ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240876

Combining Diverse Information for Coordinated Action: Stochastic Bandit Algorithms for Heterogeneous Agents

Lucia Gordon[a,*], **Esther Rolf**[a] **and Milind Tambe**[a]

[a]Harvard University
ORCID (Lucia Gordon): https://orcid.org/0000-0003-3219-6960, ORCID (Esther Rolf): https://orcid.org/0000-0001-5066-8656, ORCID (Milind Tambe): https://orcid.org/0000-0003-3296-3672

Abstract. Stochastic multi-agent multi-armed bandits typically assume that the rewards from each arm follow a fixed distribution, regardless of which agent pulls the arm. However, in many real-world settings, rewards can depend on the *sensitivity* of each agent to their environment. In medical screening, disease detection rates can vary by test type; in preference matching, rewards can depend on user preferences; and in environmental sensing, observation quality can vary across sensors. Since past work does not specify how to allocate agents of heterogeneous but known sensitivity of these types in a stochastic bandit setting, we introduce a UCB-style algorithm, MIN-WIDTH, which aggregates information from diverse agents. In doing so, we address the joint challenges of (i) aggregating the rewards, which follow different distributions for each agent-arm pair, and (ii) coordinating the assignments of agents to arms. MIN-WIDTH facilitates efficient collaboration among heterogeneous agents, exploiting the known structure in the agents' reward functions to weight their rewards accordingly. We analyze the regret of MIN-WIDTH and conduct pseudo-synthetic and fully synthetic experiments to study the performance of different levels of information sharing. Our results confirm that the gains to modeling agent heterogeneity tend to be greater when the sensitivities are more varied across agents, while combining more information does not always improve performance.

1 Introduction

The setting of stochastic multi-agent multi-armed bandits (MAB) [16, 15, 20] is characterized by multiple agents taking actions simultaneously in each time step. This setting serves as a natural model for diverse domains, from COVID test allocation [3] to preference matching [8, 22] and poaching prevention [23, 24]. These real-world problems involve unknown characteristics about the environment that are learned *online* while the planner figures out the optimal action for each agent. The resulting explore-exploit tradeoff lends itself well to UCB-style algorithms [2], which estimate unknown quantities optimistically with an upper confidence bound (UCB) in an effort to maximize cumulative reward over time.

We introduce a new stochastic MAB problem wherein a planner specifies actions for agents of heterogeneous but known sensitivities to their unknown environment. The "environment" comprises a set of arms, each of which takes on a state of 0 or 1 at each time step following a Bernoulli distribution, which models a binary outcome as in Solanki et al. [18] and Xu et al. [23]. The mean of the Bernoulli is an unknown parameter that must be learned online by the agents. The agents differ in their **sensitivity**, which is their probability of receiving a reward of 1 upon pulling an arm given that its state is 1. In this way, the utility of the agents' actions is a function of their sensitivity as well as the arm mean.

Several key ideas help us tackle the core challenges of sequential decision-making with multiple agents with heterogeneous sensitivities to their environment. First, we address the combinatorial challenge of the many ways of allocating agents to arms by decomposing our combinatorial problem into learning the means of the individual arms. Second, we address the learning challenge by combining rewards across agents of varying sensitivity to speed up learning in a sensitivity-aware manner. Third, we address the problem of how to match *heterogeneous* agents with arms by assigning the highest-sensitivity agents to the arms with the highest UCBs, which we experimentally show is an effective strategy. In contrast, applying past work to our problem without these insights would either naively combine all the agents' rewards and ignore their sensitivities [5] or slowly learn the optimal assignments by approaching the problem at the coarse, super-arm level [2].

We introduce the MIN-WIDTH algorithm designed for this new problem (§5). For each arm, MIN-WIDTH combines all the agents' rewards to generate a mean estimator with the tightest UCB, which is nontrivial since rewards are drawn from different distributions for each agent-arm pair. We derive an instance-independent $\mathcal{O}(\sqrt{T\log(T)})$ regret upper bound for the MIN-WIDTH algorithm, where T is the time horizon and there are additional factors for the numbers of agents and their sensitivities (Theorem 2). We also evaluate MIN-WIDTH through pseudo-synthetic experiments with realistic parameter values for diverse domains including COVID test allocation, hotel recommendation, and poaching prevention along with fully synthetic experiments (§8). To compare algorithms with different levels of information sharing, we introduce two sensitivity-aware baselines that we evaluate against MIN-WIDTH. We find that MIN-WIDTH outperforms classical baseline algorithms (CUCB [5] and UCB [2]) not designed for heterogeneous agents as well as our sensitivity-aware baselines in many settings. Moreover, we show experimentally that the performance of MIN-WIDTH is robust to having only approximate knowledge of the agent sensitivities.

