Planning to Fairly Allocate: Probabilistic Fairness in the Restless Bandit Setting

Christine Herlihy\textsuperscript{*}  
Aviva Prins\textsuperscript{*}  
cherlihy@cs.umd.edu  
aviva@cs.umd.edu  
University of Maryland  
College Park, Maryland, USA  

Aravind Srinivasan  
arsriv1@umd.edu  
University of Maryland  
College Park, Maryland, USA  

John P. Dickerson  
johnd@umd.edu  
University of Maryland  
College Park, Maryland, USA  

\textbf{ABSTRACT}

Restless and collapsing bandits are often used to model budget-constrained resource allocation in settings where arms have action-dependent transition probabilities, such as the allocation of health interventions among patients. However, SOTA Whittle-index-based approaches to this planning problem either do not consider fairness among arms, or incentivize fairness without guaranteeing it. We thus introduce \textit{ProbFair}, a probabilistically fair policy that maximizes total expected reward and satisfies the budget constraint while ensuring a strictly positive lower bound on the probability of being pulled at each timestep. We evaluate our algorithm on a real-world application, where interventions support continuous positive airway pressure (CPAP) therapy adherence among patients, as well as on a broader class of synthetic transition matrices. We find that \textit{ProbFair} preserves utility while providing fairness guarantees.

\textbf{CCS CONCEPTS}

\begin{itemize}
  \item Computing methodologies → Partially-observable Markov decision processes; Multi-agent planning; Applied computing
  \item Health care information systems.
\end{itemize}

\textbf{KEYWORDS}

probabilistic fairness; resource allocation; restless multi-armed bandits; collapsing multi-armed bandits; Whittle index; intervention planning; sequential decision making; POMDP; healthcare

\textbf{ACM Reference Format:}


\section{1 INTRODUCTION}

Restless multi-armed bandits (RMABs) are used to model budget-constrained resource allocation tasks in which a decision-maker must select a subset of arms (e.g., projects, patients, assets) to receive a beneficial intervention at each timestep, while the state of each arm evolves over time in an action-dependent, Markovian fashion. Such problems are common in healthcare, where clinicians may be tasked with monitoring large, distributed patient populations and determining which individuals to expend scarce resources on so as to maximize total welfare. RMABs have been proposed to determine which inmates should be prioritized to receive hepatitis C treatment in U.S. prisons [2], and which tuberculosis patients should receive medication adherence support in India [24].

Current state-of-the-art approaches to solving RMABs rely on the indexing work introduced by Whittle [42]. While the Whittle index solves an otherwise PSPACE-complete problem in an asymptotically optimal fashion by decoupling arms [41], it fails to provide any guarantees about how pulls will be distributed among arms.

Though the intervention is canonically assumed to be beneficial for every arm, the marginal benefit (i.e., relative increase in the probability of a favorable state transition) varies in accordance with each arm’s underlying state transition function. Consequently, Whittle index-based maximization of total expected reward \textit{without regard for distributive fairness} empirically allocates all available interventions to a small subset of arms, ignoring the rest [29].

There are many application domains where a bimodal distributive outcome may be perceived as unfair or undesirable by beneficiaries and decision-makers, thus motivating efforts to incentivize or guarantee distributive fairness. In the aforementioned healthcare examples, resource constraints and variation in transition dynamics interact. A practical consequence is that a majority of patients will never receive the beneficial intervention(s) in question. This, in turn, means that their clinical outcomes will be strictly worse in expectation than they would be under a policy that guaranteed a non-zero probability of receiving the intervention at each timestep.

To improve distributive fairness, we explore whether it is possible to modify the Whittle index to guarantee each arm at least one pull per user-defined time interval, but find this to be intractable. We then introduce \textit{ProbFair}, a state-agnostic policy that maps each arm to a fairness-constraint satisfying, stationary probability distribution over actions that takes the arm’s transition matrix into account. At each timestep, we then use a dependent rounding algorithm [39] to sample from this probabilistic policy to produce a budget-constraint satisfying discrete action vector.

We evaluate \textit{ProbFair} on a randomly generated dataset and a realistic dataset derived from obstructive sleep apnea patients tasked with nightly self-administration of continuous positive airway pressure (CPAP) therapy [16, 17].
Our core contributions include:

(i) A novel approach that is both efficiently computable and reward maximizing, subject to the guaranteed satisfaction of budget and probabilistic fairness constraints.

(ii) Empirical results demonstrating that ProbFair is competitive vis-à-vis other fairness-inducing policies, and stable over a range of cohort composition scenarios.

2 RESTLESS MULTI-ARMED BANDIT MODEL

Here, we give an overview of the restless multi-armed bandit (RMAB) framework, along with our proposed extension, which takes the form of a fairness-motivated constraint. A restless multi-armed bandit consists of \( N \in \mathbb{N} \) independent arms, each of which evolves over a finite horizon \( T \in \mathbb{N} \), according to an associated Markov Decision Process (MDP). Each arm’s MDP is characterized by a 4-tuple \((S, A, P, r)\) where \( S \) represents the state space, \( A \) represents the action space, \( P \) represents an \( |S| \times |A| \times |S| \) transition matrix, and \( r : S \times A \rightarrow \mathbb{R} \) represents a local reward function that maps states to real-valued rewards. Appendix A summarizes notation; note that \( [N] \) denotes the set \( \{1, 2, \ldots, N\} \).

States, actions, and observability: We specifically consider a discrete two-state system \( S := \{0, 1\} \) where \( 1 \) represents being in the “good” ("bad") state, and a set of two possible actions \( A := \{0, 1\} \) where 1 represents the decision to select ("pull") arm \( i \in [N] \) at time \( t \in [T] \), and 0 represents the choice to be passive (not pull). In the general RMAB setting, each arm’s state \( s_t^i \) is observable. We consider the partially-observable extension introduced by Mate et al. [24], where arms’ states are only observable when they are pulled. Otherwise, an arm’s state is replaced with the probabilistic belief \( b_t^i \in [0, 1] \) that it is in state 1. Such partial observability captures uncertainty regarding patient status and treatment efficacy associated with outpatient or remotely-administered interventions.

Transition matrices: Each arm \( i \) is characterized by a set of transition matrices \( P \), where \( P_{s_t^i}^{s_{t+1}^i} \) represents the probability of transitioning from state \( s_t^i \) to state \( s_{t+1}^i \) when action \( a_t^i \) is taken. We assume \( P \) to be (a) static and (b) known by the agent at planning time. Assumptions (a) and (b) are likely to be violated in practice; however, they provide a useful modeling foundation, and can be modified to incorporate additional uncertainty, such as the requirement that transition matrices must be learned [15]. Clinical researchers often use longitudinal data to construct risk-adjusted transition matrices that encode cohort-specific transition probabilities. These can guide patient-level decision-making [40].

Consistent with previous literature, we assume strictly positive transition matrix entries, and impose four structural constraints: (a) \( P_{0,0}^1 < P_{0,1}^1 \); (b) \( P_{1,0}^1 < P_{1,1}^1 \); (c) \( P_{0,0}^1 < P_{0,1}^1 \); (d) \( P_{1,0}^1 < P_{1,1}^1 \) [24]. These constraints are application-motivated, and imply that arms are more likely to remain in a “good” state than change from a bad state to a good one, and that a pull is helpful when received. In the absence of such constraints, the effect of the intervention may be superfluous or harmful, rather than desirable.

Objective and constraints: In the canonical RMAB setting, the agent’s goal is to find a policy \( \pi^* \) that maximizes total expected reward \( \arg \max \mathbb{E}_{\pi}[R(\tau(\pi))] \) while satisfying a budget constraint, \( k \ll N \in \mathbb{N} \), which allows the agent to select at most \( k \) arms at each timestep. We consider a cumulative reward function, \( R(\cdot) := \sum_{t \in [N]} \sum_{t \in [T]} \beta^{t-1} r(s_t^i) \), for some discount rate \( \beta \in [0, 1] \), and non-decreasing \( r(s) \).

We extend this model by introducing a Boolean-valued, distributive fairness-motivated constraint, which may take one of two general forms:

1. **Time-indexed**: A function \( g(\cup_{t \in [T]} \{a_t^i\}) \) that is satisfied if each arm is pulled at least once within each user-defined time interval \( v \leq T \) (e.g., at least once every seven days), or a minimum fraction \( \gamma \in (0, 1) \) of times over the entire time horizon [20].

2. **Probabilistic**: A function \( g'(\tilde{P} | \tilde{a}_t^i \sim \tilde{P} \forall t) \) which operates on the stationary probability vector \( \tilde{P} \), from which discrete actions are drawn, by requiring the probability that each arm receives a pull at any given \( t \) to fall within an interval \( [\ell, u] \) where \( 0 < \ell \leq \frac{T}{N} \leq u \leq 1 \).

3 CONTEXT, MOTIVATION & RELATED WORK

In this section, we motivate our ultimate focus on probabilistic fairness by revisiting the distribution of pulls under Whittle-index based policies. We begin by providing background information on the Whittle index, and then proceed to ask: (1) Which arms are ignored, and why does it matter? (2) Is it possible to modify the Whittle index so as to provide a time-indexed fairness guarantee for each arm? In response to the latter, we demonstrate that time-indexed fairness guarantees necessitate the coupling of arms, which undermines the indexability of the problem. We then identify prior work at the intersection of algorithmic fairness, constrained resource allocation, and multi-armed bandits, and identify desiderata that characterize our own approach.

3.1 Background: Whittle Index-based Policies

Pre-computing the optimal policy for a given set of restless or collapsing arms is PSPACE-hard in the general case [28]. However, as established by Whittle [42] and formalized by Weber and Weiss [41], if the set of arms associated with a problem are indexable, we can decouple the arms and efficiently solve the problem using an asymptotically-optimal heuristic index policy.

Mechanics: At each timestep \( t \in [T] \), the value of a pull, in terms of both immediate and expected discounted future reward, is computed for each decoupled arm, \( i \in [N] \). This value-computation step relies on the notion of a subsidy, \( m \), which can be thought of as the opportunity cost of passivity. Formally, the Whittle index is the subsidy required to make the agent indifferent between pulling and not pulling arm \( i \) at time \( t \). (Per Section 2, \( b \) denotes the probabilistic belief that an arm is in state \( s = 1 \); for restless arms, \( b_t^i = s_t^i \in \{0, 1\} \).

\[
W(b_t^i) = \inf_m \left\{ m \mid V_m(b_t^i, a_t^i = 0) = V_m(b_t^i, a_t^i = 1) \right\}
\]

The value function \( V_m(b_t^i) \) represents the maximum expected discounted reward under passive subsidy \( m \) and discount rate \( \beta \) for arm \( i \) with belief state \( b_t^i \in [0, 1] \) at time \( t \):

\[
V_m(b_t^i) = \max \left\{ m + r(b_t^i) + \beta V_m(b_{t+1}^i) \right\} \text{ passive}
\]

\[
V_m(b_t^i) = \max \left\{ r(b_t^i) + \beta b_t^i V_m(p_{1,1}^i) + (1 - b_t^i) V_m(p_{0,1}^i) \right\} \text{ active}
\]
Once the Whittle index has been computed for each arm, the agent sorts the indices, and the k arms with the greatest index values receive a pull at time t, while the remaining N – k arms are passive. Weber and Weiss [41] give sufficient conditions for indexability:

**Definition 3.1.** An arm is indexable if the set of beliefs for which it is optimal to be passive for a given m, \( \mathcal{B}^*(m) = \{ \pi \mid \forall r \in \Pi_m, \pi(b) = 0 \} \), monotonically increases from 0 to the entire belief space as m increases from \(-\infty\) to \(+\infty\). An RMAB is indexable if every arm is indexable.

Indexability is often difficult to establish, and computing the Whittle index can be complex [21]. Prevailing approaches rely on proving the optimality of a threshold policy for a subset of transition matrices [27]. A forward threshold policy pulls an arm when its state is at or below a given threshold, and makes the arm passive otherwise; the converse is true for a reverse threshold policy. Mate et al. [24] give such conditions for this RMAB setting, when \( r(b) = b \), and provide an algorithm, Threshold Whittle, that is asymptotically optimal for forward threshold-optimal arms. Mate et al. [25] expand on this work for any non-decreasing \( r(b) \) and present the Risk-Aware Whittle algorithm.

### 3.2 Motivation: Individual Welfare & Whittle

**Bimodal allocation:** Existing theory does not offer any guarantees about how the sequence of actions will be distributed over arms under Whittle index-based policies, nor about the probability with which a given arm can expect to be pulled at any particular timestep. Prins et al. [29] demonstrate that Whittle-based policies tend to allocate all pulls to a small number of arms, neglecting most of the population. We present similar findings in Appendix B.

