Scalable Decision-Focused Learning in Restless Multi-Armed Bandits with Application to Maternal and Child Health

Kai Wang[*†],[1] **Shresth Verma**[*],[2] **Aditya Mate**[†],[1] **Sanket Shah**,[1] **Aparna Taneja**,[2]
Neha Madhiwalla,[3] **Aparna Hegde**,[3] **Milind Tambe**[1,2]

[1]Harvard University, [2]Google Research, [3]ARMMAN
{kaiwang, aditya_mate, sanketshah}@g.harvard.edu, {aparnataneja, milindtambe}@google.com,
{neha, aparnahegde}@armman.org

Abstract

This paper studies restless multi-armed bandit (RMAB) problems with unknown arm transition dynamics but with known correlated arm features. The goal is to learn a model to predict transition dynamics given features, where the Whittle index policy solves the RMAB problems using predicted transitions. However, prior works often learn the model by maximizing the predictive accuracy instead of final RMAB solution quality, causing a mismatch between training and evaluation objectives. To address this shortcoming, we propose a novel approach for decision-focused learning in RMAB that directly trains the predictive model to maximize the Whittle index solution quality. We present three key contributions: (i) we establish differentiability of the Whittle index policy to support decision-focused learning; (ii) we significantly improve the scalability of decision-focused learning approaches in sequential problems, specifically RMAB problems; (iii) we apply our algorithm to a previously collected dataset of maternal and child health to demonstrate its performance. Indeed, our algorithm is the first for decision-focused learning in RMAB that scales to real-world problem sizes.

1 Introduction

Restless multi-armed bandits (RMABs) (Weber and Weiss 1990; Tekin and Liu 2012) are composed of a set of heterogeneous arms and a planner who can pull multiple arms under budget constraint at each time step to collect rewards. Different from the classic stochastic multi-armed bandits (Gittins, Glazebrook, and Weber 2011; Bubeck and Cesa-Bianchi 2012), the state of each arm in an RMAB can change even when the arm is not pulled, where each arm follows a Markovian process to transition between different states with transition probabilities dependent on arms and the pulling decision. Rewards are associated with different arm states, where the planner's goal is to plan a sequential pulling policy to maximize the total reward received from all arms. RMABs are commonly used to model sequential scheduling problems where limited resources must be strategically assigned to different tasks sequentially to maximize performance. Examples include machine maintenance (Glazebrook, Ruiz-Hernandez, and Kirkbride 2006), cognitive radio sensing problem (Bagheri and Scaglione 2015), and healthcare (Mate et al. 2022).

In this paper, we study offline RMAB problems with unknown transition dynamics but with given arm features. The goal is to learn a mapping from arm features to transition dynamics, which can be used to infer the dynamics of unseen RMAB problems to plan accordingly. Prior works (Mate et al. 2022; Sun et al. 2018) often learn the transition dynamics from the historical pulling data by *maximizing the predictive accuracy*. However, RMAB performance is evaluated *by its solution quality* derived from the predicted transition dynamics, which leads to a mismatch in the training objective and the evaluation objective. Previously, decision-focused learning (Wilder, Dilkina, and Tambe 2019) has been proposed to directly optimize the solution quality rather than predictive accuracy, by integrating the one-shot optimization problem (Donti, Amos, and Kolter 2017; Perrault et al. 2020) or sequential problems (Wang et al. 2021; Futoma, Hughes, and Doshi-Velez 2020) as a differentiable layer in the training pipeline. Unfortunately, while decision-focused learning can successfully optimize the evaluation objective, it is computationally extremely expensive due to the presence of the optimization problems in the training process. Specifically, for RMAB problems, the computation cost of decision-focused learning arises from the complexity of the sequential problems formulated as Markov decision processes (MDPs), which limits the applicability to RMAB problems due to the PSPACE hardness of finding the optimal solution (Papadimitriou and Tsitsiklis 1994).

Our main contribution is a novel and scalable approach for decision-focused learning in RMAB problems using Whittle index policy, a commonly used approximate solution in RMABs. Our three key contributions are (i) we establish the differentiability of Whittle index policy to support decision-focused learning to directly optimize the RMAB solution quality; (ii) we show that our approach of differentiating through Whittle index policy improves the scalability of decision-focused learning in RMAB; (iii) we apply our algorithm to an anonymized maternal and child health RMAB dataset previously collected by ARMMAN (2022) to evaluate the performance of our algorithm in simulation.

[*]These authors contributed equally.
[†]Work done during an internship at Google Research.

We establish the differentiability of Whittle index by showing that Whittle index can be expressed as a solution to a full-rank linear system reduced from Bellman equations with transition dynamics as entries, which allows us to compute the derivative of Whittle index with respect to transition dynamics. On the other hand, to execute Whittle index policy, the standard selection process of choosing arms with top-k Whittle indices to pull is non-differentiable. We relax this non-differentiable process by using a differentiable soft top-k selection to establish differentiability. Our differentiable Whittle index policy enables decision-focused learning in RMAB problems to backpropagate from final policy performance to the predictive model. We significantly improve the scalability of decision-focused learning, where the computation cost of our algorithm $O(NM^{\omega+1})$ scales linearly in the number of arms N and polynomially in the number of states M with $\omega \approx 2.373$, while previous work scales exponentially $O(M^{\omega N})$. This significant reduction in computation cost is crucial for extending decision-focused learning to RMAB problems with large number of arms.

In our experiments, we apply decision-focused learning to RMAB problems to optimize importance sampling-based evaluation on synthetic datasets as well as an anonymized RMAB dataset about a maternal and child health program previously collected by (ARMMAN 2022) – these datasets are the basis of comparing different methods in simulation. We compare decision-focused learning with the two-stage method that trains to minimize the predictive loss. The two-stage method achieves the best predictive loss but significantly degraded solution quality. In contrast, decision-focused learning reaches a slightly worse predictive loss but with a much better importance sampling-based solution quality evaluation and the improvement generalizes to the simulation-based evaluation that is built from the data. Lastly, the scalability improvement is the crux of applying decision-focused learning to real-world RMAB problems: our algorithm can run decision-focused learning on the maternal and child health dataset with hundreds of arms, whereas state of the art is a 100-fold slower even with 20 arms and grows exponentially worse.

Related Work

Restless multi-armed bandits with given transition dynamics This line of research primarily focuses on solving RMAB problems to get a sequential policy. The complexity of solving RMAB problems optimally is known to be PSPACE hard (Papadimitriou and Tsitsiklis 1994). One approximate solution is proposed by Whittle (1988), where they use Lagrangian relaxation to decompose arms and compute the associated Whittle indices to define a policy. Specifically, the indexability condition (Akbarzadeh and Mahajan 2019; Wang et al. 2019) guarantees this Whittle index policy to be asymptotically optimal (Weber and Weiss 1990). In practice, Whittle index policy usually provides a near-optimal solution to RMAB problems.

Restless multi-armed bandits with missing transition dynamics When the transition dynamics are unknown in RMAB problems but an interactive environment is available, prior works (Tekin and Liu 2012; Liu, Liu, and Zhao 2012; Oksanen and Koivunen 2015; Dai et al. 2011) consider this as an online learning problem that aims to maximize the expected reward. However, these approaches become infeasible when interacting with the environment is expensive, e.g., healthcare problems (Mate et al. 2022). In this work, we consider the offline RMAB problem, and each arm comes with an arm feature that is correlated to the transition dynamics and can be learned from the past data.

Decision-focused learning The predict-then-optimize framework (Elmachtoub and Grigas 2021) is composed of a predictive problem that makes predictions on the parameters of the later optimization problem, and an optimization problem that uses the predicted parameters to come up with a solution, where the overall objective is the solution quality of the proposed solution. Standard two-stage learning method solves the predictive and optimization problems separately, leading to a mismatch of the predictive loss and the evaluation metric (Huang et al. 2019; Lambert et al. 2020; Johnson and Khoshgoftaar 2019). In contrast, decision-focused learning (Wilder, Dilkina, and Tambe 2019; Mandi et al. 2020; Elmachtoub, Liang, and McNellis 2020) learns the predictive model to directly optimize the solution quality by integrating the optimization problem as a differentiable layer (Amos and Kolter 2017; Agrawal et al. 2019) in the training pipeline. Our offline RMAB problem is a predict-then-optimize problem, where we first (offline) learn a mapping from arm features to transition dynamics from the historical data (Mate et al. 2022; Sun et al. 2018), and the RMAB problem is solved using the predicted transition dynamics accordingly. Prior work (Mate et al. 2022) is limited to using two-stage learning to solve the offline RMAB problems. While decision-focused learning in sequential problems were primarily studied in the context of MDPs (Wang et al. 2021; Futoma, Hughes, and Doshi-Velez 2020) they come with an expensive computation cost that immediately becomes infeasible in large RMAB problems.

2 Model: Restless Multi-armed Bandit

An instance of the restless multi-armed bandit (RMAB) problem is composed of a set of N arms, each is modeled as an independent Markov decision process (MDP). The i-th arm in a RMAB problem is defined by a tuple $(\mathcal{S}, \mathcal{A}, R_i, P_i)$. $\mathcal{S}$ and $\mathcal{A}$ are the identical state and action spaces across all arms. $R_i, P_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ are the reward and transition functions associated to arm i. We consider finite state space with $|\mathcal{S}| = M$ fully observable[1] states, and action set $\mathcal{A} = \{0, 1\}$ corresponding to not pulling or pulling the arm, respectively. For each arm i, the reward is denoted by $R_i(s_i, a_i, s_i') = R(s_i)$, i.e., the reward $R(s_i)$ only depends on the current state s_i, where $R : \mathcal{S} \to \mathbb{R}$ is a vector of size M. Given the state s_i and action a_i, $P_i(s_i, a_i) = [P_i(s_i, a_i, s_i')]_{s_i' \in \mathcal{S}}$ defines the probability distribution of transitioning to all possible next states $s_i' \in \mathcal{S}$.

