Biostatistics (2021), **0**, 0, *pp.* 1–22
doi:10.1093/biostatistics/output

A Markov Decision Process for Adaptive Multistage Trial Design

David Merrell*, Thevaa Chandereng†, Yeonhee Park‡

**Department of Computer Sciences, University of Wisconsin - Madison. †Department of Biostatistics, Columbia University. ‡ Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison.*

dmerrell@cs.wisc.edu, tc3123@cumc.columbia.edu, ypark56@wisc.edu

Summary

This paper presents TrialMDP, an algorithm for designing adaptive, multistage clinical trials. We build on past approaches that compute optimal trial designs via dynamic programming. However, our method differs in that it optimizes the *number of stages* as well as their treatment allocations. That is, the algorithm yields a policy that adaptively chooses the *size and composition* of the next stage, based on results seen up to that point in the trial.

The algorithm maximizes a utility function balancing *(i)* statistical power, *(ii)* patient outcomes, and *(iii)* the number of trial stages. We show that it attains significant improvements in utility over a suite of baseline designs, and offers control over the tradeoff between statistical power and patient outcomes. It is well suited for small trials that assign high cost to failures.

We provide TrialMDP as an R package: `https://github.com/dpmerrell/TrialMDP`.

*To whom correspondence should be addressed.

Key words:

1. Introduction

Introduction section here...

Some RAR references: (Rosenberger and Lachin, 2015), (Villar *and others*, 2018), (Rosenberger *and others*, 2001)

Some Bayesian RAR references: (Yin *and others*, 2012)

Some bandit references: (Villar *and others*, 2015*a*), (Villar *and others*, 2015*b*)

Some dynamic programming references: (Hardwick and Stout, 2002), (Hardwick and Stout, 1999), (Hardwick and Stout, 1995), (Woodroofe and Hardwick, 1990)

2. Proposed method

2.1 *Problem formulation*

Class of trials. In this paper we focus on multi-stage adaptive trials with two arms and binary outcomes. We label the arms A and B ("treatment" and "control", respectively) and outcomes 0 and 1 ("failure" and "success"). A trial has a fixed maximum number of patients, N. The trial proceeds in K stages; we require that all results from the current stage are observed before the next stage begins. Importantly, we allow K to adapt as the trial progresses. For example, if earlier results are inconclusive then it may be optimal to increase the number of remaining stages.

Let p_A, p_B denote the treatments' success probabilities. We assume a frequentist superiority test is performed at the end of the trial, with null hypothesis $p_A{=}p_B$. In this paper we focus specifically on the Cochran-Mantel-Haenzsel (CMH) test, with strata corresponding to the trial stages. It has been argued that multistage trials with CMH tests are more robust to confounding effects than other adaptive designs, e.g., RAR trials without stratified tests (Chandereng and Chappell, 2019).

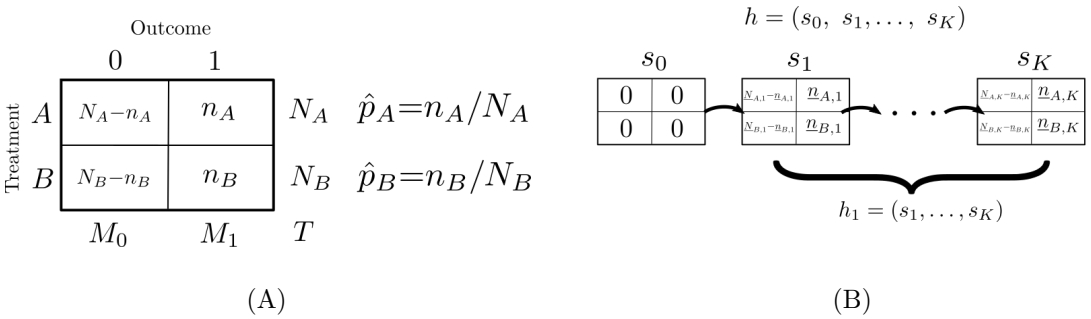

Fig. 1. (A) Contingency table notation. (B) Trial history notation. A history h is a sequence of cumulative contingency tables, $(s_0, \ldots, s_K)$. A subscript h_k indicates a history's kth *suffix*.

Notation: tables and histories. We establish some notation for clarity. A 2×2 contingency table has the following attributes: N_A and N_B, the number of patients assigned to each treatment; n_A and n_B, the number of successes for each treatment; M_0 and M_1, the total numbers of failures and successes; and T, the total number of outcomes recorded in the table. Symbols $\hat{p}_A$, $\hat{p}_B$ represent maximum-likelihood estimates of the treatment success probabilities. See Figure 1 for illustration.

Each stage of the trial has its own contingency table with corresponding quantities. We use a subscript to indicate the stage. For example, the kth stage of the trial has its own table with quantities $N_{A,k}$, $n_{A,k}$, T_k, and so on.

At any point we can summarize the state of the trial in a 2×2 contingency table, s, of *cumulative* results (i.e., the sum of stage-wise tables). We typically refer to such tables as *states*. We use an underline to indicate a *cumulative* quantity. For example, after completing k stages we have quantities $\underline{N}_{A,k} = \sum_{i=1}^{k} N_{A,i}$; $\underline{n}_{A,k} = \sum_{i=1}^{k} n_{A,i}$; $\hat{\underline{p}}_{A,k} = \underline{n}_{A,k} / \underline{N}_{A,k}$; and so on.

The sequence of states (i.e., cumulative tables) occupied by a trial forms a *trial history* $h = (s_0, \ldots, s_K)$, where s_0 is always the empty contingency table and s_K always has sum $\underline{T}_K = N$. We use a subscript to denote the *suffix* of a history. For example, $h_k = (s_k, \ldots, s_K)$ is the sequence of states *after* the kth stage of the trial. It is useful to think of a history as a random object, subject to uncertainty in the patient outcomes and the values of p_A, p_B.

Utility function. We aim to design multistage trials that balance *(i)* statistical power and *(ii)* patient successes. We also recognize that each stage entails a cost in time and other overhead. As such, we wish to avoid an excessive number of stages. We formalize these goals with the following utility function:

$$U(h) = V(h) \; - \; \lambda_F \cdot F(h) \; - \; \lambda_K \cdot K(h) \tag{2.1}$$

where $V(h)$ is a proxy for the trial's statistical power; $F(h)$ measures the number of patient failures; and $K(h)$ is the number of stages. This utility function promotes a high statistical power while penalizing patient failures and trial stages. The coefficients λ_F and λ_K control the relative importance of patient failures and trial stages, respectively.

