# Flexible Budgets in Restless Bandits: A Primal-Dual Algorithm for Efficient Budget Allocation

**Paula Rodriguez-Diaz**[1], **Jackson A. Killian**[1], **Lily Xu**[1]
**Arun Sai Suggala**[2], **Aparna Taneja**[2], **Milind Tambe**[1,2]
[1]Harvard University
[2]Google Research
{prodriguezdiaz, jkillian, lily_xu}@g.harvard.edu
{arunss,aparnataneja}@google.com, milind_tambe@harvard.edu

## Abstract

Restless multi-armed bandits (RMABs) are an important model to optimize allocation of limited resources in sequential decision-making settings. Typical RMABs assume the budget — the number of arms pulled — to be fixed for each step in the planning horizon. However, for realistic real-world planning, resources are not necessarily limited at each planning step; we may be able to distribute surplus resources in one round to an earlier or later round. In this paper we define a general class of *RMABs with flexible budget*, which we term *F-RMABs*, and provide an algorithm to optimally solve for them. We derive a min-max formulation to find optimal policies for F-RMABs and leverage gradient primal-dual algorithms to solve for reward-maximizing policies with flexible budgets. We introduce a scheme to sample expected gradients to apply primal-dual algorithms to the F-RMAB setting and make an otherwise computationally expensive approach tractable. Additionally, we provide heuristics that trade off solution quality for efficiency and present experimental comparisons of different F-RMAB solution approaches.

## 1 Introduction

Restless multi-armed bandits (RMABs), a model for constrained resource allocation among $N$ evolving and independent processes, are gaining increased attention, in part for their ability to capture challenging real-world planning problems. Salient examples include scheduling (Bagheri and Scaglione 2015; Yu, Xu, and Tong 2018; Yang et al. 2018), machine replacement (Ruiz-Hernández, Pinar-Pérez, and Delgado-Gómez 2020), aerial vehicle routing (Zhao, Krishnamachari, and Liu 2008), anti-poaching patrol planning (Qian et al. 2016), and healthcare (Lee, Lavieri, and Volk 2019; Mate et al. 2020; Killian et al. 2021; Biswas et al. 2021).

Most previous literature in RMABs assumes that resource constraints are fixed at each step in the planning horizon, i.e., there are a fixed maximum number of arms we can pull in each round. In some real-world settings, resources are not strictly constrained at each planning round but rather over multiple time steps. Existing RMAB planning techniques are therefore unable to take advantage of such flexibility in planning. Accordingly, applying the classic RMAB model in flexible-budget real-world settings may result in policies that either do not make efficient use of resources or lead to sub-optimal rewards.

We consider a general class of RMABs which we call *flexible budget restless multi-armed bandits*, or F-RMABs. In an F-RMAB instance, rather than the standard per-round budget constraint, the total resources that can be used is budgeted over some time *window*. The classic RMAB is a special case of the F-RMAB where the flexible window is a single timestep, so a planner may act on some subset of the $N$ arms such that the total cost of acting is less than or equal to $B$. In the F-RMAB class, the per-round budget may be flexible over a time window of length $F$ within the horizon $H$, where $F \leq H$, but the total cost of all actions *over that flexible window* must be less than or equal to $FB$, to preserve the per-round budget constraint on average.

Solving an RMAB is PSPACE-hard in general (Papadimitriou and Tsitsiklis 1999). To overcome this complexity, a common approach is to consider the Lagrangian relaxation of the problem in which the budget constraint is dualized (Hawkins 2003). Solving the relaxed problem gives Lagrange multipliers which act as a greedy index heuristic, known as the *Whittle index*, to solve the original problem. We show that introducing per-round budget variables $b_t$ and total budget limit of $FB$ over a flexible time window $F$ and performing Lagrangian relaxation results in a min-max problem that upper bounds the original problem, can be solved efficiently using primal-dual algorithms that we provide, and performs well in practice.

To summarize, our key contributions in this paper are: (i) We define the F-RMAB model; (ii) we provide an algorithm to compute well-performing F-RMAB policies; (iii) we introduce heuristics that trade off solution quality for efficiency; and (iv) we experimentally compare different F-RMAB solution approaches and show that our approach achieves an increase in reward of up to 24%, 72%, and 11% respectively for the three synthetic domains tested.

## 2 Related Work

Restless multi-armed bandits, introduced by Whittle (1988), are known to be PSPACE-hard in their full generality (Papadimitriou and Tsitsiklis 1994), but have a well-studied heuristic solution known as the Whittle index policy, which is asymptotically optimal (Weber and Weiss 1990). Many works study the Whittle index policy across a wide variety of application settings (Mate et al. 2020; Biswas et al. 2021; Lee, Lavieri, and Volk 2019). However, all of the settings where Whittle index policy has been studied, including multi-action settings (Hodge and Glazebrook 2015; Hawkins 2003; Killian, Perrault, and Tambe 2021) assume a fixed budget for all rounds in the planning horizon.

A specific form of flexibility, i.e., expected budget constraints over complete horizons, has been studied under the lens of Constrained Markov Decision Processes (CMDPs). However, standard CMDP techniques (Altman 1999) have exponential complexity for RMABs. CMDPs with weak coupling have been studied to address this (Boutilier and Lu 2016) but consider only one resource constraint over the total horizon. Our formulation newly addresses settings where the resource constraints are defined over shorter time periods than the complete planning horizon, e.g., planning one year of weekly interventions with per-month budget constraints.

Flexibility in decision making, including resource flexibility, budget design and flexible strategies, have shown to be useful for manufacturing processes and decision making in production flow control (Benjaafar, Morin, and Talavage 1995; Tomlin and Wang 2005; Boyabatlı, Leng, and Toktay 2016). However, while these models aim to compute policies that are flexible in the face of new information, they are not required to satisfy an overall resource constraint.

On the contrary, in our work, while the planner will take steps to be flexible in the face of new information, e.g., after observing the given state of arms, they must make choices that reason over multiple timesteps to avoid violating the multi-step budget constraint, making the problem more challenging.

In this sense, our work also relates to a broader literature on optimization with look-ahead (Wu and Frazier 2019; Lam and Willcox 2017; Atkinson 1994; Shmueli and Feitelson 2003), but these methods both do not consider multi-step budget constraints and scale poorly in the length of the time horizon, whereas our method is linear in the flexible horizon.

# 3 Flexible Budget RMABs

Here, we define *restless multi-armed bandits with flexible budget* (F-RMABs) and provide algorithms to solve for reward maximizing policies in this setting. In §3.1 we give a background on classic RMABs and in §3.2 we define F-RMABs as a general class of RMABs with flexible per-round budget. We also quantify the benefits of F-RMAB policies for two-state MDPs in §3.3.

## 3.1 Background: Restless Multi-Armed Bandits

An RMAB instance consists of $N$ independent Markov Decision Processes (MDPs), each corresponding to an arm of the instance (Puterman 2014). Each MDP is defined by the tuple $\{\mathcal{S}, \mathcal{A}, R, \mathcal{P}\}$. $\mathcal{S}$ denotes the state space, $\mathcal{A}$ is the set of possible actions, $R$ is the reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, and $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the transition function. We use $P^a_{s,s'}$ to denote the probability of transitioning from state $s$ to state $s'$ under the action $a$.

We let $s^t = [s^t_1, s^t_2 \dots s^t_N]$ denote the vector of states of the $N$ MDPs at time step $t$. A policy is a mapping $\pi^t : \mathcal{S}^N \to \mathcal{A}^N$ that informs the action to take at a given state, at time step $t$. We consider the more general multi-action case in which $|\mathcal{A}| \geq 2$ and define an action-cost matrix $\boldsymbol{c}$ of size $N \times |\mathcal{A}|$, i.e., $c_{nj}$ is the cost of taking action $j \in \mathcal{A}$ on arm $n$. Let $\mathbb{1}_{\pi^t(s^t)}$ be the one-hot encoder of size $N \times |\mathcal{A}|$, where each row $n$ indicates which action to perform on arm $n$ at time step $t$. The planner's goal is to find reward maximizing policies $\{\pi^t\}^H_{t=1}$ under the budget constraint $\mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq B$ for each $t \in [H]$. Here $H$ is the horizon length and $\cdot$ is the Frobenius inner product.

The total reward accrued can be measured using discounted, average, or total reward criteria in the finite- or infinite-horizon settings; we consider the total reward criterion in the finite-horizon setting, which enables the clearest analysis of our method. The expected *total reward* from initial state $s^0$ is defined as $V^1_\pi(s^0) = \mathbb{E}\left[\sum^H_{t=1} \sum^N_{n=1} R(s^{t-1}_n, [\pi^t(s^{t-1})]_n, s^t_n)\right]$ where the next state is drawn according to $s^t_n \sim P^{[\pi^t(s^{t-1})]_n}_{s^{t-1}_n, s^t_n}$. The planner's goal is to find policies $\pi = \{\pi^t\}^H_{t=1}$ that maximize the total reward.

## 3.2 Definition

In F-RMABs, we define the MDP followed by each arm using the tuple $\{\mathcal{S}, \mathcal{A}, R, \mathcal{P}\}$ just as in the classic RMAB setting. We now consider a flexible-budget time window of length $F$ where $F \leq H$. Our goal is to find optimal policies $\{\pi^t\}^H_{t=1}$ such that $\sum^F_{t=1}(\mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c}) \leq FB$ and $\mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq B$ for $t = F + 1, \dots, H$. That is, we consider an exhaustible budget $FB$ that is available to spend over the flexible window $1, \dots, F$ and think of $B$ as the one-step budget at every time step $t$ after the flexible window $t = F + 1, \dots, H$.