In a RMAB problem, at each time step $t \in [T]$, the learner

[1]Fully observable RMAB can be generalized to partially observable RMAB using belief states as seen in Section 4.6.

observes $\boldsymbol{s}_t = [s_{t,i}]_{i\in[N]} \in \mathcal{S}^N$, the states of all arms. The learner then chooses action $\boldsymbol{a}_t = [a_{t,i}]_{i\in[N]} \in \mathcal{A}^N$ denoting the pulling actions of all arms, which has to satisfy a budget constraint $\sum_{i\in[N]} a_{t,i} \leq K$, i.e., the learner can pull at most K arms at each time step. Once the action is chosen, arms receive action $\boldsymbol{a}_t$ and transitions under P with rewards $\boldsymbol{r}_t = [r_{t,i}]_{i\in[N]}$ accordingly. We denote a full trajectory by $\tau = (\boldsymbol{s}_1, \boldsymbol{a}_1, \boldsymbol{r}_1, \cdots, \boldsymbol{s}_T, \boldsymbol{a}_T, \boldsymbol{r}_T)$. The total reward is defined by the summation of the discounted reward across T time steps and N arms, i.e., $\sum_{t=1}^{T} \gamma^{t-1} \sum_{i\in[N]} r_{t,i}$, where $0 < \gamma \leq 1$ is the discount factor.

A policy is denoted by π, where $\pi(\boldsymbol{a} \mid \boldsymbol{s})$ is the probability of choosing action $\boldsymbol{a}$ given state $\boldsymbol{s}$. Additionally, we define $\pi(a_i = 1 \mid \boldsymbol{s})$ to be the marginal probability of pulling arm i given state $\boldsymbol{s}$, where $\pi(\boldsymbol{s}) = [\pi(a_i = 1 \mid \boldsymbol{s})]_{i\in[N]}$ is a vector of arm pulling probabilities. Specifically, we use π^* to denote the optimal policy that optimizes the cumulative reward, while π^{solver} to denote a near-optimal policy solver.

3 Problem Statement

This paper studies the RMAB problem where we do not know the transition probabilities $P = \{P_i\}_{i\in[N]}$ in advance. Instead, we are given a set of features $\boldsymbol{x} = \{x_i \in \mathcal{X}\}_{i\in[N]}$, each corresponding to one arm. The goal is to learn a mapping $f_w : \mathcal{X} \to \mathcal{P}$, parameterized by weights w, to make predictions on the transition probabilities $P = f_w(\boldsymbol{x}) \coloneqq \{f_w(x_i)\}_{i\in[N]}$. The predicted transition probabilities are later used to solve the RMAB problem to derive a policy $\pi = \pi^{\text{solver}}(f_w(\boldsymbol{x}))$. The performance of the model f is evaluated by the performance of the proposed policy π.

3.1 Training and Testing Datasets

To learn the mapping f_w, we are given a set of RMAB instances as training examples $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}, \mathcal{T})\}$, where each instance is composed of a RMAB problem with feature $\boldsymbol{x}$ that is correlated to the unknown transition probabilities P, and a set of realized trajectories $\mathcal{T} = \{\tau^{(j)}\}_{j\in J}$ generated from a given behavior policy π_{beh} that determined how to pull arms in the past. The testing set $\mathcal{D}_{\text{test}}$ is defined similarly but hidden at training time.

3.2 Evaluation Metrics

Predictive loss To measure the correctness of transition probabilities $P = \{P_i\}_{i\in[N]}$, we define the predictive loss as the average negative log-likelihood of seeing the given trajectories $\mathcal{T}$, i.e., $\mathcal{L}(P, \mathcal{T}) \coloneqq -\log \Pr(\mathcal{T} \mid P) = -\mathbb{E}_{\tau\sim\mathcal{T}} \sum_{t\in[T]} \log P(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$. Therefore, we can define the predictive loss of a model f_w on dataset $\mathcal{D}$ by:

$$\mathbb{E}_{(\boldsymbol{x},\mathcal{T})\sim\mathcal{D}}\, \mathcal{L}(f_w(\boldsymbol{x}), \mathcal{T}) \tag{1}$$

Policy evaluation On the other hand, given transition probabilities P, we can solve the RMAB problem to derive a policy $\pi^{\text{solver}}(P)$. We can use the historical trajectories $\mathcal{T}$ to evaluate how good the policy performs, denoted by $\text{Eval}(\pi^{\text{solver}}(P), \mathcal{T})$. Given dataset $\mathcal{D}$, we can evaluate the predictive model f_w on dataset $\mathcal{D}$ by:

$$\mathbb{E}_{(\boldsymbol{x},\mathcal{T})\sim\mathcal{D}}\, \text{Eval}(\pi^{\text{solver}}(f_w(\boldsymbol{x})), \mathcal{T}) \tag{2}$$

Two common types of policy evaluation are importance sampling-based off-policy policy evaluation and simulation-based evaluation, which will be discussed in Section 5.

3.3 Learning Methods

Two-stage learning To learn the predictive model f_w, we can minimize Equation 1 by computing gradient $\frac{d\mathcal{L}(f_w(\boldsymbol{x}),\mathcal{T})}{dw}$ to run gradient descent. However, this training objective (Equation 1) differs from the evaluation objective (Equation 2), which often leads to suboptimal performance.

Decision-focused learning In contrast, we can directly run gradient ascent to maximize Equation 2 by computing the gradient $\frac{d\text{Eval}(\pi^{\text{solver}}(f_w(\boldsymbol{x})),\mathcal{T})}{dw}$. However, in order to compute the gradient, we need to differentiate through the policy solver π^{solver} and the corresponding optimal solution. Unfortunately, finding the optimal policy in RMABs is expensive and the policy is high-dimensional. Both of these challenges prevent us from computing the gradient to achieve decision-focused learning.

4 Decision-focused Learning in RMABs

In this paper, instead of grappling with the optimal policy, we consider the Whittle index policy (Whittle 1988) – the dominant solution paradigm used to solve the RMAB problem. Whittle index policy is easier to compute and has been shown to perform well in practice. In this section we establish that it is also possible to backpropagate through the Whittle index policy. This differentiability of Whittle index policy allows us to run decision-focused learning to directly maximize the performance in the RMAB problem.

4.1 Whittle Index and Whittle Index Policy

Informally, the Whittle index of an arm captures the added value derived from pulling that arm. The key idea is to determine the Whittle indices of all arms and to pull the arms with the highest values of the index.

To evaluate the value of pulling an arm i, we consider the notion of 'passive subsidy', which is a hypothetical exogenous compensation m rewarded for not pulling the arm (i.e. for choosing action $a = 0$). Whittle index is defined as the smallest subsidy necessary to make pulling as rewarding as not pulling, assuming indexability (Liu and Zhao 2010):

Definition 4.1 (Whittle index)**.** Given state $u \in \mathcal{S}$, we define the Whittle index associated to state u by:

$$W_i(u) \coloneqq \inf_m \{Q_i^m(u; a = 0) = Q_i^m(u; a = 1)\} \tag{3}$$

where the value functions are defined by the following Bellman equations, augmented with subsidy m for action $a = 0$.

$$V_i^m(s) = \max_a Q_i^m(s; a) \tag{4}$$

$$Q_i^m(s; a) = m\mathbf{1}_{a=0} + R(s) + \gamma \sum\nolimits_{s'} P_i(s, a, s') V_i^m(s') \tag{5}$$

Given the Whittle indices of all arms and all states $W = [W_i(u)]_{i\in[N],u\in\mathcal{S}}$, the Whittle index policy is denoted by $\pi^{\text{whittle}} : \mathcal{S}^N \longrightarrow [0, 1]^N$, which takes the states of all arms as input to compute their Whittle indices and output the probabilities of pulling arms. This policy repeats for every time step to pull arms based on the index values.

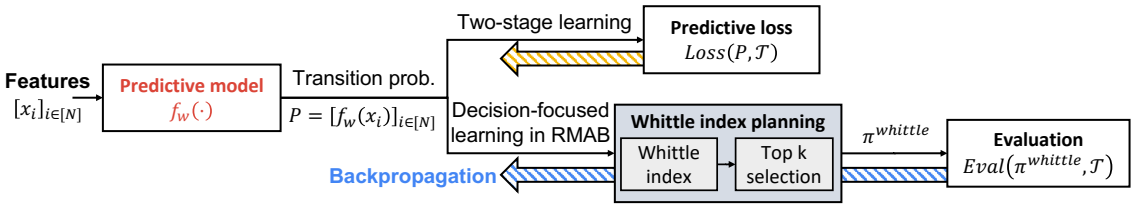


Figure 1: This flowchart visualizes different methods of learning the predictive model. Two-stage learning directly compares the predicted transition probabilities with the given data to define a predictive loss to run gradient descent. Decision-focused learning instead goes through a policy solver using Whittle index policy to estimate the final evaluation and run gradient ascent.

4.2 Decision-focused Learning Using Whittle Index Policy

Instead of using the optimal policy π^* to run decision-focused learning with expensive computation cost, we use Whittle index policy π^{whittle} to determine how to pull arms as an approximate solution. In this case, in order to run decision-focused learning, we need to compute the derivative of the evaluation metric by chain rule:

$$\frac{d\text{Eval}(\pi^{\text{whittle}}, \mathcal{T})}{dw} = \frac{d\text{Eval}(\pi^{\text{whittle}}, \mathcal{T})}{d\pi^{\text{whittle}}} \frac{d\pi^{\text{whittle}}}{dW} \frac{dW}{dP} \frac{dP}{dw} \tag{6}$$

where W is the Whittle indices of all states under the predicted transition probabilities P. The policy π^{whittle} is the Whittle index policy induced by W. The flowchart is illustrated in Figure 1.