The functions V, F, and K have the following forms:

$$V(h) = \frac{1}{N} \sum_{i=1}^{K} \frac{w_i}{\frac{1}{2}(\hat{\underline{p}}_{A,i} + \hat{\underline{p}}_{B,i}) \frac{1}{2}(\hat{\underline{q}}_{A,i} + \hat{\underline{q}}_{B,i})} \qquad F(h) = \frac{1}{N}(\underline{N}_{A,K} - \underline{N}_{B,K})(\hat{\underline{p}}_{B,K} - \hat{\underline{p}}_{A,K})$$

$$K(h) = K$$

Function K simply returns the number of stages in the trial. Function F quantifies the number of patient failures as a fraction of all patients. It is a function only of the final state, s_K, and becomes small when the estimates $\hat{\underline{p}}_{A,K}$ and $\hat{\underline{p}}_{B,K}$ are close.

Function V serves as a proxy for the trial's statistical power. It is crafted such that maximizing V also maximizes the power of the Cochran-Mantel-Haenzsel test (Cochran, 1954) to an acceptable approximation. Each $w_i = N_{A,i}N_{B,i}/(N_{A,i}+N_{B,i})$ is the *harmonic mean* of that stage's treatment allocations. V takes larger values when the allocations are balanced; and when p_A, p_B are close to each other, and far from $\frac{1}{2}$. The factor $\frac{1}{N}$ makes V consistent across trials with differing sample sizes. See Appendix A for a more detailed justification of V.

Our goal is to maximize the expected value of $U(h)$, subject to the randomness in h.

Markov Decision Process formulation. In our effort to maximize the expected utility (Equation 2.1), we find it useful to model a multistage adaptive trial as a *Markov Decision Process* (MDP).

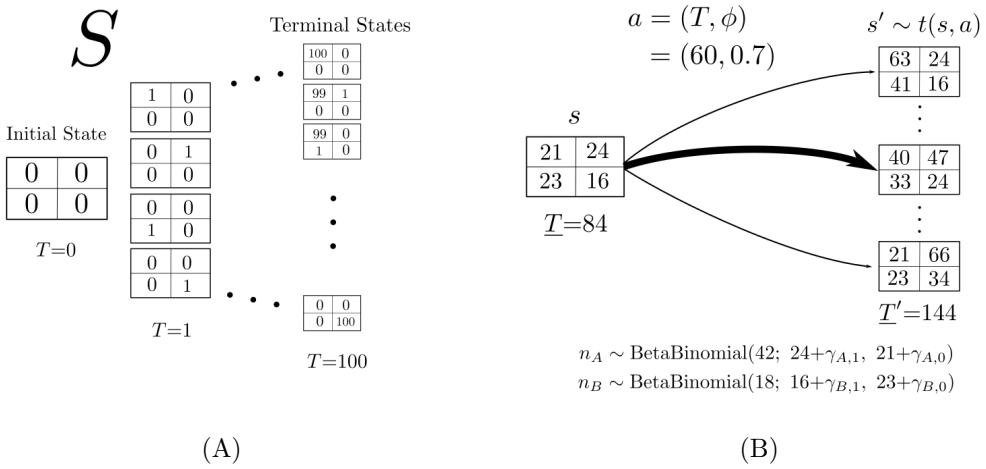

Fig. 2. (A) State space. At any point, the state of the trial is summarized by its contingency table. We can order the set of all contingency tables by their sums, T. The trial begins with T=0; the trial ends when it reaches a state with T=N (in this case N=100). (B) Transition distribution. In this example, current state s and action a=$(60, 0.7)$ induce a distribution $s' \sim t(s, a)$ for the next state. The next state necessarily has $\underline{T}$=144=84 + 60. Its entries are governed by Beta-Binomial distributions, parameterized by the entries of the current contingency table.

An MDP is a simple model of sequential decision-making. It consists of an agent navigating a *state space*. At each time-step, the agent chooses an *action*; given the agent's current state and chosen action, the agent *transitions* to a new state and collects a *reward*. In general the transition is stochastic, governed by a *transition distribution*. One *solves* an MDP by obtaining a *policy* maximizing the expected total reward. We refer the reader to Chapter 38 of Lattimore and Szepesvari's text for detailed information about MDPs (Lattimore and Szepesvari, 2020).

We formulate adaptive multistage trial design as an MDP with the following components:

- *State space.* In our setting the state space S consists of every possible 2×2 contingency table with $\leqslant N$ observations. We can order the states by their sums (i.e., numbers of observations). We let S_i denote the subset of S containing tables with exactly i observations. The trial always begins at the empty contingency table in S_0 and terminates at some table in S_N. The state space grows quickly with N, $|S|$=$O(N^4)$. Figure 2(A) illustrates S for N=100.

- *Actions.* At each stage of the trial we choose an action a=(T, ϕ), the stage's *size* and *alloca-*

tion. Suppose we have completed k stages; then T may take any value between 1 and $N-\underline{T}_k$. The allocation ϕ is the fraction of patients assigned to treatment A in this stage. We restrict ϕ to a finite set Φ, e.g., $\Phi = \{0.2, 0.3, \ldots, 0.8\}$. Importantly, exactly $T{\cdot}\phi$ patients (rounded to the nearest integer) are assigned to treatment A. In other words, patients are assigned to treatments "without replacement." Contrast this with other randomized designs—RAR, blocked RAR, etc.—that assign each patient to A with independent probability ϕ.

For example, action (T=60, ϕ=0.7) implies that the next stage will treat 60 patients, assigning exactly $T{\cdot}\phi$=42 of them to treatment A and 18 to treatment B.

- *Transitions.* Given the current contingency table s_i and the chosen stage design a_i=(T, ϕ), the next contingency table s_{i+1} is randomly distributed. In particular, the quantities $n_{A,i+1}$ and $n_{B,i+1}$ for this stage would have Binomial distributions parameterized by the true success probabilities p_A and p_B:

$$n_{A,i+1}|p_A \sim \text{Binomial}\left(T{\cdot}\phi,\ p_A\right) \qquad n_{B,i+1}|p_B \sim \text{Binomial}\left(T{\cdot}(1-\phi),\ p_B\right).$$

  However, we only have imperfect knowledge of p_A and p_B, encoded in the entries of the current table s_i. We harness this information by giving p_A and p_B Beta distributions:

$$p_A \sim \text{Beta}(\underline{n}_{A,i}{+}\gamma_{A1},\ \underline{N}_{A,i}{-}\underline{n}_{A,i}{+}\gamma_{A0}) \qquad p_B \sim \text{Beta}(\underline{n}_{B,i}{+}\gamma_{B1},\ \underline{N}_{B,i}{-}\underline{n}_{B,i}{+}\gamma_{B0})$$

  where each γ_* is a smoothing hyperparameter typically set to 1. Together, these distributional assumptions assign independent Beta-Binomial probabilities to $n_{A,i+1}$ and $n_{B,i+1}$, which in turn define the distribution for s_{i+1}. See Figure 2(B) for illustration. We sometimes use the notation $s_{i+1} \sim t(s_i, a_i)$ to indicate s_{i+1}'s distributional dependence on s_i, a_i.

