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 Tambe1,2

1Harvard University, 2Google 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 real-world planning settings, this flexibility in budget is often constrained to within a subset of consecutive planning steps, e.g., weekly planning of a monthly budget. 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 ≤ 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 [Pa-
We use 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 (MDPs). 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 2003; Tomlin and Wang 2005; Boyabati, 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.

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 \( (S, \mathcal{A}, R, \mathcal{P}) \). \( S \) denotes the state space, \( \mathcal{A} \) is the set of possible actions, \( R \) is the reward function \( R : S \times \mathcal{A} \times S \rightarrow \mathbb{R} \), and \( \mathcal{P} : S \times \mathcal{A} \times S \rightarrow [0, 1] \) represents the transition function. We use \( P_{s,a}^{s'} \) to denote the probability of transitioning from state \( s \) to state \( s' \) under the action \( a \).

We let \( s_t = [s_1^t, s_2^t, \ldots, s_N^t] \) denote the vector of states of the N MDPs at time step \( t \). A policy is a mapping \( \pi^t : \mathcal{S}^N \rightarrow \mathcal{A}^N \) 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 \( c \) of size \( N \times |\mathcal{A}| \), i.e., \( c_{ij} \) is the cost of taking action \( j \in \mathcal{A} \) on arm \( i \). Let \( \mathcal{I}_{\pi^t(s)} \) be the one-hot encoder of size \( N \times |\mathcal{A}| \), where each row indicates which action to perform on arm \( n \) at time step \( t \). The planner’s goal is to find reward maximizing policies \( \{\pi^t\}_{t=1}^H \) under the budget constraint \( \mathcal{I}_{\pi^t(s)} \cdot c \leq B \) 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 \( V_n^*(s_0) = \mathbb{E} \left[ \sum_{t=1}^H \sum_{n=1}^N R(s_n^{t-1}, c_{\pi^t(s_n^{t-1})_n}, s_n^t) \right] \) where the next state is drawn according to \( s_n^t \sim \mathcal{P}[\pi^t(s_n^{t-1})]_{s_n^{t-1}} \). The planner’s goal is to find policies \( \pi = \{\pi^t\}_{t=1}^H \) that maximize the total reward.

3.2 Definition

In F-RMABs, we define the MDP followed by each arm using the tuple \( \{S, \mathcal{A}, R, \mathcal{P}\} \) 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 \( \{\pi^t\}_{t=1}^H \) such that \( \sum_{t=1}^F (\mathcal{I}_{\pi^t(s)} \cdot c) \leq FB \) for \( t = F + 1, \ldots, H \). That is, we consider an exhaustible budget \( FB \) that is available to spend over the flexible window \( 1, \ldots, F \) and think of \( B \) as the one-step budget at every time step \( t \) after the flexible window \( t = F + 1, \ldots, H \).

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 [Sombu et al. 2020] (see Fig. 8 in Appendix). In this process, we model the MDP followed by each arm using two states, \( \text{good} \) (s = 1) and \( \text{bad} \) (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 \( F = H, H \rightarrow \infty \) 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
\[ B = \left( \frac{1+\sigma(1)}{p_{t}} \right) N. \] Define normalized cumulative reward as \( \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{H} \mathbb{E} \left[ R(s_{n}^{(t-1)}, a_{n}^{(t-1)}, s_{n}^{(t)}) \right], \) where \( s_{n}^{(t)} \), \( a_{n}^{(t)} \) is the state and action of arm \( n \) at time \( t \). Let \( R_{F-RMAB} \) and \( R_{RMAB} \) be the maximum normalized cumulative rewards that can be achieved under the budget constraints imposed by F-RMAB and RMAB. Then,

\[
R_{F-RMAB}^* \geq 1 - o(1) \quad \frac{1+\sigma(1)}{p_{t}}, \quad R_{RMAB}^* \leq 1 - \frac{c}{1+\sigma(1)}.
\]

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 flexibility, we extend Lagrangian relation to the flexible setting 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:

\[
\sum_{t=1}^{F} \mathbb{1}_{\pi^{t}}(a^{t}) \cdot c \leq F B,
\]

where \( \mathbb{1}_{\pi^{t}}(a^{t}) \) is the one-hot encoded matrix of size \( N \times |A| \), 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:

\[
\sum_{t=1}^{F} b_{t} \leq F B.
\]