This bimodal distribution is a consequence of how the Whittle index prioritizes arms. Whittle favors arms for whom a pull is most beneficial to achieving sustained occupancy in the “good” state, regardless of whether this results in the same subset of arms repeatedly receiving pulls. While the structural constraints in Sec. 2 ensure that a pull is beneficial for every arm, marginal benefit varies. Since reward is a function of each arm’s underlying state, arms whose trajectories are characterized by a relative—but not absolute—indifference to the intervention are likely to be ignored.

**Ethical implications:** This zero-valued lower bound on the number of pulls an arm can receive aligns with a utilitarian approach to distributive justice, in which the decision-maker seeks to allocate resources so as to maximize total expected utility [3, 23]. This may be incompatible with competing pragmatic and ethical desiderata, including egalitarian and prioritarian notions of distributive fairness, in which the decision-maker seeks to allocate resources equally among arms (e.g., Round-Robin), or prioritize arms considered to be worst-off under the status quo, for some quantifiable notion of worst-off that induces a partial ordering over arms [33, 38]. We consider the worst-off to be arms who would be deprived of algorithmic attention (e.g., not receive any pulls), or, from a probabilistic perspective, would have a zero-valued lower bound on the probability of receiving a pull at any given timestep.

**Why algorithmic attention?** This choice is motivated by our desire to improve equality of opportunity (i.e., access to the beneficial intervention) rather than equality of outcomes (i.e., observed adherence). The agent directly controls who receives the intervention, but has only indirect control (via actions) over the sequence of state transitions an arm experiences. Additionally, proclivity towards adherence may vary widely in the absence of restrictive assumptions about cohort homogeneity, and focusing on equality of outcomes could thus entail a significant loss of total welfare.

**Distributive fairness and algorithmic acceptability:** To realize the benefits associated with an algorithmically-derived resource allocation policy, practitioners tasked with implementation must find the policy to be acceptable (i.e., in keeping with their professional and ethical standards), and potential beneficiaries must find participation to be rational.

With respect to practitioners, many clinicians report experiencing mental anguish when resource constraints force them to categorically deny a patient access to a beneficial treatment, and may resort to providing improvised and/or sub-optimal care [6]. Providing fairness-aware decision support can improve acceptability [18, 32] and minimize the loss of reward associated with ethically-motivated deviation to a sub-optimal but equitable approach such as Round-Robin [8, 9]. For beneficiaries, we posit that an arm may consider participation rational when it results in an increase in expected time spent in the adherent state relative to non-participation (e.g., due to receiving a strictly positive number of pulls in expectation).

### 3.3 Time-indexed Fairness and Indexability

We now consider whether it is possible to modify the Whittle index to guarantee time-indexed fairness while preserving our ability to decouple arms. Unfortunately, the answer is no—we provide an overview here and a detailed discussion in Appendix D.1. Recall that structural constraints ensure that when an arm is considered in isolation, the optimal action will always be to pull, and that a Whittle-index approach computes the infimum subsidy, m, an arm requires to accept passivity at time t. Whether or not arm i is actually pulled at time t depends on how the subsidy of one arm compares to the infimum subsidies required by other arms. Thus, any modification intended to guarantee time-indexed fairness must be able to alter the ordering among arms, such that any arm i which would otherwise have a subsidy with rank > k when sorted in descending order will now be in the top-k arms. Even if we could construct such a modification for a single arm without requiring time-stamped system information, if every arm had this same capability, then a new challenge would arise: we would be unable to distinguish among arms, and arbitrary tie-breaking could again jeopardize fairness constraint satisfaction.

### 3.4 Additional Related Work

While multi-armed bandit problems are canonically framed from the perspective of the decision-maker, interest in individual and group fairness in this setting has grown in recent years [7, 14, 20].

In the stochastic multi-armed bandit setting, each arm is characterized by a fixed but unknown average reward rather than by an MDP. The decision-maker thus faces uncertainty about the true utility of each arm and must balance exploration (i.e., pulling arms to gain information about their reward distributions) with exploitation (i.e., pulling the optimal arm(s)) to maximize expected reward. Joseph et al. [14] examine fairness among arms in this setting, and
introduce a definition that requires the decision-maker to favor (i.e., select) arms with higher average reward over arms with lower average reward, even in the face of uncertainty. As the authors note, this definition is consistent with reward maximization, but imposes a cost in terms of per-round regret when learning the optimal policy, due to the fact that arms with overlapping confidence intervals are chained until they can be resolved with high confidence.

Prior work in other non-restless bandit settings demonstrates that alternative definitions—i.e., those which center on a distributional fairness among arms as opposed to the principle that arms with similar average rewards should be treated similarly [10], generally entail deviation from optimal behavior. Li et al. [20] study the combinatorial sleeping bandit setting, in which arms are stochastic but may be unavailable at any given timestep. They introduce the minimum selection fraction constraint, which we adapt and refer to as the state-indexed fairness (see Section 2). Chen et al. [7] consider the contextual bandit setting, and propose an algorithm that guarantees each arm a minimum probability of selection at each timestep.

In the restless setting that we consider, prior works have tended toward opposite ends of the reward-fairness spectrum by either: (1) redistributing pulls without providing arm-level guarantees [19, 25]; or (2) guaranteeing time-indexed fairness without providing optimality guarantees [29]. Recent work has also considered the adjacent problem of fairness among intervention providers (i.e., workers) [5]. In contrast to prior work, we aim to guarantee rather than incentivize fairness, without incurring an exponential dependency on the time horizon or sacrificing optimality guarantees. We thus seek an efficient policy that is reward maximizing, subject to the satisfaction of both budget and probabilistic fairness constraints.

4 METHODOLOGICAL APPROACH

Here we introduce ProbFair, an approximately optimal solution to a relaxed version of the allocation task in which we guarantee the satisfaction of probabilistic rather than time-indexed fairness, along with the budget constraint. This relaxation is necessary for tractability, as it allows us to precompute a stationary, state-agnostic probability vector, \( \tilde{p} \), from which constraint-satisfying discrete actions are drawn.

ProbFair maps each arm \( i \) to an arm-specific, stationary probability distribution over actions, such that for each timestep \( t \), \( P(a_i^t = 1) = p_i \) and \( P(a_i^t = 0) = 1 - p_i \), where \( p_i \in [0, 1] \) for all \( i \in [N] \) and \( \sum_i p_i = k \). Here, \( t \) and \( u \) are user-defined fairness parameters satisfying \( 0 < t \leq \frac{1}{N} \leq u \leq 1 \), per Section 2. Note that \( tT \) and \( uT \) can be interpreted as lower and upper bounds on the expected number of pulls an arm will receive over the time horizon.

In Section 4.1, we describe how to construct the \( p_i \)'s so as to efficiently approximate our constrained reward-maximization objective within a multiplicative factor of \( (1 - \epsilon) \), for any given constant \( \epsilon > 0 \). We use a dependent rounding approach detailed in Section 4.2 to sample from this distribution at each timestep independently, to produce a discrete action vector, \( \tilde{a}_i \in [0, 1]^N \), which is guaranteed to satisfy the budget constraint, \( k \) [39].

To motivate our approach, note that when we take the union of each arm’s stationary probability vector, we obtain a system-level policy, \( \pi_{PF} : \{ i \mid i \in N \} \rightarrow [1 - p_i, p_i]^N \). Regardless of the system’s initial state, repeated application of this policy will result in convergence to a steady-state distribution in which (WLOG) arm \( i \) is in the adherent state (i.e., state 1) with probability \( x_i \in [\ell, u] \), and the non-adherent state (i.e., state 0) with probability \( 1 - x_i \in [0, 1] \). By definition, for any arm \( i \), \( x_i \) will satisfy the equation:

\[
\begin{align*}
x_i &= \mathbb{P}(\text{arm } i \text{ is adherent at time } T) \\
&= \argmax_{p_i \in [\ell, u]} \left[ (1 - p_i)P_{0,1}^i + p_iP_{1,1}^i \right] + (1 - x_i)(1 - p_i)P_{0,1}^0 + p_iP_{1,1}^0 \in [\ell, u].
\end{align*}
\]

Thus, \( x_i = f_i(p_i) \), where

\[
f_i(p_i) = \frac{(1 - p_i)P_{0,1}^0 + p_iP_{1,1}^i}{1 - (1 - p_i)P_{0,1}^0 + p_i P_{1,1}^i} = \frac{(1 - p_i)P_{0,1}^0 + p_iP_{1,1}^i}{1 - (1 - p_i)P_{0,1}^0 + p_i P_{1,1}^i}
\]

We seek the policy which maximizes total expected reward, where reward is non-decreasing in \( s \) (i.e., with time spent in the adherent state). Thus, ProbFair is defined as:

\[
\pi_{PF} = \arg \max_{p_i \in [\ell, u]} \sum_i f_i(p_i) \text{ s.t. } \sum_i p_i \leq k
\]

Solving this constrained maximization problem is thus consistent with maximizing the expected number of timesteps each arm will spend in the adherent state, subject to satisfying the budget and probabilistic fairness constraints. We emphasize that our construction process takes the transition matrices of each arm \( i \) into account via \( f_i \) (Equation 3).

4.1 Computing the \( p_i \)'s: Algorithmic Approach

Overview: To construct \( \pi_{PF} \), we: (1) partition the arms based on the shapes of their respective \( f_i \) functions (Eq. 3); (2) perform a grid search over possible ways to allocate the budget, \( k \), between the two subsets of arms; (2a) solve each sub-problem to produce a probabilistic policy for the arms in that subset; (2b) compute the total expected reward of the policy; (3) take the argmax over this set of grid search values to determine the approximately optimal budget allocation; and (4) form \( \pi_{PF} \) by taking the union over the policies produced by evaluating each sub-problem at its approximately optimal share of the budget. Figure 1 visualizes; the remainder of this section provides technical details.

Figure 1: ProbFair: constructing and sampling from \( \pi_{PF} \)

\( k \quad \epsilon \quad \) arms \( \{ i | i \in [N] \} \)

\( \) candidate \( x \) values \( x : \{ k, \ell, \ldots, (k - (k - 1)u) \} \)

\( \) concave arms \( \mathcal{X} := \{ f_i \text{ concave} \} \)

\( \) strictly convex arms \( \mathcal{Y} := \{ f_i \text{ strictly convex} \} \)

\( \) solve \( P_1 \) via gradient descent

\( \) candidate \( \tilde{f}_P \)

\( \sum_{p_i \in \mathcal{X}} f_i(p_i) + \sum_{p_i \in \mathcal{Y}} f_i(p_i) \)

\( \) compute \( \mathcal{Y}_e := \{ f_i \text{ strictly convex} \} \)

\( \) argmax \( \mathcal{Y}_e \)

\( \) for \( t \in T \)

\( \pi_{PF} := \pi_{\mathcal{X}_e} \cup \pi_{\mathcal{Y}_e} \)

\( \pi_{PF} : \{ i \mid i \in N \} \rightarrow [1 - p_i, p_i]^N \)

where \( \pi_{PF} : \{ \} \rightarrow \mathcal{X} \) s.t. \( \pi_{PF} : \{ i \mid i \in [N] \} \rightarrow \mathcal{Y} \)
To begin, we introduce two theorems (see App. E for full proofs):

**Theorem 4.1.** For every arm \( i \in [N] \), \( f_i(p_i) \) is either concave or strictly convex in all of \( p_i \in [0, 1] \).

**Proof Sketch.** WLOG, fix an arm \( i \in [N] \). For notational convenience, let us define the following constants derived from the arm’s transition matrix \( P^i \): \( c_1 := p_{0,1}^i, c_2 := p_{1,0}^i - p_{0,1}^i, c_3 := 1 - p_{1,0}^i + p_{0,1}^i \), and \( c_4 := p_{0,1}^i - p_{1,1}^i - p_{0,1}^i + p_{1,0}^i \). Then

\[
f''_i(p_i) = \frac{2c_2^2}{c_3^3} \left( c_1 (1 - c_2) - c_2 c_4 \right)^2.
\]

By the structural constraints introduced in Section 2 ensure that \( f_i(p_i) \) is monotonically non-decreasing in \( p_i \) over the interval \([0, 1]\).

**Theorem 4.2.** For each arm \( i \in [N] \), the structural constraints introduced in Section 2 ensure that \( f_i(p_i) \) is either concave or strictly convex in all of \( p_i \).