- *Rewards.* Given the current state s_i and the chosen action a_i, we transition to state s_{i+1} and receive a reward $R(s_i, a_i, s_{i+1})$. In an MDP the goal is to maximize expected total reward. Recall, however, that our ultimate goal is to maximize the expected utility U (Equation

2.1). We craft a reward function R consistent with U, as follows:

$$R(s_i, a_i, s_{i+1}) = \begin{cases} \frac{1}{N} \frac{w_{i+1}}{\frac{1}{2}(\underline{\hat{p}}_{A,i+1} + \underline{\hat{p}}_{B,i+1}) \frac{1}{2}(\underline{\hat{q}}_{A,i+1} + \underline{\hat{q}}_{B,i+1})} - \lambda_K & s_{i+1} \notin S_N \\ \frac{1}{N} \frac{w_{i+1}}{\frac{1}{2}(\underline{\hat{p}}_{A,i+1} + \underline{\hat{p}}_{B,i+1}) \frac{1}{2}(\underline{\hat{q}}_{A,i+1} + \underline{\hat{q}}_{B,i+1})} - \lambda_K - \lambda_F \cdot F(s_{i+1}) & s_{i+1} \in S_N \end{cases} \tag{2.2}$$

Maximizing the expected total reward (Equation 2.2) is identical to maximizing expected utility (Equation 2.1). With each stage, the reward function produces that stage's contribution to the total utility. This includes the stage's term for V; the stage's cost λ_K; and the final failure penalty $F(s_{i+1})$ when s_{i+1} is terminal.

Notice that our particular R is a function only of a_i and s_{i+1}. We sometimes write $R(a_i, s_{i+1})$ for compactness.

Casting our problem into the MDP framework helps us design algorithmic solutions. Our particular MDP lends itself to a straightforward dynamic programming approach, since there are no cycles in its directed graph of possible transitions.

2.2 *Solution via dynamic programming*

The MDP described in Section 2.1 can be solved by a relatively simple dynamic programming algorithm. This makes our method a close relative of past dynamic programming approaches for trial design (Woodroofe and Hardwick, 1990; Hardwick and Stout, 1995, 1999, 2002). However, our method differs from them in an important respect: we seek to maximize an objective that is a *function of the trial history*, and not just a function of the final state. Concretely, our objective function (Equation 2.1) includes $V(h)$ and $K(h)$, which are functions of stage-wise attributes. Formulating the problem as an MDP gives us the flexibility to consider such an objective.

Recurrence relations. Like any dynamic programming algorithm, ours divides the problem at hand into subproblems and solves them in an order that efficiently reuses computation. This

dependence between subproblems is defined by a set of recurrence relations. In our case we have a single recurrence based on the Bellman equation (Lattimore and Szepesvari, 2020):

$$\overline{U}(s) = \begin{cases} \max_a \left\{ \mathbb{E}_{s' \sim t(s,a)} \left[R(s,a,s') + \overline{U}(s') \right] \right\} & s \notin S_N \\ \\ 0 & s \in S_N \end{cases} \tag{2.3}$$

The algorithm computes this recurrence at every state in S, iterating through the state space in order of *decreasing* $\underline{T}$. In other words the algorithm evaluates the recurrence at each state in S_N, S_{N-1}, and so on, until it finally computes $\overline{U}(s_0)$ for the the empty table $s_0 \in S_0$ and terminates. At each state $s \notin S_N$ the algorithm also stores the maximizing action a^*. This table of optimal actions is the algorithm's most important output, as it constitutes the optimized trial design—for every possible contingency table, it gives the best design for the next stage of the trial. Figure 3 illustrates the algorithm in detail with pseudocode.

The trial design (i.e., policy) yielded by this recurrence is guaranteed to maximize the expected utility, since our optimization problem has the *optimal substructure property*. See Appendix C for more discussion and a proof of optimal substructure.

Computational expense. At a high level TRIALMDP is a nested loop over every possible state, action, and transition. For each state the algorithm stores a set of values, along with the optimal action. Hence the algorithm uses $O(|S|)$=$O(N^4)$ space. The number of possible actions and transitions varies between states; summing across all states yields total time cost $O(|\Phi|N^7)$, where Φ is the set of allocation fractions mentioned in Section 2.1.

These complexities apply if we allow the algorithm to consider every possible state and action. However, there are practical ways to *prune away* states and actions, attaining much lower computational cost without sacrificing much utility. Introducing a *minimum stage size* parameter $T_{\min}$ eliminates all of the states in $S_1, \ldots, S_{T_{\min}-1}$ and $S_{N-T_{\min}+1}, \ldots S_{N-1}$; and reduces the number of possible actions at each remaining state. An additional *stage increment* parameter κ further constrains the algorithm to states where T is an integer multiple of κ, resulting in a "coarsened"

Algorithm 1 TrialMDP

```
1:  procedure MainLoop(N, λ_F, λ_K)
2:      initialize tables U, A
3:      for s ∈ S_N do
4:          U[s] = 0
5:      for s ∉ S_N do
6:          U[s] = −∞
7:          for a ∈ A_s do
8:              u = 0
9:              for (p, s′) ∈ t(s, a) do
10:                 u += p·{R(a, s′, N, λ_F, λ_K)
11:                                  + U[s′]}
12:             if u > U[s] then
13:                 U[s] = u; A[s] = a
14:     return U, A

function R(a, s′, N, λ_F, λ_K)
    r = 0
    w = a.N_A · a.N_B/(a.N_A + a.N_B)
    p̂_A = (s′.n_A+1)/(s′.N_A+2)
    p̂_B = (s′.n_B+1)/(s′.N_B+2)
    q̂_A = 1 − p̂_A
    q̂_B = 1 − p̂_B
    r += 4 · w/(N·(p̂_A+p̂_B)(q̂_A+q̂_B))
    if s′ ∈ S_N then
        r −= λ_F·(s′.N_A − s′.N_B)
                ·(p̂_B − p̂_A)/N
    r −= λ_K
    return r
```

Fig. 3. TrialMDP algorithm pseudocode. The algorithm populates tables U and A with utilities and actions, respectively. Tables U and A are indexed by states; i.e., U[s] yields the utility for state s. The for-loop on line 5 iterates through states in order of decreasing $\underline{T}$. The for-loop on line 7 iterates through all possible actions for the current state; and the loop on line 9 computes the expectation of U for the current state and action. Function R evaluates the reward function given by Equation 2.2.

state space. These parameters reduce the algorithm's space and time cost to $O((N - T_{\min})^4/\kappa)$ and $O((N-T_{\min})^7/\kappa^2)$, respectively. See Appendix D for derivations. We typically set $T_{\min}$=$N/8$ and κ=2.