## 3.3 Benefit of Flexible Budgets

We present theoretical results illustrating the benefits of flexible budget RMABs by considering a *two-state process* model, which has been previously used in many real-world problems such as treatment adherence for tuberculosis (Mate et al. 2022), intervention planning for maternal health (Biswas et al. 2021), and multichannel access/scheduling problems (Sombabu et al. 2020) (see Fig. 7 in Appendix). In this process, we model the MDP followed by each arm using two states, *good* ($s = 1$) and *bad* ($s = 0$). We consider a binary action space $\mathcal{A} = \{0, 1\}$ and reward $R(s, a, s') = s'$. We further assume the MDPs representing all arms have the same transition probabilities: $\mathcal{P}(0, 0, 0) = 1, \mathcal{P}(0, 1, 0) = 0, \mathcal{P}(1, 0, 0) = p_{10}, \mathcal{P}(1, 1, 0) = 0$. Here, $p_{10} \in (0, 1)$ is the problem hyper-parameter. Despite the simplicity of this setting, analyzing RMABs and F-RMABs turns out to be non-trivial.

**Theorem 1.** *Suppose $F = H$, $H \to \infty$ and $p_{10} \geq N^{-1/2}$. Moreover, suppose the cost of playing action $0$ is $0$ and action $1$ is $1$, and suppose the one-step budget $B = \left(\frac{(1+o(1))p_{10}}{1+p_{10}}\right) N$. Define normalized cumulative re-*

*ward as* $\frac{1}{NH}\sum_{n=1}^{N}\sum_{t=1}^{H}\mathbb{E}\left[R(s_n^{(t-1)}, a_n^{(t-1)}, s_n^{(t)})\right]$*, where* $s_n^{(t)}, a_n^{(t)}$ *is the state and action of arm* $n$ *at time* $t$*. Let* $R_*^{\textit{F-RMAB}}$ *and* $R_*^{\textit{RMAB}}$ *be the maximum* normalized cumulative rewards *that can be achieved under the budget constraints imposed by F-RMAB and RMAB. Then,*

$$R_*^{\textit{F-RMAB}} \geq \frac{1-o(1)}{1+p_{10}}, \qquad R_*^{\textit{RMAB}} \leq \frac{1-c}{1+p_{10}}.$$

*Here* $o(1)$ *goes down to* $0$ *as* $H \to \infty$*.* $c > 0$ *is a positive constant that doesn't depend on* $H$*.*

The above Theorem proves for the first time the existence of RMABs in which, when given flexibility over the amount of resources to allocate at each decision step, one can design policies that outperform optimal fixed-budget policies.

## 4 Solving Under Flexible Budgets

Existing RMAB solution approaches require a fixed budget per round, leading to suboptimal performance. To make use of budget flexibility, we extend Lagrangian relaxation to the F-RMAB formulation and solve the resulting min-max problem with gradient algorithms. We also provide heuristics to solve F-RMABs that trade off solution quality for computational efficiency.

### 4.1 Concave-Convex Optimization

Recall, the budget constraint for F-RMABs over the flexible window is given by:

$$\sum_{t=1}^{F} \mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq FB, \tag{1}$$

where $\mathbb{1}_{\pi^t(s^t)}$ is the one-hot encoded matrix of size $N \times |\mathcal{A}|$, where each row $n$ indicates the action recommended by the policy $\pi^t$ on arm $n$, at time step $t$. Since this budget constraint is over multiple timesteps, formulating the optimal Bellman equation requires expanding the state space of the F-RMAB to capture the budget remaining after a given action is taken. However, this expansion adds an additional layer of combinatorial complexity over that involved in formulating the optimal Bellman equation for classic RMABs. Moreover, it is unclear how to relax this single budget constraint, which covers multiple timesteps, in a convenient or informative manner.

We make the insight that Eq. 1 can be reformulated to the following equivalent constraint structure, which introduces per-round budget variables:

$$\mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq b_t \qquad \forall t \in \{1, \ldots, F\} \tag{2}$$

$$\sum_{t=1}^{F} b_t \leq FB\ . \tag{3}$$

We will show this reformulated set of constraints is much more convenient to solve. For this constraint structure, each $b_t$ for $t \in \{1, \ldots, F\}$ is a variable that we must solve for in the original maximization problem. The key idea is that having a constraint in each round of the problem will allow us to follow a per-round Lagrangian relaxation, enabling us to convert the problem into a more tractable form.

Thus for the finite-horizon problem with total time horizon of length $H$ and flexible time window of length $F$, the F-RMAB problem can be formulated as the following optimization problem:

$$\max_{\substack{\pi^1,\ldots,\pi^H \\ b_1,\ldots,b_F}} \mathbb{E}\left[\sum_{t=1}^{H}\sum_{n=1}^{N} R(s_n^{t-1}, [\pi^t(s^{t-1})]_n, s_n^t)\right] \tag{4}$$

$$\text{s.t.} \quad \mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq b_t, \quad \forall t \in \{1, \ldots, F\} \tag{5}$$

$$\mathbb{1}_{\pi^t(s^t)} \cdot \boldsymbol{c} \leq B, \quad \forall t \in \{F+1, \ldots, H\} \tag{6}$$

$$\sum_{t=1}^{F} b_t \leq FB \tag{7}$$

Since the optimal policies for all arms are still coupled by budget constraints, this problem is still at least as hard as standard RMABs. However, we carry out a Lagrangian relaxation that gives a new problem that upper bounds Eq. 4, but is in a far more tractable form, as we show in Theorem 2.

**Theorem 2.** *The Lagrangian relaxation of Eq. 4 gives a new first-order primal-dual optimization problem which upper bounds Eq. 4 and has structure:*

$$\min_{x \in X}\max_{y \in Y} \langle Kx, y\rangle + G(x) - H^*(y), \tag{8}$$

*where* $X$ *and* $Y$ *are finite-dimensional vector spaces equipped with inner product* $\langle \cdot, \cdot \rangle$*.* $K : X \to Y$ *is a linear operator and* $G : X \to \mathbb{R} \cup \{\infty\}$ *and* $H^* : Y \to \mathbb{R} \cup \{\infty\}$ *are convex functions.*

*Proof.* Throughout the proof, we use the shorthand notation $a^t$ to denote a vector of actions $[a_1^t, \ldots a_N^t]$ taken at time step $t$. We let $\mathbb{1}_{a^t} \in \mathbb{R}^{N \times |\mathcal{A}|}$ denote the one-hot encoding of $a^t$, where the $n^{th}$ row encodes the action $a_n^t$. We first rewrite the objective in Eq. (4) as follows. Define $J^t(s^{t-1})$, the maximum expected reward from time steps $[t, H]$, starting from state $s^{t-1}$ as:

$$J^t(s^{t-1}) = \max_{a^t} \sum_{n=1}^{N} \mathbb{E}\left[R(s_n^{t-1}, a_n^t, s_n^t) + J^{t+1}(s^t)\right]$$

$$\text{s.t. } \mathbb{1}_{a^t} \cdot \boldsymbol{c} \leq \tilde{b}_t,$$

where we follow the convention $J^{H+1}(s^H) = 0$, and define $\tilde{b}_t = b_t$ if $t \leq F$, and $B$ otherwise. The objective in Eq. (4) can be expressed in terms of $J^1$ as

$$\max_{b_1 \ldots b_F} J^1(s^0) \quad \text{s.t.} \sum_{t=1}^{F} b_t \leq FB. \tag{9}$$

Then, by attaching Lagrange multipliers for constraints (5) and (6) to the recursive definition of the objective function, we get:

$$J^t(s^{t-1}) =$$

$$\max_{a^t}\min_{\lambda_{s^{t-1}} \geq 0} \quad \sum_{n=1}^{N} \mathbb{E}\left[R(s_n^{t-1}, a_n^t, s_n^t) + J^{t+1}(s^t)\right]$$

$$+\lambda_{s^{t-1}}(\tilde{b}_t - \mathbb{1}_{a^t} \cdot \boldsymbol{c}).$$

We now make two transformations that give us a new objective that upper bounds the original objective: (a) we swap the min and max in $J^t$ and in Eq. (9). Since $\min_x \max_y f(x,y) \geq \max_y \min_x f(x,y)$ for any $f$, this gives us a new problem that upper bounds Eq. (4), and (b) we enforce the Lagrangian multipliers to be constant across states: $\lambda_{s^{t-1}} = \lambda_{t-1}$. This constraint on $\lambda$ makes the problem much more tractable. We note that similar approximations have been studied in the literature of coupled dynamic systems (Hawkins 2003). This further allows decoupling per-arm value functions (Thm. 3, (Hawkins 2003)) giving:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \succeq 0} \max_{\boldsymbol{b}} \quad & \sum_{n=1}^{N} L_n^1(s_n^0; \boldsymbol{\lambda}) + \sum_{t=1}^{F} \lambda_t b_t + \sum_{t=F+1}^{H} \lambda_t B \\ \text{s.t.} \quad & \sum_{t=1}^{F} b_t \leq FB \end{aligned} \tag{10}$$

where $\boldsymbol{b} = (b_1, \ldots, b_F)$, and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_H)$ is the vector of Lagrange multipliers, one per time step, and

$$over. \tag{11}$$

Since $L_n^1(s_n^0; \boldsymbol{\lambda})$ is a supremum of linear functions in $\boldsymbol{\lambda}$, it is convex in $\boldsymbol{\lambda}$. So, the overall problem is a convex-concave min-max problem which can be solved efficiently. We now dualize the budget constraint over the flexible time window (constraint in Eq. 10). This gives us the following unconstrained min-max problem:

$$\begin{aligned} \min_{\boldsymbol{\lambda} \succeq 0, \mu \geq 0} \max_{\boldsymbol{b}} \Bigg\{ & \sum_{n=1}^{N} L_n^1(s_n^0; \boldsymbol{\lambda}) + \sum_{t=1}^{F} \lambda_t b_t \\ & + \sum_{t=F+1}^{H} \lambda_t B + \mu \Big( FB - \sum_{t=1}^{F} b_t \Big) \Bigg\}. \end{aligned} \tag{12}$$