The term $\frac{d\text{Eval}(\pi^{\text{whittle}}, \mathcal{T})}{d\pi^{\text{whittle}}}$ can be computed via policy gradient theorem (Sutton, Barto et al. 1998), and the term $\frac{dP}{dw}$ can be computed using auto-differentiation. However, there are still two challenges remaining: (i) how to differentiate through Whittle index policy to get $\frac{d\pi^{\text{whittle}}}{dW}$ (ii) how to differentiate through Whittle index computation to derive $\frac{dW}{dP}$.

4.3 Differentiability of Whittle Index Policy

A common choice of Whittle index policy is defined by:

Definition 4.2 (Strict Whittle index policy)**.**

$$\pi_W^{\text{strict}}(\boldsymbol{s}) = \mathbf{1}_{\text{top-k}([W_i(s_i)]_{i\in[N]})} \in \{0,1\}^N \tag{7}$$

which selects arms with the top-k Whittle indices to pull.

However, the strict top-k operation in the strict Whittle index policy is non-differentiable, which prevents us from computing a meaningful estimate of $\frac{d\pi^{\text{whittle}}}{dW}$ in Equation 6. We circumvent this issue by relaxing the top-k selection to a soft-top-k selection (Xie et al. 2020), which can be expressed as an optimal transport problem with regularization, making it differentiable. We apply soft-top-k to define a new differentiable soft Whittle index policy:

Definition 4.3 (Soft Whittle index policy)**.**

$$\pi_W^{\text{soft}}(\boldsymbol{s}) = \text{soft-top-k}([W_j(s_i)]_{i\in[N]}) \in [0,1]^N \tag{8}$$

Using the soft Whittle index policy, the policy becomes differentiable and we can compute $\frac{d\pi^{\text{whittle}}}{dW}$.

4.4 Differentiability of Whittle Index

The second challenge is the differentiability of Whittle index. Whittle indices are often computed using value iteration and binary search (Qian et al. 2016; Mate et al. 2020) or mixed integer linear program. However, these operations are not differentiable and we cannot compute the derivative $\frac{dW}{dP}$ in Equation 6 directly.

Main idea After computing the Whittle indices and the value functions of each arm i, the key idea is to construct linear equations that link the Whittle index with the transition matrix P_i. Specifically, we achieve this by resolving the $\max$ operator in Equation 4 of Definition 4.1 by determining the optimal actions a from the pre-computed value functions. Plugging back in Equation 5 and manipulating as shown below yields linear equations in the Whittle index $W_i(u)$ and transition matrix P_i, which can be expressed as a full-rank linear system in P_i, with the Whittle index as a solution. This makes the Whittle index differentiable in P_i.

Selecting Bellman equation Let u and arm i be the target state and target arm to compute the Whittle index. Assume we have precomputed the Whittle index $m = W_i(u)$ for state u and the corresponding value functions $[V_i^m(s)]_{s\in\mathcal{S}}$ for all states under the same passive subsidy $m = W_i(u)$. Equation 5 can be combined with Equation 4 to get:

$$V_i^m(s) \geq \begin{cases} m + R(s) + \gamma \sum_{s'\in\mathcal{S}} P_i(s, a=0, s') V_i^m(s') \\ R(s) + \gamma \sum_{s'\in\mathcal{S}} P_i(s, a=1, s') V_i^m(s') \end{cases} \tag{9}$$

where $m = W_i(u)$.

For each $s \in S$, at least one of the equalities in Equation 9 holds because one of the actions must be optimal and match the state value function $V_i^m(s)$. We can identify which equality holds by simply plugging in values of precomputed value functions $[V_i^m(s)]_{s\in\mathcal{S}}$. Furthermore, for the target state u, both equalities must hold because by the definition of Whittle index, the passive subsidy $m = W_i(u)$ makes both actions equally optimal, i.e. in Equation 3, $V_i^m(u) = Q_i^m(u, a=0) = Q_i^m(u, a=1)$ for $m = W_i(u)$.

Thus Equation 9 can be written in matrix form:

$$\begin{bmatrix} \boldsymbol{V}_i^m \\ \boldsymbol{V}_i^m \end{bmatrix} \geq \begin{bmatrix} \mathbf{1}_M & \gamma \boldsymbol{P}_i(\mathcal{S}, a=0, \mathcal{S}) \\ \mathbf{0}_M & \gamma \boldsymbol{P}_i(\mathcal{S}, a=1, \mathcal{S}) \end{bmatrix} \begin{bmatrix} m \\ \boldsymbol{V}_i^m \end{bmatrix} + \begin{bmatrix} \boldsymbol{R}(S) \\ \boldsymbol{R}(S) \end{bmatrix} \tag{10}$$

where $\boldsymbol{V}_i^m := [V_i^m(s)]_{s\in\mathcal{S}}$, $\boldsymbol{R}(\mathcal{S}) = [R(s)]_{s\in\mathcal{S}}$, and $\boldsymbol{P}_i(\mathcal{S}, a, \mathcal{S}) := [P_i(s, a, s')]_{s,s'\in\mathcal{S}} \in \mathbb{R}^{M\times M}$.

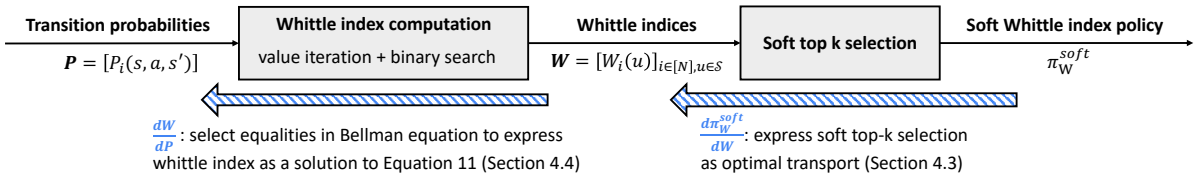


Figure 2: We establish the differentiability of Whittle index policy using a soft top-k selection to construct a soft Whittle index policy, and the differentiability of Whittle index by expressing Whittle index as a solution to a linear system in Equation 11.

By the aforementioned discussion, we know that there are at least $M+1$ equalities in Equation 10 while there are also only $M+1$ variables ($m \in \mathbb{R}$ and $\boldsymbol{V}_i^m \in \mathbb{R}^M$). Therefore, we rearrange Equation 10 and pick only the rows where equalities hold to get:

$$A \begin{bmatrix} \mathbf{1}_M & \gamma \boldsymbol{P}_i(\mathcal{S}, a=0, \mathcal{S}) - I_M \\ \mathbf{0}_M & \gamma \boldsymbol{P}_i(\mathcal{S}, a=1, \mathcal{S}) - I_M \end{bmatrix} \begin{bmatrix} m \\ \boldsymbol{V}_i^m \end{bmatrix} = A \begin{bmatrix} -\boldsymbol{R}(S) \\ -\boldsymbol{R}(S) \end{bmatrix} \tag{11}$$

where we use a binary matrix $A \in \{0,1\}^{(M+1)\times 2M}$ with a single 1 per row to extract the equality. For example, we can set $A_{ij} = 1$ if the j-th row in Equation 10 corresponds to the equality in Equation 9 with the i-th state in the state space S for $i \in [M]$, and the last row $A_{(M+1),j} = 1$ to mark the additional equality matched by the Whittle index definition (see Appendix H for more details). Matrix A picks $M+1$ equalities out from Equation 10 to form Equation 11.

Equation 11 is a full-rank linear system with $m = W_i(u)$ as a solution. This expresses $W_i(u)$ as an implicit function of $\boldsymbol{P}$, allowing for computation of $\frac{dW_i(u)}{d\boldsymbol{P}}$ via autodifferentiation, thus achieving differentiability of the Whittle index. We repeat this process for every arm $i \in [N]$ and every state u. Figure 2 summarizes the differentiable Whittle index policy and the algorithm is shown in Algorithm 1.

4.5 Computation Cost and Backpropagation

It is well studied that Whittle index policy can be computed more efficiently than solving the RMAB problem as a large MDP problem. Here, we show that the use of Whittle index policy also demonstrates a large speed up in terms of backpropagating the gradient in decision-focused learning.

In order to use Equation 11 to compute the gradient of Whittle indices, we need to invert the left-hand-side of Equation 11 with dimensionality $M+1$, which takes $O(M^\omega)$ where $\omega \approx 2.373$ (Alman and Williams 2021) is the best known matrix inversion constant. Therefore, the overall computation of all N arms and M states is $O(NM^{\omega+1})$ per gradient step.

In contrast, the standard decision-focused learning differentiates through the optimal policy using the full Bellman equation with $O(M^N)$ variables, where inverting the large Bellman equation requires $O(M^{\omega N})$ cost per gradient step. Thus, our algorithm significantly reduces the computation cost to a linear dependency on the number of arms N. This significantly improves the scalability of decision-focused learning.

4.6 Extension to Partially Observable RMAB

For partially observable RMAB problem, we focus on a subclass of RMAB problem known as collapsing bandits (Mate et al. 2020). In collapsing bandits, belief states (Monahan 1982) are used to represent the posterior belief of the unobservable states. Specifically, for each arm i, we use $b_i \in \mathcal{B} = \Delta(\mathcal{S}) \subset [0,1]^M$ to denote the posterior belief of an arm, where each entry $b_i(s_i)$ denotes the probability that the true state is $s_i \in \mathcal{S}$. When arm i is pulled, the current true state $s_i \sim b_i$ is revealed and drawn from the posterior belief with expected reward $b_i^\top R$, where we can define the transition probability on the belief states. This process reduces partially observable states to fully observable belief states with in total MT states since the maximal horizon is T. Therefore, we can use the same technique to differentiate through Whittle indices of partially observable states.

5 Policy Evaluation Metrics

In this paper, we use two different variants of evaluation metric: importance sampling-based evaluation (Sutton, Barto et al. 1998) and simulation-based (model-based) evaluation.

