# Contextual Budget Bandit for Food Rescue Volunteer Engagement

**Ariana Tang[1], Naveen Raman[2], Fei Fang[2], Zheyuan Ryan Shi[3]**
[1]University of Chicago, [2]Carnegie Mellon University, [3]University of Pittsburgh

## Abstract

Restless multi-armed bandits (RMABs) are an extension of the multi-armed bandit framework where pulling an arm results in both a reward and a Markovian state change. While RMABs are being employed in domains including public health and food insecurity, its objective of maximizing the cumulative reward comes at the expense of reward disparity across different context groups. To address this issue, we introduce contextual budget allocation, which optimizes the budget amount across different contexts, along with the traditional budget allocation within each context. This allows higher-need groups to receive larger budgets. We develop a set of novel policies: (1) COcc, an empirically fast heuristic algorithm based on the Whittle index policy, and (2) Mitosis, a provably optimal algorithm that combines a branch-and-bound search structure and no-regret algorithm framework. We conduct extensive experiments with synthetic and real food rescue datasets.

## 1 Introduction

Restless multi-armed bandits (RMABs) are a model of decision-making consisting of $N$ arms corresponding to independent Markov Decision Processes Whittle [1988], Weber and Weiss [1990a], Papadimitriou and Tsitsiklis [1994]. Arms in an RMAB are "restless" because they can change state even when not pulled, and at each timestep, we can pull up to $K$ out of $N$ such arms. The flexibility of RMABs has led to their application in a variety of domains, including public health [Mate et al., 2020], food rescue [Raman et al., 2024], autonomous driving [Li et al., 2021], email campaign Mimouni and Avrachenkov [2025] etc.

While the majority of policies for RMABs focus on maximizing rewards, they often come at a cost of significant disparity of rewards across some set of features. Such features, which distinguish one timestep from another, naturally come up in real-world applications of RMABs. For example, on food rescue platforms (FRPs), RMABs are used to optimize volunteer (arms) notifications for different "rescue trips" (timesteps). Regions with lower pickup rates would be sacrificed against more "convenient" regions while the RMAB is being optimized for cumulative reward. This geographical disparity has severe implications and has actually been observed in real-world trials Shi et al. [2021]. More formally, we can view each timestep in an RMAB as corresponding to a *context*, which details how rewards and transitions vary between timesteps. In the example above, the context captures attributes such as geographic information, but this concept of context is generic and goes far beyond the specific application of food rescue. See Appendix J for discussion of its applicability to domains such as digital agriculture and peer review.

In this work, we develop policies which respond to this disparity across contexts in RMABs. While fairness has been previously studied in RMABs Killian et al. [2023], Wang et al. [2024] (see related work in Appendix I), **our work is the first to consider inter-context rather than inter-arm fairness**. Developing policies to optimize inter-context fairness is challenging due to the stochasticity of contexts in RMABs. To overcome this, we tackle inter-context fairness through *contextual*

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

*budget allocation*, which allocates budgets based on the particular context, while maintaining an overall budget in expectation across contexts. By doing so, we can allocate higher budgets to more disadvantaged regions while still maintaining overall reward.

## 2 Preliminary Background

A Restless Multi-Armed Bandit (RMAB) is defined by $\langle N, \mathcal{S}, \mathcal{A}, \{r_i\}_{i\in[N]}, \{P_i\}_{i\in[N]}\rangle$. Each arm $i \in [N] := \{1, 2, \dots, N\}$ is an independent Markov Decision Process, with state space $\mathcal{S}_i = \{0, 1\}$ and binary action space $\mathcal{A}_i = \{0, 1\}$. Action $0$ corresponds to idling the arm while action $1$ corresponds to pulling the arm. The reward function for each arm $r_i : \mathcal{S}_i \times \mathcal{A}_i \to \mathbb{R}$ maps state-action pairs to a reward. $P_i$ is the transition kernel for each arm $i$. The overall system state at time $t$ is $\mathbf{s}^t = (s_1^t, s_2^t, \dots, s_N^t)$, and the decision maker selects action $\mathbf{a}^t = (a_1^t, a_2^t, \dots, a_N^t)$ subject to a budget constraint: $\sum_{i\in[N]} a_i^t \leq B, \quad \forall t = 1, 2, \dots$, which limits the number of arms that can be pulled in every time step. The objective is to design a policy that maps the current state $\mathbf{s}^t$ to an action vector $\mathbf{a}^t$ that maximizes the average reward over all arms and over an infinite time horizon.

Solving optimal policy for RMAB is PSPACE-hard Papadimitriou and Tsitsiklis [1999]. A widely-adopted approach for tackling the computational complexity inherent in RMABs is the Whittle Index Policy. The *Whittle Index* for each arm $i$'s state $s_i \in \mathcal{S}_i$ is $w_i(s_i) = \min_w \{w | Q_{i,w}(s_i, 0) = Q_{i,w}(s_i, 1)\}$, defined using the standard Bellman $Q$-function and $V$ (value) function. At each time step, the *Whittle Index Policy* pulls the $B$ arms with the highest Whittle Indices. The Whittle index policy is asymptotically optimal [Gittins et al., 2011].

## 3 Contextual Budget Bandit

In this work, we augment the standard RMAB model with variability in transition and reward across time, and the flexibility to adjust budget accordingly. We introduce the Contextual Budget Bandit model and multiple algorithms for it.

### 3.1 The Contextual Budget Bandit Model

A Contextual Budget Bandit (CBB) is defined by $\langle N, \mathcal{S}, \mathcal{A}, K, \{r_i^k\}_{i\in[N],k\in[K]}, \{P_i^k\}_{i\in[N],k\in[K]}, \mathcal{F}\rangle$. Departing from the standard RMAB model, we introduce the $[K] = \{1, 2, \dots, K\}$ (finite) **contexts**. A Borel measure $\mathcal{F}$ on $[K]$ specifies the distribution over these contexts, which is known by the decision maker. At each time step, a new context is sampled with respect to $\mathcal{F}$ and globally applies to all arms. $r_i^k, P_i^k, \forall k \in [K]$ are the reward function and the transition probability kernels *specific to context* $k$. We write reward function $r_i^k$ in the expanded form as $r_i(s, a; k)$ (and $P[s \to s' | a, k]$ for $P_i^k$ respectively). Context and state transitions are *independent*.

A policy $\pi$ for CBB (i) pre-specifies budget allocation $\vec{B} := (B_k)_{k\in[K]}$ and (ii) maps $(\mathbf{s}^t, k^t)$ to $\mathbf{a}^t$. The design objective is to maximize the average expected reward: $\mathcal{R}\text{eward}(\pi) := \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{(\mathbf{a},\mathbf{s})\sim\pi, k^t\sim\mathcal{F}} \left[\sum_{i\in[N]} r_i(s_i^t, a_i^t; k^t)\right]$. In standard RMAB, there is a budget for the number of arms pulled at each time point. We generalize this budget notion for CBB as the following Contextual Budget Constraint:

**Definition 3.1** (Contextual Budget Constraint)**.** A policy is said to satisfy *Contextual Budget Constraint* if the number of arms pulled at each time step is constrained by a fix quantity $B_k$ contingent on context, $\sum_{i\in[N]} a_i^t \mathbb{I}(k^t = k) \leq B_k, \forall t, k$ (Constraint I), while the expected budget usage is still bounded by $B$, $\mathbb{E}_{k\sim\mathcal{F}}[B_k] = \sum_{k\in[K]} f_k B_k \leq B$ (Constraint II).

The vanilla Whittle Index Policy for standard RMAB (henceforth Vanilla Whittle) can be directly applied to CBB: it uses a uniform budget for each context $\vec{B} = (B, \dots, B)$. It satisfies the Contextual Budget Constraint. However, the following theorem shows that its performance can be arbitrarily bad, with proof in Appendix A.

**Theorem 1.** For a CBB, denote the Vanilla Whittle Policy's reward as $\mathcal{R}\text{eward}^{\text{VanillaWhittle}}$, and the optimal policy that satisfies context-specific budget constraint as $\mathcal{R}\text{eward}^{\text{CBC-OPT}}$. There exists an instance where, $\frac{\mathcal{R}\text{eward}^{\text{CBC-OPT}}}{\mathcal{R}\text{eward}^{\text{VanillaWhittle}}} \to \infty$, as $N \to \infty$.

### 3.2 Finding the Optimal Budget Allocation

In this section, we introduce the Branch And Bound algorithm that finds the optimal budget allocation, and the Mitosis algorithm that selects the near-optimal budget allocation in a no-regret fashion.

First, we define the *occupancy measure* $\mu$ of a (possibly randomized) policy $\pi$ in CBB as the average visitation probability to a state-action-context tuple $(s, a; k)$: $\mu_i(s, a; k) := \mathbf{Pr}[s_i = s, a_i = a; k], \forall i \in [N]$. We can formulate the problem of maximizing the stationary reward of CBB as an occupancy measure linear program (occupancy-measure LP):

$$\begin{aligned}
&\max_{\mu} \sum_{i\in[N]} \sum_{k\in[K]} \sum_{s_i,a_i} \mu_i(s_i, a_i; k) r_i(s_i, a_i; k) \\
&s.t.\ f_{k'} \left[ \sum_{k\in[K]} \sum_{s_i,a_i} P[s_i \to s_i' | a_i, k] \mu_i(s_i, a_i; k) \right] = \sum_{a_i} \mu_i(s_i', a_i; k'), \forall k', s_i', i \\
&\quad \sum_{k\in[K]} \sum_{a_i,s_i} \mu_i(s_i, a_i; k) = 1, \forall i \in [N] \\
&\quad \sum_{k\in[K]} \sum_{i} \sum_{s_i} \mu_i(s_i, 1; k) \le B \qquad \text{(Relaxed Budget Constraint)} \\
&\quad \mu_i(s_i, a_i, k) \ge 0, \forall i, s_i, a_i, k.
\end{aligned}$$

Solving the occupancy-measure LP induces a policy for CBB, which pulls the arms according to the optimal solution. We call this algorithm COcc. However, this policy has a provable $5/6$ sub-optimality gap. We show the details about this algorithm and the theoretical analysis in Appendix B.

To find the optimal policy of CBB, we proceed as follows. For any budget allocation $\vec{B}$, we denote as $\mathcal{LP}(\vec{B})$ the optimal value of occupancy-measure LP with the following constraint inserted: $\frac{1}{f_k} \sum_{i,s_i} \mu_i(s_i, 1; k) = B_k, \forall B_k \in \vec{B}$. For any polytope region $\mathcal{B} \subseteq \mathcal{B}_0$, we denote as $\mathcal{LP}(\mathcal{B})$ the optimal value of $\max_{\vec{B}\in\mathcal{B}} \mathcal{LP}(\vec{B})$. Fix any budget allocation $\vec{B}$, $\mathcal{R}\text{eward}(\vec{B})$ can be evaluated by randomized simulation procedure that runs an occupancy index policy with $\vec{B}$ and returns the resulting reward of the CBB. We define such as the following two types of *oracle functions*:

- $\mathcal{O}\text{racle}(\vec{B})$: an oracle that returns an accurate estimate of $\mathcal{R}\text{eward}(\vec{B})$ (eg. running many epochs of simulation).
- $\mathcal{O}\text{racle}_{\text{small}}(\vec{B})$: a fast oracle that returns a noisy estimate $\mathcal{R}\text{eward}(\vec{B}) + \epsilon$ (eg. running one epoch of simulation). Assume $\mathbf{E}\epsilon = 0$ and its variance is bounded.

Notice that $\mathcal{LP}(\vec{B}) \ge \mathcal{O}\text{racle}(\vec{B})$, and $\mathcal{LP}(\mathcal{B}) \ge \max_{\vec{B}\in\mathcal{B}} \mathcal{O}\text{racle}(\vec{B})$, because occupancy-measure LP is a relaxation of the original problem. We design two algorithms that leverage this property search for the optimal budget allocation $\vec{B}^\star$ that maximizes $\mathcal{R}\text{eward}(\vec{B})$.

**The Branch And Bound Algorithm** Using $\mathcal{LP}(\vec{B}) \ge \mathcal{O}\text{racle}(\vec{B}), \forall \vec{B}$, this algorithm recursively splits the search region into smaller subregions and prunes subregions if its LP-based upperbound is lower than another's actual reward. Although Branch And Bound is still NP-hard in the worst case, it provides a systematic way to efficiently search. We provide the pseudocode and illustration in Appendix D.

**The Mitosis Algorithm** While Branch And Bound already cuts down the search tree dramatically compared to brute force search, it still calls too many costly $\mathcal{O}$racle evaluations on large-scale problems. To address this issue, we develop the following multi-armed bandit (MAB) framework which allows for more nuanced speed-accuracy trade-off for the evaluation and search of better budget allocations:

**Definition 3.2** (Combinatorial MAB Framework for Solving CBB)**.** To find optimal $\vec{B}$, we define an associated *Multi-Armed Bandit (MAB) problem* by identifying each arm with a vector $\vec{B} \in \mathcal{B}_0$. Pulling arm $\vec{B}$ invokes the fast oracle $\mathcal{O}\text{racle}_{\text{small}}(\vec{B})$ which returns a noisy reward $\mathcal{R}\text{eward}(\vec{B}) + \epsilon$.

For MAB, a standard UCB-type algorithm maintains empirical statistics for each arm and computes an index that serves as an upper confidence bound on the arm's true reward. For each arm (represented by $\vec{B}$) and time (in the MAB system) $\tau$, its upper-confidence-level index is given by $I_t(\vec{B}) := \hat{\mu}_t(\vec{B}) + f\big(N_t(\vec{B}), t\big)$, where $N_t(\vec{B})$ is the number of times arm $\vec{B}$ has been selected and $\hat{\mu}_t(\vec{B})$ the empirical mean reward from $\vec{B}$. $f$ is chosen so that, with high probability, $I_t(\vec{B})$ is an upper bound on the true mean reward $\mu(\vec{B})$. For example, the UCB1 algorithm sets $f\big(N_t(\vec{B}), t\big) = c\sqrt{\frac{\log t}{N_t(\vec{B})}}$, for $c > 0$..