What remains then is to show that Eq. 12 can be rewritten in the structure of Eq. 8. First, let $(\boldsymbol{\lambda}, \mu)$ be $x$ and let $\boldsymbol{b}$ be $y$. Now, let $K$ be such that $\langle K(\boldsymbol{\lambda}, \mu), \boldsymbol{b} \rangle = \sum_{t=1}^{F} (\lambda_t - \mu) b_t$ and define $H^\star(\boldsymbol{b}) = 0$, and

$$G(\boldsymbol{\lambda}, \mu) = \sum_{n=1}^{N} L_n^1(s_n^0; \boldsymbol{\lambda}) + \sum_{t=F+1}^{H} \lambda_t B + \mu FB \tag{13}$$

Then we can rewrite the min-max problem in Eq. 12 as

$$\min_{\boldsymbol{\lambda}, \mu} \max_{\boldsymbol{b}} \langle K(\boldsymbol{\lambda}, \mu), \boldsymbol{b} \rangle + G(\boldsymbol{\lambda}, \mu) - H^*(\boldsymbol{b}) \tag{14}$$

which gives us the claim. □

The key benefit of Theorem 2 is that, if $G$ is convex, there are efficient algorithms for solving optimization problems with this structure.

**Proposition 1.** *$G(\boldsymbol{\lambda}, \mu)$ is convex in $\boldsymbol{\lambda}$ and $\mu$.*

Now that we have demonstrated the underlying structure of our problem, in the next section we describe our approach for solving the Lagrangian sub-problem optimally, and using that to derive good policies for the F-RMAB.

## 4.2 Solving F-RMABs with a Gradient Algorithm

We now build our algorithm for solving F-RMABs. The first step is to solve Eq. 14. The key idea is that, for a given state in a given round, the solution will contain information about how budget within a flexible window would be best allocated, and what actions are best to take. We use that information to actually take actions each round in the environment.

We solve Eq. 14 by building from the proximal optimization method of Chambolle and Pock (2011), which is desirable for its convergence properties on concave-convex minmax optimization problems such as Eq. 14. The key challenge in implementing their approach is in efficiently computing the proximal steps.

Note first that the proximal operator (or proximal mapping) of a convex function $F$ is

$$\mathbf{prox}_{\sigma F}(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} \Big( F(\boldsymbol{u}) + \frac{1}{2\sigma} ||\boldsymbol{u} - \boldsymbol{x}||_2^2 \Big).$$

Following the notation in Chambolle and Pock (2011), $\mathbf{prox}_{\sigma F}(\boldsymbol{x}) = (\boldsymbol{I} + \sigma \partial F)^{-1}$. Then, the proximal operator of $H^\star(\boldsymbol{b})$ is $\mathbf{prox}^{\sigma}_{H^\star}(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} \Big( \frac{1}{2\sigma} ||\boldsymbol{u} - \boldsymbol{x}||_2^2 \Big) = \boldsymbol{x}$. Hence, the proximal operator of the zero function $H^\star$ is the identity. The proximal operator of $G$ does not have any analytical form. However, it is a piecewise-liear function. Since the proximal operator of linear functions is simply $\boldsymbol{x} - \sigma \nabla F(\boldsymbol{x})$, a good approximation of $\mathbf{prox}_{\sigma G}(\boldsymbol{x})$ is $x - \sigma \nabla G(x)$. Though an approximation, as we show next, computing $\nabla G$ is convenient, and performs well in practice.

**Proposition 2.** *The gradient of $G$ at $(\boldsymbol{\lambda}, \mu)$ is given by $\nabla G((\boldsymbol{\lambda}, \mu)) = [D^1, D^2, \ldots, D^F, D^{F+1} + B, \ldots, D^H + B, FB]$ where $D^t = \mathbb{E}[\sum_{n \in [N]} -c_n^t]$ is the expected sum of costs over all arms in step $t$ under the optimal policy for $\boldsymbol{\lambda}$.*

The main challenge then is in computing $D^t$ which has no convenient closed form. However, as long as we can compute the optimal policy $\pi^*(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda}$, we can get unbiased samples of each $D^t$ via Monte Carlo simulation of $\pi^*(\boldsymbol{\lambda})$.

Combining each of these steps, we have a complete algorithm for solving Eq. 14 to our desired level of convergence (Chambolle and Pock 2011). We name this approach, which includes our derived gradient sampling method, *primal-dual gradient sampling* (PDGS) and provide pseudocode in Algorithm 1.

## 4.3 Heuristics: Compressing $F$ Steps into One

The main difference between classic RMABs and F-RMABs is that the latter considers budget constraints over periods of length $F$ with $F \geq 1$ and the former considers budget constraints at each round ($F = 1$). Therefore, we propose a natural heuristic, which is not generally optimal but is well-performing. The *compress steps* heuristic first simplifies an F-RMAB to a classical multi-action RMAB by considering all possible sequences of binary actions in each window $F$ and forcing all $b_t = FB$, then computes policies using existing multi-action RMAB techniques (Hawkins 2003).

However, this baseline has no means to reason about flexible budget. To create such a baseline, we also develop a heuristic that reasons about the expected value of future

Algorithm 1: PDGS

**Input:** Flexible window $F$, horizon $H$, initial values $b^0 \in \mathbb{R}^F, \lambda^0 \in \mathbb{R}^H, \mu^0 \in \mathbb{R}$, gradient steps $\tau, \sigma > 0$, transition probability $P$, per-round budget $B$, and number of gradient samples $N_s$ for each state $s$

```
$\bar{\nu}^0 = [\boldsymbol{\lambda}^0, \mu^0]$
while not converged do
    $b^{n+1} = b^n + \sigma K \bar{\nu}^n$
    $\hat{\nu}^{n+1} = \nu^n - \tau K' b^{n+1}$ // $\hat{\nu}^{n+1} = [\hat{\boldsymbol{\lambda}}^{n+1}, \hat{\mu}^{n+1}]$
    $\pi^{n+1} = \text{FINITEHBELLMANLP}(P, \hat{\boldsymbol{\lambda}}^{n+1}, H)$ // LP to compute value func given $\lambda$.
    $\nabla G(\hat{\nu}^{n+1}) = \text{SAMPLEGRADS}(N_s, \pi^{n+1}, P, H, F)$
    $\nu^{n+1} = \hat{\nu}^{n+1} + \tau \nabla G(\hat{\nu}^{n+1})$
    $\bar{\nu}^{n+1} = 2\nu^{n+1} - \nu^n$
end while
```

Algorithm 2: SAMPLEGRADS

**Input**: Number of gradient samples $N_s$, policy $\pi : \mathcal{S}^N \rightarrow \mathcal{A}^N$, transition function $P$, planning horizon $H$, flexible time window $F$

```
for $i \in \{1, ..., N_s\}$ do
    $\boldsymbol{x}_i = (x_{1,i}, \dots, x_{H,i})$ = MonteCarlo($\pi, P$) // Simulate $\pi$ in environment $P$, return cost at $t \in [H]$
end for
$\boldsymbol{D} = \frac{1}{N_s} \sum_{i=1}^{N_s} \boldsymbol{x}_i$
return $[\boldsymbol{D}, BF]$ // $\nabla G$
```

chains of flexible actions for each arm. This allows us to use existing tools to encode a flexible budget, though this method is exponential in $F$, and thus not very scalable. These heuristics are presented in detail in Algorithms 4 and 5 in the Appendix.

Finally, given our method and the above two baselines, we decide which actions to take each round by solving the ACTIONKNAPSACK (see appendix B) which takes in the $Q$-values and budget for the given round. For our method, the budget is determined by PDGS. For COMPRESSSTEPS, the budget at each round is the entire remaining flexible budget. After a set of actions is taken, we step the environment forward, reduce the horizon by 1, and reduce the total flexible available by the amount of budget used in the last round. This guarantees that all methods spend the same total budget over all rounds in our evaluation.

## 4.4 Complexity Analysis

The computational complexity of PDGS has desirable scaling properties. It consists of three major steps, namely, the iterative updates, the computation of $\pi^\star$ and the gradient sampling, all inside the convergence loop. The updates involving multiplications of $K$ have complexity $\mathcal{O}(FH)$. Computing $\pi^\star$ depends on the computational complexity of building and solving an LP. Specifically, there is linear cost in setting up constraints and quadratic cost in the number of variables to solve (Jiang et al. 2020). The FINITEBELLMANLP has $NSH$ value function variables, $H$ Lagrange variables, and $NS^2HA$ constraints. That gives LP complexity of $\mathcal{O}(NS^2HA + N^2S^2H^2)$. Finally, SAMPLEGRADIENTS loops over all arms and timesteps, sampling from an $S$-length distribution for each next state, and so has complexity $\mathcal{O}(NSH \times N_{steps})$. Putting this all together, inside the convergence loop with $Z$ steps gives the following computational complexity for PDGS:

$$\mathcal{O}(Z(FH + NS^2HA + N^2S^2H^2 + NSHN_s)) \, . \quad (15)$$

Notably, the linear scaling in $F$ is desirable, especially compared to the COMPRESS heuristic which computes all possible sequences of actions of length $F$ and thus has exponential complexity of $\mathcal{O}(NFA^FS^3)$.