**Proof Sketch.** WLOG, fix an arm \( i \in [N] \). Theorem 4.1 ensures that the arms can be partitioned into two disjoint sets: \( X = \{ i \mid f_i \) is concave\} and \( Y = \{ i \mid f_i \) is strictly convex\}. (P1) maximize \( \sum_{i \in X} f_i(p_i) \) subject to: \( p_i \in [t, u] \) for all \( i \in X \), and \( \sum_{i \in X} p_i = k \). (P2) maximize \( \sum_{i \in Y} f_i(p_i) \) subject to: \( p_i \in [t, u] \) for all \( i \in Y \), and \( \sum_{i \in Y} p_i = k - z \). Then, \( \pi_{PF} \) is the union of the solutions to P1 and P2 at the optimal grid search value \( z^* = \arg \max_{z} \sum_{i \in X} f_i(p_i) + \sum_{i \in Y} f_i(p_i) \). Algorithm 1 provides pseudocode.

**Algorithm 1 ProBFair**

1. procedure ProBFair\((N, k, t, u)\)
2. \( X \leftarrow \{ i \mid f_i \) is concave\} in all of \( p_i \in [0, 1]\}
3. \( Y \leftarrow \{ i \mid f_i \) is strictly convex\} in all of \( p_i \in [0, 1]\}
4. grid_search_vals \( \leftarrow \{ \ell \mid \ell \in [0, k - |Y|]\}\}
5. for each \( \ell \in \) grid_search_vals do
6. \( \pi_{X,\ell} \leftarrow \text{SolveP1}(X, k, \ell, u)\)
7. \( \pi_{Y,\ell} \leftarrow \text{SolveP2}(Y, k, \ell, u)\)
8. \( V_2 \leftarrow \sum_{p_i \in \pi_{X,\ell}} f_i(p_i) + \sum_{p_i \in \pi_{Y,\ell}} f_i(p_i)\)
9. \( z^* \leftarrow \arg \max_{\ell} V_2\)
10. \( \pi_{PF} \leftarrow \pi_{X, z^*} \cup \pi_{Y, z^*}\)
11. return \( \pi_{PF}\)

**P1** is a concave-maximization problem that can be solved efficiently via gradient descent. The computational complexity is \( O\left(\frac{|X|}{t^2}\right)\) [26]. To solve P2, we begin by introducing a lemma that we prove in Appendix E:

**Lemma 4.3.** P2 has an optimal solution in which \( p_i \in (t, u) \) for at most one \( i \) in \( Y \).

**Proof Sketch.** Suppose for contradiction there exists some optimal solution \( \hat{p} \) with distinct indices \( i, j \in Y \) such that \( p_i, p_j \in (t, u) \). Let us compare \( \hat{p} \) with a perturbed solution \( p_i := p_i + \epsilon \) and \( p_j := p_j - \epsilon \). Using a Taylor series expansion, the change in objective must be \((\epsilon^2/2) \cdot (f''_i(p_i) + f''_j(p_j)) + o(\epsilon^3)\). Since \( f_i \) and \( f_j \) are strictly convex, \( f''_i(p_i) + f''_j(p_j) > 0 \). Thus, the objective increases regardless of the sign of \( \epsilon \), a contradiction.

**Theorem 4.4.** Alg. 2 yields the mapping from arms in \( Y \) to subsets in \( \{ Y_1, Y_2, \ldots, Y_s \} \) which maximizes \( \sum_{i \in Y} f_i(p_i) \) s.t. \( \sum_{i \in Y} p_i = k - z \). (See Appendix E for the complete proof).

**Proof Sketch.** By Lemma 4.3, there exists at most one arm with optimal value \( p^*_i \in (t, u) \). By Lemma E.1, \( Y := \{ i \mid \sum_{j \in Y} f_j(p_j) \text{ at most one } i \in [N] \} \) and \( p^* = k - z - \sum_{i \in Y} f_i(p_i) + \max_{i \in Y} f_i(p_i) \). Then, we can rewrite Equation 4 as an optimization problem over set assignment:

\[
\arg \max_{\{ y_1, y_2 \}, y_1 \subseteq Y, y_2 \subseteq Y \setminus y_1} \sum_{i \in y_1} f_i(p_i) + f_j(p_j) + \sum_{i \in y_2} f_i(p_i)
\]

s.t. \( |Y_1| = y, Y_2 = \{ j \}, \sum_{x=1}^{3} \cup_{x=1}^{3} Y_x = \emptyset, \) and \( \cup_{x=1}^{3} Y_x = Y \).

By algebraic manipulation, assigning the \( |Y| - y - 1 \) arms with maximal values of \( f_i(p_i) - f_j(p_j) \) is maximal among the remaining arms. By definition, \( Y^* = Y \setminus \{ Y_2 \cup Y_3 \} \), which completes the proof.
Corollary 4.5. Alg. 2 has time complexity $O(\lceil \log |\mathcal{Y}| \rceil \log |\mathcal{Y}|)$.

With our solutions to P1 and P2 so defined, the cost of finding our probabilistic policy in this way is $O\left(\frac{\log |\mathcal{Y}|}{\epsilon} + |\mathcal{X}| \log |\mathcal{Y}|\right)$, which is at worst $O\left(\frac{\log N}{\epsilon^2}\right)$ when all $N$ arms are in $X$.

4.2 Sampling Approach

For problem instances with feasible solutions, Algorithm 1 returns $\pi^*_P$, a mapping from the set of arms to a set of stationary probability distributions over actions, such that for each arm $i$, the probability of receiving a pull at any given timestep is in $[t, u]$. By virtue of the fact that $t > 0$, this policy guarantees probabilistic fairness constraint satisfaction for all arms. We use a linear-time algorithm introduced by Srinivasan [39] and detailed in Appendix E.2 to sample from $\pi^*_P$ at each timestep, such that the following properties hold: (1) with probability one, we satisfy the budget constraint by pulling exactly $k$ arms; and (2) any given arm $i$ is pulled with probability $p_i$. Formally, each time we draw a vector of binary random variables $(X_1, X_2, \ldots, X_N)$ from the distribution $\pi^*_P$, $\Pr\left[|i: X_i = 1| = k\right] = 1$ and $\forall i, \Pr\left[X_i = 1\right] = p_i$.

5 EXPERIMENTAL EVALUATION

In this section, we empirically demonstrate that ProbFair enforces the probabilistic fairness constraint introduced in Section 2 with minimal loss in total expected reward, relative to fairness-aware alternatives. We begin by identifying our comparison policies, evaluation metrics, and datasets. We then present results from three experiments: (1) ProbFair versus fairness-inducing alternative policies, holding the cohort fixed and considering fairness-aligned sets of hyperparameters; (2) ProbFair evaluated on a breadth of cohorts representing different types of patient populations; and (3) ProbFair when fairness is not enforced (i.e., $t = 0$), to examine the cost of state agnosticism.¹

5.1 Experimental Setup

Policies: In our experiments, we compare ProbFair against a subset of the following baseline and fairness-inducing, guaranteeing, and agnostic* policies:

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random¹⁰</td>
<td>Select $k$ arms uniformly at random at each $t$.</td>
</tr>
<tr>
<td>Round-Robin¹⁰</td>
<td>Select $k$ arms at each $t$ in fixed, sequential order.</td>
</tr>
<tr>
<td>TW-based heuristics¹⁰</td>
<td>Select top-$k$ arms based on Whittle index values. Available arms vary based on time-indexed fairness constraint satisfaction [29].</td>
</tr>
<tr>
<td>Risk-aware TW (RA-TW)¹⁰</td>
<td>Select top-$k$ arms based on Whittle index values. Incentivizes fairness via concave reward function [25].</td>
</tr>
<tr>
<td>Threshold Whittle (TW)*</td>
<td>Select top-$k$ arms based on Whittle index values [24, 42].</td>
</tr>
</tbody>
</table>

We specifically consider three Threshold Whittle-based heuristics: $H_{\text{FIRST}}, H_{\text{LAST}},$ and $H_{\text{RAND}}$. These heuristics partition the $k$ pulls available at each timestep into (un)constrained subsets, where a pull is constrained if it is executed to satisfy a time-indexed fairness constraint. During constrained pulls, only arms that have not yet been pulled the required number of times within a $v$-length interval are available; other arms are excluded from consideration, unless all arms have already satisfied their constraints. $H_{\text{FIRST}}, H_{\text{LAST}},$ and $H_{\text{RAND}}$ position constrained pulls at the beginning, end, or randomly within each interval of length $v$, respectively. Appendix F.1 provides pseudocode.

Objective: In all experiments, we assign equal value to the adherence of a given arm over time. Thus, we set our objective to reward occupancy in the “good” state: a simple local reward $r_t(s_t^i) := s_t^i \in \{0, 1\}$ and undiscounted cumulative reward function, $R(r(s)) := \sum_{i \in |\mathcal{N}|} \sum_{t \in |\mathcal{T}|} r(s_t^i)$. Evaluation metrics: We are interested in comparing policies along two dimensions: reward maximization and fairness (i.e., with respect to the distribution of algorithmic attention). To this end, we rely on two performance metrics: (a) intervention benefit and (b) earth mover’s distance.

Intervention benefit ($\text{IB}$) is the total expected reward of an algorithm, normalized between the reward obtained with no interventions (0% intervention benefit) and the asymptotically optimal but fairness-agnostic Threshold Whittle algorithm (100%) [24]. Formally,

$$\text{IB}_{\text{NoAct,TW}}(\text{ALG}) := \frac{E_{\text{ALG}}[R_{\text{TW}}(\cdot) \mid \cdot] - E_{\text{NoAct}}[R(\cdot)]}{E_{\text{TW}}[R(\cdot)] - E_{\text{NoAct}}[R(\cdot)]}$$ (7)

Per Lemma F.2 (App. F.2), the price of fairness (PoF) metric [4] is inversely proportional to intervention benefit. We thus report $\text{IB}$. Earth mover’s distance ($\text{EMD}$) is a metric that allows us to compute the minimum cost required to transform one probability distribution into another [36]. We use it to compare algorithms with respect to fairness—i.e., how evenly a set of pulls are allocated among arms. (Other metrics that may measure individual distributive fairness are discussed in Appendix F.2.)

For each algorithm, we consider a discrete distribution $F$ of observed pull counts, where each bucket, $j \in \{0 \ldots T\}$, corresponds to a feasible number of total pulls that an arm could receive, and $F[j] \in \{0 \ldots N\}$ corresponds to the number of arms whose observed pull count is equal to $j$. For example, $F[0]$ corresponds to the quantity of arms never pulled, and $F[T]$ corresponds to the quantity of arms pulled at every timestep. Each algorithm produces $kT$ total pulls, so the distributions have the same total mass.

We use Round-Robin as a fair reference algorithm since it distributes pulls evenly among arms. We then compute the minimum cost required to transform each algorithm’s distribution, $F_{\text{ALG}}$, into that of Round-Robin’s, $F_{\text{RR}}$. For our application this is equivalent to:

$$\text{EMD}_{\text{RR}}(\text{ALG}) := \sum_{j=0}^{T} \sum_{h=0}^{h} |F_{\text{ALG}}[j] - F_{\text{RR}}[j]|$$ (8)

Unless otherwise noted, we normalize EMD such that the maximum distance we encounter, that of TW, is one:

$$\text{EMD}_{\text{RR}}(\text{ALG}) - \text{EMD}_{\text{RR}}(\text{RR}) = \text{EMD}_{\text{ALG}}$$ (9)

Datasets: We evaluate performance on two datasets: (a) a realistic patient adherence behavior model and (b) a general set of randomly generated synthetic transition matrices.
CPAP Adherence. Obstructive sleep apnea (OSA) is a common condition that causes interrupted breathing during sleep [30]; when used throughout the entirety of sleep, continuous positive airway pressure therapy (CPAP) eliminates nearly 100% of obstructive apneas for the majority of treated patients [37]. However, poor adherence behavior in using CPAP reduces its beneficial outcomes. CPAP non-adherence affects an estimated 30-40% of patients [35].

We derive the CPAP dataset that we use in our experiments from the work of Kang et al. [16, 17], who model the dynamics and patterns of patient adherence behavior as a basis for designing effective and economical interventions. In particular, we adapt their Markov model of CPAP adherence behavior (a three-state system based on hours of nightly CPAP usage) to a two-state system using the clinical standard for adherence—at least four hours of CPAP machine usage per night [37]. Kang et al. [16] find, via expectation-maximization on CPAP usage patterns, that patients can be divided into two groups based on this clinical standard. Though patients in the first cluster occasionally miss a night, these patients utilize a CPAP machine for more than four hours every night without assistance, while patients in the second cluster do not. We refer to the latter cluster as the non-adherent cohort in our analysis.

Kang et al. [17] consider many intervention effects. We specifically consider an intervention effect, \( \alpha \) = 1.1, that broadly characterizes supportive interventions such as telemonitoring and phone support, which are associated with a moderate 0.70 hours (95% CI ± 0.35) increase in device usage per night [1]. We add random \( \sigma = 1 \) logistic noise to the transition matrices so that there is some variance in individual arm dynamics. To prevent overlap with the general cohort we consider for contrast, added noise can only hinder the probability of adherence in the non-adherent cohort.