3. Evaluation

3.1 *Simulation study*

We performed a simulation study to compare TrialMDP against established baselines. At each point in a grid of values for $\lambda_F, \lambda_K, p_A$, and p_B, we ran 10,000 simulated trials using our algorithm and a suite of baseline designs. The baselines included *(i)* a one-to-one, single-stage traditional design; *(ii)* a Response-Adaptive-Randomized (RAR) design; and *(iii)* a blocked RAR design.

For null scenarios with p_A=p_B, we chose an arbitrary sample size of N=100. For alternative scenarios with p_A>p_B, we chose N large enough for a traditional design to attain power near 0.8.

See Tables 1 and 2 for the exact values.

The RAR baseline used a 1:1 randomization ratio for the first 25% of patients, and adaptive randomization thereafter according to the frequentist rule of Rosenberger *and others* (2001). The blocked RAR baseline used two stages of equal size. For TRIALMDP, we swept the parameter settings $(\lambda_F, \lambda_K) \in \{2,3,4,5\}\times\{0.01, 0.025, 0.05, 0.1\}$.

As an initial sanity check we visualized some trial histories to see whether the designs behave as expected. Figure 4 shows some examples. TRIALMDP always chose 1:1 allocation for the first stage, increasing the allocation to A in subsequent stages when $p_A > p_B$. As λ_F increased, the designs reliably increased allocation to A, in agreement with our expectations.

Recall that TRIALMDP is supposed to optimize the utility function (Equation 2.1) in expectation. If this were true, we would expect our designs to attain higher utility than the others, averaged over the simulated histories. To verify this we computed the utility for every simulated history and for every design. Some results are shown in Table 1. We found that our algorithm does typically achieve higher utility than the baseline designs *(i)* under the alternative hypothesis and *(ii) as long as λ_F is sufficiently large.* When λ_F is not large enough, our designs have highest utility *among the adaptive designs*—but the traditional design is mathematically guaranteed to attain the highest utility. We demonstrate this mathematically in Appendix B. The key takeaway is that TRIALMDP's designs typically attain superior utility among adaptive designs, under the alternative hypothesis.

Beyond a one-dimensional comparison of utility, it is useful to compare the designs in two dimensions: statistical power and patient outcomes. As we vary the parameter λ_F, TRIALMDP designs trials that balance these quantities differently. We visualize this by plotting the algorithm's *frontier* in that two-dimensional space. Figure 5 gives examples. The key takeaways are *(i)* in some cases, our algorithm *dominates* the other designs; and *(ii)* when our designs don't dominate the others, λ_F still provides a useful way to control the balance between power and patient outcomes.

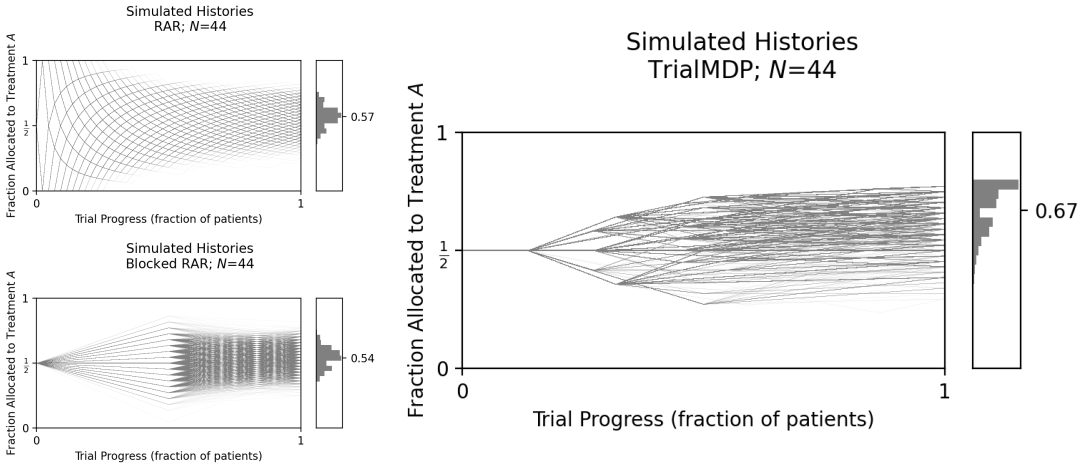

Fig. 4. Simulated trial histories. Each plot traces the treatment allocation of 10,000 simulated trials. Histograms on the right give distributions of final allocations and report the mean. For each of these plots, (p_A, p_B)=(0.4, 0.1). (A) Histories for RAR and blocked RAR trial designs. (B) Histories for a TrialMDP design, with parameter settings λ_F=4.0 and λ_K=0.01. Under these settings our algorithm allocates many more patients on average to the superior treatment.

			Power Z-score			$F(h)$ Z-score			Number of Stages			Utility Z-score		
p_A	p_B	N	RAR	BRAR	MDP	RAR	BRAR	MDP	RAR	BRAR	MDP	RAR	BRAR	MDP
0.3	0.1	92	-0.30	-0.25	-0.19	-13.49	-8.99	-11.95	92	2	3.34	-58.23	6.01	8.12
0.4	0.1	44	-0.29	-0.19	-0.55	-5.98	-3.46	-11.15	44	2	3.74	-8.45	2.36	8.11
	0.2	122	-0.13	-0.08	-0.51	-12.85	-9.35	-15.69	122	2	3.39	-100.90	6.03	11.71
0.5	0.3	142	-0.29	-0.32	-0.82	-11.85	-8.26	-18.70	142	2	3.34	-152.93	4.96	14.16
0.7	0.4	60	0.00	-0.11	-0.61	-6.15	-4.08	-14.31	60	2	4.05	-15.82	2.65	10.60
	0.5	142	-0.41	-0.36	-0.91	-8.40	-6.03	-18.56	142	2	3.25	-169.59	3.23	14.16
0.9	0.6	44	-0.04	-0.11	-0.23	-3.46	-2.24	-8.76	44	2	3.35	-8.54	1.43	6.51
	0.7	92	0.02	-0.09	-0.19	-4.26	-2.95	-11.28	92	2	3.19	-69.77	1.30	7.79

Table 1. Simulation study alternative scenarios. Labels RAR, BRAR, and MDP refer to adaptive trials designed by RAR, blocked RAR, and TrialMDP, respectively. TrialMDP used parameter settings λ_F=4.0 and λ_K=0.01. The multicolumns marked as "Z-scores" report gain relative to the traditional 1:1 trial design, computed as $Z=(\mu_1-\mu_2)/\sqrt{(\sigma_1^2+\sigma_2^2)/10{,}000}$. With these particular parameter settings our algorithm attains lower power than the other designs, but its superior patient outcomes give it the greatest utility across all scenarios.