**Addressing the Combinatorial Explosion with StemArm** Naively applying UCB to our setting would result in combinatorial explosion, because the first step of UCB is to initialize all arms' empirical statistics by pulling each arm once. In CBB, the number of arms is the number of integer solutions to $\sum_{k\in[K]} f_k B_k \leq B$, which is $\mathcal{O}((BK)^K)$, so this initialization step is prohibitively expensive. To address this issue, we incorporate the hierarchical tree structure from Branch And Bound to speed up our algorithms.

We design StemArm as a special arm that represents a *group* of candidate budget allocations. Instead of tracking every $\vec{B}$, we use StemArm to encapsulate less-promising budget allocations, which are grouped in polytope regions $\mathcal{B}_m \subseteq \mathcal{B}_0$. Formally,

**Definition 3.3** (StemArm)**.** A *StemArm* represents a union of polytopes subregions StemArm $= \cup_{m=1}^{M} \mathcal{B}_m$. When it is pulled, it splits out a most promising child arm: $\vec{B}_{\text{new}} := \arg\max_{\vec{B}\in\text{StemArm}} \mathcal{L}\text{P}(\vec{B})$, and updates itself by excluding the new arm from itself (StemArm $\leftarrow$ StemArm $\setminus \{\vec{B}_{\text{new}}\}$).

The StemArm's UCB index is set as $I_t(\text{StemArm}) := \max_{\vec{B}\in\text{StemArm}} \mathcal{L}\text{P}(\vec{B})$.

All arms $\vec{B} \in$ StemArm (before they are split out) will never be pulled. While these encapsulated arms have no empirical history for UCB value, , which upperbounds all encapsulated arms' rewards with probability 1. For convenience, denote it as $\mathcal{L}$P(StemArm). Note that because every pull of the StemArm splits out the child arm with the highest $\mathcal{L}$P value, the StemArm's UCB index $\mathcal{L}$P(StemArm) decreases when it splits each time.

Mitosis runs within the Combinatorial MAB Framework to find optimal budget allocation (see Appendix D for the pseudocode and illustration). It begins with the set of candidate arms containing only one StemArm, representing the entire feasible region $\mathcal{B}_0$. At each round, the algorithm selects from candidate arms (either a standard arm $\vec{B}$ or a StemArm) with the highest index. When a standard arm is pulled, we run $r_t(\vec{B}) \leftarrow \mathcal{O}\text{racle}_{\text{small}}(\vec{B})$ to update its empirical statistics. When a StemArm is selected, it splits out a new arm into candidate arms. is named for how the StemArm 'buds' new arms progressively during the algorithm, similar to cell division in mitosis. Mitosis retains the no-regret guarantees of classical MAB algorithms. The proof is in Appendix E.

**Theorem 2.** [No-Regret of the Mitosis Algorithm] Let $\mathcal{A}$ denote the set of arms that have been pulled. After running the algorithm for $T$ rounds, the cumulative regret $R(T) \triangleq \sum_{t=1}^{T} (\mu^\star - \mu_t)$ satisfies

$$R(T) = \sum_{\vec{B}\in\mathcal{A}} \mathbb{E}[N_T(\vec{B})]\,\Delta(\vec{B}) = O\left(\sum_{\vec{B}\in\mathcal{A}} \frac{\log T}{\Delta(\vec{B})}\right).$$

## 4 Experiments

We empirically evaluated the performance of COcc, Branch And Bound and Mitosis in CBB, and compare them against three baseline algorithms: Random, Greedy and the Vanilla Whittle Policy. We first demonstrate the algorithms' performance on numerical simulations. Then, we show that the performance of some of these algorithms could be sensitive to problem instance parameters, while our Mitosis algorithm consistently performs the best, robust against all these different setups that could arise in the real world. Finally, we run CBB experiment on real-world food rescue data which further confirms the superiority of Mitosis. Due to page limit, we detail the results in Appendix G.

## Acknowledgement

We thank 412 Food Rescue for thoughtful discussions and continued support of this work. Co-author Tang would like to thank the faculty and students of the Institute of Theoretical Computer Science at Shanghai University of Finance and Economics for their support and valuable comments. Co-author Raman is also supported by an NSF GRFP Fellowship. Co-author Shi is partially supported by a Google Academic Research Award and a John C. Mascaro Faculty Fellowship.

## References

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

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

Christos H Papadimitriou and John N Tsitsiklis. The complexity of optimal queueing network control. In *Proceedings of IEEE 9th annual conference on structure in complexity Theory*, pages 318–322. IEEE, 1994.

Aditya Mate, Jackson Killian, Haifeng Xu, Andrew Perrault, and Milind Tambe. Collapsing bandits and their application to public health intervention. *Advances in Neural Information Processing Systems*, 33:15639–15650, 2020.

Naveen Janaki Raman, Zheyuan Ryan Shi, and Fei Fang. Global rewards in restless multi-armed bandits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Mushu Li, Jie Gao, Lian Zhao, and Xuemin Shen. Adaptive computing scheduling for edge-assisted autonomous driving. *IEEE Transactions on Vehicular Technology*, 70(6):5318–5331, 2021.

Ibtihal El Mimouni and Konstantin Avrachenkov. Deep q-learning with whittle index for contextual restless bandits: Application to email recommender systems. In Tetiana Lutchyn, Adín Ramírez Rivera, and Benjamin Ricaud, editors, *Proceedings of the 6th Northern Lights Deep Learning Conference (NLDL)*, volume 265 of *Proceedings of Machine Learning Research*, pages 176–183. PMLR, 07–09 Jan 2025. URL https://proceedings.mlr.press/v265/mimouni25a.html.

Zheyuan Ryan Shi, Leah Lizarondo, and Fei Fang. A recommender system for crowdsourcing food rescue platforms. In *Proceedings of the Web Conference 2021*, pages 857–865, 2021.

Jackson A. Killian, Manish Jain, Yugang Jia, Jonathan Amar, Erich Huang, and Milind Tambe. Equitable restless multi-armed bandits: A general framework inspired by digital health, 2023. URL https://arxiv.org/abs/2308.09726.

Shufan Wang, Guojun Xiong, and Jian Li. Online restless multi-armed bandits with long-term fairness constraints. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15616–15624, Mar. 2024. doi: 10.1609/aaai.v38i14.29489. URL https://ojs.aaai.org/index.php/AAAI/article/view/29489.

Christos H. Papadimitriou and John N. Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999. doi: 10.1287/moor.24.2.293. URL https://doi.org/10.1287/moor.24.2.293.

John Gittins, Kevin Glazebrook, and Richard Weber. *Restless Bandits and Lagrangian Relaxation*, chapter 6, pages 149–172. John Wiley & Sons, Ltd, 2011. ISBN 9780470980033. doi: https://doi.org/10.1002/9780470980033.ch6. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470980033.ch6.

Guojun Xiong, Jian Li, and Rahul Singh. Reinforcement learning augmented asymptotically optimal index policy for finite-horizon restless bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8726–8734, Jun. 2022. doi: 10.1609/aaai.v36i8.20852. URL https://ojs.aaai.org/index.php/AAAI/article/view/20852.

Richard R. Weber and Gideon Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27(3):637–648, 1990b. doi: 10.2307/3214547.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2–3):235–256, May 2002. ISSN 0885-6125. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Qing Zhao. *Variants of the Bayesian Bandit Model*, pages 31–56. Springer International Publishing, Cham, 2020. ISBN 978-3-031-79289-2. doi: 10.1007/978-3-031-79289-2_3. URL https://doi.org/10.1007/978-3-031-79289-2_3.

Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE congress on evolutionary computation (CEC)*, pages 1–8. IEEE, 2020.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007.

Biyonka Liang, Lily Xu, Aparna Taneja, Milind Tambe, and Lucas Janson. Context in public health for underserved communities: A bayesian approach to online restless bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):28195–28203, Apr. 2025. doi: 10.1609/aaai.v39i27.35039. URL https://ojs.aaai.org/index.php/AAAI/article/view/35039.

Zhanqiu Guo and Wayne Wang. Contextwin: Whittle index based mixture-of-experts neural model for restless bandits via deep rl. *arXiv preprint arXiv:2410.09781*, 2024.

Xin Chen, I Hou, et al. Contextual restless multi-armed bandits with application to demand response decision-making. *arXiv preprint arXiv:2403.15640*, 2024.

Dimitris Bertsimas, Vivek F. Farias, and Nikolaos Trichakis. On the efficiency-fairness trade-off. *Management Science*, 58(12):2234–2250, 2012. doi: 10.1287/mnsc.1120.1549. URL https://doi.org/10.1287/mnsc.1120.1549.

Zhi Liu and Nikhil Garg. Redesigning service level agreements: Equity and efficiency in city government operations. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 309, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400707049. doi: 10.1145/3670865.3673624. URL https://doi.org/10.1145/3670865.3673624.

Dexun Li and Pradeep Varakantham. Towards soft fairness in restless multi-armed bandits, 2022a. URL https://arxiv.org/abs/2207.13343.

Dexun. Li and Pradeep Varakantham. Efficient resource allocation with fairness constraints in restless multi-armed bandits. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1158–1167. PMLR, 01–05 Aug 2022b. URL https://proceedings.mlr.press/v180/li22e.html.

Shresth Verma, Yunfan Zhao, Sanket Shah, Niclas Boehmer, Aparna Taneja, and Milind Tambe. Group fairness in predict-then-optimize settings for restless bandits. In Negar Kiyavash and Joris M. Mooij, editors, *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pages 3448–3469. PMLR, 15–19 Jul 2024. URL https://proceedings.mlr.press/v244/verma24a.html.

Roch Guérin, Amy McGovern, and Klara Nahrstedt. Report on the nsf workshop on sustainable computing for sustainability (nsf wscs 2024), 2024. URL https://arxiv.org/abs/2407.06119.

Justin Payan and Yair Zick. I will have order! optimizing orders for fair reviewer assignment. *arXiv preprint arXiv:2108.02126*, 2021.

# A Proof of Theorem 1

**Theorem 1.** For a CBB, denote the Vanilla Whittle Policy's reward as $\mathcal{R}\text{eward}^{\text{VanillaWhittle}}$, and the optimal policy that satisfies context-specific budget constraint as $\mathcal{R}\text{eward}^{\text{CBC-OPT}}$. There exists an instance where, $\frac{\mathcal{R}\text{eward}^{\text{CBC-OPT}}}{\mathcal{R}\text{eward}^{\text{VanillaWhittle}}} \to \infty$, as $N \to \infty$.

*Proof.* Consider a CBB instance with $N$ stochastically identical arms. For simplicity we assume that the transition probabilities are such that each arm is always active ($s_i = 1$). Let there be two contexts, where context 1 occurs with probability $f_1 = 1 - \frac{1}{N}$ and context 2 occurs with probability $f_2 = \frac{1}{N}$. For each arm $i$, context 1 generates reward $r_i(s_i = 1, a_i = 1) = \frac{1}{N}$, context 2 generates reward $r_i(s_i = 1, a_i = 1) = N$. Suppose budget $B = 1$.

Consider the policy that leaves all arms idle at context 1, and pulls all $N$ arms at context 2. The policy is feasible because its budget constraint $\vec{B} = (0, N)$ satisfies $f_1 \times 0 + f_2 \times N = \frac{1}{N} \times N = 1 = B$. Its average reward is $N$.

For Vanilla Whittle Index Policy, the good context 2 that has the high reward only occurs with probability$\frac{1}{N}$. And when it happens, we can only pull and get reward from *one* arm. So its average reward is $(1 - \frac{1}{N})\frac{1}{N} + \frac{1}{N} \times N = O(1)$. As $N \to \infty$, the gap between the two policies goes to infinity. □

# B The COcc Policy

## B.1 Contextual Occupancy Index and the Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy

Theorem 1 shows that the Vanilla Whittle Policy can be arbitrarily bad for CBB compared to the optimal policy that satisfies the Contextual Budget Constraint—we need new algorithms that can determine the optimal budget allocation $\vec{B}$ across contexts. In this section, we introduce the class of Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy, which first determines and commit a budget allocation $\vec{B}$, and then pulls arms according to descending orders of Contextual Occupancy Index.

First, we introduce the occupancy-measure LP that is used to compute the Contextual Occupancy Index.

**Definition B.1.** The *occupancy measure* $\mu$ of a (possibly randomized) policy $\pi$ in CBB is the average visitation probability to a state-action-context tuple $(s, a; k)$:

$$\mu_i(s, a; k) := \mathbf{Pr}\left[s_i = s, a_i = a; k\right], \forall i \in [N].$$

We can formulate the problem of maximizing the stationary reward of CBB as a linear program (LP) over occupancy measures:

**Definition B.2.** For a given CBB instance, its *occupancy-measure LP* is

$$\max_{\mu} \sum_{i\in[N]} \sum_{k\in[K]} \sum_{s_i,a_i} \mu_i(s_i, a_i; k) r_i(s_i, a_i; k) \qquad \text{(occupancy-measure LP)}$$

$$s.t.\ f_{k'} \left[ \sum_{k\in[K]} \sum_{s_i,a_i} P[s_i \to s_i' | a_i, k] \mu_i(s_i, a_i; k) \right] = \sum_{a_i} \mu_i(s_i', a_i; k'), \forall k', s_i', i$$

$$\sum_{k\in[K]} \sum_{a_i,s_i} \mu_i(s_i, a_i; k) = 1, \forall i \in [N]$$

$$\sum_{k\in[K]} \sum_{i} \sum_{s_i} \mu_i(s_i, 1; k) \le B \qquad \text{(Relaxed Budget Constraint)}$$

$$\mu_i(s_i, a_i, k) \ge 0, \forall i, s_i, a_i, k.$$

Solving the occupancy-measure LP induces a policy for CBB, which pulls the arms according to optimal solution variables. We refer to it as the following Contextual Occupancy Soft Budget Policy:

**Definition B.3** (Contextual Occupancy Soft Budget Policy (adapted from [Xiong et al., 2022]))**.** Given the optimal solution $\mu^\star(\cdot,\cdot;k)$ to the occupancy-measure LP, the *Contextual Occupancy Soft Budget Policy* $\pi^{\text{soft}}$ pulls an arm $i$ in state $s_i$ and context with probability $\chi_i^\star(s_i,k)$, where

$$\chi_i^\star(s_i,k) = \frac{\mu_i^\star(s_i,1;k)}{\mu_i^\star(s_i,0;k)+\mu_i^\star(s_i,1;k)}.$$

The Contextual Occupancy Soft Budget Policy behaves as if it pulls all arms whose Contextual Occupancy Index defined below, are above a certain threshold.

