In *Computing Science and Statistics* **27** (1995)

Determining Optimal Few-Stage Allocation Procedures

Janis P. Hardwick[1] Statistics Department
Quentin F. Stout[1] EECS Department
University of Michigan, Ann Arbor, MI 48109

Abstract

This paper gives algorithms for the design of optimal experiments involving Bernoulli populations in which allocation is done in stages. It is assumed that the outcomes of the previous stage are available before the allocations for the next stage are decided and that the total sample size for the experiment is fixed. At each stage, one must decide how many observations to take and how many to sample from each of two alternative populations. Of particular interest are 2- and 3-stage experiments.

The algorithms can be used to optimize experiments of useful sample sizes and are applied here to two estimation 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 that initial stages should be much larger than previously believed. This information suggests that one might approach large problems by extrapolating optimal solutions for moderate sample sizes; and, that approaches of this sort could give design guidelines that are far more explicit (and hopefully more accurate) than those obtained through asymptotic analyses alone.

Keywords: sequential allocation, selection, estimation, dynamic programming

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, where all decisions are made in advance. Adaptive allocations 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. Problems attributed to fully sequential methods include the following:

- they are difficult to design and analyze,
- they are complex to carry out, and often require that one consult computers at each step,
- one needs timely responses, since the $i+1^{\text{st}}$ allocation cannot be decided until the i^{th} outcome is known, and
- it is difficult to randomize since the allocations are determined one at a time.

While advances in computing hardware and algorithms make it easier to optimize and analyze certain fully sequential designs, and while use of networks or portable computers can ameliorate the second concern, the remaining points are more problematic.

One way to address these concerns is to incorporate a restricted form of sequential allocation, where decisions are made in stages. The most common of these is a two-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 many observations to take from each population in the last stage. Within each stage one can use constrained randomization and have concurrent observations, thus reducing the impact of response delay.

Two- and three-stage designs have received extensive analytical treatment, with results typically proving that the designs are first- and second-order asymptotically optimal, respectively. See, for example, [1, 3, 4, 5, 6].

Despite this interest in few-stage designs, we know of no attempts to optimize such designs fully. Published analyses tend to give fairly vague guidelines, which make it difficult to determine good allocations for specific sample sizes, much less to determine fully optimal ones. We address these points in Section 3 by giving an efficient algorithm that determines optimal few-stage designs. In Section 4 we apply the algorithm to two sample problems in estimation, and show that the optimal stage sizes do not always behave as predicted. 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. We also discuss relationships between analysis, computation, and visualization of allocation routines.

[1] Research supported in part by National Science Foundation under grant DMS-9157715.

2 Definitions

Throughout, we assume that the sample size n is fixed. There are two Bernoulli populations, Population 1 and Population 2, and our only option is to choose which of these to observe. We use a Bayesian approach, where the success parameters of the two populations have independent prior 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 Population 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 where k is small. In a 1-stage design, the only decision required is the number of observations to sample from Population 1. All remaining observations are sampled from Population 2. If $k > 1$, one first determines how many observations to take from Populations 1 and 2. These are denoted as o_1 and o_2 respectively. Once the initial observations have been obtained, one is left with a $(k-1)$-stage experiment of size $n - o_1 - o_2$, where the priors have been updated to include the initial observations. To eliminate nuisance cases, we require that $k \leq n$. Without loss of generality, we require that each stage have at least one observation.

There is an objective function $R_0^*(v)$ that 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_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.

3 Optimal Few-Stage Allocation

Our algorithm, given in Figure 2, 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. This 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 back towards the initial stage.

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

$$R_{st}^*(v) = \min\{R_{st}(o_1, o_2; v) : o_1, o_2 \text{ legal}\}.$$

"Legal" values are determined by the constraints that there are exactly k stages, 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.

k : number of stages

n : sample size, $n \geq k$

s_i, f_i : successes and failures on Population i, $i = 1, 2$

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

$|v|$: for state $v = (s_1, f_1, s_2, f_2)$, $|v|$ is the total number of observations, i.e., $|v| = s_1 + f_1 + s_2 + f_2$

o_i : number of new observations assigned to Pop. i

$p_i(s, o; v)$: probability of s successes among o observations on Pop. i, starting at state v

$R_{st}^*(v)$: value of starting st-stage experiment, $1 \leq st \leq k$, at state v and proceeding optimally ($R_0^*(v)$ is the objective function)

$R_{st}(o_1, o_2; v)$: value of starting st-stage experiment at state v, assigning o_i observations to Pop. i, and proceeding optimally.