3.2 *Trial redesign*

We demonstrate TrialMDP's practical usage by applying it to a historical trial. We chose the phase-II thymoglobulin trial described by Bashir *and others* (2012) because it *(i)* had two arms, *(ii)* had a small sample size (N=20), and *(iii)* suggested a large treatment effect, with nontrivial ethical implications. This combination makes the trial well-suited to our algorithm.

		Size Z-score			Number of Stages			Utility Z-score		
p_A=p_B	N	RAR	BRAR	MDP	RAR	BRAR	MDP	RAR	BRAR	MDP
0.1	100	-0.12	-0.04	0.03	100	2	2.77	-538.91	-5.22	-8.79
0.3	100	0.01	0.11	0.20	100	2	3.77	-506.73	-5.06	-13.55
0.5	100	0.19	0.17	0.37	100	2	3.88	-487.79	-5.16	-13.26
0.6	100	0.20	-0.13	0.17	100	2	3.85	-494.45	-5.14	-13.31
0.7	100	0.08	0.03	0.18	100	2	3.68	-486.53	-4.97	-12.52
0.9	100	-0.09	-0.00	0.00	100	2	2.77	-520.00	-5.21	-8.64

Table 2. Simulation study null scenarios. Our algorithm produces designs with somewhat inflated Type-I error. This table excludes $F(h)$, which is always 0 since p_A=p_B. A 1:1 design is optimal under the null hypothesis; hence every adaptive design reports a relative decrease in utility. As in Table 1, these results use λ_F=4.0 and λ_K=0.01.

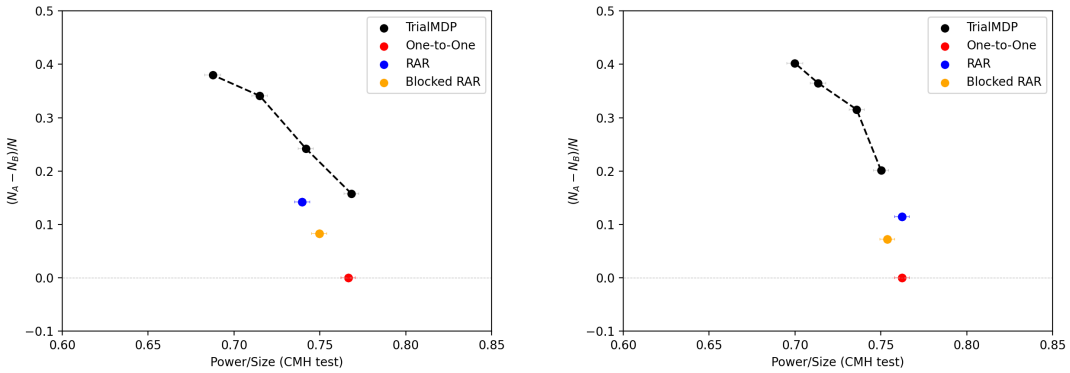

Fig. 5. Frontier plots. Trial designs are shown as points in two dimensions, with statistical power on the horizontal axis and allocation to A on the vertical axis—higher and to the right is better. As we vary λ_F our algorithm traces a frontier in this space. (A) In this case (p_A=0.4, p_B=0.1) TRIALMDP dominates the other adaptive designs when λ_F=2; it attains higher power *and* allocation to the superior treatment. (B) In other cases (e.g., p_A=0.7, p_B=0.4) TRIALMDP does not dominate the other designs, but enriches the set of tradeoffs between power and patient outcomes. In both plots $\lambda_F \in \{2, 3, 4, 5\}$ and λ_K=0.01.

The redesign proceeded in two phases: "parameter tuning" and "testing." In the parameter tuning phase we swept through the same grid of $\lambda_F, \lambda_K, p_A, p_B$ values used in our simulation study, but with the sample size fixed at N=20. We ran our algorithm and simulated 10,000 trials at each grid point, and generated frontier plots similar to those in Figure 5. Visual inspection suggested that λ_F=3.0 and λ_K=0.05 would yield reasonable power and patient outcomes for a variety of p_A, p_B scenarios.

In the testing phase we simulated the thymoglobulin trial by computing point estimates of

p_A	p_B	N	Number of Stages	Size/Power (CMH test)	$N_A - N_B$	(10%)	(90%)
0.4	0.4	20	2.577	0.055	-0.056	-10.000	6.000
0.8	0.4	20	2.290	0.557	5.096	0.000	10.000

Table 3. Results of trial redesign. Running our algorithm with λ_F=3.0, λ_K=0.05 yielded a trial design with Type-I error of 0.055 under the null, and power 0.557 under the alternative. Compare with a power of ∼0.61 for a traditional trial. The last two columns show the 10%-ile and 90%-ile of $N_A - N_B$. Notice that the design had probability <10% of assigning more patients to the inferior treatment.

p_A=0.8 and p_B=0.4 from its results. We included two scenarios: the null where p_A=p_B=0.4, and the alternative where p_A=0.8 and p_B=0.4. Using the design from our algorithm with λ_F=3.0 and λ_K=0.05, we ran 10,000 trials for each scenario. The results are aggregated in Table 3.

4. Discussion

Key takeaways.

Practical recommendations. The user of TrialMDP immediately faces a question: *what values of λ_F and λ_K should be used?* Consider Equation 2.1. Since $V(h)$ is only a proxy for the statistical power, there isn't a clear way to assign practical meaning to λ_F, λ_K. For example, we cannot interpret λ_F as a literal "conversion rate" between units of failure and units of statistical power. This makes it difficult to set λ_F, λ_K in a principled way. Instead we recommend *tuning* λ_F and λ_K through a process like the one demonstrated in Section 3.2: *(i)* use the algorithm to design trials for a grid of λ_F, λ_K values; *(ii)* simulate trials for each design, for a set of scenarios p_A, p_B; *(iii)* examine the simulation results and choose λ_F, λ_K that yield acceptable power and patient outcomes across scenarios. As a starting point, λ_F=3.0, λ_K=0.01 yielded reasonable characteristics across all the scenarios in this paper.