# 5 Experimental Evaluation

We evaluate the algorithms presented in sections 4.2 and 4.3 on three synthetic domains. Our results show these domains all benefit from per-round budget flexibility.

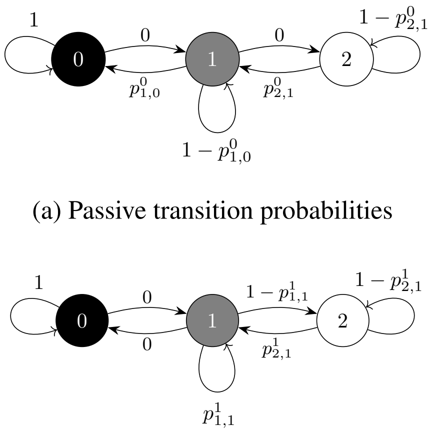
## 5.1 Domains Aided by Flexibility

**Dropout State** This first domain characterizes settings with potential urgent interventions, such as clinical health settings in which patients are likely to never return after dropping out of a program (Biswas et al. 2021). We consider three states: dropout ($s = 0$), at-risk ($s = 1$), and safe ($s = 2$). We consider a binary action set $\mathcal{A} = \{0, 1\}$ corresponding to a passive action ($a = 0$) and active action ($a = 1$). The reward function $R : \mathcal{S} \rightarrow \mathbb{R}$ is defined as $R(0) = 0$ and $R(1) = R(2) = 1$. Once an arm reaches the dropout state, it can not transition to any other state, i.e. $P^a_{0,0} = 1$ for all $a \in \mathcal{A}$. Fig. 1 illustrates the remaining active and passive transition probabilities in this domain.

In this dropout domain, interventions may be more urgent in certain rounds depending on the combined state of all arms. For example, if at time $t$, $k$ arms are at risk of transitioning to a dropout state, i.e. $k$ arms are in state 1 as shown in Figure 1, acting on these $k$ arms at $t$ is more urgent than acting on them at $t + 1$ or other near future steps. Thus this domain illustrates a key instance when the F-RMAB class introduces essential flexibility.

**Immediate Recovery** This setting models maintenance-style RMAB problems. Each arm corresponds to one item that gradually decays over time, and intervention is guaranteed to restore that item to peak condition (see Fig. 5 in Appendix). For example, a bikeshare program must maintain a fleet of bikes, and a mechanic could fully restore any bike, but only if that item has not decayed beyond the point of repair. There is a *good* state, a *dropout* state, and $S - 2$ *intermediate* states. An active action on all states other than $s = 0$ will transition back to the good state with probability 1. We consider the first half of the states, closer to the *dropout* state, to have reward $-1$ and the second half, closer to the *good* state, to have reward 1. If the amount of states is odd then we consider the median state to have reward 0.

**Two-state Process** The two-state process models approaches in health intervention planning such as maternal health care (Mate et al. 2022). This domain models an entity with two states, a *good* and a *bad* state, with reward



(a) Passive transition probabilities

(b) Active transition probabilities

Figure 1: Drop out state domain with three states: drop out ($s = 0$), risk (1), and safe (2). Passive transition probabilities are presented in Figure (a) and active transition probabilities are shown in Figure (b). We take $p^0_{0,0} \in [0.85, 0.95]$, $p^1_{0,0} = 0$, $p^0_{1,1} \in [0.35, 0.5]$ and $p^1_{1,1} = 1$ in our experiments.

$R(1) = 1$ for each arm in the good state and $R(0) = 0$ for the bad state. See Figure 7 in Appendix for a diagram of the model, with four transition probability parameters.

## 5.2 Baselines

We compare our algorithms against Lagrange-based policies for classic RMABs presented in Hawkins (2003): *Hawkins* 2003 computes values of Lagrange multipliers after relaxing per round budget constraints with fixed budget $B$, then solves a knapsack with budget $B$ per round over Q-values adjusted for the solved Lagrange multipliers. *Compress (static)* (Algorithm 5 in Appendix B) plans across the flexible window $F$ in each timestep and executes the first action before recomputing. *Compress (closing)* (Algorithm 4 in Appendix B) plans across a flexible window of size $F$, then $F - 1$, and so on until a window of size 1, repeating every $F$ steps.

Our method, *PDGS-N*, solves the primal-dual optimization problem presented in Theorem 2 using Algorithm 1 with $N$ iterations. Then solves a knapsack with the solved budget variables $b_t$ over Q-values adjusted for the solved Lagrange multipliers. Planning is done using the closing window framework.

## 5.3 Results

We test PDGS (Algorithm 1) and Compress heuristics (Algorithms 4 and 5 in appendix B) to solve for F-RMABs and compare them against a classic RMAB solution algorithm with fixed per round budget on the three domains described above. For each domain we consider a planning horizon of length $H$, an initial per round budget of $B = 1$, and vary the length of the flexible time window $F$.

In Fig. 2 [left] we see that optimal policies that allow for budget flexibility attain higher reward than optimal policies restricted to a fixed budget at every round. The *Hawkins* approach demonstrates the optimal reward achieved with a fixed budget. Our *PDGS* algorithm to solve for optimal policies in F-RMABs attains higher cumulative reward than Hawkins across all settings. Notably, PDGS progressively obtains higher rewards with longer flexible time windows, especially in the immediate recovery domain (Fig. 2(b)), demonstrating the additional planning power that can be gained with wider windows of flexibility.

As shown in Fig. 2(a), the flexible algorithms *Compress (closing)* and *PDGS-200* obtain a maximum increase in reward of 21.50% and 23.56% respectively for $F = 5$ compared to the reward obtained by *Hawkins*. This increase in reward is obtained by designing a more efficient allocation of resources over a planning horizon of length $H = 30$. This allocation is done by mostly assigning 0 to 2 resources at each step of the planning horizon, even for $F = 3$ and $F = 5$, in contrast to *Hawkins* which is restricted to use 1 resource at each step (see Fig. 6(a.1)–(a.3) in appendix D.3).

For the immediate recovery domain with $S = 5$ states (one dropout state, one fully recovered state, and three intermediate states), the average cumulative reward significantly increases as the length of the flexible window increases up to the point where the whole planning horizon is flexible, as shown in Fig. 2(b). *PDGS-200* attains an increase in average cumulative reward of 15.57%, 63.28% and 72% for $F = 2, 5, 10$ respectively compared to *Hawkins.PDGS* attains higher rewards by allocating up to 2 to 5 resources at different steps even for flexible windows as wide as $F = 10$ (see Fig. 6(b.1)–(b.3) in appendix D.3).

As shown in Fig. 2(c) for the two-state process domain, our method to solve for policies with flexible budget (*PDGS-200*) attains an increase in reward of 6.71%, 5.66%, and 11.32% for flexible time windows of length $F = 2, 3, 6$ respectively in contrast to the per round budget policy derived by *Hawkins*. We observe that RMABs can also benefit from flexibility in settings with as few as two states, which are relevant settings for health intervention planning, in contrast to the other two domains considering more than two states and having intermediate states that directly characterize waiting steps until reaching a bad state.

For flexible time windows as short as $F = 2$, our proposed heuristic on compressing from $F$ to 1 steps into one round by closing the flexible time window *Compress (closing)* (algorithm 4 in appendix B) attains almost as much reward as the near optimal algorithms (*PDGS*) under significantly smaller runtime (Fig. 2 [right]), implying that the exponential factor of $F$ is relatively negligible for such small windows. One limitation from *Compress (closing)* is that, when reasoning on what actions to take at $F - t$, i.e. when the flexible window is $t$ steps closed, it maximizes for the reward obtained at the last step of the current window and does not reason about the cumulative reward obtained at all intermediate steps of the window. This limitation explains the increase in gap between *Compress (closing)* and *PDGS-200* as $F$ increases, suggesting that for flexible windows as small as $F = 2$, *Compress (closing)* may be used to obtain almost as much reward as optimal algorithms under significantly less runtime as shown in Fig. 2 [right].

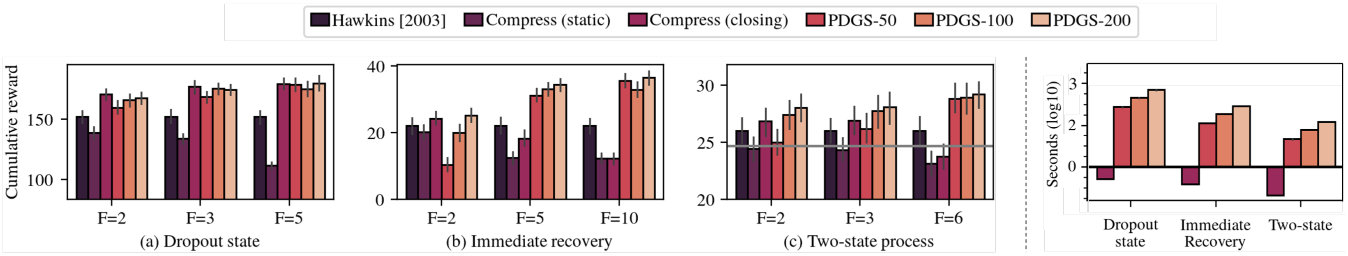


Figure 2: [Left] Cumulative reward for (a) dropout state with $H = 30$, $N = 10$, $B = 1$, (b) immediate recovery with 5 states, $H = 10$, $N = 10$, $B = 1$, and (c) two-state process with $H = 6$, $N = 10$, $B = 1$.The cumulative reward axis range in (a) and (b) starts at the average value for a policy taking $B$ random actions. The horizontal gray line in (c) denotes this same value for the two-step process domain. [Right] Runtime for flexible budget algorithms with $F = 2$. Plotted on a log scale.