Synthetic. In addition, we construct a synthetic dataset of randomly generated arms such that the structural constraints outlined in Section 2 are preserved. We conjecture that forward (reverse) threshold-optimal arms are a subset of concave (strictly convex) arms (see Appendix F.3).

### 5.2 ProbFair vs. Fairness-aware Alternatives

Here we compare ProbFair to policies which either induce or guarantee fairness. The former includes Risk-aware Whittle (RA-TW), which incentivizes fairness via concave reward \( r(b) \) [25]. We use the authors’ suggested reward function \( r(b) = -e^\lambda (1-b) \), \( \lambda = 20 \). This imposes a large negative utility on lower belief values, which motivates preemptive intervention. However, RA-TW does not guarantee time-indexed or probabilistic fairness for individual arms. The latter includes Round-Robin and the First, Last, and Random heuristics, which guarantee time-indexed fairness but do not provide any optimality guarantees.

In Table 1, we report average results for each policy, along with margins of error for 95% confidence intervals, computed over 100 simulation seeds for a synthetic cohort of 100 collapsing arms, with \( k = 20 \) and \( T = 180 \). To facilitate meaningful comparisons between ProbFair and the heuristics, we consider combinations of values for \( t \) and \( v \) that produce equivalent, integer-valued lower bounds on the number of pulls any arm can expect to receive—i.e., \( \min_t E[\sum_i I(a_i^t = 1)] = t \times T = \frac{t}{v} \).

<table>
<thead>
<tr>
<th>( \min_t E[# pulls] )</th>
<th>Policy</th>
<th>( E[IB] ) (%)</th>
<th>( E[EMD] ) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 ( t = 0.056 ) ( v = 18 )</td>
<td>PF ( t )</td>
<td>88.73 ± 0.26</td>
<td>81.78 ± 0.18</td>
</tr>
<tr>
<td></td>
<td>HFIRST ( v )</td>
<td>86.11 ± 0.26</td>
<td>71.53 ± 0.13</td>
</tr>
<tr>
<td></td>
<td>HLAST ( v )</td>
<td>87.37 ± 0.28</td>
<td>70.48 ± 0.12</td>
</tr>
<tr>
<td></td>
<td>HRAND ( v )</td>
<td>90.79 ± 0.22</td>
<td>74.12 ± 0.15</td>
</tr>
<tr>
<td>18 ( t = 0.1 ) ( v = 10 )</td>
<td>PF ( t )</td>
<td>80.80 ± 0.30</td>
<td>59.96 ± 0.19</td>
</tr>
<tr>
<td></td>
<td>HFIRST ( v )</td>
<td>76.62 ± 0.30</td>
<td>49.54 ± 0.09</td>
</tr>
<tr>
<td></td>
<td>HLAST ( v )</td>
<td>77.95 ± 0.30</td>
<td>49.26 ± 0.08</td>
</tr>
<tr>
<td></td>
<td>HRAND ( v )</td>
<td>81.53 ± 0.30</td>
<td>52.73 ± 0.10</td>
</tr>
<tr>
<td>30 ( t = 0.167 ) ( v = 6 )</td>
<td>PF ( t )</td>
<td>66.12 ± 0.35</td>
<td>23.61 ± 0.12</td>
</tr>
<tr>
<td></td>
<td>HFIRST ( v )</td>
<td>63.58 ± 0.31</td>
<td>18.98 ± 0.03</td>
</tr>
<tr>
<td></td>
<td>HLAST ( v )</td>
<td>64.63 ± 0.34</td>
<td>19.47 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>HRAND ( v )</td>
<td>65.21 ± 0.32</td>
<td>19.64 ± 0.04</td>
</tr>
<tr>
<td>comparison</td>
<td>RA-TW</td>
<td>85.12 ± 0.42</td>
<td>95.80 ± 0.42</td>
</tr>
<tr>
<td>baseline</td>
<td>RW</td>
<td>50.02 ± 0.35</td>
<td>10.08 ± 0.10</td>
</tr>
<tr>
<td></td>
<td>NoAct</td>
<td>0.00 ± 0.00</td>
<td>73.48 ± 0.13</td>
</tr>
<tr>
<td></td>
<td>RR</td>
<td>56.96 ± 0.33</td>
<td>0.00 ± 0.00</td>
</tr>
</tbody>
</table>

Table 1: Expected intervention benefit and normalized earth mover’s distance by policy and fairness bracket.

Key findings from this experiment include:

- **Fairer** hyperparameter values (i.e., \( t \uparrow, v \downarrow \)), correspond to decreases in \( E[IB] \) and \( E[EMD] \), reflecting improved individual fairness at the expense of total expected reward.
- **ProbFair** is competitive with respect to RA-TW, outperforming on both metrics when \( t = 0.056 \), and incurring a slight loss in \( E[IB] \) but improvement in \( E[EMD] \) for \( t = 0.1 \).
- For each \( (t, v) \) combination, ProbFair performs competitively with respect to the best-performing heuristic (which, like TW, are state-aware, see Section 5.4).

### 5.3 ProbFair on a Breadth of Cohorts

In this section we conduct sensitivity analysis with respect to cohort composition. For each dataset, we identify a transition matrix characteristic that can be modified during the generation process to produce a subset of arms that will exhibit less favorable transition dynamics than their peers. For the synthetic dataset, this characteristic is strict convexity. For the CPAP dataset, it is non-adherence, a mnemonic coined by Kang et al. [16] to characterize a cluster of study participants, and contrast this to a model fit on the general patient population.

For each dataset, we generate ten different cohorts, each of which is characterized by the percentage of unfavorable arms that it contains. We use a seed to control the generation process such that each cohort contains 100 collapsing arms in total. A sliding window of the unfavorable arms we can generate with this seed are included as we increase the cardinality of the unfavorable subset. For ease of interpretation, we present unnormalized results over 100 simulation seeds with \( k = 20 \) and \( T = 180 \) in Figure 2, and then proceed to summarize normalized performance.
Figure 2: Expected total reward (left) and unnormalized earth mover's distance (right) on a breadth of cohorts.

Key findings from this experiment include:
- Per Figure 2, for each dataset, expected total reward predictably declines for all policies as the percentage of unfavorable arms increases, while unnormalized EMD increases for TW and ProbFair.
  - Synthetic: As the proportion of strictly convex arms increases, ProbFair’s allocation of resources tends towards the bimodality of TW.
  - CPAP: As the proportion of non-adherent arms increases, the level of intervention required to improve trajectories rises, but the budget constraint is static.
- For each dataset, ProbFair’s normalized performance remains stable even as cohort composition is varied:
  - Synthetic: With respect to IB (EMD), ProbFair achieves an average (over all cohorts) of averages (over 100 simulations per cohort) of 80.69% ± 1.42% (58.98% ± 1.29%).
  - CPAP: The values for IB (EMD) are: 79.84% ± 0.68% (59.68% ± 1.08%).

5.4 ProbFair: Price of State Agnosticism

Here, we investigate the cost associated with ProbFair’s state agnosticism, relative to state-aware Threshold Whittle. To ensure a fair comparison, we set $\ell = 0$ and $u = 1$, effectively constructing a version of ProbFair in which probabilistic fairness is not enforced. (Recall that TW is fairness-agnostic; in the previous results, we do not expect ProbFair to obtain the same total reward as TW).

Although ProbFair incorporates each arm’s structural information (i.e., transition matrices), it produces a set of stationary probability distributions over actions from which all discrete actions are subsequently drawn. TW, in contrast, ingests each arm’s current state at each timestep, and is thus able to exploit realized sequences of state transitions.

While we thus expect ProbFair to incur some loss in intervention benefit, our results (computed over 100 simulation seeds, with $k = 20$, $N = 100$, and $T = 180$) indicate that this loss is acceptable rather than catastrophic. Relative to TW, ProbFair $_{\ell=0}$ obtains $97.41\% \pm 0.26$ of $\mathbb{E}[IB]$ and incurs an increase of only $4.56\% \pm 0.19$ with respect to $\mathbb{E}[EMD]$.

6 Conclusion and Future Work

In this paper, we introduce ProbFair, a novel, probabilistically fair algorithm for constrained resource allocation. Our theoretical results prove that this policy is reward-maximizing, subject to the guaranteed satisfaction of both budget and tunable probabilistic fairness constraints. Our empirical results demonstrate that ProbFair preserves utility while providing fairness guarantees. Promising future directions include: (1) extending ProbFair to address larger state and/or action spaces; and (2) relaxing the requirement for stationarity in the construction of $\pi_{PF}$.


A NOTATION

In Table 2, we present an overview of the notation used in the paper. \([N]\) denotes the set \(\{1, 2, \ldots, N\}\).

Table 2: Notation used in our model and notes on their interpretation.

<table>
<thead>
<tr>
<th>MDP Variables</th>
<th>Here, timestep (t \in [T] = {1, 2, \ldots, T}) (subscript) and arm index (i \in [N] = {1, 2, \ldots, N}) (superscript) are implied.</th>
</tr>
</thead>
<tbody>
<tr>
<td>State space</td>
<td>(s \in S = {0, 1}) [s = \begin{cases} 1 &amp; \text{arm is in the ‘good’ state.} \ 0 &amp; \text{else} \end{cases}]</td>
</tr>
<tr>
<td>Belief space</td>
<td>(b \in B = {0, 1}) [b_{t+1} = \begin{cases} s_{t+1} &amp; \text{if known} \ b_t P_{0,1}^0 + (1 - b_t) P_{1,1}^0 &amp; \text{else} \end{cases}]</td>
</tr>
<tr>
<td>Action space</td>
<td>(a \in A = {0, 1}) [a = \begin{cases} 1 &amp; \text{pull arm (i.e., provide intervention)} \ 0 &amp; \text{else, don’t pull} \end{cases}]</td>
</tr>
</tbody>
</table>

| MDP Functions | | |
|---------------|---------------------------------------------------------------------------------------------------------|
| Transition function | \(P: S \times A \times S \to [0, 1]\) \(s_t, a_t, s_{t+1} \mapsto \Pr(s_{t+1} | s_t, a_t)\) | The probability of an arm going from state \(s_t\) to \(s_{t+1}\), given action \(a_t\). Equivalent (matrix) notation: \(P_{s_t,a_t,s_{t+1}}^i\). |
| Reward function | \(r: S \text{ or } B \to \mathbb{R}\) \(r(b)\) is used in computing the Whittle index. |
| Policy function | \(\pi: S \to A\) A policy for actions. The set of optimal policies is \(\pi^* \in \Pi^*\). |

| RMAB Variables | | |
|---------------|---------------------------------------------------------------------------------------------------------|
| Timestep      | \(\{t \in \mathbb{N} \mid t \leq T\}\) This timestep is implicit in the MDP. |
| Arm index     | \(i \in \{1, 2, \ldots, N\}\) Each arm can represent a patient. \(k\) arms can be pulled at any timestep \(t\). |

<table>
<thead>
<tr>
<th>Objective Functions</th>
<th>The objective is to find a policy (\pi^* = \max_{\pi} \mathbb{E}_{\pi}[R(\cdot)]).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discounted reward function</td>
<td>(R^\beta_{\theta}: S^N \to \mathbb{R}) (\sum_{i=0}^N \sum_{t=1}^T \beta^{t-1} r(s^i_t)) (\beta \in [0, 1]) is some discount parameter.</td>
</tr>
</tbody>
</table>

| Fairness-motivated Constraint Functions | | |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------|
| Integer periodicity                  | \(\bigwedge_{j=0}^{v} \sum_{i=t+j+1}^{\infty} a_i^j \geq 1\) A form of time-indexed fairness. \(\sum_{i=t+j+1}^{\infty} a_i^j \geq 1\) Guarantees arm \(i\) is pulled at least once within each period of \(v\) timesteps. |
| Minimum selection fraction            | \(\bigwedge_{i \in [N]} \frac{1}{T} \sum_{t=1}^{T} a_i^j \geq \psi\) A form of time-indexed fairness. \(\sum_{t=1}^{T} a_i^j \geq \psi\) Arm \(i\) should be pulled at least some minimum fraction \(\psi \in (0, 1)\) of timesteps. |
| Probabilistic                        | \(\bigwedge_{i \in [N]} \bigwedge_{t \in [T]} \Pr(a_i^j = 1 | i, t) \in [\ell, u]\) Pull each arm with probability \(p_i \in [\ell, u]\), where \(\ell \in (0, \frac{k}{N})\) and \(u \in \left(\frac{k}{N}, 1\right]\). |
Here, we present numerical results confirming Prins et al. [29]'s findings that Threshold Whittle (TW) tends to allocate pulls according to a bimodal distribution: a small subset of arms are pulled frequently, while others are largely ignored.