Figure 1: Notation

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 (sample from either Population 1 or Population 2) that need to be evaluated, 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 Populations 1 and 2, creating $O(n^2)$ options. Further, to evaluate $R_{st}(o_1, o_2; v)$ one must consider $O(n^2)$ outcomes:

$$R_{st}(o_1, o_2; v) = \sum_{s_1'=0}^{o_1} \sum_{s_2'=0}^{o_2} p_1(s_1', o_1; v)\, p_2(s_2', o_2; v) \cdot R_{st-1}^*(v + (s_1', o_1 - s_1', s_2', o_2 - s_2'))$$

Thus, if straightforward implementations are used, it takes $O(n^2)$ time to evaluate $R_{st}(o_1, o_2; v)$; $O(n^4)$ time to evaluate $R_{st}(v)$; and $\Theta(n^8)$ time to evaluate the en-

{Evaluate last (kth) stage}

For all states v with $k-1 \le |v| \le n-1$,

$$R_1^*(v) = \min_{o_1+o_2=n-|v|} R_1(o_1, o_2; v)$$

{Evaluate middle stages}

For $st = 2$ to k-1

For all states v with $k - st \le |v| \le n - st$,

$$R_{st}^*(v) = \min_{1 \le o_1+o_2 \le n-st+1-|v|} R_{st}(o_1, o_2; v)$$

{Evaluate initial stage}

$$R_k^*(0) = \min_{1 \le o_1+o_2 \le n-k+1} R_k(o_1, o_2; 0)$$

Figure 2: Few-stage Allocation Algorithm

tire stage. (There is an implicit assumption that one can compute $p_i(s_i', o_i; v)$ in constant time. In general this can be done by a recursive computation that we omit. We also assume that R_0^* can be computed in constant time.)

3.1 Time/Space Reductions

To reduce the time per stage, one needs to reuse calculations among the states. To do so, note that at any state v, if $o_1 \ge 1$ then

$$\begin{aligned} R_{st}(o_1, o_2; v) &= p_1(1,1;v) \cdot R_{st}(o_1-1, o_2; v+\mathbf{s}_1) \\ &\quad + p_1(0,1;v) \cdot R_{st}(o_1-1, o_2; v+\mathbf{f}_1), \end{aligned}$$

and if $o_2 \ge 1$ then

$$\begin{aligned} R_{st}(o_1, o_2; v) &= p_2(1,1;v) \cdot R_{st}(o_1, o_2-1; v+\mathbf{s}_2) \\ &\quad + p_2(0,1;v) \cdot R_{st}(o_1, o_2-1; v+\mathbf{f}_2). \end{aligned}$$

Thus, if one computes and stores $R_{st}(o_1, o_2; v)$ 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 3. Using this order, one need only store arrays corresponding to $R_{st}^*(\cdot)$, $R_{st}(o_1, 0; \cdot)$ for a fixed value of o_1, and $R_{st}(o_1, o_2; \cdot)$ for fixed values of o_1 and o_2. $R_{st}(o_1, 0; \cdot)$ is written on top of $R_{st}(o_1-1, 0; \cdot)$, and $R_{st}(o_1, o_2; \cdot)$ is written on top of $R_{st}(o_1, o_2-1; \cdot)$.

for all states v, initialize $R_{st}^*(v) = \infty$

for all st_end in $k - st + 1, \ldots, n - st + 1$

for $o_1 = 0$ to $st\_end - k + st$

for all states v with $|v| = st\_end - o_1$

compute $R_{st}(o_1, 0; v)$, using $R_{st}(o_1-1, 0; \cdot)$

$R_{st}^*(v) = \min\{R_{st}^*(v), R_{st}(o_1, 0; v)\}$

for $o_2 = 1$ to $st\_end - k + st - o_1$

for all states with $|v| = st\_end - o_1 - o_2$

compute $R_{st}(o_1, o_2; v)$, using $R_{st}(o_1, o_2-1; \cdot)$

$R_{st}^*(v) = \min\{R_{st}^*(v), R_{st}(o_1, o_2; v)\}$

Figure 3: Mid-stage Evaluation Order

Note that one must also keep track of the values of o_1 and o_2 for which the minimum $R_{st}^*(v)$ is obtained. This requires an additional $\Theta(n^4)$ space per stage.