**Definition B.4** (Contextual Occupancy Index)**.** For an arm $i \in [N]$ in a CBB problem instance, solve its occupancy-measure LP for the instance and obtain its *Contextual Occupancy Index* contingent with state and context as follows:

$$\begin{aligned}\rho_i(s_i,k) &:= \chi_i^\star(s_i,k) r_i(s_i,a_i;k)\\ &= \frac{\mu_i^\star(s_i,1;k)}{\mu_i^\star(s_i,0;k)+\mu_i^\star(s_i,1;k)} r_i(s_i,a_i;k).\end{aligned}$$

Occupancy Index is a more robust index for RMAB compared to the Whittle Index. It behaves the same as Vanilla Whittle when the RMAB is indexable, but the asymptotic optimality for standard RMAB does not require indexability in general (Xiong et al. [2022]). Despite its many appealing properties, the Contextual Occupancy Soft Budget Policy does not satisfies the Contextual Budget Constraint because occupancy-measure LP-Constraint III is a relaxed version—only the *expected* number of arms pulled at each time point for a given context falls within the budget quantity. Therefore, the optimal value of occupancy-measure LP, which can be achieved by the Contextual Occupancy Soft Budget Policy, is an upperbound for all policies that satisfies Contextual Budget Constraint:

$$\mathcal{R}\text{eward}^{\text{VanillaWhittle}} \leq \mathcal{R}\text{eward}^{\text{CBC-OPT}} \leq \mathcal{R}\text{eward}^{\text{LP}}.$$

Therefore, we define the class of Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy as follows:

**Definition B.5** (Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy)**.** A *Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy* determines a budget allocation $\vec{B} \in \mathcal{B}_0 := \{\vec{B} \in \mathbb{N}^K : \sum_{k\in[K]} f_k B_k \leq B\}$ given total budget $B$ and context probabilities $\vec{f}$, and pulls the $B_k$ arms with the highest Contextual Occupancy Index.

We consider within the class of Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy and focus on determining the optimal budget allocation $\vec{B}$. When there is no confusion, we write as $\mathcal{R}\text{eward}(\vec{B})$ the reward of CBB running Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy with budget allocation $\vec{B}$ We aim at designing efficient algorithms that solve the following optimization problem:

$$\begin{aligned}&\max_{\vec{B}} \mathcal{R}\text{eward}(\vec{B})\\ &s.t. \sum_{k\in[K]} f_k B_k \leq B. \end{aligned} \tag{Constraint II}$$

## B.2 The Contextual Occupancy Index (COcc) Policy and its Suboptimality

Contextual Occupancy Soft Budget Policy achieves higher reward than Vanilla Whittle because it shifts budget across time — by saving up budget at bad context and using them when context is good. Guided by this insight, we design the Contextual Occupancy Index Policy (COcc) that mimics the Contextual Occupancy Soft Budget Policy but also satisfies Contextual Budget Constraint. Specifically, it assigns each context the average budget that Contextual Occupancy Soft Budget Policy would have used for each context, and pulls arms with similar priority.

**Definition B.6** (The <u>C</u>ontextual <u>**Occ**</u>upancy Index (COcc) Policy)**.** COcc determines $\vec{B}$ by assigning the budget that Contextual Occupancy Soft Budget Policy would uses (on average) to each context, which can be obtained from the optimal solution of occupancy-measure LP:

$$B_k = \frac{1}{f_k} \sum_{i,s_i} \mu_i^\star(s_i, 1; k), \quad \text{for} k \in [K].$$

At each time step, if the context is $k$, it pulls top-$B_k$ arm s that have the highest positive *Contextual Occupancy Index*.

Standard RMAB problems and the Vanilla Whittle policy correspond to $B_k$ being the same across all $k$, and in such situation, the COcc is equivalent to the Whittle Index Policy, and is asymptotically optimal.[1] However, while numerical studies shows that COcc's performance is usually close to occupancy-measure LP's upperbound, the following theorem shows that the COcc is not optimal for CBB, with proof in Appendix C:

**Theorem 3.** The COcc's asymptotic approximation ratio compared to $\mathcal{R}\text{eward}^{\text{CBC-OPT}}$ is bounded above by $\frac{5}{6}$.

The source of the suboptimality comes from when there are more than one contexts. The proof in Appendix C presents an original mathematical framework for asymptotic analysis.

# C Proof of Theorem 3

**Theorem 3.** The COcc's asymptotic approximation ratio compared to $\mathcal{R}\text{eward}^{\text{CBC-OPT}}$ is bounded above by $\frac{5}{6}$.

*Proof.* **Outline** First, we formally establish the asymptotic framework, which is where the Whittle Index Policy for standard RMAB achieves optimality. Then we introduce how to analyze CBB's in this asymptotic regime. Finally, we present the instance where the $\frac{5}{6}$ bound is achieved.

### Asymptotic Notion

We define the asymptotic notion for analyzing (sub-)optimality for CBB. It is same to the approach for standard RMAB, originally proposed by Weber and Weiss [1990b] for RMAB with stochastically identical arms and generalized to heterogeneous arms by Xiong et al. [2022]:

**Definition C.1** ($\rho$-scaled CBB)**.** Fix a *Base* CBB instance with $M$ arms

$$\langle M, \mathcal{S}, \mathcal{A}, K, \{r_i^k\}_{i\in[M],k\in\mathcal{K}}, \{P_i^k\}_{i\in[M],k\in\mathcal{K}}, \mathcal{F}\rangle.$$

With budget $B \in \mathbb{N}$.

Now, consider each arm being replicated $\rho$ times, with the budget scaled by $\rho$ as well. The new CBB instance has $\rho \times M$ arms, with each of the $M$ arms in the base CBB repeated $\rho$ times. Budget is scaled to $\rho B$.

For a base contextual RMAB instance scaled with $\rho$, when there is no confusion about the base instance we're referring to, denote its reward for any policy $\pi$ as

$$\mathcal{R}\text{eward}^\pi(\rho) := \lim_{T\to\infty} \mathbb{E}_{\vec{a}\sim\pi(\cdot),k\sim\mathcal{F}}\Big[\frac{1}{T}\sum_{t=1}^{T}\sum_{i\in[\rho M]} r_i^k(s_i^t, a_i^t)\Big].$$

For $\rho$-scaled CBB, notice that its reward upperbound from solving the occupancy-measure LP simply scales with $\rho$:

$$\mathcal{R}\text{eward}^{\text{LP}}(\rho) = \rho\overline{\mathcal{R}\text{eward}}(1).$$

We refer to every arm $i \in [M]$ in the base instance as a **type-**$i$ arm, and its $\rho$ replicates in the $\rho$-scaled CBB as the $\rho$ type-$i$ arms.

[1] The asymptotic notion is usually to repeat all arms of an RMAB instance infinitely, along with the budget for the same repeats.

### Asymptotic System Behavior for CBB

In the following section we introduce a new method for analyzing the asymptotic behavior of CBB as $\rho \to \infty$. It is different from the standard approach of Weber and Weiss [1990b].

To ease the complication of notations, we describe our method with CBB that $r_i(s_i = 0, a = 0; k) = r_i(s_i = 0, a = 1; k) = 0, \forall i, k$, and transition probabilities $P_i^k s_i^{t+1} \mid s_i^t, a_i^t = 1] = P_i^k s_i^{t+1} \mid s_i^t, a_i^t = 0], \forall s_i^{t+1}, s_i^t$. In this way it is meaningless to pull inactive arms, since it makes no difference in rewards nor transition probabilities. Generalization to general CBB is without loss of generality.

We care about the proportion of active arms as $\rho \to \infty$ of the $\rho$. The following technical lemma characterizes the dynamic of arms:

**Lemma 1.** Denote as $a_i^t$ the proportion of type-$i$ active arms at any time point $t$ under given policy $\pi$. Conditional on $a_i^t$ and context $k$, $a_i^{t+1}$'s distribution converges to a Direc Delta function $\delta_{\text{shift}}(\cdot)$, shifted with $\mathbb{E}[a_i^{t+1} \mid a_i^t, k]$ as the total number of arms $\rho \to \infty$. In other words,

$$f(a_i^{t+1} \mid a_i^t, k) = \delta_{\mathbb{E}[a_i^{t+1} \mid a_i^t, k]}(a_i^{t+1}), \tag{1}$$

where, let $B_{i,k}$ be the number of active type-$i$ arms pulled by the policy at context $k$:

$$\mathbb{E}[a_i^{t+1} \mid a_i^t, k] = \min(a_i^t, \frac{B_{i,k}}{\rho}) P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1] \tag{2}$$

$$+ (a_i^t - \min(a_i^t, \frac{B_{i,k}}{\rho}) \tag{3}$$

$$\times P_i^k s_i^{t+1} = 0 \mid s_i^t = 1, a_i^t = 1] \tag{4}$$

$$+ (1 - a_i^t) P_i^k s_i^{t+1} = 1 \mid s_i^t = 0] \tag{5}$$

*Proof.* The sketch of the proof is that, the **number** of active arms is sum of Binomial random variables, with parameters given the by the policy and transition probabilities. As total number of arms $\rho \to \infty$, each Binomial variable divided by $\rho$ converges to (shifted) Direc Delta. Therefore, the *proportion* of active arms is also (shifted) Direc Delta.

**Notes on Binomial Distribution** To make later analysis clear, first consider a single binomial random variable $X$ with $N$ experiments and success rate $p$ (i.e. $X \sim \text{Bin}(N, p)$). For any $x \in [0, 1]$ (assume $Nx$ is integer):

$$P[X = Nx] = \binom{N}{Nx} p^{Nx} (1-p)^{N(1-x)}$$

$$\text{Apply Stirling's Formula: } n! \sim \sqrt{2\pi n} \cdot (\frac{n}{e})^n$$

$$= \sqrt{\frac{1}{x(1-x)N}} \cdot \left( (\frac{p}{x})^x (\frac{1-p}{1-x})^{(1-x)} \right)^N$$

It can be verified that $(\frac{p}{x})^x (\frac{1-p}{1-x})^{(1-x)} < 1$ for $x \neq p$. Therefore, as $N \to \infty$

$$P[X = xN] = \begin{cases} \sqrt{\frac{1}{x(1-x)N}} \to \infty & x = p \\ \sqrt{\frac{1}{x(1-x)N}} & \\ \quad \times \mathcal{O}(\left( (\frac{p}{x})^x (\frac{1-p}{1-x})^{(1-x)} \right)^N) \to 0 & x \neq p \end{cases}$$

Therefore, say if we let $f(x) = P[X = xN]$, $f(\cdot)$ is a shifted-to-$p$ Direc Delta function.

**Stationary Distribution Contextual Budget Bandit** Let $A_i^t := a_i^t \rho$ denote the **number of active type-$i$ arms** at time point $t$. Conditional on current $A_i^t$ and context $k$,

- $\min(A_i^t, B_{i,k})$ arms are pulled, where each arm remains active w.p. $P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1]$.

- Each of $\rho - A_i^t$ inactive arms transfers back to active w.p. $P_i^k s_i^{t+1} = 1 \mid s_i^t = 0]$,

Therefore, the number of active arms at next period $A_i^{t+1}$ is the sum of three binomial random variables:

$$\begin{aligned}
&\{A_i^{t+1} \mid A_i^t, k\} && (6)\\
\sim\ & \underbrace{\text{Bin}(\min\{A_i^t, B_{i,k}\}, P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1])}_{\text{active arms pulled staying active}} && (7)\\
&+ \underbrace{\text{Bin}(A_i^t - \min\{A_i^t, B_{i,k}\}, P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 0])}_{\text{idle active arms staying active}} && (8)\\
&+ \underbrace{\text{Bin}(\rho - A_i^t, P_i^k s_i^{t+1} = 1 \mid s_i^t = 0])}_{\text{inactive arms transfer back to active}}. && (9)
\end{aligned}$$

Scaled by $\rho \to \infty$, each of the above binomial distribution converges to a Direc Delta function centered on its mean. Since adding up random variables is equivalent to taking convolution of their probability mass functions—Direc Delta functions are closed under convolution—random variable $a_i^{t+1} = \frac{A_i^{t+1}}{\rho}$'s probability mass function is a Direc Delta shifted by $\frac{1}{\rho}\mathbb{E}[A_i^{t+1} \mid A_i^t, k]$. □

The lemma implies, the *proportion* of active arms $a_i^t$ evolve "almost deterministically"—more precisely speaking, fix any policy $\pi$, if at current time step the proportion of active arms is $a_i^t$, context is $k$, the next time step will have $(\mathbb{E}[a_i^{t+1} \mid a_i^t, k])\%$ active arms almost surely, where $(\mathbb{E}[a_i^{t+1} \mid a_i^t, k])$ is given by the following:

$$\begin{aligned}
&\mathbb{E}[a_i^{t+1} \mid a_i^t, k]\\
=\ & \frac{1}{\rho}\mathbb{E}[A_i^{t+1} \mid A_i^t, k]\\
=\ & \frac{1}{\rho}\mathbb{E}[\underbrace{\text{Bin}(\min\{A_i^t, B_{i,k}\}, P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1])}_{\text{active arms pulled}}\\
&+ \underbrace{\text{Bin}(A_i^t - \min\{A_i^t, B_{i,k}\}, P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 0])}_{\text{untouched active arms}}\\
&+ \underbrace{\text{Bin}(\rho - A_i^t, q_i)}_{\text{inactive arms}}]\\
=\ & \frac{1}{\rho}(\min\{A_i^t, B_{i,k}\} \cdot P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1]\\
&+ (A_i^t - \min\{A_i^t, B_{i,k}\}) \cdot P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 0]\\
&+ (\rho - A_i^t)q_i)
\end{aligned}$$

Denote $\beta_{i,k} := \frac{B_{i,k}}{\rho}$:

$$\begin{aligned}
&\mathbb{E}[a_i^{t+1} \mid a_i^t, k] && (10)\\
=\ & \min(a_i^t, \beta_{i,k}) \cdot P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 1] && (11)\\
&+ \max(a_i^t - \beta_{i,k}, 0) \cdot P_i^k s_i^{t+1} = 1 \mid s_i^t = 1, a_i^t = 0] && (12)\\
&+ (1 - a_i^t)P_i^k s_i^{t+1} = 1 \mid s_i^t = 0] && (13)
\end{aligned}$$

If, current time step's proportion of active arms is $x \in [0, 1]$, with probability $f_k$ context $k$ occurs, then the next time step's active-arm proportion will be $y = \mathbb{E}[a_i^{t+1} \mid x, k]$ (as given in 11-13) w.p. $f_k$. And for each $y$, define its inverse

$$\mathcal{X}(y) := \{(x, k) : \mathbb{E}[a_i^{t+1} \mid x, k] = y\}. \quad (14)$$

Denote the stationary distribution of proportion of active arms as $\pi : [0, 1] \to [0, 1]$, it should satisfy:

$$\pi(y) = \sum_{(x,k) \in \mathcal{X}(y)} f_k \pi(x). \tag{15}$$

### C.1 A 5/6 Approximation Upperbound.

**An adversarial instance** Consider a base CBB example with only one type of arm (i.e., $M = 1$). Let there be $\rho$ copies of this arm in the scaled setting as $\rho \to \infty$. We drop the index $i$ for convenience. Suppose there are two contexts, $k \in \{1, 2\}$, each occurring with probability $f_1 = f_2 = 0.5$.