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:

\[
\max_{\pi^{1}, \ldots, \pi^{F}} \mathbb{E} \left[ \sum_{t=1}^{F} \sum_{n=1}^{N} R(s_{n}^{(t-1)}, a_{n}^{(t-1)}, s_{n}^{(t)}) \right]
\]

s.t. \( \mathbb{1}_{\pi^{t}}(a^{t}) \cdot c \leq b_{t}, \quad \forall t \in \{ 1, \ldots, F \} \)

\[
\sum_{t=1}^{F} b_{t} \leq F B
\]

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:

\[
\min_{x \in X} \max_{y \in Y} \left[ K(x, y) + G(x) - H^{*}(y) \right]
\]

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 \( \mathbb{1}_{a^{t}} \in \mathbb{R}^{N \times |A|} \) 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:

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

s.t. \( \mathbb{1}_{a^{t}} \cdot c \leq b_{t} \),

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

\[
\max_{b_{1}, \ldots, b_{F}} J^{1}(s^{0}) \quad \text{s.t.} \quad \sum_{t=1}^{F} b_{t} \leq F B
\]

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

\[
J^{t}(s^{t-1}) = \max_{a^{t}} \min_{\lambda_{t+1} \geq 0} \sum_{n=1}^{N} \mathbb{E} \left[ R(s_{n}^{t-1}, a_{n}^{t}, s_{n}^{t}) + J^{t+1}(s^{t}) \right] + \lambda_{t+1}(b_{t} - \mathbb{1}_{a^{t}} \cdot c).
\]
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 $J'$ 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), (b) we enforce the Lagrangian multipliers to be constant across states: $\lambda_{t-1} = \lambda_{t-1}$. 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:

$$\min_{\lambda \geq 0} \max_b \left\{ \sum_{n=1}^{N} L_n(s^n_0; \lambda) + \sum_{t=1}^{F} \lambda_t b_t + \sum_{t=F+1}^{H} \lambda_t B \right\},$$

s.t. $\sum_{t=1}^{F} b_t \leq FB \quad (10)$

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

$$L_n(s^n_{t-1}; \lambda) = \max_{a_n} E \left[ R(s^n_{t-1}, a^n_{t}, s^n_{t}) - \lambda_t \alpha_n \cdot c_n + L^{t+1}_n(s^n_{t}, \lambda) \right]. \quad (11)$$

Since $L^t_n(s^n_0; \lambda)$ is a supremum of linear functions in $\lambda$, it is convex in $\lambda$. 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:

$$\min_{\lambda \geq 0, \mu \geq 0} \max_b \left\{ \sum_{n=1}^{N} L_n(s^n_0; \lambda) + \sum_{t=1}^{F} \lambda_t b_t + \sum_{t=F+1}^{H} \lambda_t B + \mu \left( FB - \sum_{t=1}^{F} b_t \right) \right\}. \quad (12)$$

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

$$G(\lambda, \mu) = \sum_{n=1}^{N} L_n(s^n_0; \lambda) + \sum_{t=F+1}^{H} \lambda_t B + \mu FB \quad (13)$$

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

$$\min_{\lambda, \mu} \max_b \left( K(\lambda, \mu) + G(\lambda, \mu) - H^*(b) \right) \quad (14)$$

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.** $G(\lambda, \mu)$ is convex in $\lambda$ 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 min-max 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

$$\text{prox}_F(x) = \arg \min_u \left( F(u) + \frac{1}{2\sigma} ||u - x||^2 \right).$$

Following the notation in Chambolle and Pock [2011],

$$\text{prox}_{\sigma F}(x) = (I + \sigma \partial F)^{-1} x.$$  

Hence, the proximal operator of the zero function $H^*$ is the identity. The proximal operator of $G$ does not have any analytical form. However, it is a piecewise-linear function. Since the proximal operator of linear functions is simply $x - \sigma \nabla F(x)$, a good approximation of $\text{prox}_{\sigma G}(x)$ 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 $(\lambda, \mu)$ is given by $\nabla G((\lambda, \mu)) = [D^1, D^2, \ldots, D^F, D^{F+1} + B, \ldots, D^H + B, FB]$, where $D^t = \E[\sum_{n} e_n^{t}]$ is the expected sum of costs over all arms in step $t$ under the optimal policy for $\lambda$.