Importance sampling-based Evaluation We adopt Consistent Weighted Per-Decision Importance Sampling (CWPDIS) (Thomas 2015) as our importance sampling-based evaluation. Given target policy π and a trajectory $\tau = \{s_1, a_1, r_1, \cdots, s_T, a_T, r_T\}$ executed by the behavior policy π_{beh}, the importance sampling weight is defined by $\rho_{ti} = \prod_{t'=1}^{t} \frac{\pi(a_{t',i}|s_{t'})}{\pi_{\text{beh}}(a_{t',i}|s_{t'})}$. We evaluate the policy π by[2]:

$$\text{Eval}_{\text{IS}}(\pi, \mathcal{T}) = \sum\nolimits_{t\in[T], i\in[N]} \gamma^{t-1} \frac{\mathbb{E}_{\tau\sim\mathcal{T}}\left[r_{t,i}\rho_{ti}(\tau)\right]}{\mathbb{E}_{\tau\sim\mathcal{T}}\left[\rho_{ti}(\tau)\right]} \tag{12}$$

Importance sampling-based evaluations are often unbiased but with a larger variance due to the unstable importance sampling weights. CWPDIS normalizes the importance sampling weights to achieve a consistent estimate.

Simulation-based Evaluation An alternative way is to use the given trajectories to construct an empirical transition probability $\bar{P}$ to build a simulator and evaluate the target policy π. The variance of simulation-based evaluation is small, but it may require additional assumptions on the missing transition when the empirical transition $\bar{P}$ is not fully reconstructed.

[2]This is slightly modified when there is only one trajectory. Please refer to Section F for more details.

Algorithm 1: Decision-focused Learning in RMAB

Input: training set $\mathcal{D}_{\text{train}}$, learning rate r, model f_w
for epoch $= 1, 2, \cdots$ and $(x, \mathcal{T}) \in \mathcal{D}_{\text{train}}$ **do**
  Predict $P = f_w(x)$ and compute Whittle indices $W(P)$.
  Let $\pi^{\text{whittle}} = \pi^{\text{soft}}_W$ and compute $\text{Eval}(\pi^{\text{whittle}}, \mathcal{T})$.
  Update $w = w + r \frac{d\text{Eval}(\pi^{\text{whittle}}, \mathcal{T})}{d\pi^{\text{whittle}}} \frac{d\pi^{\text{whittle}}}{dW} \frac{dW}{dP} \frac{dP}{dw}$, where $\frac{dW}{dP}$ is computed from Equation 11.
end for
Return: predictive model f_w

6 Experiments

We compare two-stage learning (**TS**) with our decision-focused learning (**DF-Whittle**) that optimizes importance sampling-based evaluation directly. We consider three different evaluation metrics including predictive loss, importance sampling evaluation, and simulation-based evaluation to evaluate all learning methods. We perform experiments on three synthetic datasets including 2-state fully observable, 5-state fully observable, and 2-state partially observable RMAB problems. We also perform experiments on a real dataset on maternal and child health problem modelled as a 2-state fully observable RMAB problem with real features and historical trajectories. For each dataset, we use 70%, 10%, 20% of the RMAB problems as the training, validation, and testing sets, respectively. All experiments are averaged over 50 independent runs.

Synthetic datasets We consider RMAB problems composed of $N = 100$ arms, M states, budget $K = 20$, and time horizon $T = 10$ with a discount rate of $\gamma = 0.99$. The reward function is given by $R = [\frac{i-1}{M-1}]_{i \in [M]}$, while the transition probabilities are generated uniformly at random but with a constraint that pulling the arm ($a = 1$) is strictly better than not pulling the arm ($a = 0$) to ensure the benefit of pulling. To generate the arm features, we feed the transition probability of each arm to a randomly initialized neural network to generate fixed-length correlated features with size 16 per arm. The historical trajectories $\mathcal{T}$ with $|\mathcal{T}| = 10$ are produced by running a random behavior policy π_{beh}. The goal is to predict transition probabilities from the arm features and the training trajectories.

Real dataset The Maternal and Child Healthcare Mobile Health program operated by ARMMAN (2022) aims to improve dissemination of health information to pregnant women and mothers with an aim to reduce maternal, neonatal and child mortality and morbidity. ARMMAN serves expectant/new mothers in disadvantaged communities with *median daily family income of $3.22 per day* which is seen to be below the world bank poverty line (World Bank 2020). The program is composed of multiple enrolled beneficiaries and a planner who schedules service calls to improve the overall engagement of beneficiaries; engagement is measured in terms of total number of automated voice (health related) messages that the beneficiary engaged with. More precisely, this problem is modelled as a $M = 2$-state fully observable RMAB problem where each beneficiary's behavior is governed by an MDP with two states - Engaging and Non-Engaging state; engagement is determined by whether the beneficiary listens to an automated voice message (average length 115 seconds) for more than 30 seconds. The planner's task is to recommend a subset of beneficiaries every week to receive service calls from health workers to further improve their engagement behavior. We do not know the transition dynamics, but we are given beneficiaries' socio-demographic features to predict transition dynamics.

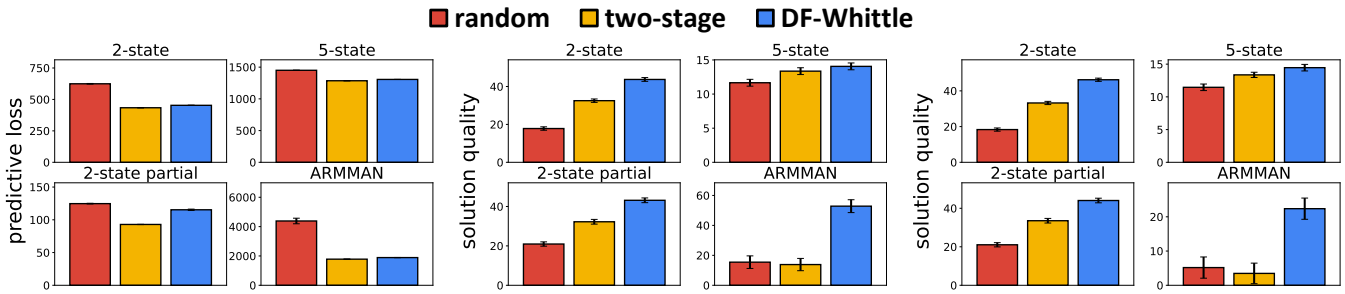
We use a subset of data from the large-scale anonymized quality improvement study performed by ARMMAN for $T = 7$ weeks, obtained from Mate et al. (2022), with beneficiary consent. In the study, a cohort of beneficiaries received Round-Robin policy, scheduling service calls in a fixed order, with a single trajectory $|\mathcal{T}| = 1$ per beneficiary that documents the calling decisions and the engagement behavior in the past. We randomly split the cohort into 8 training groups, 1 validation group, and 3 testing groups each with $N = 639$ beneficiaries and $K = 18$ budget formulated as an RMAB problem. The demographic features of beneficiaries are used to infer the missing transition dynamics.

Data usage All the datasets are anonymized. The experiments are secondary analysis using different evaluation metrics with approval from the ARMMAN ethics board. There is no actual deployment of the proposed algorithm at ARMMAN. For more details about the dataset, consent of data collection, please refer to Appendix B and C.

7 Experimental Results

Performance improvement and justification of objective mismatch In Figure 3, we show the performance of random policy, two-stage, and decision-focused learning (DF-Whittle) on three evaluation metrics - predictive loss, importance sampling-based evaluation and simulation-based evaluation for all domains. For the evaluation metrics, we plot the improvement against the no-action baseline that does not pull any arms throughout the entire RMAB problem. We observe that two-stage learning consistently converges to a smaller predictive loss, while DF-Whittle outperforms two-stage on all solution quality evaluation metrics significantly (p-value < 0.05) by alleviating the objective mismatch issue. This result also provides evidence of aforementioned objective mismatch, where the advantage of two-stage in the predictive loss does not translate to solution quality.

Significance in maternal and child care domain In the ARMMAN data in Figure 3, we assume limited resources that we can only select 18 out of 638 beneficiaries to make service call per week. Both random and two-stage method lead to around 15 more (IS-based evaluation) listening to automated voice messages among all beneficiaries throughout the 7-week program by $18 \times 7 = 126$ service calls, when compared to not scheduling any service call; this low improvement also reflects the hardness of maximizing the effectiveness of service calls. In contrast, decision-focused learning achieves an increase of beneficiaries listening to 50 more voice messages overall; DF-whittle achieves a much



a Predictive loss b IS-based evaluation c Simulation-based evaluation

Figure 3: Comparison of predictive loss, importance sampling-based evaluation, and simulation-based evaluation on all synthetic domains and the real ARMMAN dataset. For the evaluation metrics, we plot the improvement against the no-action baseline that does not pull any arm. Although two-stage method achieves the smallest predictive loss, decision-focused learning consistently outperforms two-stage method in both *solution quality* evaluation metrics across all domains.

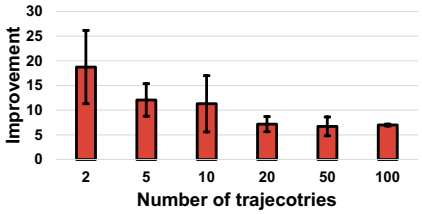
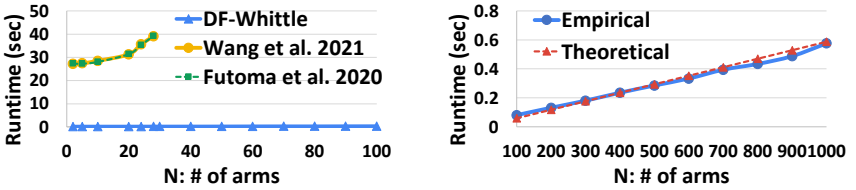


Figure 4: The performance improvement of decision-focused v.s. two-stage method with varying number of trajectories.



a Comparing our algorithm to other decision-focused baselines.

b Computation cost with varying number of arms N.