### 5.4 Scale-Up Results

We extend our experimental results to settings with greater amount of arms ($N$) and longer time horizon ($H$). As seen in Fig. 3, flexible budget policies solved via PDGS obtain higher or equal reward than optimal fixed budget policies when scaling up the amount of arms to $N = 10, 20, 50$ for the three domains considered. The highest reward is attained at the maximum length of the flexible window ($F = 10$), where the complete planning horizon of length $H = 10$ is flexible.

In the dropout state and immediate recovery domains, PDGS's superior performance (see Fig. 3) shows that there continues to be benefit from flexibility as the number of arms increases. However, in the two state domain, the benefit from flexibility seems to decrease with the number of arms. Promisingly though, policies computed via PDGS attain equally as much reward as the optimal fixed budget policies solved with *Hawkins*, demonstrating its robustness. Moreover, PDGS has no significant increase in runtime when increasing the flexible time window $F$ as seen in Fig. 3 (bottom row). Most of the increase in runtime comes from increasing the amount of arms $N$ which we expected to result in a quadratic increase in runtime (see §4.4).

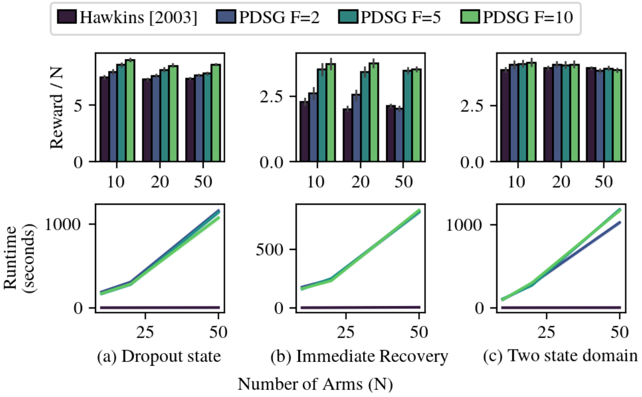


Figure 3: Average reward per arm (top row) and runtime in seconds (bottom row) for Hawkins and PDGS with flexible windows of length $F$, time horizon $H = 10$ and per-round budget $B = \frac{N}{10}$.

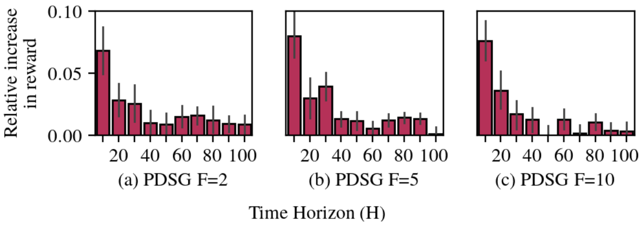


Figure 4: *Two-state process domain*. Relative increase in reward compared to Hawkins over total horizon $H$ for policy solved via PDGS.

Finally, we evaluate the performance of PDGS for longer time horizons reaching values where $F \ll H$ in the two-state process domain. PDGS consistently attains higher reward than Hawkins when extending the time horizon, showing that our proposed algorithm is able to arrive at policies that find benefit in flexibility even when $F \ll H$, e.g. $F = 2$ and $H = 100$ (see 4). However, the relative increase in reward decreases as $F/H$ decreases, suggesting wider lengths of flexibility could be considered for longer time horizons to obtain higher benefits from flexible budgets.

## 6 Conclusion

We introduce the flexible budget restless multi-armed bandits (F-RMAB) problem and derive a method, which we call *PDGS*, for optimally solving Lagrangian-relaxation in the F-RMAB formulation via a gradient primal-dual algorithm, which translates into a scalable approach for computing well-performing policies in this new domain. Further, we empirically verify that our method outperforms a suite of challenging baselines across a range of scenarios inspired by classic RMAB applications and real-world settings, emphasizing the additional planning power that F-RMABs represent over traditional RMAB methods. We define heuristics that trade off solution quality for efficiency in runtime. These heuristics perform almost as good as the optimal methods (*PDGS*) for small flexible time windows requiring significantly less runtime. We hope that this work further contributes to the real-world applicability of RMAB methods.

## Ethical Statement

We propose an algorithm to solve for reward-maximizing policies under flexible per round budget. The per round budget flexibility is solved under the assumption that there exists a flexible time window of length $F$ for which a planner can allocate $FB$ resources. Our approach is limited to consider complete flexibility over the flexible time window of length $F$, i.e. for each $t = 1, \dots, F$, the per round budget $b_t$ can take values from 0 to $FB$. This approach could be limiting in settings where it is not desirable to not spend any resource at some step or to spend all available resources at only one step. For instance, one could consider admitting up to $(\alpha, \beta)$–flexibility and imposing the additional constraint $\alpha FB \leq b_t \leq \beta FB$ where $\alpha, \beta \in [0, 1]$ and $\alpha < \beta$. However, solving the optimization problem in Eq. 4-7 with this additional constraint is non-trivial. Our algorithm takes a major step towards solving for reward-maximizing policies under real-world characteristics such as budget flexibility but is limited to complete flexibility over a given time window.

The characteristic on having budget flexibility has been thought as desirable from a planner's perspective. However, when the resources at hand are humans —human working time—, the capacity of human employees to be flexible with their working hours should also be taken into account. One ethical concern would be imposing flexibility that arises in negative impacts from having longer shifts, or being unfair among different workers in terms of their work load distribution. Accordingly, choosing a value for $F$ and $B$ that takes these considerations into account is an important ethical consideration for planners potentially using our tool. Additionally, imposing additional constraints such as the one described previously in this section could help tackle this ethical consideration.

## Acknowledgments

P. Rodriguez-Diaz was supported by the NSF under grant SES-2200228. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF. J.A. Killian was supported by an NSF Graduate Research Fellowship under grant DGE1745303. L. Xu was supported by a Google PhD Fellowship, and was a Student Researcher at Google while this work was completed.

## References

Altman, E. 1999. *Constrained Markov decision processes*. Stochastic modeling. Boca Raton ; London: Chapman & Hall/CRC. ISBN 978-0-8493-0382-1.

Atkinson, J. B. 1994. A greedy look-ahead heuristic for combinatorial optimization: An application to vehicle scheduling with time windows. *Journal of the Operational Research Society*, 45(6): 673–684.

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.

Benjaafar, S.; Morin, T. L.; and Talavage, J. J. 1995. The strategic value of flexibility in sequential decision making. *European Journal of Operational Research*, 82(3): 438–457.

Biswas, A.; Aggarwal, G.; Varakantham, P.; and Tambe, M. 2021. Learn to Intervene: An Adaptive Learning Policy for Restless Bandits in Application to Preventive Healthcare. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*. Number: arXiv:2105.07965 arXiv:2105.07965 [cs].

Boutilier, C.; and Lu, T. 2016. Budget Allocation using Weakly Coupled, Constrained Markov Decision Processes. In *UAI'16: Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*.

Boyabatlı, O.; Leng, T.; and Toktay, L. B. 2016. The Impact of Budget Constraints on Flexible vs. Dedicated Technology Choice. *Management Science*, 62(1): 225–244.

Chambolle, A.; and Pock, T. 2011. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1): 120–145.

Chung, F.; and Lu, L. 2006. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1): 79–127.

Gurobi Optimization, L. 2021. Gurobi Optimizer Reference Manual.

Hawkins, J. T. 2003. *A Langrangian decomposition approach to weakly coupled dynamic optimization problems and its applications*. Ph.D. thesis, Massachusetts Institute of Technology. Operations Research Center; Sloan School of Management.

Hodge, D. J.; and Glazebrook, K. D. 2015. On the asymptotic optimality of greedy index heuristics for multi-action restless bandits. *Advances in Applied Probability*, 47(3): 652–667.

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

Killian, J. A.; Biswas, A.; Shah, S.; and Tambe, M. 2021. Q-Learning Lagrange Policies for Multi-Action Restless Bandits. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 871–881. Virtual Event Singapore: ACM. ISBN 978-1-4503-8332-5.

Killian, J. A.; Perrault, A.; and Tambe, M. 2021. Beyond" To Act or Not to Act": Fast Lagrangian Approaches to General Multi-Action Restless Bandits. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 710–718.

Lam, R.; and Willcox, K. 2017. Lookahead Bayesian optimization with inequality constraints. *Advances in Neural Information Processing Systems*, 30.

Lee, E.; Lavieri, M. S.; and Volk, M. 2019. Optimal Screening for Hepatocellular Carcinoma: A Restless Bandit Model. *Manufacturing & Service Operations Management*, 21(1): 198–212.

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

Health Interventions. *Advances in Neural Information Processing Systems*. ArXiv: 2007.04432.