**Experimental Setup:** For each iteration, we generate $N = 2$ forward threshold-optimal arms and run TW for a $T = 365$ horizon simulation, where the budget constraint $k = 1$. We run 1,000 such iterations.

**Results:** In 515 out of 1,000 (51.5%) simulations, the arms’ Whittle indices never overlap, meaning that for any combination of initial states, state transitions, and pulls, TW would pull one arm for all timesteps $t \in T$ and completely ignore the second arm. We visualize one such case in Figure 3. Recall that TW precomputes the infimum subsidy $m$ per arm and belief combination. Since belief is a function of last known state $s \in \{0, 1\}$ and time-since-seen $u \in [T]$ (using the notation of Mate et al. [24]), we plot the infimum subsidy of each arm-state combination with time-since-seen, $u$, on the $x$-axis. There exists a horizontal line that divides the two arms, so arm $i = 1$ will be pulled for every timestep and arm $i = 2$ will never be pulled.

![Figure 3](image)

Figure 3: The Whittle index values for Arm 1 and 2 can be separated by a horizontal line, meaning that (WLOG) Arm 1 will always be chosen over Arm 2 because its index value dominates.

In order to modify the Whittle index to guarantee time-indexed fairness constraint satisfaction, one would need to ensure that no such horizontal line exists. Additionally, if we consider a specific form of time-indexed fairness known as an integer periodicity constraint, which allows a decision-maker to guarantee that arm $i$ is pulled at least once within each period of $\nu$ days, the lines associated with the arms in Figure 3 must cross before $\nu$ timesteps elapse to guarantee fairness constraint satisfaction.

Another perspective we can take is to ask: what’s the smallest interval $\nu_i$ for each arm $i$ we could have specified such that Threshold Whittle would have satisfied the integer periodicity constraint? Note that this is retrospective, as there is no way to enforce this constraint at planning time. We visualize the minimum such $\nu_i$ in Figure 4. On the far right, we see the 515 cases where (WLOG) the second arm is never pulled—that is, the minimum $\nu_i$ such that Threshold Whittle satisfies the hard integer periodicity constraint must be larger than the horizon, $T = 365$. There is one case where arm $i = 2$ is pulled exactly once. In a majority of the remaining simulations, TW pulls each arm with approximately equal frequency.

![Figure 4](image)

Figure 4: The smallest interval $\nu_i$ such that TW satisfies an integer periodicity definition of time-indexed fairness, given $N = 2$ random arms. In over 50% of iterations, no such fairness constraint satisfaction is possible (i.e., $\exists i \text{ s.t. } \nu_i > T$).
C MOTIVATING A FOCUS ON DISTRIBUTIVE FAIRNESS

In Section 3.2 and Appendix B, we discuss and empirically demonstrate how Whittle index-based policies produce bimodal distributive outcomes in which a small subset of arms receive all available interventions, and the rest of the arms are ignored. Here, we note that there are many application domains where such skewed resource allocation may be perceived as unfair or undesirable by beneficiaries and decision-makers, thus motivating efforts to incentivize or guarantee distributive fairness.

In the healthcare examples we present in the main paper (i.e., where interventions support medication or CPAP adherence), resource constraints and variation in transition dynamics interact. A practical consequence is that a majority of patients will never receive the beneficial intervention(s). This, in turn, means that their clinical outcomes will be strictly worse in expectation than they would be under a policy that guaranteed a non-zero probability of receiving the intervention at each timestep.

These considerations are not restricted to healthcare contexts. In poaching prevention, the planner must be cognizant of the effect of choosing against patrolling a particular area (i.e., pulling an arm) will have on poacher strategies [31]. In wireless scheduling, multiple processes compete to transmit packets over a shared wireless channel. When scheduling the transmissions, the agent must not only maximize reward but also ensure the performance of critical applications codified in Quality of Service (QoS) guarantees [13, 20, 22].

D INTRACTABILITY OF ALTERNATIVE APPROACHES

In this section, we motivate the algorithmic design choices we have made when constructing ProbFair by discussing the feasibility of possible alternatives, including: (1) direct modification of the Whittle index to guarantee time-indexed fairness constraint satisfaction, and (2) a math programming-based approach.

D.1 Why not modify the Whittle index to guarantee time-indexed fairness constraint satisfaction?

In Section 3.3, we demonstrate that it is not possible to guarantee time-indexed fairness when arms are decoupled. If arms cannot be decoupled, the tractability of a Whittle index-based approach breaks down. Here, we discuss this topic in greater detail, and provide specific examples of possible Whittle index modifications. We also take this opportunity to emphasize that our focus is on guaranteeing fairness rather than incentivizing it. Mate et al. [25] provide an example of the latter, which we discuss and include as a comparison algorithm in Section 5.2.

To begin, recall that the efficiency of Whittle index-based policies stems from our ability to decouple arms when we are only concerned with maximizing total expected reward [41, 42]. However, guaranteeing time-indexed fairness (as defined in Section 2) in the planning setting requires time-stamped record keeping. It is no longer sufficient to compute each arm’s infimum subsidy in isolation and order the resulting set of values. Instead, for an optimal index policy to be efficiently computable, it must be possible to modify the value function (Equation 3.1) so as to ensure that the infimum subsidy each arm would require in the absence of fairness constraints is minimally perturbed via augmentation or “donation”, so as to maximize total expected reward while ensuring its own fairness constraint satisfaction or the constraint satisfaction of other arms, respectively, without requiring input from other arms.

Plausible modifications include altering the conditions under which an arm receives the subsidy associated with passivity, $m$, or introducing a modified reward function, $r'(b)$ that is capable of accounting for an arm’s fairness constraint satisfaction status in addition to its state at time $t$. For example, we might use an indicator function to “turn off” the subsidy until arm $i$ has been pulled at least once within the interval in question, or increase reward as an arm’s time-since-pulled value approaches the interval cut-off, so as to incentivize a constraint-satisfying pull. When these modifications are viewed from the perspective of a single arm, they appear to have the desired effect: if no subsidy is received, it will be optimal to pull for all belief states; similarly, for a fixed $m$, as reward increases it will be optimal to pull for an increasingly large subset of the belief state space.

Recall, however, that structural constraints ensure that when an arm is considered in isolation, the optimal action will always be to pull. Whether or not arm $i$ is actually pulled at time $t$ depends on how the infimum subsidy, $m$, it requires to accept passivity at time $t$ compares to the infimum subsidies required by other arms. Thus, any modification intended to guarantee time-indexed fairness constraint satisfaction must be able to alter the ordering among arms, such that any arm $i$ which would otherwise have a subsidy with rank $> k$ when sorted in descending order will now be in the top-$k$ arms. Even if we were able to construct such a modification for a single arm without requiring system state, if every arm had this same capability, then a new challenge would arise: we would be unable to distinguish among arms, and arbitrary tie-breaking could again jeopardize fairness constraint satisfaction.

If it is not possible to decouple arms, then we must consider them in tandem. Papadimitriou and Tsitsiklis [28] prove that the RMAB problem is FSPACE-complete even when transition rules are action-dependent but deterministic, via reduction from the halting problem.

D.2 Why not use a math-programming approach?

Our constrained maximization problem can be readily formulated as an integer program (IP) with a totally unimodular (TU) constraint matrix. However, this approach is intractable because the objective function coefficients of this IP cannot be efficiently enumerated. To support this intractability claim, we begin by presenting an integer program (IP) that maximizes total expected reward under both budget and time-indexed fairness constraints, for problem instances with feasible hyperparameters. We then prove that any problem instance with feasible hyperparameters yields a totally unimodular (TU) constraint matrix, which ensures that the linear program (LP) relaxation of our IP will yield an integral solution. We proceed to demonstrate that tractability issues arise because we incur an exponential dependency on the
time horizon, $T$, when we construct the IP’s objective function coefficients. We conclude by comparing PRBFair to the IP for small values of $N$ and $T$.

D.2.1 Integer Program Formulation. To leverage a math programming approach for our constrained reward maximization task, we seek to construct an integer program (IP) whose solution is the policy $\mathbf{X} \in \{0, 1\}^{|\mathcal{A}| \times |\mathcal{T}|}$. We require this policy to be reward-maximizing, subject to the guaranteed satisfaction of both budget and time-indexed fairness constraints. To begin, let each decision variable $x_{i,a,t} \in \{0, 1\}$ represent whether or not we take action $a \in \mathcal{A} = \{0, 1\}$ for arm $i \in [N]$ at time $t \in [T]$. Then, let each objective function coefficient $c_{i,a,t}$ represent the expected reward associated with an arm-action-timestep combination.

To formalize the objective function, recall that the agent seeks to maximize total expected reward, $\mathbb{E}_x[R(\cdot)]$. For clarity of exposition, we specifically consider the linear global reward function $R(r(s)) = \sum_{i=1}^{N} \sum_{t=0}^{T} s_i^t$. Note that this implies the discount rate, $\beta = 1$; however, the approach outlined here can be extended in a straightforward manner for $\beta \in (0, 1)$. In order to compute the expected reward associated with taking action $a$ for arm $i$ at time $t$, we must consider: (1) what state is the arm currently in (i.e., what is the realized value of $s_i^t$? (2) when the arm transitions from $s_i^t$ to $s_{i+1}^t$ by virtue of taking action $a$, what reward, $r(\cdot)$, should we expect to earn?

Because we define $r(s) = s$, (2) can be reframed as: what is the probability $p(s_{i+1}^t = 1|s_i^t, a_j^t)$ that action $a$ causes a transition from $s_i^t$ to the adherent state? Because each arm’s state at time $t$ is stochastic, depending not only on the sequence of actions taken in previous timesteps, but the associated set of stochastic transitions informed by the arm’s underlying MDP, each coefficient of our objective function must be computed as the expectation over the possible values of $s_i^t$.

\[
\bar{c} = \mathbb{E}_s[p(s_{i+1}^t = 1|x_{i,a,t}, s_i^t)] \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] \\
= \frac{1}{2^t} \sum_{s \in S} p(s_t = s) \sum_{s' \in S} p(s_{t+1}^t = s'|x_{i,a,t}, s_t = s)r(s') \quad \forall i, a, t \in [N] \times \mathcal{A} \times [T] 
\]

Within the context of this IP, the time-indexed fairness constraint we introduce in Section 2 can be more specifically defined as either an integer periodicity or minimum selection fraction constraint. We formalize each of these below:

The integer periodicity constraint allows a decision-maker to guarantee that arm $i$ is pulled at least once within each period of $v$ days. We define this constraint as a function $g$, over the vector of actions, $\vec{a}$ associated with arm $i$, and user-defined interval length $v \in [1, T]$:

\[
g(\vec{a}) = \sum_{t=j+v+1}^{(j+1)v} a_j^t \geq 1 \quad \forall j \in \left\{0, 1, 2, \ldots \left\lfloor \frac{T}{v} \right\rfloor \right\}; \forall i \in \{1, 2, \ldots N\}
\]

The minimum selection fraction constraint introduced by Li et al. [20] forces the agent to pull arm $i$ at least a minimum fraction, $\psi \in (0, 1)$, of the total number of steps, but is agnostic to how these pulls are distributed over time. We define this constraint, $g'$, as a function over the vector of actions, $\vec{a}$ associated with arm $i$ and user-defined $\psi$:

\[
g'(\vec{a}) = \frac{1}{T} \sum_{t=1}^{T} a_i^t \geq \psi \forall i \in \{1, 2, \ldots N\}
\]
The resulting integer program is given by:

\[
\begin{align*}
\text{max} & \quad c^T x \\
\text{s.t.} & \quad \sum_{a=1}^{|A|} x_{i,a,t} = 1 & \forall i \in [N], t \in [T] & \quad (a) \text{ Select exactly one action per arm } i \text{ at each } t \\
& \quad \sum_{t \in I} x_{i,1,t} \geq 1 & \forall j \in \left\{ \frac{T - \nu}{v} \right\} & \quad (b.i) \text{ Pull each arm } i \text{ at least once during each interval of length } \nu \\
& \quad \frac{1}{T} \sum_{t=1}^{T} x_{i,1,t} \geq \psi & \psi \in (0,1), \forall i \in N & \quad (b.ii) \text{ Pull each arm } i \text{ at least a minimum fraction } \psi \text{ of } T \text{ rounds} \\
& \quad \sum_{i=1}^{N} x_{i,1,t} = k & \forall t \in [T] & \quad (c) \text{ Pull exactly } k \text{ arms at each } t \\
& \quad x_{i,a,t} \in \{0,1\} & \forall i \in [N], a \in A, t \in [T] & \quad (d) \text{ Each arm-action-timestep choice is a binary decision variable}
\end{align*}
\]

D.2.2 LP Relaxation and Integrality of Solution. We now prove that the IP we have formulated in Section D.2.1 has an attractive property: namely, any feasible problem instance will produce a totally unimodular constraint matrix. Our proof leverages a theorem introduced by Ghouila-Houri and restated below for convenience, which can be used to determine whether a matrix, \( A \in \mathbb{R}^{m \times n} \) is totally unimodular:

**Lemma D.1.** (Ghouila-Houri [1]) A matrix \( A \in \mathbb{Z}^{m \times n} \) is totally unimodular (TU) if and only if for every subset of the rows \( R \subseteq [m] \), there is a partition \( R = R_1 \cup R_2 \) such that for every \( j \in [n] \),

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} \in \{-1,0,1\}
\]

**Theorem D.2.** Within the context of the integer program outlined in Appendix D.2.1, any feasible problem instance will produce a constraint matrix that is totally unimodular (TU).