Figure 5: We compare the computation cost of our decision-focused learning with other baselines and the theoretical complexity $O(NM^{\omega+1})$ with varying number of arms N.

higher increase by strategically assigning the limited service calls using the right objective in the learning method. The improvement is statistically significant (p-value < 0.05).

In the testing set, we examine the difference between those selected for service call in two-stage and DF-Whittle. We observe that there are some interesting differences. For example, DF-Whittle chooses to do service calls to expectant mothers earlier in gestational age (22% vs 37%), and to a lower proportion of those who have already given birth (2.8% vs 13%) compared to two-stage. In terms of the income level, there is no statistic significance between two-stage and DFL (p-value = 0.20 see Appendix B). In particular, 94% of the mothers selected by both methods are below the poverty line (World Bank 2020).

Impact of Limited Data Figure 4 shows the improvement between decision-focused learning and two-stage method with varying number of trajectories given to evaluate the impact of limited data. We notice that a larger improvement between decision-focused and two-stage learning is observed when fewer trajectories are available. We hypothesize that less samples implies larger predictive error and more discrepancy between the loss metric and the evaluation metric.

Computation cost comparison Figure 4a, compares the computation cost per gradient step of our Whittle index-based decision-focused learning and other baselines in decision-focused learning (Wang et al. 2021; Futoma, Hughes, and Doshi-Velez 2020) by changing N (the number of arms) in $M = 2$-state RMAB problem. The other baselines fail to run with $N = 30$ arms and do not scale to larger problems like maternal and child care with more than 600 people enrolled, while our approach is 100x faster than the baselines as shown in Figure 4a and with a linear dependency on the number of arms N.

In Figure 4b, we compare the empirical computation cost of our algorithm with the theoretical computation complexity $O(NM^{\omega+1})$ in N arms and M states RMAB problems. The empirical computation cost matches with the linear trend in N. Our computation cost significantly improves the computation cost $O(M^{\omega N})$ of previous work as discussed in Section 4.5.

8 Conclusion

This paper presents the first decision-focused learning in RMAB problems that is scalable for large real-world datasets. We establish the differentiability of Whittle index policy in RMAB by providing new method to differentiate through Whittle index and using soft-top-k to relax the arm selection process. Our algorithm significantly improves the performance and scalability of decision-focused learning, and is scalable to real-world RMAB problem sizes.

References

Agrawal, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, Z. 2019. Differentiable convex optimization layers. *arXiv preprint arXiv:1910.12430*.

Akbarzadeh, N.; and Mahajan, A. 2019. Restless bandits with controlled restarts: Indexability and computation of Whittle index. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 7294–7300. IEEE.

Alman, J.; and Williams, V. V. 2021. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 522–539. SIAM.

Amos, B.; and Kolter, J. Z. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, 136–145. PMLR.

ARMMAN. 2022. ARMMAN Helping Mothers and Children. https://armman.org/. Accessed: 2022-05-19.

Bagheri, S.; and Scaglione, A. 2015. The restless multi-armed bandit formulation of the cognitive compressive sensing problem. *IEEE Transactions on Signal Processing*, 63(5): 1183–1198.

Bai, S.; Kolter, J. Z.; and Koltun, V. 2019. Deep equilibrium models. *arXiv preprint arXiv:1909.01377*.

Benamou, J.-D.; Carlier, G.; Cuturi, M.; Nenna, L.; and Peyré, G. 2015. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2): A1111–A1138.

Bubeck, S.; and Cesa-Bianchi, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*.

Dai, W.; Gai, Y.; Krishnamachari, B.; and Zhao, Q. 2011. The non-Bayesian restless multi-armed bandit: A case of near-logarithmic regret. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2940–2943. IEEE.

Donti, P. L.; Amos, B.; and Kolter, J. Z. 2017. Task-based end-to-end model learning in stochastic optimization. *arXiv preprint arXiv:1703.04529*.

Elmachtoub, A.; Liang, J. C. N.; and McNellis, R. 2020. Decision trees for decision-making under the predict-then-optimize framework. In *International Conference on Machine Learning*, 2858–2867. PMLR.

Elmachtoub, A. N.; and Grigas, P. 2021. Smart "predict, then optimize". *Management Science*.

Futoma, J.; Hughes, M. C.; and Doshi-Velez, F. 2020. Popcorn: Partially observed prediction constrained reinforcement learning. *arXiv preprint arXiv:2001.04032*.

Gittins, J.; Glazebrook, K.; and Weber, R. 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.

Glazebrook, K. D.; Ruiz-Hernandez, D.; and Kirkbride, C. 2006. Some indexable families of restless bandit problems. *Advances in Applied Probability*, 38(3): 643–672.

Huang, C.; Zhai, S.; Talbott, W.; Martin, M. B.; Sun, S.-Y.; Guestrin, C.; and Susskind, J. 2019. Addressing the loss-metric mismatch with adaptive loss alignment. In *International Conference on Machine Learning*, 2891–2900. PMLR.

Jiang, S.; Song, Z.; Weinstein, O.; and Zhang, H. 2020. Faster dynamic matrix inverse for faster lps. *arXiv preprint arXiv:2004.07470*.

Johnson, J. M.; and Khoshgoftaar, T. M. 2019. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1): 1–54.

Krishnan, S.; Garg, A.; Patil, S.; Lea, C.; Hager, G.; Abbeel, P.; and Goldberg, K. 2017. Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning. *The International Journal of Robotics Research*, 36(13-14): 1595–1618.

Lambert, N.; Amos, B.; Yadan, O.; and Calandra, R. 2020. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*.

Liu, H.; Liu, K.; and Zhao, Q. 2012. Learning in a changing world: Restless multiarmed bandit with unknown dynamics. *IEEE Transactions on Information Theory*, 59(3): 1902–1916.

Liu, K.; and Zhao, Q. 2010. Indexability of restless bandit problems and optimality of whittle index for dynamic multichannel access. *IEEE Transactions on Information Theory*, 56(11): 5547–5567.

Mandi, J.; Stuckey, P. J.; Guns, T.; et al. 2020. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1603–1610.

Mate, A.; Killian, J. A.; Xu, H.; Perrault, A.; and Tambe, M. 2020. Collapsing Bandits and Their Application to Public Health Intervention. In *NeurIPS*.

Mate, A.; Madaan, L.; Taneja, A.; Madhiwalla, N.; Verma, S.; Singh, G.; Hegde, A.; Varakantham, P.; and Tambe, M. 2022. Field Study in Deploying Restless Multi-Armed Bandits: Assisting Non-Profits in Improving Maternal and Child Health. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Monahan, G. E. 1982. State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms. *Management science*, 28(1): 1–16.

Oksanen, J.; and Koivunen, V. 2015. An order optimal policy for exploiting idle spectrum in cognitive radio networks. *IEEE Transactions on Signal Processing*, 63(5): 1214–1227.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1994. The complexity of optimal queueing network control. In *Proceedings of IEEE 9th Annual Conference on Structure in Complexity Theory*, 318–322. IEEE.

Perrault, A.; Wilder, B.; Ewing, E.; Mate, A.; Dilkina, B.; and Tambe, M. 2020. End-to-end game-focused learning of adversary behavior in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1378–1386.

Qian, Y.; Zhang, C.; Krishnamachari, B.; and Tambe, M. 2016. Restless poachers: Handling exploration-exploitation tradeoffs in security domains. In *Proceedings of the 2016*

International Conference on Autonomous Agents & Multiagent Systems, 123–131.

Ranchod, P.; Rosman, B.; and Konidaris, G. 2015. Nonparametric bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 471–477. IEEE.

Sun, Y.; Feng, G.; Qin, S.; and Sun, S. 2018. Cell association with user behavior awareness in heterogeneous cellular networks. *IEEE Transactions on Vehicular Technology*, 67(5): 4589–4601.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Tekin, C.; and Liu, M. 2012. Online learning of rested and restless bandits. *IEEE Transactions on Information Theory*, 58(8): 5588–5611.

Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts Libraries.

Wang, K.; Shah, S.; Chen, H.; Perrault, A.; Doshi-Velez, F.; and Tambe, M. 2021. Learning MDPs from Features: Predict-Then-Optimize for Sequential Decision Making by Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34.

Wang, K.; Yu, J.; Chen, L.; Zhou, P.; Ge, X.; and Win, M. Z. 2019. Opportunistic scheduling revisited using restless bandits: Indexability and index policy. *IEEE Transactions on Wireless Communications*, 18(10): 4997–5010.

Weber, R. R.; and Weiss, G. 1990. On an index policy for restless bandits. *Journal of applied probability*, 27(3): 637–648.

Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A): 287–298.

Wilder, B.; Dilkina, B.; and Tambe, M. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1658–1665.

World Bank, . 2020. *Poverty and shared prosperity 2020: Reversals of fortune*. The World Bank.

Xie, Y.; Dai, H.; Chen, M.; Dai, B.; Zhao, T.; Zha, H.; Wei, W.; and Pfister, T. 2020. Differentiable top-k operator with optimal transport. *arXiv preprint arXiv:2002.06504*.

Appendix

A Hyperparameter Setting and Computation Infrastructure

We run both Decision Focused Learning and Two-Stage Learning for 50 epochs in 2-state and 5-state synthetic domain problems, 30 epochs in ARMMAN domain and 18 epochs in 2-state partially observable setting. The learning rate r is kept at 0.01 and $\gamma = 0.59$ is used in all experiments. All the experiments are performed on an Intel Xeon CPU with 64 cores and 128 GB memory.

Neural Network Structure

The predictive model f_w we use to predict the transition probability is a neural network with an intermediate layer of size 64 with ReLU activation function, and an output layer of size of the transition probability followed by a softmax layer to match probability distribution. Dropout layers are added to avoid overfitting. The same neural network structure is applied to all domains and all training methods.

In the synthetic datasets, given the generated transition probabilities, we feed the transition probability of each arm into a randomly initialized neural network with two intermediate layers each with 64 neurons, and an output dimension size 16 to generate a feature vector of size 16. The randomly initiated neural network uses ReLU layers as nonlinearity followed by a linear layer in the end.