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 Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*.

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.

Papadimitriou, C. H.; and Tsitsiklis, J. N. 1999. The Complexity of Optimal Queuing Network Control. *Mathematics of Operations Research*, 24(2): 293–305.

Puterman, M. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qian, Y.; Zhang, C.; Bhaskar, K.; and Tambe, M. 2016. Restless Poachers: Handling Exploration-Exploitation Tradeoffs in Security Domains. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Ruiz-Hernández, D.; Pinar-Pérez, J. M.; and Delgado-Gómez, D. 2020. Multi-machine preventive maintenance scheduling with imperfect interventions: A restless bandit approach. *Computers & Operations Research*, 119: 104927.

Shmueli, E.; and Feitelson, D. G. 2003. Backfilling with lookahead to optimize the performance of parallel job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 228–251. Springer.

Sombabu, B.; Mate, A.; Manjunath, D.; and Moharir, S. 2020. Whittle index for AoI-aware scheduling. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 630–633. IEEE.

Tomlin, B.; and Wang, Y. 2005. On the Value of Mix Flexibility and Dual Sourcing in Unreliable Newsvendor Networks. *Manufacturing & Service Operations Management*, 7(1): 37–57.

Weber, R. R.; and Weiss, G. 1990. On an index policy for restless bandits. *J. Appl. Probab.*, 27(3): 637–648.

Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *J. Appl. Probab.*, 25(A).

Wielandt, H. 1950. Unzerlegbare, Nicht Negativen Matrizen. *Mathematische Zeitschrift*, 52(1): 642–648.

Wu, J.; and Frazier, P. 2019. Practical two-step lookahead Bayesian optimization. *Advances in neural information processing systems*, 32.

Yang, F.; Zhu, Z.; Zhao, S.; Yang, Y.; and Luo, X. 2018. Optimal task offloading in fog-enabled networks via index policies. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 688–692. Anaheim, CA, USA: IEEE. ISBN 978-1-72811-295-4.

Yu, Z.; Xu, Y.; and Tong, L. 2018. Deadline Scheduling as Restless Bandits. *IEEE Transactions on Automatic Control*, 63(8): 2343–2358.

Zhao, Q.; Krishnamachari, B.; and Liu, K. 2008. On myopic sensing for multi-channel opportunistic access: structure, optimality, and performance. *IEEE Transactions on Wireless Communications*, 7(12): 5431–5440.

# A Theory

## A.1 Proof of Proposition 1

**Proposition 1.** *$G(\boldsymbol{\lambda}, \mu)$ is convex in $\boldsymbol{\lambda}$ and $\mu$.*

*Proof.* $G(\boldsymbol{\lambda}, \mu)$ is given by

$$G((\boldsymbol{\lambda}, \mu)) = \sum_{n=1}^{N} L_n^1(s_n^0; \boldsymbol{\lambda}) + \sum_{t=F+1}^{H} \lambda_t B + \mu F B \ .$$

The second and third terms are linear in $\boldsymbol{\lambda}$ and $\mu$ and thus are convex. Each of the first terms, i.e., $L_n^1(s_n^0; \boldsymbol{\lambda})$ are a maximum over piece-wise linear convex functions of $\boldsymbol{\lambda}$ and thus are also piece-wise linear convex functions $\boldsymbol{\lambda}$. Since $G$ is a sum of functions that are convex in $(\boldsymbol{\lambda}, \mu)$, $G$ is also convex in $(\boldsymbol{\lambda}, \mu)$. □

## A.2 Proof of Theorem 1

**Theorem 1.** *Suppose $F = H$, $H \to \infty$ and $p_{10} \geq N^{-1/2}$. Moreover, suppose the cost of playing action $0$ is $0$ and action $1$ is $1$, and suppose the one-step budget $B = \left(\frac{(1+o(1))p_{10}}{1+p_{10}}\right) N$. Define normalized cumulative reward as $\frac{1}{NH}\sum_{n=1}^{N}\sum_{t=1}^{H} \mathbb{E}\left[R(s_n^{(t-1)}, a_n^{(t-1)}, s_n^{(t)})\right]$, where $s_n^{(t)}, a_n^{(t)}$ is the state and action of arm $n$ at time $t$. Let $R_*^{\textit{F-RMAB}}$ and $R_*^{\textit{RMAB}}$ be the maximum* normalized cumulative rewards *that can be achieved under the budget constraints imposed by F-RMAB and RMAB. Then,*

$$R_*^{\textit{F-RMAB}} \geq \frac{1 - o(1)}{1 + p_{10}}, \qquad R_*^{\textit{RMAB}} \leq \frac{1 - c}{1 + p_{10}}.$$

*Here $o(1)$ goes down to $0$ as $H \to \infty$. $c > 0$ is a positive constant that doesn't depend on $H$.*

*Proof.* Let $N_{t,0}, N_{t,1}$ be the number of arms in states $0, 1$ at time $t$ respectively. The normalized cumulative reward can be expressed in terms of $N_{t,0}, N_{t,1}$ as follows

$$\frac{1}{NH}\sum_{n=1}^{N}\sum_{t=1}^{H} \mathbb{E}\left[R(s_n^{(t-1)}, a_n^{(t-1)}, s_n^{(t)})\right] = \frac{1}{NH}\sum_{t=1}^{H} \mathbb{E}\left[N_{t,1}\right].$$

This follows from the following fact

$$R(s_n^{(t-1)}, a_n^{(t-1)}, s_n^{(t)}) = s_n^{(t)}.$$

**Lower bound on $R_*^{\textbf{F-RMAB}}$.** To derive a lower bound, we carefully construct a policy that satisfies the budget constraints imposed by F-RMAB and compute its normalized cumulative reward. Since $R_*^{\text{F-RMAB}}$ is the maximum achievable normalized cumulative reward, it should be lower bounded by the normalized cumulative reward of this policy.

At any round, our policy simply selects the arms that are in state 0 and plays action 1 for the arms. For the rest of the arms, the policy plays action 0. So, the budget $b_t$ used in round $t$ by this policy is equal to $N_{t,0}$. In the proof, we show that this policy satisfies the budget constraint of F-RMAB with high probability[1]

We first derive recurrences for how $N_{t,0}, N_{t,1}$ evolve with time for the policy defined above. Let $b_t$ be the budget used in round $t$ (note that $b_t = N_{t,0}$). From the definition of the MDP, we have

$$N_{t+1,1} = N_{t,0} + \text{Bin}(N - N_{t,0}, 1 - p_{10}),$$

where $\text{Bin}(N, p)$ is a Binomial random variable with $N$ trials and success probability of $p$. Rewriting this, we get

$$N_{t+1,1} = N - \text{Bin}(N_{t,1}, p_{10})$$

Here, we used the facts that $N_{t,0} + N_{t,1} = N$, and $\text{Bin}(N, p) \overset{(d)}{=} N - \text{Bin}(N, 1 - p)$. This shows that

$$\mathbb{E}\left[N_{t+1,1}\right] = N - p_{10}\mathbb{E}\left[N_{t,1}\right].$$

Summing the LHS and RHS over all possible values of $t$ gives us

$$\frac{1}{H}\sum_{t=0}^{H-1} \mathbb{E}\left[N_{t+1,1}\right] = N - \frac{p_{10}}{H}\sum_{t=0}^{H-1} \mathbb{E}\left[N_{t,1}\right].$$

Rearranging the terms in the above expression gives us

$$\frac{1}{H}\sum_{t=1}^{H} \mathbb{E}[N_{t,1}] = \frac{N}{1 + p_{10}} - \frac{p_{10}N_{0,1}}{H(1 + p_{10})} + \frac{p_{10}N_{H,1}}{H(1 + p_{10})}.$$

As $H \to \infty$, the last two terms in the RHS above approach 0. So we have

$$\lim_{H\to\infty} \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}[N_{t,1}] = \frac{N}{1 + p_{10}}.$$

This shows that the normalized cumulative reward accrued by the policy is $\frac{1}{1+p_{10}}$. It remains to be shown that this policy indeed satisfies the budget constraint of F-RMAB. To this end, we rely on martingale concentration inequalities (Chung and Lu 2006). Since the budget used in each round is $b_t = N_{t,0}$, it needs to be shown that $\frac{1}{H}\sum_{t=0}^{H-1} N_{t,0} \leq B$ with high probability. Let $Z_t = N_{t,0} - \mathbb{E}_{s<t}[N_{t,0}]$. Note that $Z_t$ is a bounded random variable with variance bounded by $p_{10}N$. Moreover, $\{Z_t\}_{t=1}^{H}$ is a martingale difference sequence. Using standard concentration inequalities for martingale difference sequences, we get the following, which holds with probability at least $1 - \delta$

$$\frac{1}{H}|\sum_{t=1}^{H} Z_t| \leq \sqrt{\frac{p_{10}N \log\frac{1}{\delta}}{H}}.$$

Next, note that $\mathbb{E}_{s<t}[N_{t,0}] = p_{10}(N - N_{t-1,0})$. This gives us the following bound on the cost of the policy

$$\begin{aligned}\frac{1}{H}\sum_{t=0}^{H-1} N_{t,0} &= \frac{1}{H}\sum_{t=0}^{H-1} Z_t + \frac{1}{H}\sum_{t=0}^{H-1} \mathbb{E}_{s<t}[N_{t,0}] \\ &\overset{(a)}{=} p_{10}N + \frac{1}{H}\sum_{t=0}^{H-1} Z_t - \frac{p_{10}}{H}\sum_{t=0}^{H-1} N_{t,0},\end{aligned}$$