Let $\epsilon > 0$. The transition probabilities and rewards are defined as follows.

- Context 1: transition probabilities is

$$\begin{aligned}
&P^1[s^{t+1} = 1 \mid s = 1, a = 1] = 1 - \epsilon \\
&P^1[s^{t+1} = 0 \mid s = 1, a = 1] = \epsilon \\
&P^1[s^{t+1} = 1 \mid s = 1, a = 0] = 1 \\
&P^1[s^{t+1} = 0 \mid s = 1, a = 0] = 0 \\
&P^1[s^{t+1} = 1 \mid s = 1, \forall a = 0, 1] = 1 \\
&P^1[s^{t+1} = 0 \mid s = 1, \forall a = 0, 1] = 0
\end{aligned}$$

  reward for context 1:

$$\begin{aligned}
&r(s^t = 1, a^t = 1; k = 1) = 1 \\
&r(s^t = 1, a^t = 0; k = 1) = 0 \\
&r(s^t = 0, a^t = 1; k = 1) = 0 \\
&r(s^t = 1, a^t = 0; k = 1) = 0
\end{aligned}$$

- Context 2: transition probabilities is

$$\begin{aligned}
&P^2[s^{t+1} = 1 \mid s = 1, a = 1] = 0 \\
&P^2[s^{t+1} = 0 \mid s = 1, a = 1] = 1 \\
&P^2[s^{t+1} = 1 \mid s = 1, a = 0] = 1 \\
&P^2[s^{t+1} = 0] \mid s = 1, a = 0] = 0 \\
&P^2[s^{t+1} = 1 \mid s = 1, \forall a = 0, 1] = 1 \\
&P^2[s^{t+1} = 0 \mid s = 1, \forall a = 0, 1] = 0
\end{aligned}$$

  reward for context 2:

$$\begin{aligned}
&r(s^t = 1, a^t = 1; k = 1) = 1 + \epsilon \\
&r(s^t = 1, a^t = 0; k = 1) = 0 \\
&r(s^t = 0, a^t = 1; k = 1) = 0 \\
&r(s^t = 1, a^t = 0; k = 1) = 0
\end{aligned}$$

**Bugdet** Assume that budget is $1/3$ of the number of total arms. I.e. in the $\rho$-scaled instance, $B = \lfloor \frac{1}{3} \rfloor$. As the scaling factor $\rho \to \infty$, we can without loss of generality assumes that it's an interger.

**The Reward for COcc** The occupancy-measure LP for the base instance simplifies to

$$\max_{\mu, B_k} \quad \mu(1,1,1) + \mu(1,1,2)(1+\epsilon)$$

subject to

$$(1 - P[s=1]) = \epsilon\mu(1,1,1) + \mu(1,1,2)$$
$$\mu(1,1,k) \leq \frac{1}{2}P[s=1], \forall k = 1,2$$
$$\mu(1,1,1) + \mu(1,1,2) \leq \frac{1}{3}$$

The COcc then allocate budget following the optimal solution ($\mu^\star$) of the occupancy-measure LP. For the $\rho$-scaled CBB with total budget $B = \frac{1}{3}\rho$, the budget allocation of COcc is

$$B_1 = \rho \times \frac{1}{f_1}\mu^\star(1,1,1) = 0,$$
$$B_2 = \rho \times \frac{1}{f_2}\mu^\star(1,1,2) = \frac{2}{3}.$$

From (11-13) we obtain, as $\rho \to \infty$, the transition dynamic of the proportion of active arms $x^t \to x^{t+1}$ in RMAB:

- With probability $f_1 = 0.5$, context $k = 1$:
$$x^{t+1} = \mathbb{E}[a_i^{t+1} \mid x^t, k] = 1;$$
- With probability $f_2 = 0.5$, context $k = 2$:
$$x^{t+1} = \mathbb{E}[a_i^{t+1} \mid x^t, k] = \max(x^t - \frac{2}{3}, 0) + 1 - x^t.$$

From 14 and 15 we obtain the stationary distribution $\pi$ under `Policy`$^\star$: (actually, guess-and-verify)

$$\pi(\frac{1}{3}) = \frac{1}{3},$$
$$\pi(\frac{2}{3}) = \frac{1}{6}$$
$$\pi(1) = \frac{1}{2},$$
$$\pi(x) = 0, \text{otw.}$$

When the proportion of active arms= $\frac{1}{3}$—only half of the budget is utilized. This happens, as give above, w.p. $\pi(\frac{1}{3}) = \frac{1}{3}$. So the reward as $\rho \to \infty$ is

$$\mathcal{R}\text{eward}^{\text{COcc}}(\rho)$$
$$= f_2(\frac{1}{3}\rho\pi(\frac{1}{3}) + \frac{2}{3}\rho(\pi(\frac{2}{3}) + \pi(1))$$
$$= 0.5\rho(\frac{1}{9} + \frac{4}{9}) = \frac{5}{18}\rho$$

**Optimal Budget Allocation** However, notice that the other context $k = 1$ is almost always active (it has probability $p = \epsilon$ of transfer to inactive). Therefore, if we allocate all budget to context 1:

$$B_1 = \frac{2}{3}, B_2 = 0$$

. The stationary reward for the optimal budget allocation is

$$\mathcal{R}\text{eward}^{\text{ContextOpt}}(\rho) = \frac{1}{3}\rho$$

Therefore, the COcc's approximation is bounded above by $\frac{5}{6}$.

$$\lim_{\rho \to \infty} \frac{\mathcal{R}\text{eward}^{\text{COcc}}(\rho)}{\mathcal{R}\text{eward}^{\text{ContextOpt}}(\rho)} = \frac{5}{6}.$$

□

### C.2 Remark: Closed-form unavailable

Ending remark for this Appendix section, and as a complement to the asymptotic analysis of CBB, we provided the following example, where, the closed-form solution of the staionary distribution of the proportion of active arms can only be calculated numerically but not characterized in clean closed-form as the above example.

**Single-Type Base Example 2** Consider a base CBB example with only one type of arm (i.e., $M = 1$). Let there be $\rho$ copies of this arm in the scaled setting as $\rho \to \infty$. We drop the index $i$ for convenience. Suppose there are two contexts, $k \in \{1, 2\}$, each occurring with probability $f_1 = f_2 = 0.5$. **Transition Probabilities and Rewards.** Let $\epsilon > 0$. The transition probabilities and rewards are defined as follows.

**Transition Probabilities and Rewards.** Let $\epsilon > 0$. The transition probabilities and rewards are defined as follows.

- **Context 1:** *Transition probabilities*:

$$\begin{aligned}
P^1[s^{t+1} = 1 \mid s = 1, a = 1] &= 1 - \epsilon,\\
P^1[s^{t+1} = 0 \mid s = 1, a = 1] &= \epsilon,\\
P^1[s^{t+1} = 1 \mid s = 1, a = 0] &= 1,\\
P^1[s^{t+1} = 0 \mid s = 1, a = 0] &= 0,\\
P^1[s^{t+1} = 1 \mid s = 0, \forall a = 0, 1] &= \frac{1}{2}\\
P^1[s^{t+1} = 0 \mid s = 0, \forall a = 0, 1] &= \frac{1}{2}
\end{aligned}$$

  *Rewards*:

$$\begin{aligned}
r(s^t = 1, a^t = 1; k = 1) &= 1,\\
r(s^t = 1, a^t = 0; k = 1) &= 0,\\
r(s^t = 0, a^t = 1; k = 1) &= 0,\\
r(s^t = 0, a^t = 0; k = 1) &= 0.
\end{aligned}$$

- **Context 2:** *Transition probabilities*:

$$\begin{aligned}
P^2[s^{t+1} = 1 \mid s = 1, a = 1] &= 0,\\
P^2[s^{t+1} = 0 \mid s = 1, a = 1] &= 1,\\
P^2[s^{t+1} = 1 \mid s = 1, a = 0] &= 1,\\
P^2[s^{t+1} = 0 \mid s = 1, a = 0] &= 0,\\
P^2[s^{t+1} = 1 \mid s = 0, \forall a = 0, 1] &= \frac{1}{2}\\
P^2[s^{t+1} = 0 \mid s = 0, \forall a = 0, 1] &= \frac{1}{2}
\end{aligned}$$

  *Rewards*:

$$\begin{aligned}
r(s^t = 1, a^t = 1; k = 2) &= 1 + \epsilon,\\
r(s^t = 1, a^t = 0; k = 2) &= 0,\\
r(s^t = 0, a^t = 1; k = 2) &= 0,\\
r(s^t = 0, a^t = 0; k = 2) &= 0.
\end{aligned}$$

**Budget Constraint.** Assume the budget in each round is a fraction of the total number of arms. For concreteness, let the budget be

$$B = \left\lfloor \tfrac{1}{4}\,\rho \right\rfloor,$$

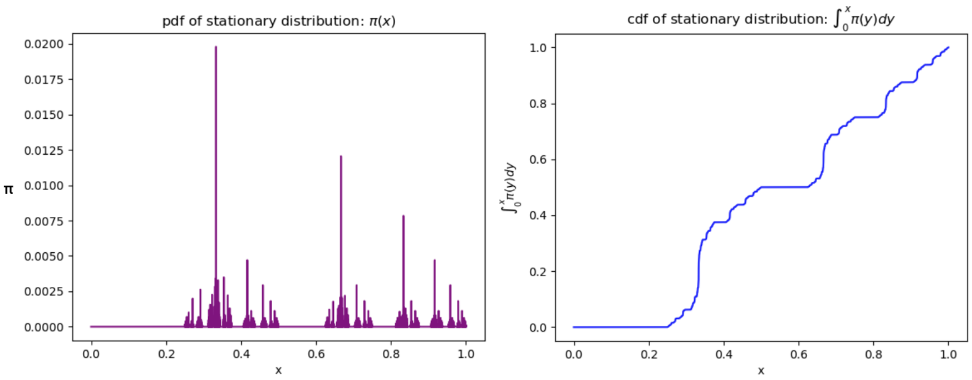

Figure 1: Calculated stationary distribution.

so that we may activate at most $\lfloor \frac{\rho}{4} \rfloor$ arms (out of $\rho$). As $\rho \to \infty$, we can assume without loss of generality that $B = \frac{\rho}{3}$ is an integer.

The occupancy-measure LP simplifies to

$$\begin{aligned} \max_{\mu,\, B_k} \quad & \mu(1,1,1) \;+\; \mu(1,1,2)\,\big(1+\epsilon\big) \\ \text{subject to} \quad & \tfrac{1}{2}\big(1 - P[s=1]\big) \;=\; \epsilon\,\mu(1,1,1) \;+\; \mu(1,1,2), \\ & \mu(1,1,k) \;\le\; \tfrac{1}{2}\,P[s=1], \quad \forall\, k \in \{1,2\}, \\ & \mu(1,1,1) + \mu(1,1,2) \;\le\; 0.25 \end{aligned}$$

The optimal solution is $P^\star[s=1] = 0.5, \mu^\star(1,1,1) = 0, \mu^\star(1,1,2) = 0.25$. Similarly, COcc would allocate budget so that

$$\begin{aligned} B_1 &= 0, \\ B_2 &= 0.5\rho. \end{aligned}$$

Therefore, from 14 and 15 we obtain for the stationary distribution $\pi$:

$$\pi(y) = \begin{cases} 0 & y \in (0, \frac{1}{4}) \\ \frac{1}{2}\pi(\frac{1}{2}) & y = \frac{1}{4} \\ \frac{1}{2}\pi(1-2y) + \frac{1}{2}\pi(2y) & y \in (\frac{1}{4}, \frac{1}{2}] \\ \frac{1}{2}\pi(2y-1) & y \in (\frac{1}{2}, 1]. \end{cases} \tag{16}$$

It doesn't have a clean closed-form solution. But the stationary of proportion of active arms ($\pi(\cdot)$ can be solved numerically, as shown in Figure 1. As shown in Figure 1, for nontrivial probability, the proportion of active arms is less than $0.5$—less than the required active arms to pull. The stationary reward can be calculated as

$$\begin{aligned} & \mathcal{R}\text{eward}^{\text{COcc}} \\ &= \int_0^1 \sum_k f_k r(k) \rho \min(\beta_k, x) \pi(x)\, dx \\ &= \rho \int_0^1 \frac{1}{2}(1+\epsilon) \min(\frac{1}{2}, x) \pi(x)\, dx \\ &\approx \rho(1+\epsilon) 0.214. \end{aligned}$$

However, notice that the other context $k = 1$ is almost always acive (it has probability $p = \epsilon$ of transfer to inactive). Therefore, if we allocate all budget to it—almost all arms will be active all the time, and reward of $r(1) = 1$ can be accured at every pull. By back-on-the-envelope calculation, under this budget allocation (all to context 1) the system generate (almost) exactly $\frac{1}{4}$ reward. Therefore, this instance give an lowerbound of $0.214/0.25 = 0.856$ impossibility lowerbound for the LP-induced budgets.