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 $\pi^*(\lambda)$ for $\lambda$, we can get unbiased samples of each $D^t$ via Monte Carlo simulation of $\pi^*(\lambda)$.

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 stochastic gradient (PDSG) 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.

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
chains of flexible actions for each arm. This allows us to use existing tools to encode a flexible budget, though this requires setting up constraints and quadratic cost in the number of variables, and thus has exponential complexity \(\mathcal{O}(NS^2HA + N^2S^2H^2)\). Finally, \textsc{SampleGrad} loops over all arms and timesteps, sampling from an \(S\)-length distribution for each next state, and so has complexity \(\mathcal{O}(NSH \times N_{\text{steps}})\). Putting this all together, inside the convergence loop with \(Z\) steps gives the following computational complexity for PDSG:

\[
\mathcal{O}(Z(FH + NS^2HA + N^2S^2H^2 + NSHN_N))
\]

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

## 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

**Dropdown 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 \(A = \{0, 1\}\) corresponding to a passive action \((a = 0)\) and active action \((a = 1)\). The reward function \(R : S \to \mathbb{R}\) is defined as \(R(0) = 0\) and \(R(1) = R(2) = 1\). Once an arm reaches the dropout state, it cannot transition to any other state, i.e. \(P_{a,0}^n = 1\) for all \(a \in A\). 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 Fig. 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. 6 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

---

**Algorithm 1: PDSG**

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

1: \(\hat{v}^0 = [\lambda^0, \sigma^0]\)
2: while not converged do
3: \(b^{n+1} = b^n + \sigma K \hat{v}^n\)
4: \(\hat{v}^{n+1} = v^n - \tau K b^{n+1} / \hat{v}^{n+1} = [\lambda^{n+1}, \sigma^{n+1}]\)
5: \(\pi^{n+1} = \text{FiniteBellmanLP}(P, \lambda^{n+1}, H) / \text{LP}\)
6: \(\nabla G(\hat{v}^{n+1}) = \text{SampleGrads}(N_g, \pi^{n+1}, P, H, F)\)
7: \(\nu^{n+1} = \nu^n + \tau \nabla G(\hat{v}^{n+1})\)
8: \(\bar{v}^{n+1} = 2\nu^{n+1} - \nu^n\)
9: end while

**Algorithm 2: SampleGrads**

**Input:** Number of gradient samples \(N_g\), policy \(\pi : S^N \to A^N\), transition function \(P\), planning horizon \(H\), flexible time window \(F\)

1: for \(i \in \{1, \ldots, N_g\}\) do
2: \(x_i = (x_{1,i}, \ldots, x_{H,i}) = \text{MonteCarlo}(\pi, P)\) // Simulate \(\pi\) in environment \(P\), return cost at \(t \in [H]\)
3: end for
4: \(D = \frac{1}{N_g} \sum_{i=1}^{N_g} x_i\)
5: return \([D, BF] / \nabla G\)
The solution problem presented in Theorem 2 using Algorithm 1. Hawkins with fixed per round budget on the three domains described compare them against a classic RMAB solution algorithm. We test PDSG (Algorithm 1) and Compress heuristics (Algorithm 5 in Appendix C) plans across the flexible window framework. Q-values adjusted for the solved Lagrange multipliers. Plans solves a knapsack with the solved budget variables. 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 C) plans across the flexible window \( F \) in each timestep and executes the first action before recomputing. Compress (closing) (Algorithm 4 in Appendix C) 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, PDSG-N, solves the primal-dual optimization problem presented in Theorem 2 using Algorithm 1) with \( N \) iterations and gradient sampling for \( \nabla G \). 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.