**Proof.** To begin, we establish the dimensions of any such constraint matrix \( A \) and note the maximum possible column-wise sum that each of its component submatrices may contribute. Note that the minimum selection fraction constraint \((b.ii)\) in Equation 16, which requires the agent to pull each arm \( i \) at least a minimum fraction, \( \psi \in (0,1) \), of \( T \) rounds, can be thought of as a special case of the integer periodicity constraint, \((b.i)\) in Equation 16, where \( \nu = T \) and each arm must be pulled at least \( T\psi \) times. As such, we assume that at most one of the time-indexed fairness constraints can be selected, and focus on the more general of the two, which is the integer periodicity constraint. For notational convenience, we refer to constraints by their alphabetic identifiers. Let \( b \) represent the integer periodicity constraint, and define a function \( \phi : r \in R \subseteq A \mapsto e \in \{a,b,c\} \) that maps each row to its corresponding constraint type.

First, recall that each \( x_{i,a,t} \) represents a single binary decision variable, and corresponds to a column in \( A \). There are \( N \times |A| \times T \) such columns. Next, note that constraint \((a)\) enforces the requirement that we select exactly one action per arm per timestep. Formally, \( \forall i, t \in N \times T, \exists a \in A \text{ s.t. } x_{i,a,t} = 1 \). Correspondingly, \( \forall a' \in A \setminus a, x_{i,a',t} = 0 \). The column vectors of the associated sub-matrix, \( A_a \in \mathbb{Z}^{NT \times N|A|T} \), are indexed by disjoint \((i,a,t)\) \( N \times |A| \times T \); thus, each column vector contains a single non-zero entry and for \( A_a \subseteq A_a \), taking the column-wise sum will yield a vector \( \vec{v} \in \mathbb{Z}^{N|A|T} \) with every entry equal to 1.

In a similar vein, equity constraint \((b)\) enforces the requirement that we must pull each arm, \( i \) at least once during each interval \( I_j \) of length \( \nu_i \). Within the associated sub-matrix, \( A_{i,j} \in \mathbb{Z}^{N|T/\nu_i|\times N|A|T} \), each column that corresponds to a passive action (e.g., \( x_{i,a=0,t} \)) will have only zero-valued entries, since passive action decision variables are not impacted by constraint \((b)\). Conversely, each column that corresponds to an active action (e.g., \( x_{i,a=1,t} \)) will have a single non-zero entry. Each active action column corresponding to a specific arm-timestep can be mapped to exactly one interval. Thus, for \( R_b \subseteq A_b \), taking the column-wise sum will yield a vector \( \vec{v} \in \mathbb{Z}^{N|A|T} \) with every entry taking a value \( \in \{0,1\} \).

The budget constraint \((c)\) enforces the requirement that we must pull exactly \( k \) of the \( N \) arms at each timestep. Much like equity constraint \((b)\), only columns corresponding to active actions are impacted. Thus, within the associated sub-matrix, \( A_c \in \mathbb{Z}^{T \times N|A|T} \), each column that corresponds to a passive action (e.g., \( x_{i,a=0,t} \)) will have only zero-valued entries, while each column that corresponds to an active action (e.g., \( x_{i,a=1,t} \)) can be mapped to a single timestep, and will have a single non-zero entry. Thus, for \( R_c \subseteq A_c \), taking the column-wise sum also yield a vector \( \vec{v} \in \mathbb{Z}^{N|A|T} \) with every entry taking a value \( \in \{0,1\} \).

The complete constraint matrix \( A \) thus contains \( NT + N|T/\nu_i| + T \) rows. Three possible cases arise when we consider every subset of these rows: (1) \( R \subseteq A = \emptyset \); (2) \( R \subseteq A \); \( R \cap A = \emptyset \); (3) \( R \subseteq A \).
Case 1. \( R \subseteq A = \emptyset \). To satisfy Lemma D.1, partition \( R \) such that \( R = R_1 \cup R_2 = \emptyset \cup \emptyset \). Then, for every \( j \in [n] \),

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 0 - 0; 0 \in \{-1, 0, 1\} \quad \square
\]

Case 2. \( R \subseteq A; R \cap A \neq \emptyset \). If we consider \( \bigcup \{ r \in R \} \phi (r) \), there are \( \left( \begin{array}{c} 3 \\ 3 \end{array} \right) \) possible sets of observed constraint types: \( \{ a \} \lor \{ b \} \lor \{ c \} \lor \{ a, b \} \lor \{ a, c \} \lor \{ b, c \} \lor \{ a, b, c \} \).

(1) If \( | \bigcup \{ r \in R \} \phi (r) | = 1 \), then any partition of \( R \) will satisfy Lemma D.1. Without loss of generality, let each row \( r \in R \) belong to \( R_1 \) and \( R_2 = \emptyset \).

i. If \( \bigcup \{ r \in R \} \phi (r) = \{ a \} \), taking the column-wise sum of \( R_1 \) will yield a vector \( \vec{d} \in \mathbb{Z}^{|A|T} \) with every entry \( \in \{ 1 \} \) if \( R \subseteq A_a \), and \( \in \{ 0, 1 \} \) otherwise. Thus, \( \forall j \in [N|A|T] \),

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 0 - 0 \lor 1 - 0; \{ 0, 1 \} \subseteq \{-1, 0, 1\} \quad \square
\]

ii. If \( \bigcup \{ r \in R \} \phi (r) = \{ b \} \lor \{ c \} \), taking the column-wise sum of \( R_1 \) will yield a vector \( \vec{d} \in \mathbb{Z}^{|A|T} \) with every entry corresponding to a passive action column \( \in \{ 0 \} \) and every entry corresponding to an active action column \( \in \{ 1 \} \) if \( R \subseteq A_{b, c} \), and \( \in \{ 0, 1 \} \) otherwise. Thus, \( \forall j \in [N|A|T] \),

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 0 - 0 \lor 1 - 0; \{ 0, 1 \} \subseteq \{-1, 0, 1\} \quad \square
\]

(2) If \( | \bigcup \{ r \in R \} \phi (r) | = 2 \), without loss of generality, partition as follows: sort the elements \( \in \bigcup \{ r \in R \} \phi (r) \) lexicographically, and let \( R_1 = \{ r | \phi (r) = \min \bigcup \{ r \in R \} \phi (r) \} \) and \( R_2 = R \setminus R_1 \). Per Case 2.1 (i) and (ii), taking the column-wise sums of \( R_1 \) and \( R_2 \) will yield two vectors, \( \vec{d}_1, \vec{d}_2 \in \mathbb{Z}^{|A|T} \), each of which will contain only entries \( \in \{ 0, 1 \} \). Thus, \( \forall j \in [N|A|T] \),

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 0 - 0 \lor 0 - 1 \lor 1 - 0 \lor 1 - 1; \{-1, 0, 1\} \subseteq \{-1, 0, 1\} \quad \square
\]

(3) If \( | \bigcup \{ r \in R \} \phi (r) | = 3 \), partition according to Algorithm 3:
Algorithm 3 Partition for Case 2.3

1: procedure Partition(N, A, T, R)
2:  \(C_R_1; C_R_2 \leftarrow \{0\}^{N \times |A|T}\)  // Initialize two sets of counters
3:  \(R_1; R_2 \leftarrow \emptyset\)
4:  for element \(e \in \{a, b, c\}\) do
5:      \(R_e \leftarrow \{r | \varphi(r) = e\}\)
6:  for \(r \in R_a\) do
7:      \(R_1 \leftarrow R_1 \cup r\)
8:      for \(i \in 0; N \times |A|T\) do
9:          if \(r[i] > 0\) then
10:              \(C_R_1[i] \leftarrow C_R_1[i] + 1\)
11:  for \(r \in R_b\) do
12:      \(\text{flag} \leftarrow \text{false}\)
13:      for \(i \in 0; N \times |A|T\) do
14:          if \(r[i] > 0\) and \(C_R_1[i] > 0\) then
15:              \(\text{flag} \leftarrow \text{true}\)
16:      if \(\text{flag}\) then
17:          \(R_2 \leftarrow R_2 \cup r\)
18:          for \(i \in 0; N \times |A|T\) do
19:              if \(r[i] > 0\) then
20:                  \(C_R_2[i] \leftarrow C_R_2[i] + 1\)
21:      else
22:          \(R_1 \leftarrow R_1 \cup r\)
23:          for \(i \in 0; N \times |A|T\) do
24:              if \(r[i] > 0\) then
25:                  \(C_R_1[i] \leftarrow C_R_1[i] + 1\)
26:  for \(r \in R_c\) do
27:      \(R_2 \leftarrow R_2 \cup r\)
28:      for \(i \in 0; N \times |A|T\) do
29:          if \(r[i] > 0\) then
30:              \(C_R_2[i] \leftarrow C_R_2[i] + 1\)
31: return \(R_1; R_2\)

Taking the column-wise sums of the resulting \(R_1\) and \(R_2\) will yield two vectors \(\tilde{v}_1, \tilde{v}_2 \in \mathbb{Z}^{N \times |A|T}\), which can contain entries \(\in \{0, 1\}\) and \(\{0, 1, 2\}\), respectively. Note that since \(\tilde{v}_2\) is constructed by taking only rows with constraint types \(\in \{b, c\}\), only entries corresponding to active action columns can take values \(\geq 1\). Moreover, \(\forall j \in \{N \times |A|T\}, \sum_{i \in R_2} A_{ij} = 2 \to \sum_{i \in R_1} A_{ij} = 1\). Thus, \(\forall j \in \{N \times |A|T\},\)

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 0 - 0 \vee 0 - 1 \vee 1 - 0 \vee 1 - 1 \vee 1 - 2; \{0, 1, 2\} \subseteq \{0, 1, 2\} \tag{25}
\]

\(\square\)

Case 3. \(R \subseteq A\). Since \(|\cup_{r \in R} \varphi(r)| = 3\), proceed as outlined in Case 2.3. Only a slight modification is required: since \(R\) is now equal to \(A\), taking the column-wise sums of the resulting \(R_1\) and \(R_2\) will yield two vectors \(\tilde{v}_1, \tilde{v}_2 \in \mathbb{Z}^{N \times |A|T}\), which can contain entries \(\in \{1\}\) and \(\{0, 2\}\), respectively. Thus, \(\forall j \in \{N \times |A|T\},\)

\[
\sum_{i \in R_1} A_{ij} - \sum_{i \in R_2} A_{ij} = 1 - 0 \vee 0 - 1 \vee 1 - 1 \vee 1 - 2; \{0, 0, 1\} \subseteq \{0, 1, 2\} \tag{26}
\]

\(\square\)

D.2.3 Enumeration of Objective Function Coefficients. The key challenge we encounter when we seek to enumerate the IP outlined in Section D.2.1 is that exact computation of the objective function coefficients, \(c \in \mathbb{R}^{N \times |A|T}\) is intractable. Each arm contributes \(|A| \times T\) coefficients, and while calculation is trivially parallelizable over arms, we must consider a probability tree like the one in Figure 5 for each arm.

The number of decision variables required to enumerate each arm’s game tree is of order \(O(|A||S|^T)\) and there are \(N\) such trees, so even a linear program (LP) relaxation is not tractable for larger values of \(T\) and \(N\), which motivates us to propose ProbFair (Section 4) as an efficient alternative.