We do not recommend using our algorithm to decide *whether* to conduct an adaptive trial. The choice between adaptive and non-adaptive designs should be based on ethical considerations. Then, given the choice to conduct an adaptive trial, our algorithm can generate a design with

useful characteristics—in many cases, superior characteristics to other adaptive designs.

Future improvements. Although our current implementation is single-threaded, the dynamic programming algorithm is highly parallelizable and would have a speedup roughly linear in the number of threads. A multi-threaded parallel implementation is a natural next step.

Our current MDP formulation assumes that the trial terminates after all patients have been treated. A more sophisticated formulation could incorporate interim analyses, accounting for the possibility of early stopping for success or futility.

5. Software

We provide a C++ implementation of TrialMDP, as an R package on GitHub: `https://github.com/dpmerrell/TrialMDP`. We also provide the analysis code for our Section 3 evaluations in another repository: `https://github.com/dpmerrell/TrialMDP-analyses`. This includes a Snakemake workflow (Mölder *and others*, 2021) that fully reproduces the results in this paper.

6. Supplementary Material

URLs for any supplementary material

Acknowledgments

We thank Zhu Xiaojin and Blake Mason for conversations about bandit algorithms. DM was funded by the National Institutes of Health (award T32LM012413).

Conflict of Interest: None declared.

References

Bashir, Qaiser, Munsell, Mark F., Giralt, Sergio, Silva, Leandro de Padua, Sharma, Manish, Couriel, Daniel, Chiattone, Alexandre, Popat, Uday, Qazilbash, Muzaffar H., Fernandez-Vina, Marcelo, Champlin, Richard E. *and others.* (2012, May). Randomized phase II trial comparing two dose levels of thymoglobulin in patients undergoing unrelated donor hematopoietic cell transplant. *Leukemia & Lymphoma* **53**(5), 915–919. Publisher: Taylor & Francis _eprint: https://doi.org/10.3109/10428194.2011.634039.

Chandereng, Thevaa and Chappell, Rick. (2019, September). Robust Blocked Response-Adaptive Randomization Designs. *arXiv:1904.07758 [stat]*. arXiv: 1904.07758.

Cochran, William G. (1954). Some Methods for Strengthening the Common chi-squared Tests. *Biometrics* **10**(4), 417–451. Publisher: [Wiley, International Biometric Society].

Hardwick, Janis and Stout, Quentin F. (2002, May). Optimal few-stage designs. *Journal of Statistical Planning and Inference* **104**(1), 121–145.

Hardwick, Janis P. and Stout, Quentin F. (1995). Exact Computational Analyses for Adaptive Designs. *Lecture Notes-Monograph Series* **25**, 223–237. Publisher: Institute of Mathematical Statistics.

Hardwick, Janis P. and Stout, Quentin F. (1999, January). Using Path Induction to Evaluate Sequential Allocation Procedures. *SIAM Journal on Scientific Computing* **21**(1), 67–87.

Lattimore, Tor and Szepesvari, Csaba. (2020, July). *Bandit Algorithms*. Cambridge University Press.

Mölder, Felix, Jablonski, Kim Philipp, Letcher, Brice, Hall, Michael B., Tomkins-Tinch, Christopher H., Sochat, Vanessa, Forster, Jan, Lee, Soohyun, Twardziok,

Sven O., Kanitz, Alexander, Wilm, Andreas, Holtgrewe, Manuel, Rahmann, Sven, Nahnsen, Sven *and others*. (2021, January). Sustainable data analysis with Snakemake. *F1000Research* **10**, 33.

Rosenberger, William F. and Lachin, John M. (2015, November). *Randomization in Clinical Trials: Theory and Practice*. John Wiley & Sons. Google-Books-ID: ZJEvCgAAQBAJ.

Rosenberger, William F., Stallard, Nigel, Ivanova, Anastasia, Harper, Cherice N. and Ricks, Michelle L. (2001). Optimal Adaptive Designs for Binary Response Trials. *Biometrics* **57**(3), 909–913. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0006-341X.2001.00909.x.

Villar, Sofía S., Bowden, Jack and Wason, James. (2015*a*, May). Multi-armed Bandit Models for the Optimal Design of Clinical Trials: Benefits and Challenges. *Statistical Science* **30**(2), 199–215.

Villar, Sofía S., Bowden, Jack and Wason, James. (2018). Response-adaptive designs for binary responses: How to offer patient benefit while being robust to time trends? *Pharmaceutical Statistics* **17**(2), 182–197.

Villar, Sofía S., Wason, James and Bowden, Jack. (2015*b*, December). Response-Adaptive Randomization for Multi-arm Clinical Trials Using the Forward Looking Gittins Index Rule. *Biometrics* **71**(4), 969–978.

Woodroofe, Michael and Hardwick, Janis. (1990, September). Sequential Allocation for an Estimation Problem with Ethical Costs. *The Annals of Statistics* **18**(3), 1358–1377.

Yin, Guosheng, Chen, Nan and Lee, J. Jack. (2012). Phase II trial design with Bayesian adaptive randomization and predictive probability. *Journal of*

the Royal Statistical Society: Series C (Applied Statistics) **61**(2), 219–235. _eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9876.2011.01006.x.

APPENDIX

A. Derivation of V

Equation 2.1 uses the following function, $V(h)$, as a proxy for a trial's statistical power:

$$V(h) = \frac{1}{N} \sum_i \frac{w_i}{\frac{1}{2}(\underline{p}_{A,i} + \underline{p}_{B,i})\frac{1}{2}(\underline{q}_{A,i} + \underline{q}_{B,i})}$$

where

$$w_i = N_{A,i} \cdot N_{B,i} / (N_{A,i} + N_{B,i}).$$

This appendix provides some justification for $V(h)$.

We're interested in multistage trials where the final analysis uses a Cochran-Mantel-Haenzsel (CMH) superiority test. Recall that the CMH statistic takes this form:

$$\text{CMH}(h) = \frac{\sum_i w_i d_i}{\sqrt{\sum_i w_i \hat{p}_i \hat{q}_i}}$$

where

$$d_i = p_{A,i} - p_{B,i} \qquad \hat{p}_i = \frac{N_{A,i} \cdot p_{A,i} + N_{B,i} \cdot p_{B,i}}{T_i} \qquad \hat{q}_i = 1 - \hat{p}_i$$

Under the null hypothesis, CMH $\sim \mathcal{N}(0, 1)$ asymptotically. Intuitively, we maximize the power of the test by choosing $N_{A,i}, N_{B,i}$ such that when $p_A \neq p_B$, the distribution of CMH has large mean without inflated variance. Our goal is to find an objective function V that, when maximized, yields trial designs with those characteristics.