[1] In case the low-probability event where the policy violates the budget constraint happens at some time step $t$, we simply play action 0 for all arms, for the rest of the rounds.

where $(a)$ follows from the expression for $\mathbb{E}_{s<t}[N_{t,0}]$ derived above. Rearranging the terms in the above expression and substituting the above concentration inequality gives us

$$\lim_{H\to\infty} \frac{1+p_{10}}{H}\sum_{t=0}^{H-1} N_{t,0} \leq \sqrt{\frac{p_{10}N\log\frac{1}{\delta}}{H}} + p_{10}N.$$

This shows that

$$\lim_{H\to\infty} \frac{1}{NH}\sum_{t=0}^{H-1} N_{t,0} \leq \frac{(1+o(1))p_{10}}{1+p_{10}}.$$

This shows that with high probability, the algorithm satisfies the budget constraints. In case the low-probability event (where the policy violates the budget constraint) happens at some time step $t$, we simply play action 0 for all arms, for the rest of the rounds. It is easy to see that this only changes the normalized cumulative regret of the policy by $o(1)$ factors. This shows that $R_*^{\text{F-RMAB}}$ is lower bounded by $\frac{1-o(1)}{1+p_{10}}$

**Upper bound on $R_*^{\text{RMAB}}$.** In this setting, we have a fixed budget of $B$ at each round. Since the MDPs of all the arms are the same, it is easy to see that the following policy is optimal for the RMAB problem: (a) if $N_{t,0} \leq B$, then play action 1 for all the arms that are in state 0 and for $(B - N_{t,0})$ randomly selected arms that are in state 1. For the rest of the arms, play action 0, (b) on the other hand, if $N_{t,0} \geq B$, then play action 1 for $B$ randomly selected arms that are in state 0. For the rest of the arms, play action 0. For this policy, we have the following recurrences for $N_{t,0}, N_{t,1}$

$$N_{t+1,1} = \begin{cases} B + \text{Bin}(N-B, 1-p_{10}), & \text{if } B \geq N_{t,0}, \\ B + \text{Bin}(N_{t,1}, 1-p_{10}), & \text{otherwise.} \end{cases} \tag{16}$$

This recurrence again follows from the definition of the MDP. From the above equations, it is easy to see that the conditional expectation $\mathbb{E}[N_{t+1,1}|\text{history}]$ has the following recurrence

$$\mathbb{E}[N_{t+1,1}|\text{history}] = p_{10}B + (1-p_{10})N \tag{17}$$
$$+ (p_{10}-1)\max\{0, N_{t,0} - B\} \tag{18}$$

Since the last term is always non-positive, we have the following upper bound for $\mathbb{E}[N_{t+1,1}|\text{history}]$

$$\mathbb{E}[N_{t+1,1}|\text{history}] \leq p_{10}B + (1-p_{10})N = \frac{1+o(1)}{1+p_{10}}N.$$

where the last equality follows from our definition of $B$. This shows that as $H \to \infty$

$$R_*^{\text{RMAB}} = \frac{1}{NH}\sum_{t=1}^{H}\mathbb{E}[N_{t,1}] \leq \frac{1}{1+p_{10}}$$

This also shows that as $H \to \infty$ $R_*^{\text{RMAB}} \leq R_*^{\text{F-RMAB}}$, thus showing that flexibility is helpful in this setting.

We can further show that FRMAB strictly improves upon standard RMAB by deriving a tighter upper bound for $R_*^{\text{RMAB}}$. To this end, consider the stochastic process in Equation (16). It can be rewritten as follows

$$N_{t+1,1} = \begin{cases} B + \text{Bin}(N-B, 1-p_{10}), & \text{if } N_{t,1} \geq N-B \\ B + \text{Bin}(N_{t,1}, 1-p_{10}), & \text{otherwise.} \end{cases}$$

The process $\{N_{t,1}\}_{t=1}^{\infty}$ is supported on a finite state space of $\{B, \dots, N\}$. Let $P \in \mathbb{R}^{(N-B+1)\times(N-B+1)}$ be the transition matrix of this process, where $P_{ij}$ denotes the probability of transitioning from state $(B+i-1)$ to state $(B+j-1)$. For any pair of states $s, s'$, it is easy to see that there is a non-zero probability of the process going from $s$ to $s'$ in $\lceil N/B \rceil$ rounds (this follows from the fact that in every round, there is a non-zero probability of $N_{t,1}$ increasing by $B$). This shows that all the elements of the matrix $P^{\lceil N/B \rceil}$ are non-zero. Applying Wielandt's theorem for our choice of $p_{10}$ shows that the stochastic process is ergodic (Wielandt 1950).

From theory of ergodic processes, we know that our process visits each of the states inifintely many times. Let $\pi : \{B, \dots, N\} \to \mathbb{R}_{>0}$ be the stationary distribution of the process, where $\pi(s)$ is the long-run proportion of time the process is in state $s$. $R_*^{\text{RMAB}}$ can be written in terms of $\pi$ as

$$R_*^{\text{RMAB}} = \frac{1}{N}\sum_{s=B}^{N}\pi(s)s.$$

We already know that $\lim_{H\to\infty} R_*^{\text{RMAB}} \leq \frac{1}{1+p_{10}}$. To show that this is strictly less than $\frac{1}{1+p_{10}}$, consider the following

$$\begin{aligned}&\mathbb{E}[N_{t+1,1}] = \\ &\quad \mathbb{P}(N_{t,1} \geq N-B)\left(B + (N-B)(1-p_{10})\right) \\ &\quad + \mathbb{P}(N_{t,1} < N-B)\mathbb{E}\left[B + N_{t,1}(1-p_{10})|N_{t,1} < N-B\right].\end{aligned}$$

This follows from Equation (16). From our definition of $B$, we have $B + (N-B)(1-p_{10}) = \frac{N}{1+p_{10}}$. Moreover, we have

$$\begin{aligned}&\mathbb{E}\left[B + N_{t,1}(1-p_{10})|N_{t,1} < N-B\right] \\ &\quad < B + (N-B)(1-p_{10}) = \frac{N}{1+p_{10}}.\end{aligned}$$

Since our process is ergodic, we know that $\lim_{H\to\infty}\frac{1}{H}\sum_{t=1}^{H}\mathbb{P}(N_{t,1} < N-B) > 0$. Combining these results and plugging them into the previous display gives us

$$\lim_{H\to\infty}\sum_{t=1}^{H}\frac{1}{NH}\mathbb{E}[N_{t+1,1}] < \frac{1}{1+p_{10}}.$$

This shows that $R_*^{\text{RMAB}} \leq \frac{1-c}{1+p_{10}}$, where $c > 0$ is a positive constant that is independent of $H$[2]. This finishes the proof of the Theorem. □

[2] a more careful analysis can be used to show that $c = 1/\text{poly}(N)$

**FRMAB vs RMAB with large budget.** The simple setting considered in this Theorem also sheds light on when flexibility is helpful. One can show that for $B \geq \left(\frac{p_{10}}{1+p_{10}}\right) N + \sqrt{p_{10} N \log H}$, both FRMAB and RMAB achieve the same reward. This is because with such high budget, the number of arms in state 0 at any point in time is lower than $B$ with very high probability. Consequently, both RMAB and FRMAB can perform action 1 on all the arms whose state is 0 in all the rounds. Consequently, both achieve the same rewards.

$$B \geq \left(\frac{p_{10}}{1+p_{10}}\right) N + \sqrt{p_{10} N \log H}$$

# B Algorithms

Algorithm 3: COMPRESSSTEPS

```
Input: $F, N, \mathcal{S}, \mathcal{A}, P$
$\mathcal{A}^F = \{(a_i)_{i \in [F]} \mid a_i \in \mathcal{A}\}$
for $n \in [N]$ do
    for $\mathbf{a} \in \mathcal{A}^F$ do
        $\hat{P}_n^{F-1} = \prod_{i \in [F-1]} P_n(s, a_i, z_i)$
        $P_n^F(s, \mathbf{a}, s') = \sum_{\mathbf{z} \in \mathcal{S}^{F-1}} \hat{P}_n^{F-1} \cdot P_n(z_{F-1}, a_{F-1}, s')$
    end for
end for
return $\mathcal{A}^F, P^F$
```

Algorithm 4: COMPRESSSTEPS + HAWKINS with **closing** flexible window

```
Input: $N$, MDP $\langle \mathcal{S}, \mathcal{A}, \mathbf{c}, P, r \rangle, B, H, F, s_0$ //$H \bmod F = 0$
for $t = 1, \ldots, H$ do
    $F' = F - t \bmod F$ // Close window
    if $F' = 0$ then
        $u = 0$ // Used budget in current window
    end if
    $\mathcal{A}^{F'}, P^{F'}$ = COMPRESSSTEPS($F', N, \mathcal{S}, \mathcal{A}, P$)
    $B' = F'B - u$
    $\lambda_{\min}, Q$ = HAWKINS($\mathcal{S}, \mathcal{A}^{F'}, P^{F'}, B'$)
    $\mathbf{a}$ = ACTIONKNAPSACK($Q(s_{t-1}, \cdot, \lambda_{\min}), \mathbf{c}, B'$) // $\mathbf{a} = (\mathbf{a_1}, \ldots, \mathbf{a_{F'}})$
    $s_{t+1} = $ STEP($\mathbf{a_1}, s_t, P$)
    $u$ += $\mathbf{a_1} \cdot \mathbf{c}$
end for
```

Algorithm 5: COMPRESSSTEPS + HAWKINS with **static** flexible window

```
Input: $N$, MDP $\langle \mathcal{S}, \mathcal{A}, \mathbf{c}, P, r \rangle, B, H, F, s_0$ //$H \bmod F = 0$
$\mathcal{A}^F, P^F$ = COMPRESSSTEPS($F, N, \mathcal{S}, \mathcal{A}, P$)
for $t = 0, F, 2F, \ldots, H$ do
    $Q, \lambda_{\min}$ = HAWKINS($\mathcal{S}, \mathcal{A}^F, P^F, FB$)
    $(\mathbf{a_1}, \ldots, \mathbf{a_F})$ = ACTIONKNAPSACK ($Q(s_{t-1}, \cdot), \mathbf{c}, TB$)
    for $i = 1, \ldots, F$ do
        $s_{i+1} = $ STEP($\mathbf{a_1}, s_i, P$)
    end for
end for
```

# C Experimental Domain Diagrams

We present the diagrams for the **Immediate recovery** (Fig. 5) and **Two-state process** (Fig. 7) domains. Please see section 5 for additional descriptions.