### 5.2 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 C) plans across the flexible window \( F \) in each timestep and executes the first action before recomputing. Compress (closing) (Algorithm 4 in Appendix C) plans across a flexible window of size \( F \), then \( F-1 \), and so on until a window of size 1, repeating every \( F \) steps.

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). PDSG-200 attains an increase in average cumulative reward of 15.57%, 63.28% and 72% for \( F = 2, 5, 10 \) respectively compared to Hawkins. PDSG attains higher rewards by allocating up to 2 to 5 resources at the last step of the current window and does not obtain a maximum increase in reward 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. 7(a.1)–(a.3) in appendix E.3).

As shown in Fig. 2(c) for the two-state process domain, our method to solve for policies with flexible budget (PDSG-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 C) attains almost as much reward as the near optimal algorithms (PDSG) under significantly smaller runtime (fig. 3), 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 PDSG-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. 3.

**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 \( p_{0,0}^0 \in [0.85, 0.95], p_{0,0}^1 = 0, p_{1,1}^0 \in [0.35, 0.5] \) and \( p_{1,1}^1 = 1 \) in our experiments.

\[
R(1) = 1 \text{ for each arm in the good state and } R(0) = 0 \text{ for the bad state. See Figure 8 in Appendix for a diagram of the model, with four transition probability parameters.}
\]
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. 4, flexible budget policies solved via PDSG 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, PDSG’s superior performance (see fig. 4) 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 PDSG attain equally as much reward as the optimal fixed budget policies solved with Hawkins, demonstrating its robustness. Moreover, PDSG has no significant increase in runtime when increasing the flexible time window F as seen in fig. 4 (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).

Finally, we evaluate the performance of PDSG for longer time horizons reaching values where F ≪ H in the two-state process domain. PDSG 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 ≪ H, e.g. F = 2 and H = 100 (see 5). 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 PDSG, 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 (PDSG) for small flexible time windows requiring significantly less runtime. We hope that this work further contributes to the real-world applicability of RMAB methods.
7 Acknowledgments

J.A.K. was supported by an NSF Graduate Research Fellowship under grant DGE1745303. L.X. was supported by a Google PhD Fellowship, and was a Student Researcher at Google for part of the project.

References


A 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 $t = 1, \ldots, F$, 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. ?? 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.

B Theory

B.1 Proof of Proposition 1

Proposition 1. $G(\lambda, \mu)$ is convex in $\lambda$ and $\mu$.

Proof. $G(\lambda, \mu)$ is given by

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

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

B.2 Proof of Theorem 1

Theorem 1. Suppose $F = H$, $H \to \infty$ and $p_{t0} \geq N^{-1/2}$. Moreover, suppose the cost of playing action 0 is 0 and action 1 is 1, and suppose the one-step budget $B = \left(\frac{(1+o(1))p_{t0}}{1+p_{t0}}\right)N$. Define normalized cumulative reward as $\frac{1}{NH} \sum_{n=1}^{N} \sum_{t=1}^{H} E \left[R(s_n^{(t-1)}, a_n^{(t-1)}, s_n^{(t)})\right]$, where $s_n^{(t-1)}, a_n^{(t-1)}$ is the state and action of arm $n$ at time $t$. Let $R^*_{F-RMAB}$ and $R^*_{RMAB}$ be the maximum normalized cumulative rewards that can be achieved under the budget constraints imposed by F-RMAB and RMAB. Then,

$$R^*_{F-RMAB} \geq \frac{1 - o(1)}{1 + p_{t0}}, \quad R^*_{RMAB} \leq \frac{1 - c}{1 + p_{t0}}.$$ 

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

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

This follows from the following fact

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

Lower bound on $R^*_{F-RMAB}$. 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 $R^*_{F-RMAB}$ 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.