3.2 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. Even when the optimal cannot be found in $\Theta(1)$ time one can often evaluate each option in only $\Theta(1)$ time and then take the minimum, yielding $O(n)$ time per state and a total time of $\Theta(n^5)$. For those cases where no algebraic simplification is possible, the ordering in Figure 3 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.

Putting all of these results together gives the following:

Theorem 3.1 *The optimal k-stage allocation for an experiment of n observations 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* $\Theta(kn^4)$ *space, if* $k \ge 3$.

4 Examples

The few-stage optimization algorithm is applicable to a wide range of problems, but due to space limitations only two estimation problems are described here. We have chosen problems for which prior asymptotic analyses give one a framework for comparison.

4.1 Two-Stage Example

Here the goal is to minimize the mean squared error in estimating the product of the success probabilities of two independent Bernoulli populations, where the success probabilities are modeled as independent beta random variables. This problem has been extensively analyzed [1, 2, 3, 4, 5], and has applications to problems such as estimating reliability or area. We refer the reader to these papers for relevant derivations, applications, and additional references.

For two-stage allocations, Rekab [4] shows that one should choose the length of the first stage, L_1, so that

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

This is of scant use in determining an optimal L_1 for any specific n, and it does not predict the order of growth. More specific guidelines appeared in Noble [3], in which, using a frequentist analysis, the author showed that

$$\sqrt{\frac{n}{4\sigma_1\sigma_2}} \leq L_1 \leq \sqrt{\frac{n}{2\sigma_1\sigma_2}} \qquad \text{for } \sigma_i = p_i(1-p_i).$$

Using Noble's results, one finds that, for $n = 100$,

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

While our approach is Bayesian, one would expect that uniform priors would correspond reasonably well to $p_1 = p_2 = 0.5$ in Noble's analysis. However, our program shows that, for uniform priors, the optimal choice of L_1 is *42*! It is doubtful that an investigator, having read the literature on this problem, would have chosen an L_1 value this large. Further, if one centers the priors more heavily around 0.5 then the value deviates even further from the prediction. For example, if both populations have a prior of Be(5,5), then the optimal L_1 is 62.

More significantly, the growth of the optimal first stage size is not as predicted. This can be seen in Figure 4 where, for sample sizes ranging from 10 through 1000, the optimal size first stage length is plotted. 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 $\Theta(n^{0.817})$, instead of the predicted $\Theta(n^{0.5})$. Since our analyses have been carried out as far as $n = 1000$, we conclude that this is not merely a small-sample anomaly.

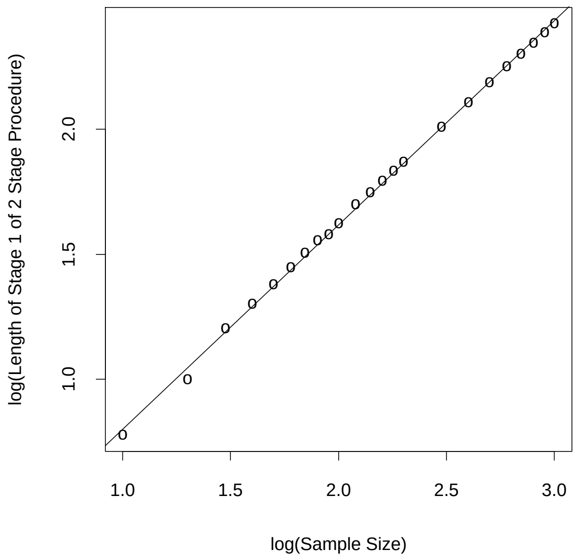

Figure 4: Growth Rate of Stage 1 of 2-Stage Procedure, Product of Means with Uniform Priors

4.2 Three-Stage Example

For our three-stage example, we use the same population model as the earlier example, but now are estimating the difference in population means, using a squared error loss plus a cost per failure. I.e., the loss of the estimator $\hat{\theta}$ is

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

where n_i is the number of observations on Population i. Such loss functions arise, for example, in clinical trials or destructive testing.