## D Pseudocode and Visualizations

**Algorithm 1** Branch And Bound

```
Input: Feasible Region $\mathcal{B}_0$, $\mathcal{L}$P, $\mathcal{O}$racle
Output: Budget Allocation $\vec{B}^*$
Initialize: $R^{\text{OPT}} \leftarrow -\infty$, $\vec{B}^{\text{OPT}} \leftarrow$ None, Queue $Q \leftarrow \{\mathcal{B}_0\}$.
while $Q \neq \varnothing$ do
    Dequeue $\mathcal{B} \leftarrow Q$.pop().
    if $\mathcal{L}$P$(\mathcal{B}) < R^{\text{OPT}}$ then
        continue.{prune $\mathcal{B}$}.
    end if
    $\vec{B}^* \leftarrow \vec{B}^{\mathcal{L}\text{P}}(\mathcal{B})$ {most promising $\vec{B} \in \mathcal{B}$}
    if $\mathcal{O}$racle$(\vec{B}^*) > L^*$ then
        Update $L^* \leftarrow \mathcal{O}$racle$(\vec{B}^*)$ and $\vec{B}^{\text{OPT}} \leftarrow \vec{B}^*$.
    end if
    Branch: Partition $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$
    $Q \leftarrow Q \cup \{\mathcal{B}_1, \mathcal{B}_2\}$.
end while
return $\vec{B}^*$.
```

**Algorithm 2** Mitosis

```
Input: Feasible region $\mathcal{B}_0$, LP upper-bound function $\mathcal{L}$P, fast oracle $\mathcal{O}$racle$_{\text{small}}$
Auxiliary: UCB-type no-regret algorithm $I_t(\cdot)$
Output: Budget allocation $\vec{B}^*$ with high empirical reward
Initialize:
        • Initialize StemArm := $\{\mathcal{B}_0\}$.
        • Candidate set (heap) $\mathcal{A} \leftarrow \{$StemArm$\}$.
        • Set time $t \leftarrow 0$.
while $t < T$ and stopping condition not met do
    Select from candidate set w.r.t. UCB index:
        $$a^* \leftarrow \arg\max_{a \in \mathcal{A}} I_t(a),$$
    if $a^*$ is a standard arm (i.e., corresponds to a specific $\vec{B}$) then
        Pull arm and observe reward $r_t(a^*) \leftarrow \mathcal{O}$racle$_{\text{small}}(\vec{B})$.
        Update the empirical statistics $N_t(a^*)$ and $\hat{\mu}_t(a^*)$.
    else
        // $a^*$ is a StemArm; splits out and pull new arm.
        StemArm splits arm $a^*$ to obtain a new standard arm $a'$ with allocation $\vec{B}_{a'} = \vec{B}_{\text{new}}$.
        Insert $a'$ into the candidate set: $\mathcal{A} \leftarrow \mathcal{A} \cup \{a'\}$.
    end if
    $t \leftarrow t + 1$, update the candidate set $\mathcal{A}$ with $I_{t+1}(\cdot)$.
end while
return The allocation $\vec{B}^*$ corresponding to the arm in $\mathcal{A}$ with the highest empirical mean reward.
```

## E No-Regret Guarantee for the Mitosis Algorithm

In the MAB framework each arm represents a candidate budget allocation $\vec{B}$ from the feasible set

$$\mathcal{B}_0 \triangleq \Big\{\vec{B} \in \mathbb{N}^K : \sum_{k=1}^{K} B_k \leq B\Big\}.$$

Pulling an arm $\vec{B}$ corresponds to calling the fast oracle $\mathcal{O}\text{racle}_{\text{small}}(\vec{B})$ (with `epoch` $= 1$) which returns a noisy estimate of the reward $\mu(\vec{B})$. Thus, by running a Multi-Armed Bandit (MAB)

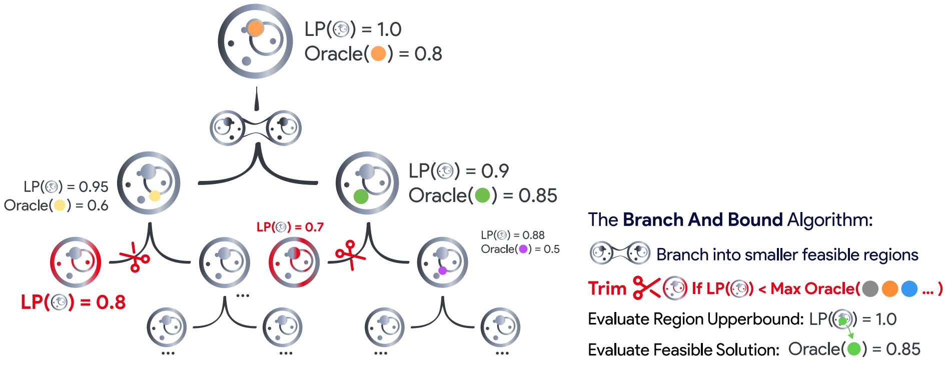
LP(◎) = 1.0
Oracle(●) = 0.8
LP(◎) = 0.95
Oracle(●) = 0.6
LP(◎) = 0.9
Oracle(●) = 0.85
LP(◎) = 0.7
LP(◎) = 0.88
Oracle(●) = 0.5
LP(◎) = 0.8
...
The Branch And Bound Algorithm:
Branch into smaller feasible regions
Trim ✂ ◎ If LP(◎) < Max Oracle(● ● ● ... )
Evaluate Region Upperbound: LP(◎) = 1.0
Evaluate Feasible Solution: Oracle(●) = 0.85

Figure 2: Visualization of Branch And Bound

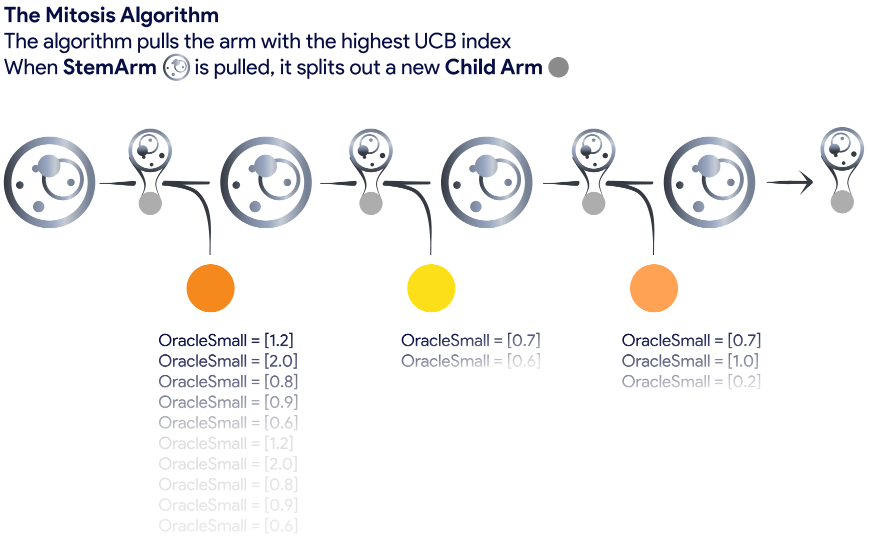

Figure 3: Visualization of Mitosis

algorithm over the arms $\vec{B} \in \mathcal{B}_0$ we aim to select the arm with the highest expected reward without having to estimate $\mu(\vec{B})$ for every $\vec{B}$.

**Definition** (Reward Regret)**.** Let

$$\mu^\star \triangleq \max_{\vec{B} \in \mathcal{B}_0} \mu(\vec{B})$$

and denote by $\vec{B}_t$ the budget allocation (arm) chosen at time $t$. Then the instantaneous regret at time $t$ is

$$\Delta_t \triangleq \mu^\star - \mu(\vec{B}_t),$$

and the cumulative (reward) regret over a time horizon $T$ is defined as

$$R(T) \triangleq \sum_{t=1}^{T} \Delta_t = \sum_{t=1}^{T} \Big( \mu^\star - \mu(\vec{B}_t) \Big).$$

The goal is to design an algorithm whose cumulative regret grows sublinearly in $T$; that is, $\frac{R(T)}{T} \to 0$ as $T \to \infty$. In our setting, the optimal budget allocation $\vec{B}^\star$ (with $\mu(\vec{B}^\star) = \mu^\star$) will be identified as $T$ increases.

**Theorem 2.** [No-Regret of the Mitosis Algorithm] Let $\mathcal{A}$ denote the set of arms that have been pulled. After running the algorithm for $T$ rounds, the cumulative regret $R(T) \triangleq \sum_{t=1}^{T} (\mu^\star - \mu_t)$ satisfies

$$R(T) = \sum_{\vec{B} \in \mathcal{A}} \mathbb{E}[N_T(\vec{B})] \, \Delta(\vec{B}) = O\left(\sum_{\vec{B} \in \mathcal{A}} \frac{\log T}{\Delta(\vec{B})}\right).$$

*Proof.* The proof is built on the classical UCB1 analysis of Auer et al. [2002]. In the Mitosis Algorithm (Algorithm 2), each arm $\vec{B}$ is initialized with its upperbound $\mathcal{L}\text{P}(\vec{B})$. The algorithm maintains two types of arms:

- **Unpromising arms:** Arms that are encapsulated in StemArm, who has not yet been pulled. Their index is given by $\mathcal{L}\text{P}(\vec{B})$.
- **Candidate arms:** Arms that have been pulled at least once. For these, the UCB index at time $t$ is defined as
$$I_t(\vec{B}) = \hat{\mu}_t(\vec{B}) + c\sqrt{\frac{\log t}{N_t(\vec{B})}},$$
where $\hat{\mu}_t(\vec{B})$ is the empirical mean, $N_t(\vec{B})$ is the number of pulls, and $c > 0$ is a constant.

The algorithm runs for $T$ rounds; by the end, let $\mathcal{A}$ denote the final set of candidate arms. We consider two cases based on the location of the optimal arm $\vec{B}^\star$.

*Case 1:* $\vec{B}^\star \in \mathcal{A}$. In this case, the optimal arm has been pulled at least once. Therefore, the candidate arms $\mathcal{A}$ form a sub-MAB instance where we can directly apply UCB1's regret bound on; the arms in StemArm are not pulled anyway, so they do not contribute to regret. Standard UCB1 analysis (using a peeling argument and concentration inequalities, see Auer et al. [2002]) shows that the expected pulls on any suboptimal arms, denoted by $N_t(\vec{B})$, satisfy

$$\mathbb{E}[N_t(\vec{B})] = \mathcal{O}\left(\frac{\log t}{\Delta(\vec{B})^2}\right). \tag{17}$$

Thus, the regret incurred by arms in $\mathcal{A}$ is

$$R(T) = \sum_{\vec{B} \in \mathcal{A}} \mathbb{E}[N_T(\vec{B})] \, \Delta(\vec{B}) = \mathcal{O}\left(\sum_{\vec{B} \in \mathcal{A}} \frac{\log T}{\Delta(\vec{B})}\right).$$

*Case 2:* $\vec{B}^\star \notin \mathcal{A}$. We prove that this case will never happen by contradiction. In other words, the optimal arm that represents the optimal budget solution for CBB will also be budded out by the StemArm in Mitosis algorithm, as the time horizon is sufficiently large.

Let $\vec{B}^{2nd} := \arg\max_{a \in \mathcal{A}} \mu(a)$ be the arm with the highest mean in the candidate arms. Since the number of pulls for all other suboptimal arms satisfies (17), the number of pulls for $\vec{B}^{2nd}$ grows linearly with $t$:

$$\mathbb{E}[N_T(\vec{B}^{2nd})] = T - \mathcal{O}(\log T).$$

Since $\vec{B}^\star \in$ StemArm, by the end of the algorithm, the StemArm's index ($\mathcal{L}$P(StemArm))is smaller than the UCB index of $\vec{B}^{2nd}$. Since $\vec{B}^\star \in$ StemArm, we have

$$\mathcal{L}\text{P}(\text{StemArm}) \geq \mathcal{L}\text{P}(\vec{B}^\star) \geq \mu(\vec{B}^\star).$$

then, the event that the suboptimal arm $\vec{B}^{2nd}$'s UCB index be strictly greater than the optimal arm's mean $\mu(\vec{B}^\star)$: for $\mu(\vec{B}^\star) > \hat{\mu}_T(\vec{B}^{2nd})$,

$$\mathbf{Pr}\left[I_T(\vec{B}^{2nd}) \geq \mu(\vec{B}^\star)\right] \tag{18}$$

$$= \mathbf{Pr}\left[\hat{\mu}_T(\vec{B}^{2nd}) + c\sqrt{\frac{\log(\mathcal{O}(T))}{N_T(\vec{B}^{2nd})}} \geq \mu(\vec{B}^\star)\right] \tag{19}$$

$$\leq \exp\left\{-\frac{\mathcal{O}(T)(\mu(\vec{B}^{2nd}) - \mu(\vec{B}^\star))^2}{c'}\right\} \tag{20}$$

$$\tag{21}$$

The probability declines exponentially. Since the probability of the event in Case 2 decays exponentially with $T$, its contribution to the overall expected regret is negligible compared to the regret in Case 1. In other words, with probability tending to one as $T \to \infty$, the optimal arm $\vec{B}^\star$ is eventually pulled and becomes a candidate arm. Therefore, the overall expected regret of the Mitosis algorithm is dominated by the regret incurred in Case 1, and we have

$$R(T) = \mathcal{O}\left(\sum_{\vec{B}\in\mathcal{A}} \frac{\log T}{\Delta(\vec{B})}\right).$$

Overall the regret of Mitosis is controlled by the classical UCB1 guarantee, up to a constant factor, and hence the algorithm achieves near-optimal performance. This completes the proof.