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

$$N_{t+1,1} = N_{t,0} + \text{Bin}(N - N_{t,0}, 1 - p_{t0}),$$

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

$$N_{t,1+1} = N - \text{Bin}(N_{t,1}, p_{t0})$$

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

$$E[N_{t+1,1}] = N - p_{t0}E[N_{t,1}].$$

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

$$\frac{1}{H} \sum_{t=0}^{H-1} E[N_{t+1,1}] = N - \frac{p_{t0}}{H} \sum_{t=0}^{H-1} E[N_{t,1}].$$

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.
Rearranging the terms in the above expression gives us
\[
\frac{1}{H} \sum_{t=1}^{H} E[N_{t,1}] = \frac{N}{1 + p_{10}} - \frac{p_{10} N_{0,1}}{H(1 + p_{10})} + \frac{p_{10} N_{H,1}}{H(1 + p_{10})}.
\]
As \(H \to \infty\), the last two terms in the RHS above approach 0. So we have
\[
\lim_{H \to \infty} \frac{1}{H} \sum_{t=1}^{H} E[N_{t,1}] = \frac{N}{1 + p_{10}}.
\]
This shows that the normalized cumulative reward achieved 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 \(Z_t = N_{t,0} - E_{\pi_t}[N_{t,0}]\). 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 \(1 - \delta\)
\[
\frac{1}{H} \left| \sum_{t=1}^{H} Z_t \right| \leq \frac{p_{10} N \log \frac{1}{\delta}}{H}.
\]
Next, note that \(E_{\pi_t}[N_{t,0}] = p_{10}(N - N_{t-1,0})\). This gives us the following bound on the cost of the policy
\[
\frac{1}{H} \sum_{t=0}^{H-1} N_{t,0} \leq \frac{1}{H} \sum_{t=0}^{H-1} Z_t + \frac{1}{H} \sum_{t=0}^{H-1} E_{\pi_t}[N_{t,0}] \leq (a) p_{10} N + \frac{1}{H} \sum_{t=0}^{H-1} Z_t - p_{10} \frac{1}{H} \sum_{t=0}^{H-1} N_{t,0},
\]
where \((a)\) follows from the expression for \(E_{\pi_t}[N_{t,0}]\) derived above. Rearranging the terms in the above expression and substituting the above concentration inequality gives us
\[
\lim_{H \to \infty} \frac{1}{H} \sum_{t=0}^{H-1} N_{t,0} \leq \frac{p_{10} N \log \frac{1}{\delta}}{H} + p_{10} N.
\]
This shows that
\[
\lim_{H \to \infty} \frac{1}{N H} \sum_{t=0}^{H-1} N_{t,0} \leq \frac{(1 + o(1))p_{10}}{1 + p_{10}}.
\]
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 \(R^*_{\text{F-RMAB}}\).** 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 \(N_{t,0}, N_{t,1}\)
\[
N_{t+1,1} = \begin{cases} B + \text{Bin}(N - B, 1 - p_{10}), & \text{if } B \geq N_{t,0} \\ B + \text{Bin}(N_{t,1}, 1 - p_{10}), & \text{otherwise}. \end{cases}
\]
**(16)**

This recurrence again follows from the definition of the MDP. From the above equations, it is easy to see that the conditional expectation \(E[N_{t+1,1}\mid \text{history}]\) has the following recurrence
\[
E[N_{t+1,1}\mid \text{history}] = p_{10} B + (1 - p_{10}) N + (p_{10} - 1) \max\{0, N_{t,0} - B\}
\]
**(17)**
Since the last term is always non-positive, we have the following upper bound for \(E[N_{t+1,1}\mid \text{history}]\)
\[
E[N_{t+1,1}\mid \text{history}] \leq p_{10} B + (1 - p_{10}) N = \frac{1 + o(1)}{1 + p_{10}} N.
\]
where the last equality follows from our definition of \(B\). This shows that as \(H \to \infty\)
\[
R^*_{\text{F-RMAB}} = \frac{1}{N} \sum_{t=1}^{H} E[N_{t,1}] \leq \frac{1}{1 + p_{10}}.
\]
This also shows that as \(H \to \infty\) \(R^*_{\text{F-RMAB}} \leq R^*_{\text{F-RMAB}}\), thus showing that flexibility is helpful in this setting.