As a first candidate we may try maximizing the expected value of of CMH:

$$\begin{aligned}
\mathbb{E}_h\left[\text{CMH}(h)\right] &\simeq \frac{\sum_i w_i (p_A - p_B)}{\sqrt{\sum_i w_i \left(\frac{N_{A,i} \cdot p_A + N_{B,i} \cdot p_B}{T_i}\right)\left(\frac{N_{A,i} \cdot q_A + N_{B,i} \cdot q_B}{T_i}\right)}} \\
&= (p_A - p_B) \frac{\sum_i w_i}{\sqrt{\sum_i w_i \left(\phi p_A + (1-\phi) p_B\right)\left(\phi q_A + (1-\phi) q_B\right)}}
\end{aligned}$$

where $\phi_i = N_{A,i}/T_i$ is the *fraction* of stage i's patients allocated to A. The trial designer has no control over p_A or p_B. So if they wish to maximize this quantity then they may ignore the factor $(p_A - p_B)$, yielding

$$\frac{\sum_i w_i}{\sqrt{\sum_i w_i \left(\phi p_A + (1-\phi)p_B\right)\left(\phi q_A + (1-\phi)q_B\right)}} \tag{A.1}$$

as a proxy objective for maximizing power. It's important to note, however, a subtle property of Expression A.1. The denominator is minimized when more patients are allocated to the treatment with more *extreme* success probability—i.e., success probability closer to 0 or 1. As a result, the maximizer of Expression A.1 exhibits a preference toward that treatment. This preference manifested itself in earlier versions of the algorithm, which would do well when $\frac{1}{2} < p_B < p_A$, but would do worse when $p_B < p_A < \frac{1}{2}$.

As a second candidate, we may try maximizing the the related quantity

$$\frac{\mathbb{E}\left[\sum_i w_i d_i\right]}{\sqrt{\mathrm{Var}\left[\sum_i w_i d_i\right]}} = (p_A - p_B)\frac{\sum_i w_i}{\sqrt{\sum_i w_i^2 \left(\frac{p_A q_A}{N_{A,i}} + \frac{p_B q_B}{N_{B,i}}\right)}} \tag{A.2}$$

$$= (p_A - p_B)\frac{\sum_i w_i}{\sqrt{\sum_i w_i \left((1-\phi_i)p_A q_A + \phi_i p_B q_B\right)}} \tag{A.3}$$

$$\propto \frac{\sum_i w_i}{\sqrt{\sum_i w_i \left((1-\phi_i)p_A q_A + \phi_i p_B q_B\right)}} \tag{A.4}$$

Cochran uses Expression A.2 as a proxy for the power of a CMH test in his original justifications for the CMH statistic (Cochran, 1954). Like Expression A.1, the new Expression A.4 also exhibits a preference based on *extremality* of the success probabilities. However, it instead favors the treatment with *less* extreme success probability, i.e., probability nearer $\frac{1}{2}$. Versions of the algorithm based on Expression A.4 would manifest this preference during simulations. The algorithm would attain superior utility when $p_B < p_A < \frac{1}{2}$, but would do worse when $\frac{1}{2} < p_B < p_A$.

Note the similarity between Expression A.4 and Expression A.1. They have identical numerators, and both denominators have the form $\sqrt{\sum_i w_i \tilde{pq}}$ where $\tilde{pq}$ is some "combined variance" computed from p_A, p_B. They differ precisely in how they compute $\tilde{pq}$. This in turn produces their

different preferences (toward the treatment with less-extreme and more-extreme success probability, respectively). Neither of these preferences are favorable. We would like a proxy for power that has simpler dependence on p_A and p_B, which are unknown. To that end we propose our final candidate:

$$\frac{\sum_i w_i}{\sqrt{\sum_i w_i \frac{1}{2}(p_A + p_B)\frac{1}{2}(q_A + q_B)}} = \frac{\sum_i w_i}{\sqrt{\frac{1}{2}(p_A + p_B)\frac{1}{2}(q_A + q_B)}\sqrt{\sum_i w_i}}$$

$$= \sqrt{\frac{\sum_i w_i}{\frac{1}{2}(p_A + p_B)\frac{1}{2}(q_A + q_B)}}$$

or, after squaring,

$$\frac{\sum_i w_i}{\frac{1}{2}(p_A + p_B)\frac{1}{2}(q_A + q_B)}$$

This new quantity lets $\tilde{pq} = \frac{1}{2}(p_A + p_B)\frac{1}{2}(q_A + q_B)$, which has neither of the preferences exhibited by Expressions A.1 or A.4. Of course in practice we don't know p_A or p_B, so we substitute their MAP estimates at each stage:

$$V(h) = \sum_i \frac{w_i}{\frac{1}{2}(\underline{\hat{p}}_{A,i} + \underline{\hat{p}}_{B,i})\frac{1}{2}(\underline{\hat{q}}_{A,i} + \underline{\hat{q}}_{B,i})}, \tag{A.5}$$

which is the expression for V used in Section 2.1 (up to a factor of $\frac{1}{N}$).

B. The utility of single-stage vs. adaptive trials

It is not always possible for an adaptive trial to attain higher utility (Equation 2.1) than a single-stage trial. The cost λ_K of an additional stage outweighs any improvements in patient outcomes, unless λ_F is large enough. In this appendix we find conditions on λ_F, λ_K that determine when a two-stage adaptive trial can attain higher utility than a single-stage one.