B Real ARMMAN Dataset

The large-scale quality improvement study conducted by ARMMAN (2022) contains 7668 beneficiries in the Round Robin Group. Over a duration of 7 weeks, 20% of the beneficiaries receive at least one active action (LIVE service call). We randomly split the 7668 beneficiaries into 12 groups while preserving the proportion of beneficiaries who received at least one active action. There are 43 features available for every beneficiary which describe characteristics such as age, income, education level, call slot preference, language preference, phone ownership etc.

B.1 Protected and Sensitive Features

ARMMAN's mobile voice call program has long been working with socially disadvantaged populations. ARMMAN does not collect or include constitutionally protected and particularly sensitive categories such as caste and religion. Despite such categories not being available, in pursuit of ensuring fairness, we worked with public health and field experts to ensure indicators such as education, and income levels that signify markers of socio-economic marginalization were measured and evaluated for fairness testing.

B.2 Feature List

We provide the full list of 43 features used for predicting transition probability:

- Enroll gestation age, age (split into 5 categories), income (8 categories), education level (7 categories), language (5 categories), phone ownership (3 categories), call slot preference (5 categories), enrollment channel (3 categories), stage of pregnancy, days since first call, gravidity, parity, stillbirths, live births

B.3 Feature Evaluation

Feature	Two-stage	Decision-focused learning	p-value
age (year)	25.57	24.9	0.06
gestation age (week)	24.28	17.21	0.00

Table 1: Feature analysis of continuous features. This table summarizes the average feature values of the beneficiaries selected to schedule service calls by different learning methods. The p-value of the continuous features is analyzed using t-test for difference in mean.

In our simulation, we further analyze the demographic features of participants who are selected to schedule service calls by either two-stage learning method and decision-focused learning method. The following tables show the average value of each individual feature over the selected participants with scheduled service calls under the two-stage or decision-focused learning method. The p-value of the continuous features is analyzed using t-test for difference in mean; the p-value of the categorical values is analyzed using chi-square test for different proportions.

In Table 1 and Table 2, we can see that there is no statistical significance (p-value > 0.05) between the average feature values of income and education, meaning that there is no obvious difference in these feature values between the population selected by two different methods. We see statistical significance in some other features, e.g., gestation age, stage of maternal event, language, phone ownership, and channel type, which may be further analyzed to understand the benefit of decision-focused learning, but they do not appear to directly bear upon socio-economic marginalization; these features are more related to the health status of the beneficiaries.

Feature	Two-stage	Decision-focused learning	p-value
income (rupee, averaged over multiple categories)	10560.0	11190.0	0.20
education (categorical)	3.32	3.16	0.21
stage of pregnancy	0.13	0.03	**0.00**
language			
language (hindi)	0.53	0.6	**0.04**
language (marathi)	0.45	0.4	0.08
phone ownership			
phone ownership (women)	0.86	0.82	**0.04**
phone ownership (husband)	0.12	0.16	**0.03**
phone ownership (family)	0.02	0.02	1.00
enrollment channel			
channel type (community)	0.7	0.47	**0.00**
channel type (hospital)	0.3	0.53	**0.00**

Table 2: Feature analysis of categorical features. This table summarizes the average feature values of the beneficiaries selected to schedule service calls by different learning methods. The p-value of the categorical values is analyzed using chi-square test for different proportions.

C Consent for Data Collection and Analysis

In this section, we provide information about consent related to data collection, analyzing data, data usage and sharing.

C.1 Secondary Analysis and Data Usage

This study falls into the category of secondary analysis of the aforementioned dataset. We use the previously collected engagement trajectories of different beneficiaries participating in the service call program to train the predictive model and evaluate the performance. The evaluation of the proposed algorithm is evaluated via different off-policy policy evaluations, including an importance sampling-based method and a simulation-based method discussed in Section 5. This paper does not involve deployment of the proposed algorithm or any other baselines to the service call program.As noted earlier, the experiments are secondary analysis using different evaluation metrics with approval from the ARMMAN ethics board.

C.2 Consent for Data Collection and Sharing

The consent for collecting data is obtained from each of the participants of the service call program. The data collection process is carefully explained to the participants to seek their consent before collecting the data. The data is anonymized before sharing with us to ensure anonymity. Data exchange and use was regulated through clearly defined exchange protocols including anonymization, read-access only to researchers, restricted use of the data for research purposes only, and approval by ARMMAN's ethics review committee.

C.3 Universal Accessibility of Health Information

To allay further concerns: this simulation study focuses on improving quality of service calls. Even in the intended future application, all participants will receive the same weekly health information by automated message regardless of whether they are scheduled to receive service calls or not. The service call program does not withhold any information from the participants nor conduct any experimentation on the health information. The health information is always available to all participants, and participants can always request service calls via a free missed call service. In the intended future application our algorithm may only help schedule *additional* service calls to help beneficiaries who are likely to drop out of the program.

D Societal Impacts and Limitations

D.1 Societal Impacts

The improvement shown in the real dataset directly reflects the number of engagements improved by our algorithm under different evaluation metrics. On the other hand, because of the use of demographic features to predict the engagement behavior, we must carefully compare the models learned by standard two-stage approach and our decision-focused learning to further examine whether there is any bias or discrimination concern.

Specifically, the data is collected by ARMMAN, an India non-government organization, to help mothers during their pregnancy. The ARMMAN dataset we use in the paper does not contain information related to race, religion, caste or other sensitive features; this information is not available to the machine learning algorithm. Furthermore, examination by ARMMAN staff of the mothers selected for service calls by our algorithm did not reveal any specific bias related to these features. In particular, the program run by ARMMAN targets mothers in economically disadvantaged communities; the majority of the participants

(94%) are below the international poverty line determined by The World Bank (World Bank 2020). To compare the models learned by two-stage and DF-Whittle approach, we further examine the difference between those mothers who are selected for service call in two-stage and DF-Whittle, respectively. We observe that there are some interesting differences. For example, DF-Whittle chooses to do service calls to expectant mothers earlier in gestational age (22% vs 37%), and to a lower proportion of those who have already given birth (2.8% vs 13%) compared to two-stage, but in terms of the income level, 94% of the mothers selected by both methods are below the poverty line. This suggests that our approach is not biased based on income level, especially when the entire population is coming from economically disadvantaged communities. Our model can identify other features of mothers who are actually in need of service calls.

D.2 Limitations

Impact of limited data and the strength of decision-focused learning As shown in Section 7 and Figure 4, we notice a smaller improvement between decision-focused learning and two-stage approach when there is sufficient data available in the training set. This is because the data is sufficient enough to train a predictive model with small predictive loss, which implies that the predicted transition probabilities and the true transition probabilities are also close enough with similar Whittle indices and Whittle index policy. In this case with sufficient data, there is less discrepancy between predictive loss and the evaluation metrics, which suggests less improvement led by fixing the discrepancy using decision-focused learning. Compared to two-stage approach, decision-focused learning is still more expensive to run. Therefore, when data is sufficient, two-stage may be sufficient to achieve comparable performance while maintaining a low training cost.

On the other hand, we notice a larger improvement between decision-focused learning and two-stage approach when data is limited. When data is limited, predictive loss is less representative with a larger mismatch compared to the evaluation metrics. Therefore, fixing the objective mismatch issue using decision-focused learning becomes more prominent. Therefore, decision-focused learning may be adopted in the limited data case to significantly improve the performance.

Computation cost As we have shown in Section 4.5, our approach improves the computation cost of decision-focused learning from $O(M^{\omega N})$ to $O(NM^{\omega+1})$, where N is the number of arms and M is the number of states. This computation cost is linear in the number of arms N, allowing us to scale up to large real-world deployment of RMAB applications with larger number of arms involved in the problem. Nonetheless, the extension in terms of the number of states M is not cheap. The computation cost still grows between cubic and biquadratic as shown in Figure 6. This is particularly significant when working on partially observable RMAB problems, where the partially observable problems are reduced to fully observable problems with larger number of states. There is room for improving the computation cost in terms of the number of states to make decision-focused learning more scalable to real-world applications.

E Computation Cost Analysis of Decision-focused Learning

We have shown the computation cost of backpropagating through Whittle indices in Section 4.5. This section covers the remaining computation cost associated to other components, including the computation cost of Whittle indices in the forward pass, and the computation cost of constructing soft Whittle index policy using soft-top-k operator.

E.1 Solving Whittle Index (Forward Pass)

In this section, we discuss the cost of computing Whittle index in the forward pass. In the work by Qian et al. (2016), they propose to use value iteration and binary search to solve the Bellman equation with M states. Therefore, every value iteration requires updating the current value functions of M states by considering all the possible M^2 transitions between states, which results in a computation cost of $O(M^2)$ per value iteration. The value iteration is run for a constant number of iterations, and the binary search is run for $O(\log \frac{1}{\epsilon})$ iterations to get a precision of order ϵ. In total, the computation cost is of order $O(M^2 \log \frac{1}{\epsilon}) = O(M^2)$ where we simply use a fixed precision to ignore the dependency on ϵ.

On the other hand, there is a faster way to compute the value function by solving linear program with M variables directly. The Bellman equation can be expressed as a linear program where all the M variables are the value functions. The best known complexity of solving a linear program with M variables is $O(M^{2+\frac{1}{18}})$ by Jiang et al. (2020). Notice that this complexity is slightly larger than the one in value iteration because (i) value iteration does not guarantee convergence in a constant iterations (ii) the constant associated to the number of value iterations is large.

In total, we need to compute the Whittle index of N arms and for M possible states in $\mathcal{S}$. The total complexity of value iteration and linear program are $O(NM^3)$ with a large constant and $O(NM^{3+\frac{1}{18}})$, respectively. In any cases, the cost of computing all Whittle indices in the forward pass is still smaller than $O(NM^{1+\omega})$, the cost of backpropagating through all the Whittle indices in the backward pass. Therefore, the backward pass is the bottleneck of the entire process.