# D Experiment Setup Details

All algorithms were implemented in Python 3.7.10 and mathematical programs were solved using Gurobi version 9.0.3 via the gurobipy interface (Gurobi Optimization 2021). Experiments were run on a cluster running CentOS with Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.1 GHz with 8GB of RAM and 8 processors.

## D.1 Experimental domains

For our experimental setups we consider the three synthetic domains described in Section 5.1 with the following characteristics.

**Dropout state** This domain takes a planning horizon of length $H = 30$, $N = 10$ arms, initial per round budget of $B = 1$ and vary the flexible window length $F = 2, 3, 5$. As shown in Fig. 1, each arm $n \in [N]$ has random transition probabilities $p^0_{0,0} \in [0.85, 0.95]$, $p^1_{0,0} = 0$, $p^0_{1,1} \in [0.35, 0.5]$ and $p^1_{1,1} = 1$. Results are averaged over 30 seeds.

**Immediate recovery** This domain takes a planning horizon of length $H = 10$, $N = 10$ arms, $S = 5$ states with rewards $R(s) = -1$ for $s = 0, 1$, $R(s) = 0$ for $s = 2$, and $R(s) = 1$ for $s = 3, 4$, an initial per round budget of $B = 1$ and vary the flexible window length $F = 2, 5, 10$. Transition probabilities for each arm $n \in [N]$ are $p^1_{s',S-1} = 1$, $p^0_{0,0} = p^1_{0,0} = 1$, $p^0_{s,s} \in [0.5, 0.7]$ are sampled uniformly and $p^0_{s,s-1} = 1 - p^0_{s,s}$ for all $s = 1, \ldots, S-1$ as shown in Fig. 5. The rest of the transition probabilities are set to 0. Results are averaged over 30 seeds.

**Two-state process** This domain takes a planning horizon of length $H = 6$, $N = 10$ arms, initial per round budget of $B = 1$ and vary the flexible window length $F = 2, 3, 6$. Transition probabilities for each arm $n \in [N]$ are $p^1_{0,1} = p^1_{1,1} = 1$, $p^0_{0,0} \in [0.85, 0.95]$ and $p^0_{1,1} \in [0.5, 0.85]$ are uniformly sampled, and $p^0_{0,1} = 1 - p^0_{0,0}$ and $p^0_{1,0} = 1 - p^0_{1,1}$ as shown in Fig. 7. Results are averaged over 30 seeds.

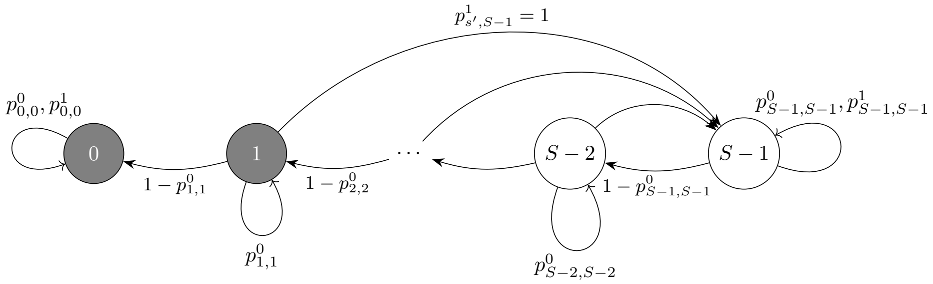
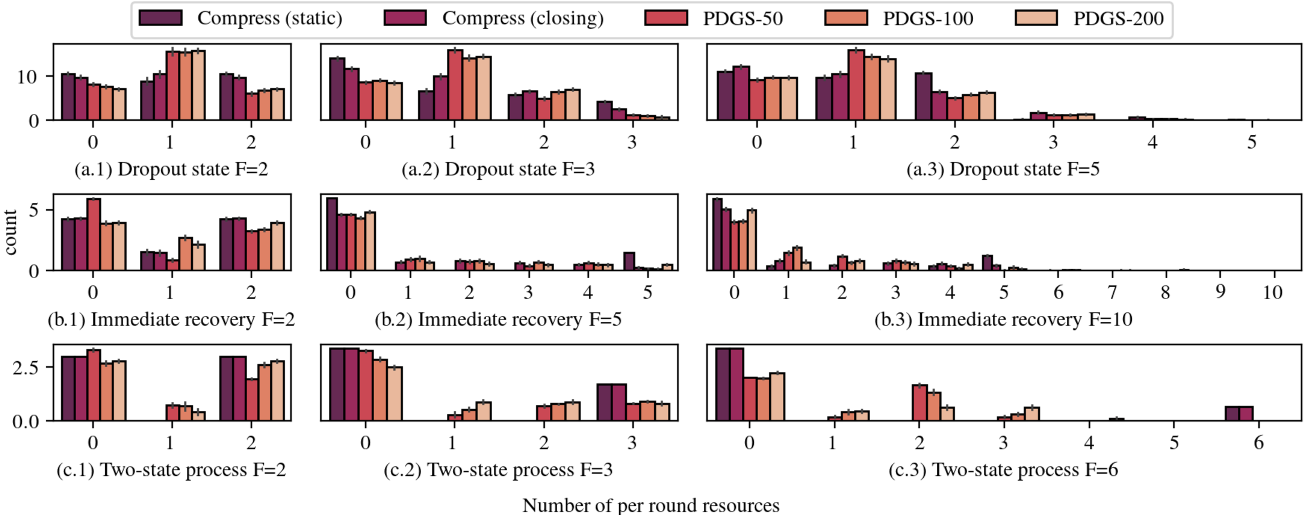


Figure 5: Immediate recovery domain.



(a.1) Dropout state F=2

(a.2) Dropout state F=3

(a.3) Dropout state F=5

(b.1) Immediate recovery F=2

(b.2) Immediate recovery F=5

(b.3) Immediate recovery F=10

(c.1) Two-state process F=2

(c.2) Two-state process F=3

(c.3) Two-state process F=6

Figure 6: Frequency of used resources at each step

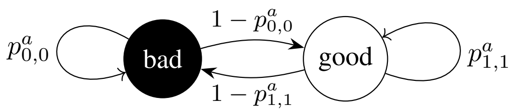


Figure 7: Two-state process domain with $p^a_{s,s'}$ the transition probability from state $s$ to $s'$ after taking action $a$.

## D.2 Algorithms

The following algorithms with the given hyper parameters were tested in the three synthetic domains.

**Hawkins [2003]** Computes values of Lagrange multipliers after relaxing per-round budget constraints with fixed budget $B$. Then solves a knapsack with budget $B$ per round over Q-values adjusted for the solved Lagrange multipliers. The only set hyperparameter is the discounting factor (gamma) of 0.95. See algos/hawkins_methods and algos/hawkins_actions in the supplementary code material for detailed implementation.

**Compress (static) and compress (closing)** Simplifies an F-RMAB to a classical multi-action RMAB by considering all possible sequences of binary actions in each window $F$ and forcing all $b_t = FB$, then computes policies using existing classic multi-action RMAB techniques. Compress (static) follows Alg. 5 and compress (closing) follows Alg. 4. Both algorithms use a discounting factor (gamma) of 0.95 in the HAWKINS function (lines 7 and 3 of Alg. 5 and Alg. 4 respectively). See algos/compressing_methods in the supplementary code material for detailed implementation.

***PDGS-N*** Implements Alg. 1 with $N$ iterations, $\tau = 0.1$, $\sigma = 0.1$, $x^0 = \vec{0}$, $y^0 = \vec{1}$, and $N_{steps} = 50$. Following Theorem 1 in Chambolle and Pock (2011), we choose $\tau$ and $\sigma$ such that $\tau\sigma L^2 < 1$ to assure that the algorithm will converge to a saddle point $(x^*, y^*)$, where $L = ||K|| = \max\{||Kx|| : x \in X \text{ with } ||x|| \leq 1\}$ for $K$ as defined in the proof of Theorem **??**. See algos/minmax_methods in the supplementary code material for detailed implementation.

## D.3 Results: frequency of used resources

Figure 6 shows the frequency with which 0 to $FB$ resources are assigned at each step of the planning horizon for the three synthetic domains considered: (a) dropout state, (b) immediate recovery, and (c) two-state process. These results are discussed in Section 5.3.