□

## F Fairness in CBB

**Definition F.1.** Define fairness index for *any* CBB policy $\pi$ as follows[2]:

$$\mathcal{F}\text{airness}(\pi) := \min_{k\in[K]} \frac{1}{f_k} \frac{\lim_{T\to\infty} \mathbf{E}_{(a,s)\sim\pi, k_t\sim\mathcal{F}}[\sum_{t=1}^{T}\sum_{i\in[N]} r_i(s_i, a_i; k_t)\mathbb{I}\{k_t = k\}]}{\lim_{T\to\infty} \mathbf{E}_{(a,s)\sim\pi, k_t\sim\mathcal{F}}[\sum_{t=1}^{T}\sum_{i\in[N]} r_i(s_i, a_i; k_t)]}.$$

This definition captures the idea that a fair algorithm should achieve reward for each context in proportion to the context's frequency. Or, it can also be understood as minimum over the average reward of each context divided by total average reward. Observe that $\mathcal{F}\text{airness} \in [0, 1]$ by construction. $\mathcal{F}\text{airness} = 1$ indicates perfect fairness, $\mathcal{F}\text{airness} = 0$ indicates extreme unfairness, where at least one context receives no reward regardless of its occurance frequency. In requirement of fairness within the Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy class, we consider the following optimization:

$$\max_{\vec{B}} \mathcal{R}\text{eward}(\vec{B})$$

$$s.t. \sum_{k\in[K]} f_k B_k \leq B \tag{Constraint II}$$

$$\mathcal{F}\text{airness}(\vec{B}) \geq \theta \tag{Constraint for Fairness}$$

**Modifying COcc, Branch And Bound and Mitosis for Fairness** We obtain a fair version of COcc by adding the following *Linear Constraint for Fairness* to the occupancy-measure LP:

$$\underbrace{\frac{1}{f_k}\sum_{i\in[N]}\sum_{s_i,a_i} \mu_i(s_i, a_i; k) r_i(s_i, a_i; k)}_{\text{reward for type } k} \geq \theta \underbrace{\left(\sum_{i\in[N]}\sum_{k\in[K]}\sum_{s_i,a_i} \mu_i(s_i, a_i; k) r_i(s_i, a_i; k)\right)}_{\text{total reward}}, \quad \forall k \in [K] \tag{Linear Constraint for Fairness}$$

[2] Assume that $f_k > 0$ for all $k \in [K]$.

Branch And Bound and Mitosis can be directly modified to incorporate the fairness constraint by adjusting the $\mathcal{L}$P$upperbounds$ and the oracle functions. Let the fairness requirement be $\theta \in [0, 1]$. We define the following methods for evaluating CBB's reward under fairness constraint:

**Definition** (Fairness-aware $\mathcal{L}$P Upperbounds and Oracles)**.** For any budget allocation $\vec{B}$, polytope region $\mathcal{B}$ and fairness requirement $\theta$, we extend the definitions of $\mathcal{L}$P upperbounds and oracles as follows:

- $\mathcal{L}\mathrm{P}_{\text{fair}}(\vec{B})$: the optimal value of the occupancy-measure LP with additional Linear Constraint for Fairness inserted.
- $\mathcal{L}\mathrm{P}_{\text{fair}}(\mathcal{B})$: the maximum of $\mathcal{L}\mathrm{P}_{\text{fair}}(\vec{B})$ over all $\vec{B} \in \mathcal{B}$.
- $\mathcal{O}\mathrm{racle}_{\text{fair}}(\vec{B}) := \mathcal{O}\mathrm{racle}_{\text{fair}}(\vec{B}) \times \mathbb{I}\{\mathcal{F}\mathrm{airness}(\vec{B}) \geq \theta\}$, where $\mathcal{F}\mathrm{airness}(\vec{B})$ is the fairness index achieved by Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy with budget allocation $\vec{B}$.
- $\mathcal{O}\mathrm{racle}_{\text{small fair}}(\vec{B}) := \mathcal{O}\mathrm{racle}_{\text{small fair}}(\vec{B}) \times \mathbb{I}\{\mathcal{F}\mathrm{airness}(\vec{B}) \geq \theta\}$. Similarly, $\mathcal{F}\mathrm{airness}(\vec{B})$ is the fairness index achieved by Flexible-Budget-Allocation-Contextual-Occupancy-Index Policy with budget allocation $\vec{B}$.

By expanding the definitions of $\mathcal{L}\mathrm{P}(\cdot)$, $\mathcal{O}\mathrm{racle}(\cdot)$ and $\mathcal{O}\mathrm{racle}_{\text{small}}(\cdot)$ to their fairness-aware versions, we can directly apply Branch And Bound and Mitosis to solve the optimal budget allocation under fairness constraint.

## G Experiments

In this section, we empirically evaluate the performance of COcc, Branch And Bound and Mitosis in CBB, and compare them against three baseline algorithms: Random, Greedy and the Vanilla Whittle Policy. We first demonstrate the algorithms' performance on numerical simulations. Then, we show that the performance of some of these algorithms could be sensitive to problem instance parameters, while our Mitosis algorithm consistently performs the best, robust against all these different setups that could arise in the real world. Finally, we run CBB experiment on real-world food rescue data which further confirms the superiority of Mitosis. For a coherent presentation and concrete argument, we describe all experiments in the food rescue context. That said, the results in Section G.1 and G.2 are evidently generalizable beyond the food rescue setting. All experiments are run on an AMD Ryzen 5955WX CPU with 128GB RAM.

### G.1 Experiments on Synthetic Data

**Food Rescue CBB Setup** Consider a food rescue platform notifying volunteers to pick up food donation delivery tasks. There are $K$ regions. Each region $k \in [K]$ is modeled as a context, and is associated with a *popularity* index $\mathrm{pop}_k \in \Re$. There are $N$ volunteers. Each $i \in [N]$ volunteer is modeled as an arm. Every volunteer has a historical rescue record vector $H_i = (k_1, k_2, \ldots), k_j \in [K]$ that records which region's tasks they have taken. Each volunteer or region is endowed with a location $(x, y)$. Depending on whether we run experiments on synthetic data or real-world data, we obtain these attributes either by sampling from random distributions or directly from food rescue data.

At each time step $t \leq T$, a food rescue trip arises from one region $k \in [K]$ chosen independently with probability $f_k$ ($\sum_{k\in[K]} f_k = 1$). The decision is to notify some volunteers via action $a_i^t = 1$, subject to the Contextual Budget Constraint. Volunteers' state space is binary: *active* ($s = 1$) or *inactive* ($s = 0$). Only by notifying active volunteers the platform obtains reward. Reward and state-action dependent transition probabilities are contingent on regions' and volunteers' attributes and distance. We introduce the high-level intuition here and defer the details to Appendix H.1. We model two types of food rescue volunteers. There are "organic" volunteers, whose reward and transition dynamics are a function of their distance from the region and their historical activities. Then there are "churner" volunteers, who could have a high likelihood of claiming a trip in some regions, but then slide to inactive states and very hard for them to become active again. In the food rescue application, as is true in many applications, both types of users exist. It is thus crucial to

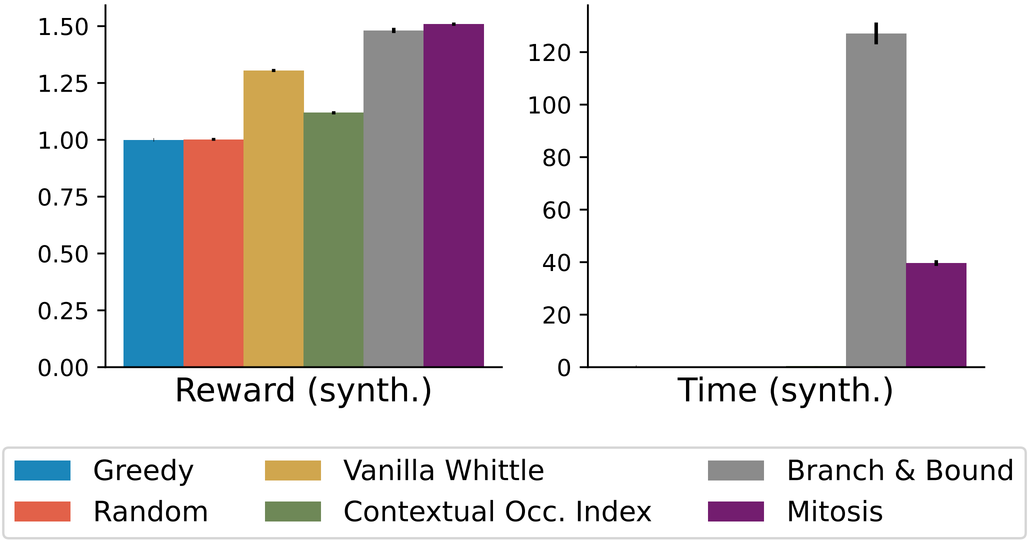

Figure 4: Main synthetic experiment with 50 arms, 3 contexts, and 0.1 budget ratio. Figures show normalized mean reward (left) and runtime in seconds (right) averaged over 32 seeds. Mitosis yields the highest reward at significantly lower computation than Branch And Bound.

model them separately. For now, we assume both types of volunteers appear equally frequently in the volunteer population. In Section G.2, we will relax this assumption.

**Experiment Setup and Results** In a synthetic CBB instance (50 arms, 3 contexts, budget=5), we run 32 seeds, each with 100 trials. We compare our proposed algorithms (COcc and Mitosis) with the following benchmarks: *Random* policy that selects arms uniformly at random; *Greedy* policy that selects arms with the highest immediate reward $r_i^k(s_i^t, 1)$, the aforementioned *Vanilla Whittle* policy that is asymptotically optimal in standard RMAB, and *Branch And Bound* that is guaranteed to compute optimal budget allocation but is computationally expensive.

Bar plots (Figure 4) compare the reward of each algorithm normalized by Random's, and runtime measured in seconds. Mitosis (purple) achieves the highest reward overall, while Branch And Bound (gray) is almost as good, but much slower. In fact, we set a timeout limit and Branch And Bound often terminates before it finds the optimal reward. The simpler baselines (Greedy, Random, Vanilla Whittle, and COcc), though taking almost no time to initialize, provide moderate to lower rewards, with the COcc notably underperforming.

We also run ablation studies by varying the number of volunteers ($N = 50, 100, 200$), the number of regions ($K = 3, 4, 5$) and budgets ($B = 2, 4, 6$). Figure 5 shows the rewards of the various algorithms when varying $N$ and $B$. Due to page limit, we defer the time plots and other reward plots to Appendix H.2. Generally, similar performance pattern holds as the scale of the instance increases. Mitosis consistently performs the best, while Branch And Bound's performance drops due to timeout.

### G.2 Sensitivity Analysis

While the ablation results in Figure 5 demonstrate the robustness of the Mitosis algorithm, some questions remain. Namely, we assumed that there are two types of volunteers – organic and churner – and they appear equally likely. In reality, for different applications, the ratio between organic and churner users may vary greatly. In this section, we aim to paint a complete picture of the algorithms' performance under various conditions.

To begin with, we refer to the proportion of organic volunteers in the volunteer population as "abundance". We systematically vary the abundance and budget $B$ in food rescue CBB, and and plot the heatmap of the ratio between COcc's and Mitosis's rewards in Figure 6. As shown in the figure, as abundance increases, the reward gap between Mitosis and COcc gradually closes, and similarly when the notification budget increases. However, if we have more churner volunteers, or if we have

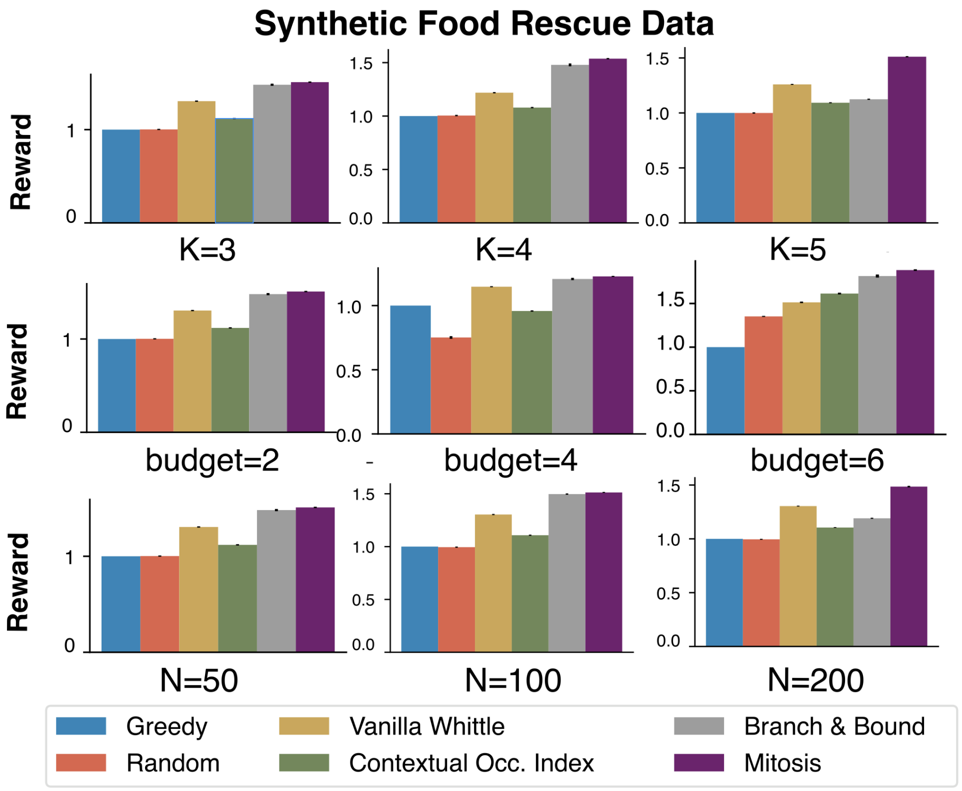

Figure 5: Ablation studies of the synthetic food rescue experiments to Figure 4. Mitosis generally performs the best across different problem sizes.

small notification budget relative to the overall number of volunteers, the performance of COcc could suffer. For churners, their different attitudes (transition dynamics) towards different regions (contexts) lead the COcc algorithm to make mistakes in action selection.