E.2 Soft-top-k Operators

In Section E.1 and Section 4.5, we analyze the cost of computing and backpropagating through Whittle indices of all states and all arms. In this section, we discuss the cost of computing the soft Whittle index policy from the given Whittle indices using soft-top-k operators.

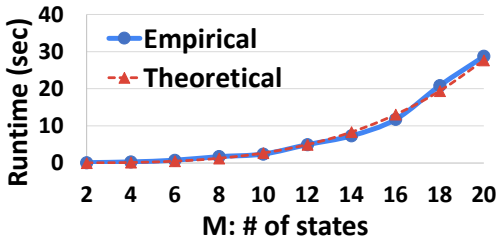


Figure 6: Computation cost comparison to the theoretical guarantee with varying number of states M.

Soft-top-k operators Xie et al. (2020) reduces top-k selection problem to an optimal transport problem that transports a uniform distribution across all input elements with size N to a distribution where the elements with the highest-k values are assigned probability 1 and all the others are assigned 0.

This optimal transport problem with N elements can be efficiently solved by using Bregman projections (Benamou et al. 2015) with complexity $O(LN)$, where L is the number of iterations used to run Bregman projections. In the backward pass, Xie et al. (2020) shows that the technique of differentiating through the fixed point equation (Bai, Kolter, and Koltun 2019; Amos and Kolter 2017) also applies, but the naive implementation requires computation cost $O(N^2)$. Therefore, Xie et al. (2020) provides a faster computation approach by leveraging the associate rule in matrix multiplication to lower the backward complexity to $O(N)$.

In summary, a single soft-top-k operator requires $O(LN)$ to compute the result in the forward pass, and $O(N)$ to compute the derivative in the backward pass. In our case, we need to apply one soft-top-k operator for every time step in T and for every trajectory in $\mathcal{T}$. Therefore, the total computation cost of computing a soft Whittle index policy and the associated importance sampling-based evaluation metric is bounded by $O(LNT|\mathcal{T}|)$, which is linear in the number of arms N, but still significantly smaller than $O(NM^{\omega+1})$, the cost of backpropagating through all Whittle indices as shown in Section 4.5. Therefore, we just need to concern the computation cost of Whittle indices in decision-focused learning.

E.3 Computation Cost Dependency on the Number of States

Figure 6 compares the computation cost of our algorithm, DF-Whittle, and the theoretical computation cost $O(NM^{\omega+1}$. We vary the number of states M in Figure 6 and we can see that the computation cost of our algorithm matches the theoretical guarantee on the computation cost. In contrast to the prior work with computation cost $O(M^{\omega N})$, our algorithm significantly improves the computation cost of running decision-focused learning on RMAB problems.

F Importance Sampling-based Evaluations for ARMMAN Dataset with Single Trajectory

Unlike the synthetic datasets that we can produce multiple trajectories of an RMAB problem, in the real problem of service call scheduling problem operated by ARMMAN, there is only one trajectory available to us for every RMAB problem. Due to the specialty of the maternal and child health domain, it is unlikely to have the exactly same set of the pregnant mothers participating in the service call scheduling program at different times and under the same engagement behavior.

Given this restriction, we must evaluate the performance of a newly proposed policy using the only available trajectory. Unfortunately, the standard CWPDIS in Equation 12 does not work because the CWPDIS estimator is canceled out when there is only one trajectory:

$$\text{Eval}_{\text{IS}}(\pi, \mathcal{T}) = \sum_{t\in[T], i\in[N]} \gamma^{t-1} \frac{\mathbb{E}_{\tau\sim\mathcal{T}}\left[r_{t,i}\rho_{ti}(\tau)\right]}{\mathbb{E}_{\tau\sim\mathcal{T}}\left[\rho_{ti}(\tau)\right]} = \sum_{t\in[T], i\in[N]} \gamma^{t-1} \frac{r_{t,i}\rho_{ti}(\tau)}{\rho_{ti}(\tau)} = \sum_{t\in[T], i\in[N]} \gamma^{t-1} r_{t,i} \tag{13}$$

which is fixed regardless what target policy π is used and the associated importance sampling weights $\frac{\pi(a_{t,i}|s_t)}{\pi_{\text{beh}}(a_{t,i}|s_t)}$ and $\rho_{ti} = \prod_{t'=1}^{t} \frac{\pi(a_{t',i}|s_{t'})}{\pi_{\text{beh}}(a_{t',i}|s_{t'})}$. This implies that we cannot use CWPDIS to evaluate the target policy when there is only one trajectory.

Accordingly, we use the following variant to evaluate the performance:

$$\text{Eval}_{\text{IS}}(\pi, \mathcal{T}) = \sum_{i\in[N], t\in[T]} \gamma^{t-1} \frac{r_{t,i}\rho'_{ti}(\tau)}{\mathbb{E}_{t'\in[T]}\left[\rho'_{t'i}(\tau)\right]} \tag{14}$$

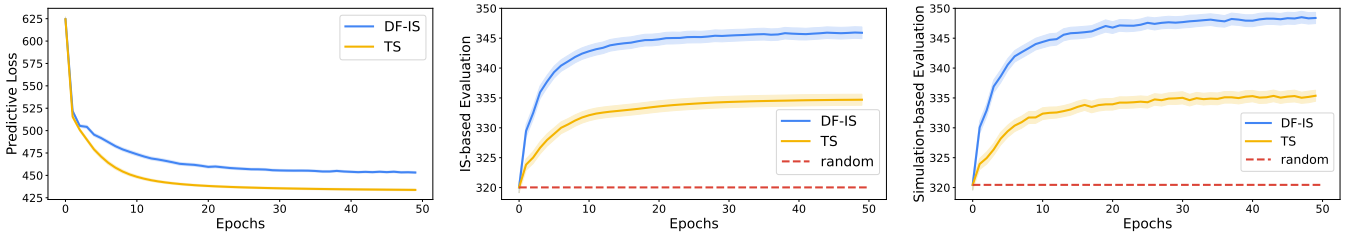
where the new importance sampling weights are defined by $\rho'_{t,i}(\tau) = \frac{\pi(a_{t,i}|s_t)}{\pi_{\text{beh}}(a_{t,i}|s_t)}$, which is not multiplicative compared to the original ones.

The main motivation of this new evaluation metric is to segment the given trajectory into a set of length-1 trajectories. We can apply CWPDIS to the newly generated length-1 trajectories to compute a meaningful estimate because we have more than

one trajectory now. The OPE formulation with segmentation is under the assumption that we can decompose the total reward into the contribution of multiple segments using the idea of trajectory segmentation (Krishnan et al. 2017; Ranchod, Rosman, and Konidaris 2015). This assumption holds when all segments start with the same state distribution. In our ARMMAN dataset, the data is composed of trajectories of the participants who have enrolled in the system a few weeks ago, which have (almost) reached a stationary distribution. Therefore, the state distribution under the behavior policy, which is a uniform random policy, does not change over time. Our assumption of identical distribution is satisfied and we can decompose the trajectories into smaller segments to perform evaluation. Empirically, we noticed that this temporal decomposition helps define a meaningful importance sampling-based evaluation with the consistency benefit brought by CWPDIS.

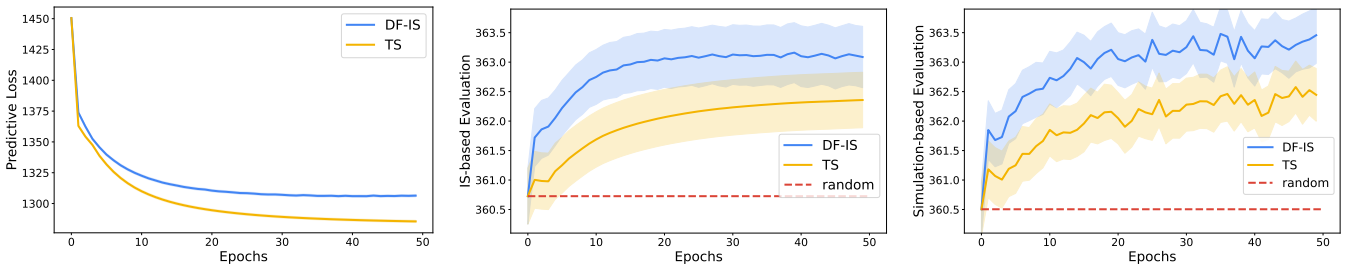
G Additional Experimental Results

We provide the learning curves of fully observable 2-state RMAB, fully observable 5-state RMAB, partially observable 2-state RMAB, and the real ARMMAN fully observable 2-state RMAB problems in Figure 7,8, 9, 10, respectively. Across all domains, two-stage method consistently converges to a lower predictive loss faster than decision-focused learning in Figure 7a,8a, 9a, 10a. However, the learned model does not produce a policy with good performance in the importance sampling-based evaluation metric in Figure 7b,8b, 9b, 10b, and similarly in the simulation-based evaluation metric in Figure 7c,8c, 9c, 10c.



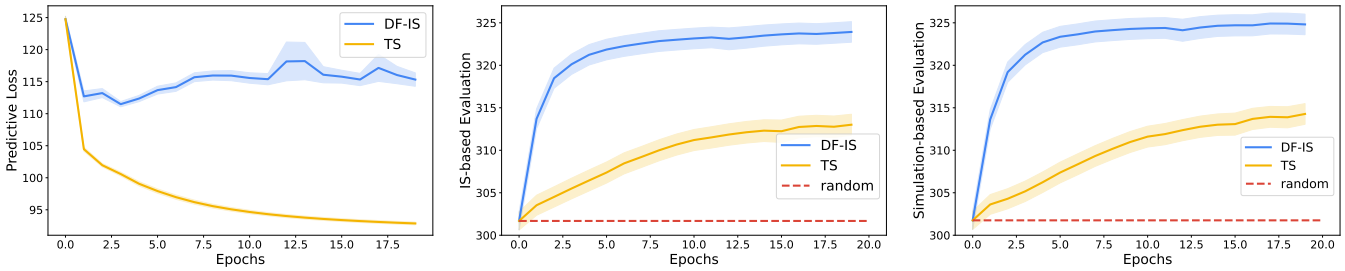
a Testing predictive loss v.s. training epoch b Testing IS-based evaluation c Testing simulation-based evaluation

Figure 7: Comparison between two-stage and decision-focused in the synthetic fully observable 2-state RMAB problems.