We can further show that F-RMAB 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 \(\{B, \ldots, N\}\). 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 \([N/B]\) 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 \(P^{[N/B]}\) 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 infinitely many times. Let \(\pi : \{B, \ldots, N\} \to \mathbb{R}_{>0}\) be the stationary distribution of the process, where \(\pi(s)\) is the long-run proportion of time
the process is in state $s$. $R^{\text{RMAB}}_s$ can be written in terms of $\pi$ as

$$R^{\text{RMAB}}_s = \frac{1}{N} \sum_{s'=B}^N \pi(s)s.$$  

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

$$\mathbb{E}[N_{t+1,1}] = \mathbb{P}(N_{t,1} \geq N - B)(B + (N - B)(1 - p_{10})) + \mathbb{P}(N_{t,1} < N - B) \mathbb{E}[B + N_{t,1}(1 - p_{10}) | N_{t,1} < N - B].$$

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

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

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

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

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

**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.

### C Algorithms

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

**Input:** $N$, MDP $\langle S, A, c, P, r \rangle$, $B$, $H$, $F$, $s_0$ // $H$ mod $F = 0$

1: for $t = 1, \ldots, H$ do
2: $F' = F - t$ mod $F$ // Close window
3: if $F' = 0$ then
4: $u = 0$ // Used budget in current window
5: end if
6: $A'' = $ COMPRESSSteps($F'$, $N$, $S$, $A$, $P$)
7: $B' = F'B - u$
8: $\lambda_{\min}, Q = $ HAWKINS($S$, $A''$, $B'$)
9: $a = $ ACTIONKnapsack($Q(s_{t-1,1}, \lambda_{\min})$, $c$, $B'$) // $a = (a_1, \ldots, a_{F'})$
10: $s_{t+1} = $ STEP($a_1, s_t, P$)
11: $u += a_1 \cdot c$
12: end for

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

**Input:** $N$, MDP $\langle S, A, c, P, r \rangle$, $B$, $H$, $F$, $s_0$ // $H$ mod $F = 0$

1: $A'' = $ COMPRESSSteps($F$, $N$, $S$, $A$, $P$)
2: for $t = 0, F, 2F, \ldots, H$ do
3: $Q, \lambda_{\min} = $ HAWKINS($S$, $A''$, $F$, $FB$)
4: $(a_1, \ldots, a_F) = $ ACTIONKnapsack($Q(s_{t-1,1}, \lambda_{\min})$, $c$, $TB$)
5: for $i = 1, \ldots, F$ do
6: $s_{t+1} = $ STEP($a_1, s_t, P$)
7: end for
8: end for

### D Experimental Domain Diagrams

We present the diagrams for the **Immediate recovery** (Fig. 6) and **Two-state process** (Fig. 8) domains. Please see section 5 for additional descriptions.

### E 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.

#### E.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 

\[2\text{a more careful analysis can be used to show that } c = 1 \text{/poly}(N)\]
Figure 6: Immediate recovery domain.

shown in Fig. 1, each arm $n \in [N]$ has random transition probabilities $p_{0,0}^n \in [0.85, 0.95]$, $p_{0,1}^n = 0$, $p_{0,1,1}^n \in [0.35, 0.5]$ and $p_{1,1}^n = 1$. 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 $F = 2, 3, 6$. Transition probabilities for each arm $n \in [N]$ are $p_{0,1}^n = p_{1,1}^n = 1$, $p_{0,0}^n \in [0.85, 0.95]$ and $p_{0,1}^n \in [0.5, 0.85]$ are uniformly sampled, and $p_{0,1}^n = 1 - p_{0,0}^n$ and $p_{0,0}^n = 1 - p_{1,1}^n$ as shown in Fig. 8. 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 $p_{0,1}^n = p_{1,1}^n = 1$, $p_{0,0}^n \in [0.85, 0.95]$ and $p_{0,1}^n \in [0.5, 0.85]$ are uniformly sampled, and $p_{0,1}^n = 1 - p_{0,0}^n$ and $p_{0,0}^n = 1 - p_{1,1}^n$ as shown in Fig. 8. Results are averaged over 30 seeds.

E.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/hawkinsactions 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, \(\tau = 0.1\), \(\sigma = 0.1\), \(x^0 = \mathbf{0}\), \(y^0 = \mathbf{1}\), and \(N_{\text{steps}} = 50\). 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.

### E.3 Results: frequency of used resources

Figure 7 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.