**Completely Random CBB.** To further stress test our main algorithm Mitosis's performance in settings most adversarial to it, we run experiments on the following simple setup with completely randomly generated CBB instances. Let transition probabilities $\Pr[s^{t+1} \mid s, a; k]$ and rewards $r(s, a; k)$ be generated from the following distributions: [3]

$$\begin{aligned}
&\Pr[s_i^{t+1} = 1 \mid s, a; k] \sim \text{Clip}(\text{Normal}(\mu_k^{(s,a)}, \sigma_k^{(s,a)}), 0, 1) \\
&\Pr[s_i^{t+1} = 0 \mid s, a; k] = 1 - \Pr[s_i^{t+1} = 1 \mid s, a; k] \\
&r_i(s = 1, a = 1; k) \sim \text{Normal}(\mu_k^r, \sigma_k^r) \\
&r_i(s, a; k) = 0 \text{ otherwise.}
\end{aligned}$$

where $\mu_k^{(s,a)}, \sigma_k^{(s,a)}$ and $\mu_k^r, \sigma_k^r, \ \forall k, s, a$ are all sampled i.i.d. from $\text{Uniform}[0, 1]$. To guarantee indexability, we further enforce $\Pr[s_i^{t+1} = 1 \mid s = 1, a = 1; k] < \Pr[s_i^{t+1} = 1 \mid s = 1, a = 0; k]$ (fatigue) and $\Pr[s_i^{t+1} = 1 \mid s = 0, a = 1; k] > \Pr[s_i^{t+1} = 1 \mid s = 0, a = 0; k]$ (recovery). Context distributions over $[K]$ is defined by sampling weights $w_k \sim \text{Uni}[0, 1]$ and normalize $f_k = \frac{w_k}{\sum_\kappa w_\kappa}, \forall k \in [K]$. With this setup, we have ripped off almost all the real-world relevance and treat this as a pure mathematical model.

We run the experiment with 50 arms, 5 contexts, and a notification budget of 5. We run 32 seeds with 100 trials each. The bar plot in Figure 7 compares the reward and runtime of Mitosis, COcc and other

[3] We use a common modeling assumption where rewards accrue only from engaged and available arms [Zhao, 2020].

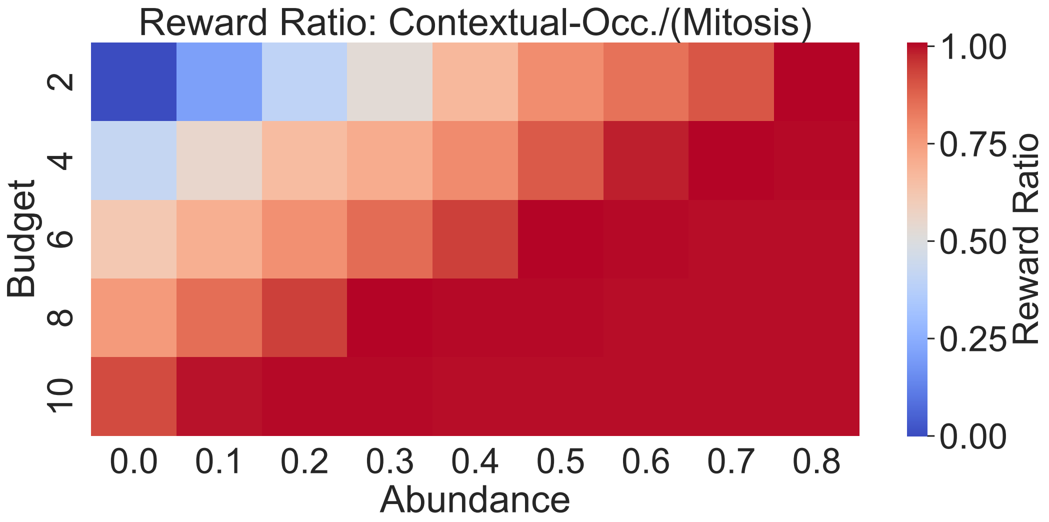

Figure 6: Heatmap showing the ratio between COcc and Mitosis's reward. Mitosis has a performance edge over COcc as the proportion of churner volunteers increases and as the notification budget decreases.

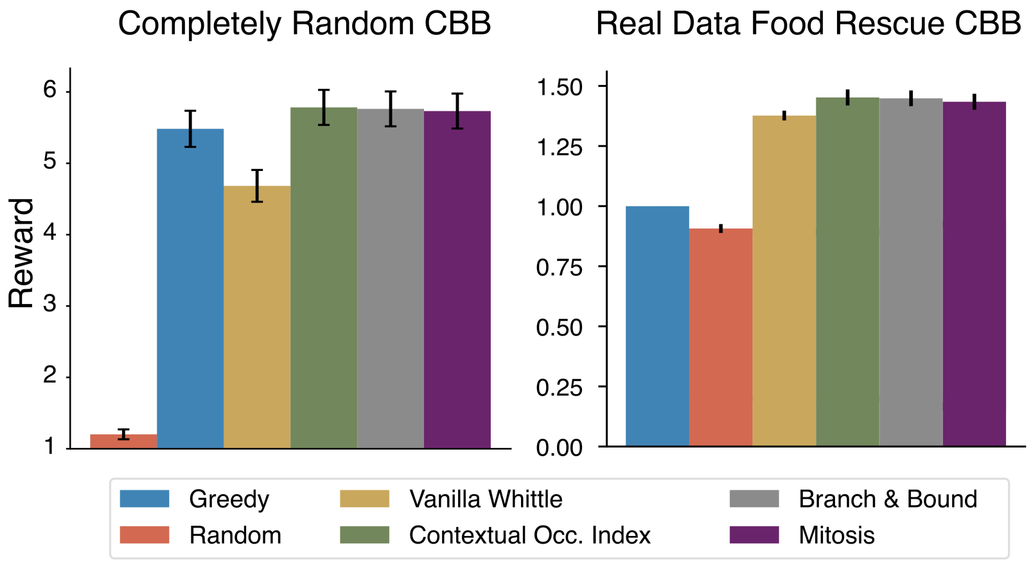

Figure 7: Reward of different policies for Completely Random CBB (Left) and Real-data Food Rescue CBB (Right)

aforementioned baselines. In Figure 7, context-agnostic Vanilla Whittle policy does not perform well, which validates Theorem 1 that Vanilla Whittle can perform arbitrarily poorly in the worst case for CBB. Meanwhile, context-aware policies – Mitosis, COcc and Branch And Bound – perform equally best. This is as expected since the instance sampling parameters are homogeneous and resemble the organic volunteer case above.

With this, we arrive at the main takeaway from Sections G.1 and G.2. COcc generally outperforms the baselines, and performs optimally in some settings. However, it fails in some other settings. On the other hand, Mitosis and Branch And Bound perform optimally in all settings, as the theoretical guarantee suggests. However, Mitosis is much more computationally efficient than Branch And Bound, making it a clear winner of all.

### G.3 Experiments on Food Rescue CBB from Real Data

We construct food rescue CBB from real data by sampling from a total pool of more than 500 thousand volunteers. Volunteers' and regions' attributes (locations and other idiosyncratic factors) are estimated from real-world data. The experiment setup is similar as the synthetic experiments in section G.1 (same set of policies, N=50, K=3, Budget=5, 32 seeds with 100 trials per seed). Results are shown in the barplot in Figure 7.

On real data, the context-aware methods (COcc, Branch And Bound and Mitosis) outperform Greedy, Random and Vanilla Whittle. Branch And Bound yields the highest average reward but requires disproportionately longer runtimes. By contrast, Mitosis nearly matches Branch And Bound while significantly reducing computation. Notably, COcc catches up with Mitosis—confirming it benefits from real-world attribute structure. The policies' performance trend is similar when we vary the number of volunteers, the number of regions and budget level in the ablation studies (see Appendix H.2 for details). This implies in application, COcc is sufficient for near-optimal performance. Mitosis guarantees optimality and is significantly faster than Branch And Bound.

## H Experiment Details: Design and Implementation

### H.1 Low/High Activeness in Synthetic Food Rescue CBB

We blend two types of synthetic setups to merge and simulate different dynamics in formulating the food rescue CBB:

#### H.1.1 Organic

In the organic instance, $N$ volunteers and $K$ regions are randomly positioned on a two-dimensional plane. Each volunteer and region is associated with a location and attributes—namely, volunteer activeness, region popularity, and a historical record $H_i$ (which, in turn, influences the context probabilities $f_k$). For every volunteer $i$ and region $k$, we define the pick-up rate as

$$p_{i,k} = \exp\left(\alpha\, \mathrm{pop}_k - \gamma\, d(i,k) + \beta\, \frac{|H_i|}{H_{\max}}\right),$$

where

- $\alpha$ is the parameter capturing the influence of region popularity (with $\mathrm{pop}_k$ denoting the popularity of region $k$), - $\gamma$ is the distance sensitivity parameter (with $d(i,k)$ representing the distance between volunteer $i$ and region $k$), - $\beta$ is the parameter reflecting volunteer activeness (with $|H_i|$ being the size of volunteer $i$'s history), and - $H_{\max}$ is a normalization constant.

Transition dynamics are such that an active volunteer (state $s = 1$) who is notified (action $a = 1$) picks up the task with probability $p_{ik}$ and may then become inactive. The immediate reward for a notification is a function of region popularity and $p_{ik}$.

#### H.1.2 Churner

In addition to the organic instance, we define a churner instance to capture more challenging dynamics within the CBB framework.

The $K$ regions are partitioned into *preferred* regions ($\mathcal{K}_{\text{preferred}} \subsetneq [K]$ and its complement. The preferred regions are designed such that, for any volunteer $i$, the pick-up probabilities $p_{ik}$ for $k \in \mathcal{K}_{\text{preferred}}$ are drawn uniformly around a high mean (e.g., centered at 0.95), making these regions very attractive and yielding a high probability of transitioning a volunteer to an inactive state. In contrast, for regions $k \notin \mathcal{K}_{\text{preferred}}$ (the "disliked" regions), the transition probabilities are concentrated around a low mean (e.g., centered at 0.05). Recovery probabilities $q_i$ for volunteers are generated around a prescribed mean (e.g., 0.2). Moreover, the reward structure is modified so that notifications in preferred regions yield an elevated immediate reward to reflect their allure despite the adverse long-term effect.

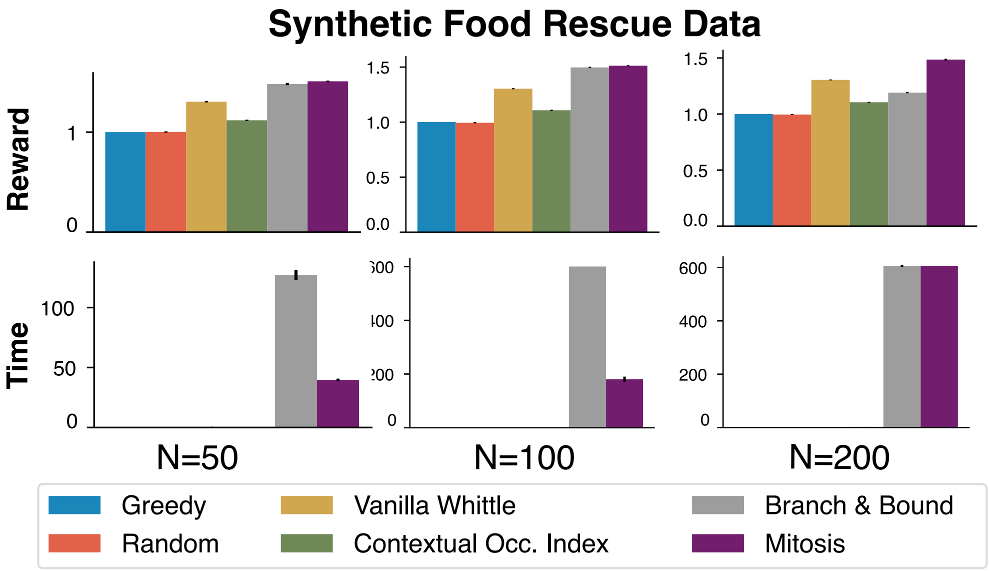

Figure 8: Ablation Experiments on Synthetic Food Rescue Experiments, Varying Number of Volunteers

### H.1.3 Blended Instance

Finally, we construct a *Blended Instance* that merges organic and churner dynamics. A fraction, referred to as abundance, $\rho_{\text{Abundance}} \in [0, 1]$ of the $N$ volunteers is designated to follow churner dynamics, while the remaining $N - N_{\text{active}}$ volunteers follow organic dynamics. Formally, we set

$$N_{\text{organic}} = \lfloor \rho_{\text{Abundance}} N \rfloor,$$

and generate two independent instances over the same set of $K$ regions:

(i) An *organic instance* with $N_{\text{organic}}$ volunteers. The transition dynamics and rewards are constructed as described in Section H.1

(ii) A *churner instance* with $N - N_{\text{organic}}$ volunteers, constructed as described in Section H.1.

## H.2 Ablation Experiments on Synthetic Food Rescue Instance

Below, we summarize the ablation study results on synthetic data, where we systematically vary the number of volunteers $(N)$, the number of regions $(K)$, and the budget $(B)$.

- **Varying number of volunteers for $N = 50, 100, 200$, fix $K = 3$ regions and budget be $5\%$ number of volunteers (Figure 8):** As $N$ increases, Mitosis (purple) consistently leads in reward and remains much faster than Branch And Bound (gray). Note that in $N = 200$ both Branch And Bound and Mitosis reach the time limit (600s) and is terminated, but sill within the same time limit, the Mitosis's solution is more than that of Branch And Bound's, demonstrating that Mitosis is much faster. COcc (green) still lags in reward, indicating it does not fully exploit the increased volunteer pool.
- **Varying number of regions for $K = 3, 4, 5$, fix number of volunteers $N = 50$, budget $B = 5$ (Figure 9):** With more regions, the search space for budget increases exponentially. Branch And Bound maintains a slight reward edge but at a steep runtime cost, it times out already at $k = 4$. Mitosis remains best and is faster compared to Branch And Bound. COcc gains some benefit but continues to underperform compared to the optimal.
- **Varying budget $B = 2, 4, 6$, fix volunteers $N = 50$, regions $K = 3$ (Figure 10):** Increasing $B$ allows more notifications, boosting Mitosis substantially while also helping COcc close some of the gap. Once again, Branch And Bound yields top-tier rewards but incurs much higher computation time.