Example 1. Suppose we wish to find the coefficient \(c’\) corresponding to \(x_{t,a=0,t=2}\). From Equation 13, we have \(c’ = \frac{1}{2} \sum_{s \in S} p(s_2 = s)p(s_3 = 1|x_{t,a=0,t=2}, s_2 = s)\). Equivalently, we sum the weight of each path from the root node to the highlighted end nodes in Figure 5 and normalize
Figure 5: Illustration of the probability tree for finding the coefficient corresponding to $x_{l,a=0,t=2}$. 
by \( \frac{1}{2^2} \):

\[
c' = \frac{1}{4} \left( p_{0,0}^0 p_{0,1}^0 + p_{0,0}^1 p_{0,1}^0 + p_{0,0}^0 p_{1,1}^0 \right)
+ \frac{1}{4} \left( p_{1,0}^0 p_{0,1}^0 + p_{1,0}^1 p_{0,1}^0 + p_{1,0}^0 p_{1,1}^0 \right)
+ \frac{1}{4} \left( p_{0,0}^0 p_{0,0}^1 p_{1,0}^0 + p_{0,0}^1 p_{0,0}^1 p_{1,1}^0 \right)
+ \frac{1}{4} \left( p_{1,0}^0 p_{0,0}^1 p_{1,0}^0 + p_{1,0}^1 p_{0,0}^1 p_{1,1}^0 \right)
+ \ldots 
\]

For each of the \(|A||S| \) paths to a blue node in Figure 5.

D.2.4 Comparison of ProbFair with the True Optimal Policy. In Section 5, we normalize intervention benefit with Threshold Whittle, which is asymptotically optimal for forward threshold-optimal transition matrices under a budget constraint \( k \) [24]. However, with the integer program (IP) we formulate in Section D.2.1, we can find the optimal policy for any set of transition matrices under budget and fairness constraints as long as \( N \) and \( T \) are small.

We generate \( N = 2 \) random arms such that the structural constraints outlined in Section 2 are satisfied. We set \( k = 1 \) and \( T = 6 \). Though the variance in reward is large due to the small \( T \), Figure 6 shows that ProbFair obtains 100% of the intervention benefit when no fairness constraints are applied. Similarly, ProbFair with \( \ell = 0.33 \) obtains the same adherence behavior as the IP policy with under hard fairness constraint \( \nu = 3 \) or minimum selection fraction constraint \( \lambda = 0.33 \). (within 95% confidence interval shown). All results shown are bootstrapped over 500 iterations.

Minimum Selection Fraction Constraints. As we discuss in Appendix B, the optimal policy is often to pull the same \( k \) arms at every timestep and ignore all other arms. Under minimum selection fraction constraints (Equation 15), each arm must be pulled at least a minimum fraction \( \lambda \) of \( T \) rounds, with no conditions on when these pulls should take place. We confirm with the optimal IP implementation our intuition that these additional pulls are allocated at the beginning or end of the simulation. That is, the optimal policy under minimum selection fraction constraints is to take advantage of the finite time horizon, which is not suitable for the applications we consider.
E PROBFAIR: A PROBABLISTICALLY FAIR POLICY

Our main contribution is the novel probabilistic policy ProBFair (see Section 4.1). Here, in Section E.1, we present complete proofs to Theorems 4.1-4.4. Then, in Section E.2, we provide additional details about how we sample from our probabilistic policy to select discrete actions at each timestep.

E.1 Proofs

In this section, we provide proofs for the theorems introduced in Section 4.1. When relevant, we begin by restating the theorem for convenience.

**Theorem 4.1.** For every arm \( i \in [N], f_i(p_i) \) is either concave or strictly convex in all of \( p_i \in [0,1] \).

*Proof.* For notational convenience, let:

\[

c_1 = p_0^0; \\
c_2 = p_1^0 - p_0^0; \\
c_3 = 1 - p_0^1 + p_0^1; \\
c_4 = p_1^0 - p_1^1 + p_0^1.
\]

Then, \( f_i(p_i) = \frac{c_1 + c_2 p_i}{c_3 + c_4 p_i} \). We observe that \( \forall i \in [N], f_i(p_i) \) is a valid probability since the term \( 1 - (1 - p_i)p_0^0 - p_i p_1^1 \) in the denominator is at least \( 1 - (1 - p_i) - p_i = 0 \) for all \( p_i \in [0,1] \). Then, there are three cases which describe the possible shapes of \( f_i(p_i) \):

**Case 1.** \( c_4 = 0 \). Here, \( f_i(p_i) \) is linear and hence, concave.

**Case 2.** \( c_4 \neq 0; c_2 = 0 \). Here, \( f_i''(p_i) = \frac{2c_i c_j^2}{(c_i + c_j p_i)^3} \geq 0 \), so \( f_i(p_i) \) is linear (hence concave) if \( c_1 = 0 \) or strictly convex (if \( c_1 > 0 \)) in the domain \( p_i \in [0,1] \).

**Case 3.** \( c_4 \neq 0; c_2 \neq 0 \). Here,

\[
f_i(p_i) = \frac{c_1}{c_4} \left( \frac{c_1 c_4 + c_3 p_i}{c_3 + c_4 p_i} \right) = c_2 + \left( \frac{c_1 - c_3 c_4}{c_3 + c_4 p_i} \right).
\]

Thus,

\[
f_i''(p_i) = \frac{2c_i c_j^2}{(c_i + c_j p_i)^3} \tag{20}
\]

The sign of \( f_i''(p_i) \) is the same as the sign of \( d = c_1 - \frac{c_3 c_4}{c_i} \). It follows that \( f_i(p_i) \) is strictly convex if \( c_1 > \frac{c_3 c_4}{c_i} \), and concave otherwise for \( p_i \in [0,1] \).

**Theorem 4.2.** For each arm \( i \in [N], the structural constraints introduced in Section 2 ensure that \( f_i(p_i) \) is monotonically non-decreasing in \( p_i \) over the interval \([0,1] \).

*Proof.* For notational convenience, let:

\[

c_1 = p_0^0; \\
c_2 = p_1^0 - p_0^0; \\
c_3 = 1 - p_0^1 + p_0^1; \\
c_4 = p_1^0 - p_1^1 + p_0^1.
\]

Then \( f_i(p_i) = \frac{c_1 + c_2 p_i}{c_3 + c_4 p_i} \) and \( f_i''(p_i) = \frac{c_2 c_3 - c_1 c_4}{(c_i + c_j p_i)^3} \).

Observe \( c_2 c_3 - c_1 c_4 \geq 0 \) implies \( f_i''(p_i) \geq 0 \).

\[
(c_2 c_3 - c_1 c_4) \geq 0 \Rightarrow c_2 c_3 \geq c_1 c_4
\]

\[
(p_0^0 - p_0^1)(1 - p_1^0 + p_0^1) \geq p_0^0 (p_0^0 - p_0^1 - p_0^0 + p_0^1)
\]

Per our structural constraints, \( p_0^0 < p_1^0 \) and \( p_1^1 > p_0^0 \).
Lemma 4.3. P2 has an optimal solution in which \( p_i \in (t, u) \) for at most one \( i \in Y \).

Proof. Note by compactness that P2 has not just a supremum, but an actual maximum solution. Suppose for contradiction that there is some optimal solution \( \bar{p} \) with distinct indices \( i, j \in Y \) such that \( p_i, p_j \in (t, u) \). Now, suppose we perturb by an infinitesimal \( \epsilon \) (of arbitrary sign but tiny positive absolute value) such that \( p_i := p_i + \epsilon \) and \( p_j := p_j - \epsilon \). This satisfies all our constraints for small enough \( |\epsilon| \). The change in the objective \( \sum_{i \in Y} f_i(p_i) \) is now \( \epsilon \cdot (f_i'(p_i) - f_j'(p_j)) + O(\epsilon^2) \); hence, if \( f_i'(p_i) - f_j'(p_j) \) is nonzero, then we can take a tiny \( \epsilon \) of the appropriate sign to increase the objective, a contradiction. Therefore, \( f_i'(p_i) - f_j'(p_j) = 0 \), and so, we now focus on lower-order terms: the change in the objective \( \sum_{i \in Y} f_i(p_i) \) is now \( (\epsilon^2/2) \cdot (f_i''(p_i) + f_j''(p_j)) + O(\epsilon^3) \). However, since \( f_i \) and \( f_j \) are strictly convex, we have that \( f_i''(p_i) + f_j''(p_j) > 0 \), and hence the objective increases regardless of the sign of (the tiny) \( \epsilon \), again a contradiction. Thus we have our structural result.

\[ \square \]

Theorem 4.4. Alg. 2 yields the mapping from arms in \( Y \) to subsets in \( \{Y_1, Y_2, Y_3\} \) which maximizes \( \sum_{i \in Y} f_i(p_i) \) s.t. \( \sum_{i \in Y} p_i = k - z \). (See Appendix E for the complete proof).

We begin by introducing Lemma E.1, which we use in our proof of Theorem 4.4:

Lemma E.1. For a given \( y, p' \in (t, u) \).

Proof. To begin, observe that \( z - k = y \ell + (|Y| - 1 - y)u + p' \). Then, to prove the lower bound, observe that:

\[
\gamma = \frac{|Y|u - (k - z)}{u - \ell} > \frac{|Y|u - (k - z)}{u - \ell} - 1 \Rightarrow \gamma > \frac{|Y|u - (y \ell + (|Y| - 1 - y)u + p')}{u - \ell} - 1 \Rightarrow \gamma > \frac{|Y|u - y \ell - |Y|u + u + yu - p' - u + \ell}{u - \ell} \Rightarrow 0 > \ell - p' \Rightarrow p' > \ell
\]

To prove the upper bound, observe that:

\[
\gamma = \frac{|Y|u - (k - z)}{u - \ell} \leq \frac{|Y|u - (k - z)}{u - \ell} \Rightarrow \gamma \leq \frac{|Y|u - (y \ell + (|Y| - 1 - y)u + p')}{u - \ell} \Rightarrow \gamma \leq \frac{y(u - \ell) + u - p'}{u - \ell} \Rightarrow 0 \leq u - p' \Rightarrow p' \leq u
\]

Thus, \( \ell < p' \leq u \).

Now, we proceed with our proof of Theorem 4.4.

Proof. Recall P2: maximize \( \sum_{i \in Y} f_i(p_i) \) such that \( p_i \in [t, u] \) for all \( i \in Y \) and \( \sum_{i \in Y} p_i = k - z \). By Lemma 4.3, there exists at most one arm with optimal value \( p_i^* \in (t, u) \).

First, we discuss an edge case. If \( k - z = |Y|\ell \), Line 2 of Algorithm 2 assigns \( \gamma = |Y| \), so Line 11 assigns

\[
\pi_y := i \mapsto \begin{cases} 
\ell, & \text{for } i \in Y_1 = \{Y\} \\
p', & \text{for } i \in Y_2 = \emptyset \\
u, & \text{for } i \in Y_3 = \emptyset 
\end{cases}
\]  

(21)

Thus Algorithm 2 returns the only valid solution to P2 in this case, which is to set \( p_i = \ell \) for all arms \( i \in Y \).

For all other cases \( k - z > |Y|\ell \), we introduce the following notation: let \( Y_1 \) be the set of arms for which \( p_i = \ell \), \( Y_2 \) be a set containing exactly one arm (WLOG \( j \)) where \( p_j = p' \in (t, u) \), and \( Y_3 \) be the remaining set of arms for which \( p_i = u \), with \( \bigcap_{x=1}^{3} Y_x = \emptyset \). Then by Lemma E.1, \( y = |Y| = \frac{|Y|u - (k - z)}{u - \ell} \) and \( p' = k - z - y \ell - (|Y| - 1 - y)u \in (t, u) \).
**P2** is then equivalent to finding a partition $Y \rightarrow Y_1 \cup Y_2 \cup Y_3$ which maximizes the following:

$$\arg \max_{Y_1, Y_2, Y_3} \sum_{i \in Y_1} f_i(t) + f_j(p') + \sum_{i' \in Y_2} f_{i'}(u)$$

s.t. $|Y_1| = \gamma$, $Y_2 = \{j\}$,

$$\sum_{x=1}^{3} y_x = 0, \quad \text{and} \quad \sum_{x=1}^{3} y_x = Y$$

(22)

Subtracting the constant $\sum_{i \in Y} f_i(t)$ and simplifying yields:

$$\arg \max_{Y_1, Y_2, Y_3} f_j(p') - f_j(t) + \sum_{i' \in Y_2} f_{i'}(u) - f_{i'}(t)$$

s.t. $|Y_1| = \gamma$, $Y_2 = \{j\}$,

$$\sum_{x=1}^{3} y_x = 0, \quad \text{and} \quad \sum_{x=1}^{3} y_x = Y$$

(23)

Suppose we sort arms in ascending order by $f_i(u) - f_i(t)$. Let us create the set $Y_3'$ from the last $|Y| - \gamma - 1$ arms. By monotonicity, for all $i \in Y_3'$ and $j \notin Y_3'$,

$$f_j(p') - f_j(t) \leq f_i(u) - f_i(t)$$

(24)

Thus, setting $Y_1^* = Y_3'$ reduces the optimization problem in Eq. 22 to finding a partition over the remaining sets $Y_1$ and $Y_2$.

$$\arg \max_{Y_1, Y_2} f_j(p') - f_j(t)$$

s.t. $|Y_1| = \gamma$, $Y_2 = \{j\}$,

$$\sum_{x=1}^{2} y_x = 0, \quad \text{and} \quad \sum_{x=1}^{2} y_x = Y \setminus Y_1^*$$

(25)

Finally, we solve Equation 25 by finding the arm $j$ with maximal value $f_j(p') - f_j(t)$. Then $Y_2^* = j$, $Y_1^* = Y \setminus \{Y_2 \cup Y_1^*\}$, and we are done. \qed

**Corollary 4.5.** Alg. 2 has time complexity $O(|Y| \log |Y|)$.

At worst, Algorithm 2 requires two sorts: once on Line 5, and a second time on Line 8, for a total computational cost of $O(2|Y| \log |Y|)$. In total, the computational cost of Algorithm 1 is at worst $O\left(\frac{kN}{\epsilon^3}\right)$ when all $N$ arms are in $X$.