Given the true values of p_A and p_B, we can compute the utility of a single-stage trial in closed form:

$$U_{\text{single}} = \frac{1}{(p_A + p_B)(q_A + q_B)} - \lambda_K$$

We can likewise compute the utility of a two-stage trial in closed form. Assume the first stage of the trial treats T patients, assigning half to each treatment. In the second stage, assume $\phi \cdot (N-T)$ patients are assigned to treatment A and $(1-\phi) \cdot (N-T)$ are assigned to treatment B. Then the two-stage trial has this utility:

$$U_{\text{two-stage}} = \frac{T + (N-T)\phi(1-\phi)}{N(p_A + p_B)(q_a + q_B)} + \frac{\lambda_F}{N}(2\phi - 1)(N-T)(p_A - p_B) - 2\lambda_K$$

We want to find conditions where $U_{\text{two-stage}} - U_{\text{single}} \geqslant 0$. Some algebra yields this condition:

$$-\frac{1}{(p_A + p_B)(q_A + q_B)}\hat{\phi}^2 + \lambda_F(p_A - p_B)\hat{\phi} - \lambda_B \frac{N}{N-T} \geqslant 0$$

where $\hat{\phi} = (2\phi - 1)$ is a convenient shorthand. The LHS of this inequality is a concave quadratic in $\hat{\phi}$. It has real roots (and hence, a feasible region) iff

$$\lambda_F^2(p_A - p_B)^2(p_A + p_B)^2(q_A + q_B)^2 - 4\frac{\lambda_B(p_A + p_B)(q_A + q_B)N}{N-T} \geqslant 0.$$

Rearranging gives the following condition on λ_F

$$\lambda_F \geqslant \frac{2}{(p_A - p_B)}\sqrt{\frac{N\lambda_K}{(N-T)(p_A + p_B)(q_A + q_B)}}.$$

The key takeaway is that λ_F must be sufficiently large before *any* adaptive design can possibly outperform the single-stage traditional trial, with respect to the utility function.

This analysis does not account for the uncertainty in p_A and p_B. The algorithm operates under this uncertainty, and will therefore not generally choose a single-stage design when it is truly optimal. In practice, the algorithm only chooses a single-stage design when λ_F is unusually small relative to λ_K.

C. Optimal Substructure

We show that our optimization problem—maximizing expected utility—possesses the *optimal substructure* property. In other words, we prove that the recurrence relation (Equation 2.3) cor-

rectly decomposes the problem into subproblems, and reuses their solutions to solve the original problem.

Suppose the algorithm is evaluating $\overline{U}(s_i)$ for some state s_i, and that it's already evaluated $\overline{U}(s_{i+1})$ for every possible successor state s_{i+1} of s_i. Let π^* denote the policy maximizing $\overline{U}$. Then optimal substructure follows from the linearity of our utility function. Assuming s_{i+1} is not terminal:

$$\begin{aligned}
\overline{U}(s_i) &= \mathbb{E}_{h_{i+1}|\pi^*,s_i}\left[\frac{1}{N}V(h_{i+1}) - \lambda_F{\cdot}F(h_{i+1}) - \lambda_K{\cdot}K(h_{i+1})\right] \\
&= \mathbb{E}_{(s_{i+1},h_{i+2})|\pi^*,s_i}\left[\frac{w(a^*)}{N\tilde{p}\tilde{q}(s_{i+1})} + \frac{1}{N}V(h_{i+2}) - \lambda_F{\cdot}F(h_{i+2}) - \lambda_K{\cdot}(1+K(h_{i+2}))\right] \\
&= \mathbb{E}_{s_{i+1}|\pi^*,s_i}\left[\mathbb{E}_{h_{i+2}|\pi^*,s_{i+1}}\left[\frac{w(a^*)}{N\tilde{p}\tilde{q}(s_{i+1})} + \frac{1}{N}V(h_{i+2}) - \lambda_F{\cdot}F(h_{i+2}) - \lambda_K{\cdot}(1+K(h_{i+2}))\right]\right] \\
&= \mathbb{E}_{s_{i+1}|\pi^*,s_i}\left[\frac{w(a^*)}{N\tilde{p}\tilde{q}(s_{i+1})} - \lambda_K + \mathbb{E}_{h_{i+2}|\pi^*,s_{i+1}}\left[\frac{1}{N}V(h_{i+2}) - \lambda_F{\cdot}F(h_{i+2}) - \lambda_K{\cdot}K(h_{i+2})\right]\right] \\
&= \mathbb{E}_{s_{i+1}|\pi^*,s_i}\left[R(a^*, s_{i+1}) + \overline{U}(s_{i+1})\right] \\
&= \max_a \left\{\mathbb{E}_{s_{i+1}|s_i}\left[R(a, s_{i+1}) + \overline{U}(s_{i+1})\right]\right\}
\end{aligned}$$

which agrees exactly with the recurrence in Equation 2.3. A similar computation covers the case when s_{i+1} is terminal.

Put another way, our dynamic program's recurrence relation computes $\overline{U}(s)$ correctly at each state, and will yield the policy maximizing $\overline{U}(s_0)$.

D. Computational complexity

We derive the space and time complexities given in Section 2.2.

Let S_i denote the set of all 2×2 contingency tables containing i observations. Define $[T_{\min} : \kappa : N - T_{\min}] = \{T_{\min}, T_{\min} + \kappa, \ldots, N - T_{\min}\}$, the set of integers ranging from $T_{\min}$ to $N - T_{\min}$

in increments of κ. Then $|S_i| = O(i^3)$, and the size of the full state space is

$$|S| \;=\; \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} |S_i| \;=\; \sum_{i=T_{\min}}^{N-T_{\min}} O\left(\frac{i^3}{\kappa}\right) \;=\; O\left(\frac{(N-T_{\min})^4}{\kappa}\right).$$

The algorithm stores data proportional to $|S|$, so this gives the space complexity.

The time complexity results from a nested sum over states, actions, and transitions:

$$\begin{aligned}
T(N,|\Phi|) &= \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} \sum_{s\in S_i} \sum_{j\in[T_{\min}:\kappa:N-T_{\min}-i]} \sum_{\phi\in\Phi} \phi{\cdot}j(1-\phi){\cdot}j \\
&= \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} \sum_{s\in S_i} \sum_{j\in[T_{\min}:\kappa:N-T_{\min}-i]} j^2 \sum_{\phi\in\Phi} \phi(1-\phi) \\
&= \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} \sum_{s\in S_i} \sum_{j\in[T_{\min}:\kappa:N-T_{\min}-i]} j^2{\cdot}O(|\Phi|) \\
&= O(|\Phi|) \cdot \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} \sum_{s\in S_i} \sum_{j\in[T_{\min}:\kappa:N-T_{\min}-i]} j^2 \\
&= O(|\Phi|) \cdot \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} O(i^3) \sum_{j\in[T_{\min}:\kappa:N-T_{\min}-i]} j^2 \\
&= O(|\Phi|) \cdot \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} O(i^3) O\left(\frac{(N-T_{\min}-i)^3}{\kappa}\right) \\
&= O(|\Phi|) \cdot O\left(\frac{(N-T_{\min})^3}{\kappa}\right) \cdot \sum_{i\in[T_{\min}:\kappa:N-T_{\min}]} O(i^3) \\
&= O\left(|\Phi| \cdot \frac{(N-T_{\min})^7}{\kappa^2}\right).
\end{aligned}$$

[*Received Month DD, 2021; revised Month DD, 2021; accepted for publication Month DD, 2021*]