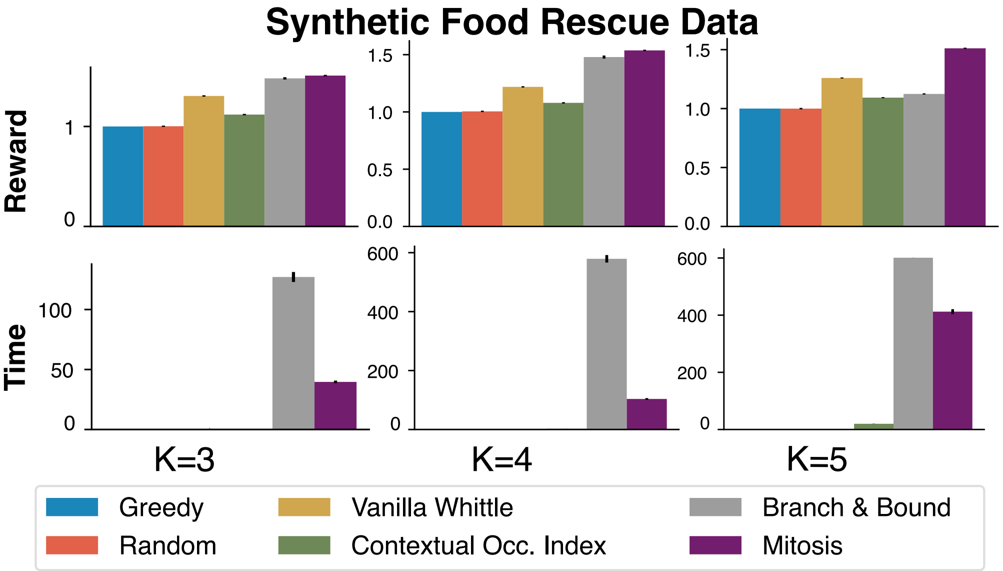

Figure 9: Ablation Experiments on Synthetic Food Rescue Experiments, Varying Number of Contexts

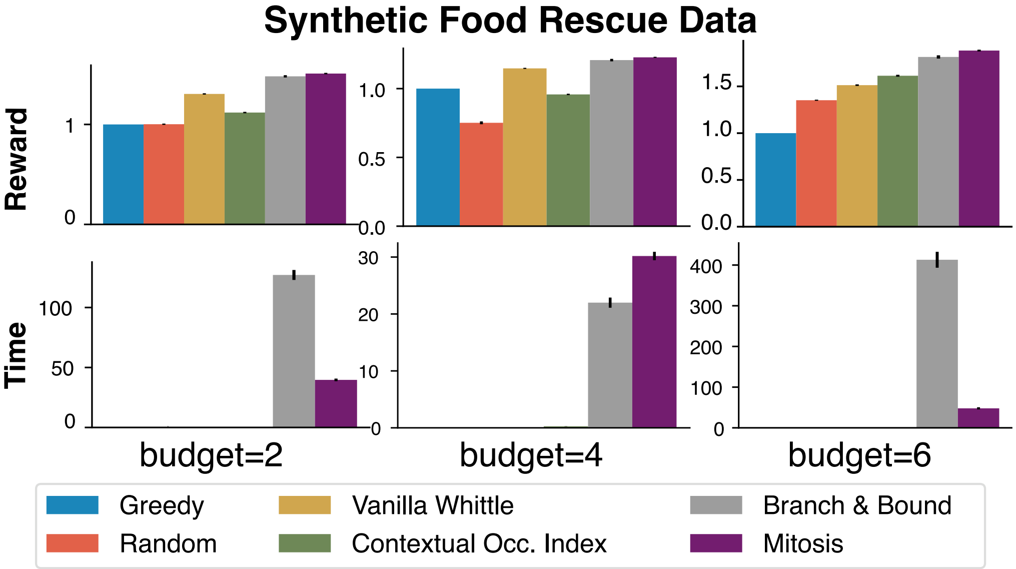

Figure 10: Ablation Experiments on Synthetic Food Rescue Experiments, Varying Budget

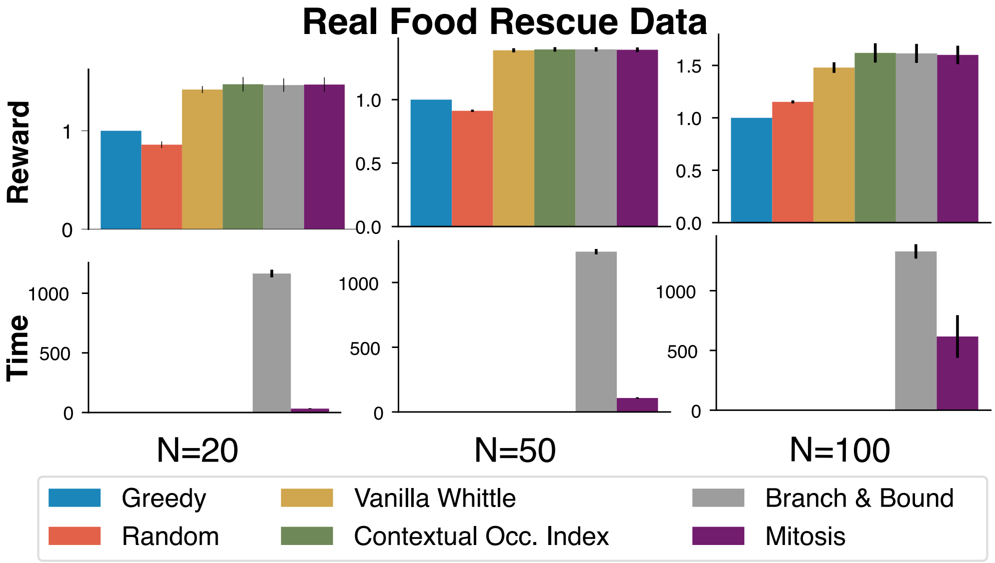

Figure 11: Ablation Experiments on Real Food Rescue Experiments, Varying Number of Volunteers

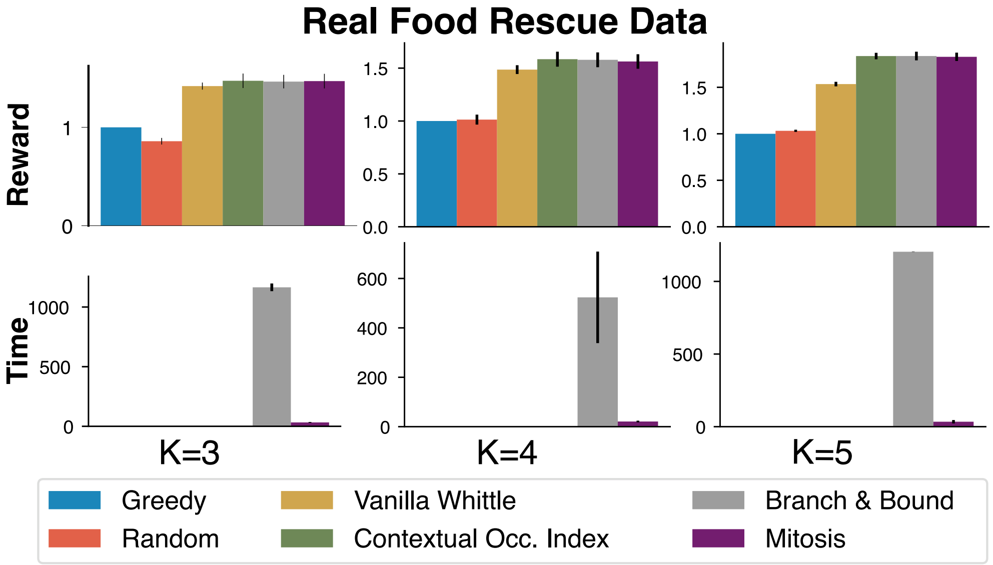

Figure 12: Ablation Experiments on Real Food Rescue Experiments, Varying Number of Contexts

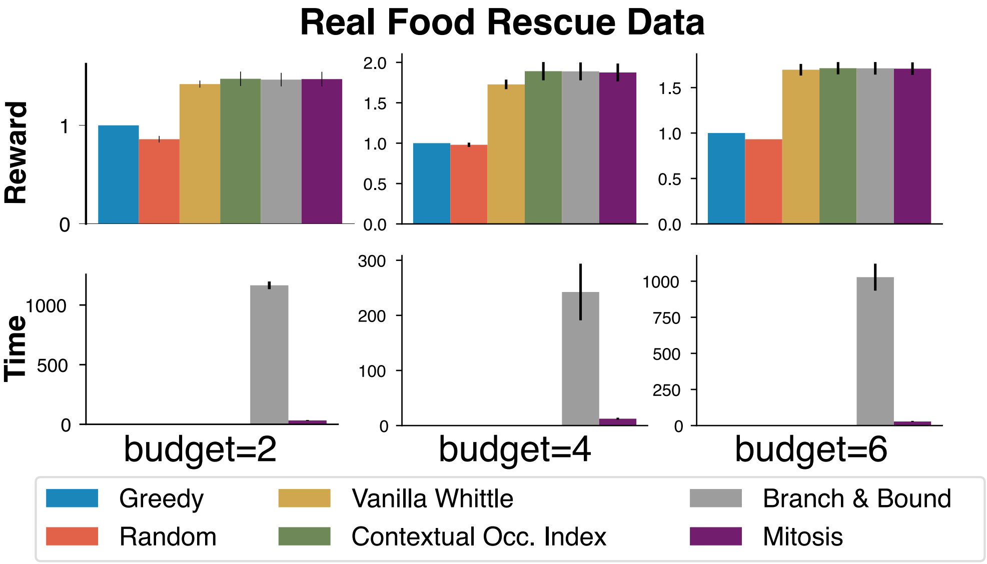

Figure 13: Ablation Experiments on Synthetic Food Rescue Experiments, Varying Budget

### H.3 Ablation Experiments on Real Food Rescue Data

Similar as synthetic data's ablations, we systematically vary the number of volunteers $(N)$, the number of regions $(K)$, and the budget $(B)$ of CBB constructed on real food rescue data. Results are shown in

- Figure 11: changing $N = 20, 50, 100$ while maintain number of regions $K = 3$, budget $B = 2$.
- Figure 12: changing $K = 3, 4, 5$ while maintaining number of regions $N = 20$, budget $B = 2$.
- Figure 13: changing $B = 2, 4, 6$ while maintaining number of volunteers $N = 20$, number of regions $K = 3$.

As the scale of the instance increases ($N, K$ or $B$ increases), Vanilla Whittle performance grows worse compared to COcc, Branch And Bound, Mitosis which they perform similarly. This shows that (i) when the scale of the problem increase, it is necessary to introduce context-aware policies to reach optimal performance (ii) in application, COcc is sufficient for near-optimal performance. Mitosis guarantees optimality and is significantly faster than Branch And Bound.

## I Related Works

**Contextual Budget Bandit** Contextual information is often present in bandit and can substantially improve decision quality, and has been extensively studied in classical Multi-Armed Bandits [Bouneffouf et al., 2020, Langford and Zhang, 2007]. Incorporating context with RMABs is a promising direction, as it enables modeling both individual arm's state dynamics and global contextual influences. To our knowledge, four main works combine context into RMABs. Liang et al. [2025], Mimouni and Avrachenkov [2025], and Guo and Wang [2024] treat context as static side information that affects arms' transitions and rewards. These studies focus on learning transition parameters online [Liang et al., 2025] or approximating Whittle indices using deep learning [Mimouni and Avrachenkov, 2025, Guo and Wang, 2024].

The most relevant prior work is Chen et al. [2024], which models context as a global Markov-evolving state and develops a dual-decomposition-based index policy, which is a generalization of Whittle Index Policy. Their approach proves optimal under the assumption that contexts evolve deterministically and periodically. In contrast, we assume context evolves randomly under a Bayesian prior, which generalizes and weakens this assumption. We theoretically demonstrate that under this setting, Whittle index policy loses asymptotic optimality, underscoring the need for careful budget design in contextual RMABs.

All prior contextual RMAB work assumes a fixed activation budget per round, independent of context. In this paper, we allow the budget to vary by context, and show—both theoretically and empirically—that context-aware budget allocation improves performance. We design algorithm for calculating optimal budget allocation.

**Fairness** is an increasingly important consideration in both business and non-profit organizations Bertsimas et al. [2012], Liu and Garg [2024]. For RMAB, fairness is typically imposed on individual arms: Wang et al. [2024] define fairness as requiring a minimum long-term activation fraction for each arm; Li and Varakantham [2022a] propose a *soft fairness constraint*, or by setting an upperbound on the number of decision epochs since an arm was last activated (Li and Varakantham [2022b]). Fairness can also be defined over groups of arms. Killian et al. [2023] study minimax and max-Nash welfare objectives by imposing fairness on groups of arms, and Verma et al. [2024] enforce fairness with respect to the reward outcomes across groups. To the best of our knowledge, although RMABs with **contextual information** have been previously studied, our work is the first to consider fairness with respect to **context**.

# J Generalization

While we ground our work in food rescue volunteer engagement, our model and algorithms are applicable to many domains.

**Digital agriculture** Agriculture chatbots empower smallholder farmers [Guérin et al., 2024]. In collaboration with Organization X, we have a chatbot which sends nudges about farming practices to over 50,000 farmers in India, Kenya, and Nigeria. However, nudges of different topics have different conversion rates. Pest control tips during the pest season address an urgent problem, resulting in high conversion rates. Meanwhile, watering tips are preventive measures, which often have lower conversion rates by the farmers. Thus, one would assign different nudging budgets to different topics of nudges, and model it as a CBB. Each farmer is an arm. At each time step, we have a nudge topic as context, and we decide on a budget of how many farmers to notify and the arm selection of who to notify. Rewards are determined based on farmer's engagement response.

**Peer review** Journals select reviewers where selection impacts future reviewer availability [Payan and Zick, 2021]. For a given paper, the goal is to select a subset of reviewers with the relevant expertise. However, submissions differ from one another. For example, submissions that are extra long, that involves heavy theoretical analysis, or that do not study the trendy topics might have lower chance of getting reviewers. Thus, when the editor plans reviewing invitations over time, they would want to send different numbers of invitations to different kinds of submissions, and model it as a CBB. Each potential reviewer is an arm. At each time step, we have a submission type as context, and we decide on a budget of how many potential reviewers to reach out to, and the arm selection of who to reach out to. Rewards are determined based on the reviewers' response.