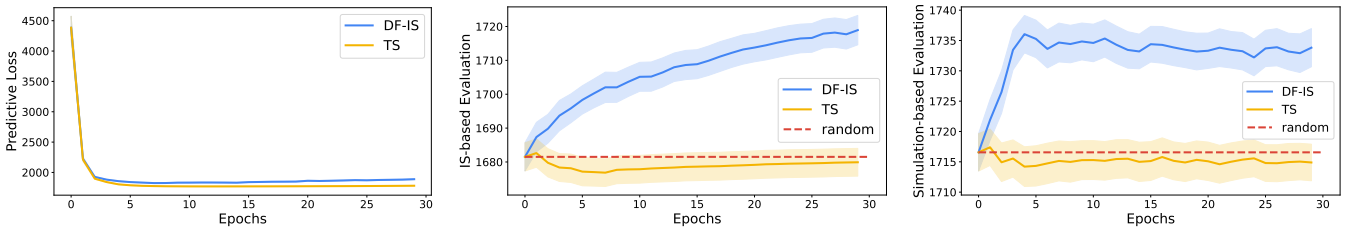
a Testing predictive loss v.s. training epoch b Testing IS-based evaluation c Testing simulation-based evaluation

Figure 8: Comparison between two-stage and decision-focused learning for fully observable 5-state RMAB problems.



a Testing predictive loss v.s. training epoch b Testing IS-based evaluation c Testing simulation-based evaluation

Figure 9: Comparison between two-stage and decision-focused learning for 2-state partially observable RMAB problems.



a Testing predictive loss

b Testing IS-based evaluation

c Testing simulation-based evaluation

Figure 10: Comparison between two-stage and decision-focused learning in the real ARMMAN service call scheduling problem. The pulling action in the real dataset is much sparser, leading to a larger mismatch between predictive loss and evaluation metrics. Two-stage overfits to the predictive loss drastically with no improvement in evaluation metrics. In contrast, decision-focused learning can directly optimize the evaluation metric to avoid the objective mismatch issue.

H Solving for and Differentiating Through the Whittle Index Computation

To solve for the Whittle index for some state $u \in S$, you have to solve the following set of equations:

$$V(u) = R(s) + m_u + \gamma \sum_{s' \in \mathcal{S}} P(s, 0, s') \cdot V(s')$$

$$V(u) = R(s) + \gamma \sum_{s' \in \mathcal{S}} P(s, 1, s') \cdot V(s') \tag{15}$$

$$V(s) = \max_{a \in \{0,1\}} [R(s) + (1-a)m_u + \gamma \sum_{s' \in \mathcal{S}} P(s, a, s') \cdot V(s')], \qquad \forall s \in \mathcal{S} - u \tag{16}$$

Here:

$\mathcal{S}$	is the set of all states
$R(s)$	is the reward for being in state s
$P(s, a, s')$	is the probability of transitioning to state s' when you begin in state s and take action a
$V(s)$	is the expected value of being in state s
m_s	is the whittle index for state s

One way to interpret these equations is to view them as the Bellman Optimality Equations associated with a modified MDP in which the reward function is changed to $R'_u(s, a) = R(s) + (1-a)m_u$, i.e., you are given a 'subsidy' for not acting (Equation 16). Then, to find the whittle index for state u, you have to find the minimum subsidy for which the value of not acting exceeds the value of acting (Whittle 1988). At this transition point, the value of not acting is equal to the value of acting in that state (Equation 15), leading to the set of equations above.

Now, this set of equations is typically hard to solve because of the $\max$ terms in Equation 16. Specifically, knowing whether $\arg\max_a = 0$ or $\arg\max_a = 1$ for some state s is equivalent to knowing what the optimal policy is for this modified MDP; such equations are typically solved using Value Iteration or variations thereof. However, this problem is slightly more complicated than a standard MDP because one also has to determine the value of m_s. The way that this problem is traditionally solved in the literature is the following:

1. One guesses a value for the subsidy m_s.
2. Given this guess, one solves the Bellman Optimality Equations associated with the modified MDP.
3. Then, one checks the resultant policy. If it is more valuable to act than to not act in state s, the value of the guess for the subsidy is increased. Else, it is decreased.
4. Go to Step 2 and repeat until convergence.

Given the monotonicity and the ability to bound the values of the whittle index, Step 3 above is typically solved using binary search. However, even with Binary Search, this process is quite time-consuming.

In this paper, we provide a much faster solution method for our application of interest. We leverage the small size of our state space to search over the space of policies rather than over the correct value of m_s. Concretely, because $S = \{0, 1\}$ and

$A = \{0, 1\}$, the whittle index equations for state $s = 0$ above boil down to:

$$V(0) = R(0) + m_{s_0} + \gamma \sum_{s' \in \{0,1\}} P(0, 0, s') \cdot V(s')$$
$$V(0) = R(0) + \gamma \sum_{s' \in \{0,1\}} P(0, 1, s') \cdot V(s')$$
$$V(1) = \max_{a \in \{0,1\}} [R(1) + (1 - a)m_{s_0} + \gamma \sum_{s' \in \{0,1\}} P(1, a, s') \cdot V(s')] \tag{17}$$

These are 3 equations in 3 unknowns $(V(0), V(1), m_{s_0})$. The only hiccup here is that Equation 17 has a $\max$ term and so this set of equations can not be solved as normal linear equations would be. However, we can 'unroll' Equation 17 into 2 different equations:

$$V(1) = R(1) + m_{s_0} + \gamma \sum_{s' \in \{0,1\}} P(1, 0, s') \cdot V(s'), \qquad \text{or} \tag{18}$$
$$V(1) = R(1) + \gamma \sum_{s' \in \{0,1\}} P(1, 1, s') \cdot V(s') \tag{19}$$

Each of these corresponds to evaluating the value function associated with the partial policies $s = 1 \rightarrow a = 0$ and $s = 1 \rightarrow a = 1$. Then to get the optimal policy, we can just evaluate both of the policies and choose the better of the two policies, i.e., the policy with the higher expected value $V(1)$. In practice, we pre-compute the Whittle index and value function using the binary search and value iteration approach studied by Qian et al. (2016). Therefore, to determine which equation is satisfied, we just use the pre-computed value functions to evaluate the expected future return of different actions, and use the one with higher value to form a set of linear equations.

This gives us a set of linear equations where Whittle index is a solution. We can therefore derive a closed-form expression of the Whittle index as a function of the transition probabilities, which is differentiable. This completes the differentiability of Whittle index. This technique is equivalent to saying that the policy does not change if we infinitesimally change the input probabilities.

H.1 Worked Example

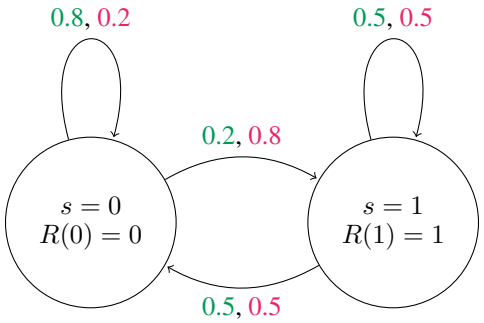


Figure 11: An MDP with the probabilities associated with the passive action $a = 0$ in red and active action $a = 1$ in green.

Let us consider the concrete example in Figure 11 with $\gamma = 0.5$. To calculate the whittle index for state $s = 0$, we have to solve the following set of linear equations:

$$V(0) = 0 + m_{s_0} + 0.5 \cdot [0.8V(0) + 0.2V(1)]$$
$$V(0) = 0 + 0.5 \cdot [0.2V(0) + 0.8V(1)]$$
$$V(1) = 1 + m_{s_0} + 0.5 \cdot [0.5V(0) + 0.5V(1)]$$

$$V(0) \approx 0.65, V(1) \approx 1.45, m_{s_0} \approx 0.25$$

$$V(0) = 0 + m_{s_0} + 0.5 \cdot [0.8V(0) + 0.2V(1)]$$
$$V(0) = 0 + 0.5 \cdot [0.2V(0) + 0.8V(1)]$$
$$V(1) = 1 + 0.5 \cdot [0.5V(0) + 0.5V(1)]$$

$$V(0) \approx 0.52, V(1) \approx 1.18, m_{s_0} \approx 0.20$$

Here the left set of equations corresponds to taking action $a = 0$ in state $s = 1$ and the right corresponds to taking the action $a = 1$. As we can see in the above calculation, given subsidy m_{s_0}, it is better to choose the passive action (a=0) on the left to obtain a higher expected future value $V(1)$. On the other hand, this can also be verified by precomputing the Whittle index and

the value function. Therefore, we know that the passive action in Equation 19 leads to a higher value, where the equality holds. Thus we can express the Whittle index as a solution to the following set of linear equations:

$$V(0) = R(0) + m_{s_0} + \gamma \sum_{s' \in \{0,1\}} P(0, 0, s') \cdot V(s')$$
$$V(0) = R(0) + \gamma \sum_{s' \in \{0,1\}} P(0, 1, s') \cdot V(s')$$
$$V(1) = R(1) + m_{s_0} + \gamma \sum_{s' \in \{0,1\}} P(1, 0, s') \cdot V(s')$$

By solving this set of linear equation, we can express the Whittle index m_{s_0} as a function of the transition probabilities. Therefore, we can apply auto-differentiation to compute the derivative $\frac{dm_{s_0}}{dP}$.