta(n^5)$ time. Note that the 2-stage problem can be viewed as the same process, where the first stage only evaluates $R'_k(0)$. This algorithm is outlined in Figure 13. It is presented recursively, but can be converted to a non-recursive implementation in a straightforward manner.

The primary difference between this algorithm and the one in Section 3 is that here, stage t repeatedly calls stage $t-1$ within a loop, rather than calling it only once. This causes the multiplicative effect of $\binom{n}{k-2}$ in the time analysis. It may appear that because one must return to stage t, $R'_t(v)$ needs to be saved, which would mean the space requirements would increase to $\Theta(kn^4)$. However, by looping through the stage lengths in increasing order, then stage $t-1$ can use the initial part of the array and not overwrite portions from stage t that are needed for later iterations. Thus the space requirements remain $\Theta(n^4)$.

In summary, one has the following:

Theorem B.2 *The optimal k-stage allocation for an experiment of n observations from 2 Bernoulli populations, in which the stage sizes must be fixed, can be determined in*

- $\Theta(n^3)$ *time and* $\Theta(1)$ *space, if* $k = 1$,
- $\Theta(n^3)$ *time and* $\Theta(n^3)$ *space, if* $k = 2$,
- $\Theta(n^5)$ *time and* $\Theta(n^3)$ *space, if* $k \geq 3$,

if the stage sizes have been fixed in advance, and in

- $\Theta(n^3)$ *time and* $\Theta(1)$ *space, if* $k = 1$,
- $\Theta(n^5)$ *time and* $\Theta(n^3)$ *space, if* $k = 2$,
- $\Theta(n^{k+3}/(k-2)!)$ *time and* $\Theta(n^4)$ *space, if* $k \geq 3$

if the stage sizes must be determined. □