The analysis in [6], slightly modified to accommodate model differences, shows second-order asymptotically optimal three-stage designs where stage lengths, L_i for $i = 1, 2, 3$, obey

$$\lim_{n\to\infty} \frac{L_2}{n} = 1 \quad \text{and} \quad \lim_{n\to\infty} \frac{\sqrt{L_1} L_3}{n \log n} = \infty. \qquad (1)$$

Once again, based only on the information in (1), it is difficult to address practical questions such as "For $n = 100$, with uniform priors, what should the stage sizes be?" Our programs show that one should choose $L_1 = 42$, $L_2 = 46$, and $L_3 = 12$ (the latter two values are averages). Figure 5 shows the optimal stage sizes, for $n = 10 \ldots 100$. The regression lines are

$$\begin{aligned} \log_{10}(L_1) &= 0.85 \log_{10}(n) - 0.07 \\ \log_{10}(L_2) &= 1.22 \log_{10}(n) - 0.77 \\ \log_{10}(L_3) &= 0.96 \log_{10}(n) - 0.84 \end{aligned}$$

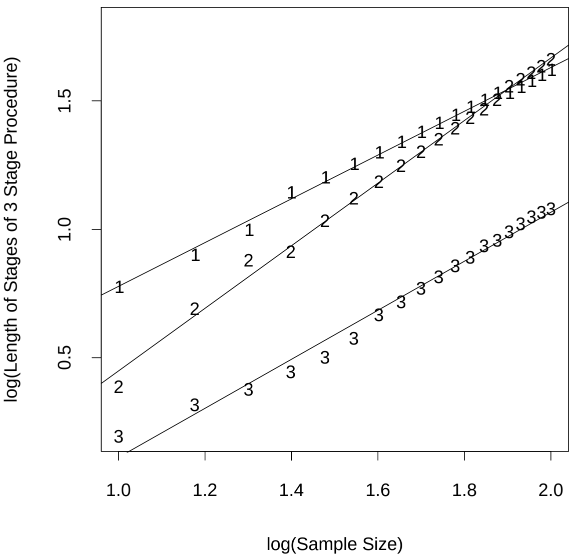

Figure 5: Optimal Stage Sizes in 3-Stage Experiment

5 Final Remarks

We have shown that it is possible to fully optimize certain few-stage allocation designs. Further, results of such 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, could not have been generated 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., but such extensions are beyond the scope of this paper. Additional points being pursued include sensitivity analysis of few-stage rules, handling multiple populations, allowing multiple endpoints, allowing additional constraints, and incorporating covariates.

As part of an ongoing project, we are using the algorithms given here, combined with various 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 psychological and statistical reasons for this attempt. Psychologically, people are often uncomfortable utilizing allocation schemes totally determined by computer. They have a better understanding of, and hence greater affinity for, fixed allocation schemes. If a user could explore an adaptive design and gain a better understanding of the decisions it makes, then the user 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 two-stage rule as a function of the number of first-stage observations on the two populations show that this is usually, though 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 required to find the optimal first stage, and hence could optimize far larger problems.

As another example, the data in Figures 4 and 5 suggest explicit growth rates that are consistent through a wide range of sample sizes. One might approach problems with large sample sizes by extrapolating the optimal allocations computed for moderate sample sizes. (This could also be coupled with hill-climbing approaches, from the preceding paragraph, to improve the initial allocation decisions.) This would give explicit constructions, rather than vague guidelines, which would hopefully produce near-optimal allocation schemes. We believe that such extrapolation techniques can compliment analytical approaches to give better insight and guidance for large problems.

References

[1] Berry, D. A. (1974), "Optimal sampling schemes for estimating system reliability by testing components – I: fixed sample sizes", *J. A. S. A.* **69**: 485–491.

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

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

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

[5] Shapiro, C. Page (1985), "Allocation schemes for estimating the product of positive parameters", *J. A. S. A.* **80**: 449–454.

[6] Woodroofe, M. and Hardwick, J. (1991), "Sequential allocation for an estimation problem with ethical cost", *Annals of Statistics* **18**: 1358-1367.