### E.2 Dependent Rounding Sampling Approach

Here we provide pseudocode for the sampling algorithm introduced in Section 4.2, along with its associated SIMPLIFY subroutine [39].

**Algorithm 4** Sampling Subroutine (adapted from Srinivasan [39])

```
1: procedure SIMPLIFY($\alpha \in [0, 1], \beta \in [0, 1]$)
2:   if $\alpha = \beta = 0$ then
3:     $p_i, p_j \leftarrow [0, 0]$
4:   else if $\alpha = \beta = 1$ then
5:     $p_i, p_j \leftarrow [1, 1]$
6:   else if $\alpha + \beta = 1$ then
7:     $flag \leftarrow X \sim B(n = 1, p = \alpha)$
8:     $p_i, p_j \leftarrow [1, 0]$ if $flag$ else $[0, 1]$
9:   else if $0 < \alpha + \beta < 1$ then
10:    $flag \leftarrow X \sim B(n = 1, p = \frac{\alpha}{\alpha + \beta})$
11:    $p_i, p_j \leftarrow [\alpha + \beta, 0]$ if $flag$ else $[0, \alpha + \beta]$
12:   else if $1 < \alpha + \beta < 2$ then
13:    $flag \leftarrow X \sim B(n = 1, p = \frac{1 - \beta}{1 - \alpha - \beta})$
14:    $p_i, p_j \leftarrow [1, \alpha + \beta - 1]$ if $flag$ else $[\alpha + \beta - 1, 1]$
15: return $p_i, p_j$
```
Algorithm 5 Sampling Algorithm (adapted from Srinivasan [39])

1: procedure Sample($G = (V, E)$)
2:   $H \leftarrow G \setminus \{v | \exists e \in G \text{ s.t. } e_{dest} = v\}$
3:   if $|H| = 1$ then return $G$
4: else if $|H| \geq 2$ then
5:   $A \subseteq G \in \{\frac{H}{2}\}$
6:   $B \leftarrow G \setminus A$
7:   pairs $\leftarrow \{(a_i, b_i) \in A \times B | i \in I\}$
8:   $H' \leftarrow (V = 0, E = \emptyset)$
9:   for $(v_i, v_j) \in \text{pairs}$ do
10:      $H' \leftarrow H' \cup \{v_i, v_j\}$
11:      $H' \leftarrow H' \cup \{v'; e_{v', \alpha} | \alpha \in \{i, j\}\}$
12:      $X_i, X_j \leftarrow \text{SIMPLIFY}(p\alpha, p\beta)$
13:      $z_\alpha \leftarrow X_j$
14:      $z_\alpha \leftarrow X_j$
15:      if $z_\alpha \in \{0, 1\}$ then
16:         $p\alpha \leftarrow X_j$
17:      else $p\alpha \leftarrow X_j$
18:   return Sample($F$)

- Select subgraph containing nodes without a parent
- $z_\nu \in \{0, 1\} \forall \nu \in G; \sum_\nu z_\nu = k$

- Defined in Algorithm 4
- If $X_i$ was fixed, $z_\alpha \in \{0, 1\}$
- If $X_j$ was fixed, $z_\beta \in \{0, 1\}$

- $\forall \nu \in G \cap H'$, update attribute values per $H'$
F ADDITIONAL EXPERIMENTAL DETAILS

In this section, we discuss additional details of our empirical study in Section 5. We provide a description and pseudocode of the heuristic policies (F.1), discuss our choice of fairness metric (F.2), and provide additional details of our Synthetic dataset (F.3).

Code and instructions needed to reproduce these experimental results are included at: https://github.com/crherlihy/prob_fair_rmab. All results presented in this paper are bootstrapped over 100 simulation iterations, with a time horizon $T = 180$, cohort size $N = 100$, and budget $k = 20$, unless otherwise noted. We utilize seeds to ensure reproducible variation for each randomized parameter, including actualized transitions in each simulation. We have run simulations on an Intel(R) Core i7 CPU with 16Gb of RAM. Simulations are configurable via configuration files; runs are trivially parallelizable via these configuration files.

F.1 Heuristic Algorithms

In Section 5, three heuristics based on the Threshold Whittle algorithm are introduced: $H_{\text{FIRST}}$, $H_{\text{LAST}}$, and $H_{\text{RAND}}$. Here, we go into more detail and provide pseudocode.

Definition F.1. Within the context of Algorithm 6, we define a constrained pull to be one that is executed to satisfy an integer periodicity constraint. Only arms that have not yet been pulled the required number of times within the $\nu$-length interval are available; other arms are excluded from consideration, unless all arms have already satisfied their constraints. In this case, all arms are available to be pulled.

If a pull is not constrained, we say it is unconstrained or residual.

The $H_{\text{FIRST}}$ heuristic requires that all constrained pulls must occur at the start of the interval. This implies that the first $N/k$ timesteps in each interval are dedicated to pulling all $N$ arms.

The $H_{\text{LAST}}$ heuristic requires that all constrained pulls must occur at the end of the interval. Unlike the $H_{\text{FIRST}}$ heuristic, not all arms will necessarily be pulled in the last $N/k$ timesteps, as some arms will have already satisfied their constraint earlier in the interval via unconstrained pulls. These leftover constrained pulls function as unconstrained pulls, per Definition F.1.

The $H_{\text{RAND}}$ heuristic chooses random positions within the interval for constrained pulls to occur. Similarly to the $H_{\text{LAST}}$ heuristic, some of the later constrained pulls may become unconstrained pulls if all arms have already satisfied their constraint earlier in the interval.

Algorithm 6 Periodicity Constraint-Enforcing Heuristic Based on Threshold Whittle

```plaintext
1: procedure SIMULATION(A, T, ν, k)
2:   for interval ∈ [0, T] with step size ν do
3:     $C_{\text{interval}}$ ← ∅
4:     for $a$ ∈ A do
5:       $a$.last observed state ← 1
6:       $a$.time since pull ← 1
7:     end for
8:     for $t$ ∈ T do
9:       $i$ ← GetInterval($t$)
10:      if $t$ is a constrained pull ∧ $C_i \subseteq A$ then
11:         $A'$ ← {$a$|$a \in A \setminus C_i$}
12:      else if $t$ is a residual pull ∨ $C_i = A$ then
13:         $A'$ ← A
14:      end if
15:      $A'_{k}$ ← SelectTopK($A'$, $k, t$)
16:      $C_i$ ← $C_i \cup A'_{k}$
17:     end for
18:   end for
19: end procedure

F.2 Fairness Metric Choices

It is not immediately obvious which evaluation metric(s) best indicate whether we have improved distributive fairness. While constraint satisfaction itself is a logical candidate, it is Boolean-valued at the arm level, and thus does not reflect to what extent a policy fairly allocates pulls. Even if we were to report population-level constraint satisfaction (i.e., by noting the proportion of arms for which a given fairness constraint is satisfied, either over the course of a single simulation, or in expectation over a set of simulation iterations), this would be tautologically biased in favor of ProbFair and the Threshold Whittle-based heuristics, which explicitly encode constraint satisfaction.
This observation motivates us to consider proxy metrics, including the price of fairness (PoF), the Herfindahl–Hirschman Index (HHI), and the earth mover’s distance (EMD).

**Price of Fairness.** Consider price of fairness, defined formally as:

\[
P_{\text{PoF}}(\text{ALG}) = \frac{E_{\text{TW}}[R(\cdot)] - E_{\text{ALG}}[R(\cdot)]}{E_{\text{TW}}[R(\cdot)]}
\]

Price of fairness is the relative loss in total expected reward associated with following a distributive fairness-enforcing policy, as compared to **Threshold Whittle** [4]. A small loss (≈ 0%) indicates that fairness has a small impact on total expected reward; conversely, a large loss means total expected reward is sacrificed in order to satisfy the fairness constraints.

**Lemma F.2.** Price of fairness is inversely proportional to intervention benefit.

**Proof.** The statement in Lemma F.2 is equivalent to the statement "Given \( y, z > 0 \), there exists \( \alpha \in \mathbb{R} \) such that \( \frac{x - z}{y - z} = \alpha \frac{x - y}{z - y} \) for all \( x > 0 \)." Here \( x = E_{\text{ALG}}[R(\cdot)], y = E_{\text{NoAct}}[R(\cdot)], \) and \( z = E_{\text{TW}}[R(\cdot)]. \) Consider \( \alpha = \frac{x - y}{z - y}. \) Then \( \alpha \frac{x - y}{z - y} = \frac{x - z}{y - z}. \) Thus, for any algorithm ALG, \( P_{\text{PoF}}(\text{ALG}) = \frac{1}{E_{\text{NoAct, TW}}(\text{ALG})}. \)

**Herfindahl–Hirschman Index (HHI).** The Herfindahl–Hirschman Index (HHI) [34], is a statistical measure of concentration useful for measuring the extent to which a small set of arms receives a large proportion of attention due to an unequal distribution of scarce pulls [12]. It is defined as:

\[
\text{HHI}(\text{ALG}) := N \left( \frac{1}{KT} \sum_{t=1}^{T} a_{t} \right)^{2}
\]

HHI ranges from 1/N to 1; higher values indicate that pulls are concentrated on a small subset of arms. However, HHI is an imperfect evaluation metric for addressing our prioritarian concern for arms that would be deprived of algorithmic attention (i.e., fail to receive any pulls) under **Threshold Whittle** (see Appendix B). Since entries are squared, reducing \( u \) offers a more direct path to lowering HHI than increasing \( \ell. \) However, reducing \( u \) will not accomplish our stated goal of guaranteeing each arm a strictly positive lower bound on the probability that it will receive a pull at any given timestep.

**Earth Mover’s Distance.** The earth mover’s distance (EMD), or Wasserstein metric, is a measure of distance between two distributions. Specifically, we measure the distance of an algorithm’s distribution of cumulative pull allocations to a fair reference distribution, **Round-Robin.** Though differences in distances are meaningful, EMD does not directly map to our fairness desiderata. That is, a given level of fairness enforcement (e.g., as characterized by the hyperparameters \( \ell \) or \( v \)) is not associated with a specific range of EMD values. Hence, our discussion of (normalized) earth mover’s distances in Section 5 focuses on relative differences between policies.

### F.3 Synthetic Dataset

**Conjecture F.3.** The set of forward (reverse) threshold-optimal arms are a subset of the set of concave (strictly convex) arms for the local reward function we consider, \( r(s) = s. \)

Mate et al. [25] provide conditions for threshold optimality. First, the arm must satisfy the structural constraints (Section 2) and the **indexability** condition \( p_{0,1}^0 - p_{0,1}^0 + p_{1,1}^0 - p_{0,1}^0 \leq 1. \) Then, the following inequalities determine forward (reverse) threshold optimality:

\[
\begin{align*}
p_{0,1}^0 - p_{0,1}^0 & \geq p_{1,1}^0 - p_{0,1}^0 \quad \text{forward threshold-optimal} \\
p_{0,1}^0 - p_{0,1}^0 & \leq p_{1,1}^0 - p_{0,1}^0 \quad \text{reverse threshold-optimal}
\end{align*}
\]

We conjecture that these conditions necessarily imply the conditions for concavity, repeated here for convenience:

\[
\begin{align*}
p_{0,1}^0 & \leq \left( p_{1,1}^0 - p_{0,1}^0 \right) \left( 1 - p_{0,1}^0 + p_{0,1}^0 \right) \quad \text{concave} \\
p_{0,1}^0 & > \left( p_{1,1}^0 - p_{0,1}^0 \right) \left( 1 - p_{0,1}^0 + p_{0,1}^0 \right) \quad \text{strictly convex}
\end{align*}
\]