2 Motivating Domains

Our setting of heterogeneous agents with known sensitivities is motivated by a diverse range of application domains, highlighted by the examples below. By explicitly incorporating agent heterogeneity and information-sharing, we introduce a more natural model for these domains.

COVID Test Allocation Consider the problem of allocating a limited number of COVID tests of varying sensitivity among floors of a college

* Corresponding Author. Email: luciagordon@g.harvard.edu

dorm with unknown virus prevalence to maximize detection of infected individuals. Different floors in a dorm, which serve as our "arms," will likely have different prevalence rates of the virus. The two primary types of tests to detect COVID are PCR (very sensitive) and antigen (less sensitive) tests, which serve as "agents" in our model. Due to limited availability, suppose a college only has enough supply to test one student on each floor per time step. Thus, when we pull a super-arm, we distribute tests (the agents) among floors (the arms) and observe the test results. Bastani et al. [3] consider a similar setup, developing a MAB system for airport COVID testing, but they do not account for varying test sensitivities.

Hotel Recommendation Consider the problem of matching customers with different preferences to hotels with limited space in order to maximize customer satisfaction, a task performed by websites such as Booking.com. Hotels, which are our "arms," differ in features such as cleanliness, service, etc., which may not be known, especially for new hotels. Customers, which serve as our "agents," vary in how much these features matter to them, information that booking platforms can request in a pre-recommendation survey. For instance, one customer may value cleanliness above other features and always be satisfied with their stay if the hotel was clean. Another customer may care equally about cleanliness and staff friendliness, so even if the hotel is clean they will not be satisfied if the staff are not very responsive. Assuming space in hotels is limited and some customer will have to be matched with a hotel that tends to be less clean, we maximize overall satisfaction by matching this latter customer with the less clean hotel and the former customer with the cleaner hotel on average. A similar setting has been modeled using contextual bandits in Wanigasekara et al. [22]. However, unlike their algorithm we combine data from post-stay cleanliness reviews across customers with different preferences to better match either the same customers or new ones with similar preferences in future time steps.

Poaching Prevention Consider the problem of maximizing the detection of animal traps by planning patrols for rangers with different detection rates in a protected area such as a national park. Poachers hide snares to trap animals, and rangers patrol the park for illegal activity and remove any snares they find, which can be modeled as a stochastic bandit problem [23]. Different parts of the park, which serve as our "arms," are more or less likely to contain snares depending on their animal density, accessibility, etc., but the poaching rates across a park are often unknown due to the vastness of the areas. Rangers, the "agents," vary in terms of their expertise, tools, and vehicles, giving them varying snare detection rates, or "sensitivities," a type of real-world heterogeneity that has not been previously modeled. Since the rangers are all patrolling the same park, we combine their observations to speed up our learning of poaching hotspots and optimize our assignments for rangers in the next round of patrols, accounting for the rangers' differing partial observabilities in snare detection.

3 Related Work

Auer et al. [2] introduce the UCB algorithm for the stochastic bandit problem that pulls the highest-UCB arm in every time step. Audibert et al. [1] introduce the UCB-V algorithm that builds on UCB by incorporating arms' empirical variances. Neither of these model multiple agents. Gai et al. [7] extend the UCB algorithm to the combinatorial setting where multiple arms are pulled in each time step with linear reward functions. Chen et al. [5] introduce an algorithm for possibly unknown reward functions. Both assume the reward obtained from pulling any given arm is an i.i.d. draw from a fixed distribution, an assumption that does *not* hold in our problem. Rejwan and Mansour [13] consider the combinatorial bandit setting with full-bandit feedback (only the sum of the rewards is observed), whereas we operate in the semi-bandit feedback setting (the reward from each pull is observed) and also have heterogeneous agents.

Existing multi-agent bandit papers differ widely in their definition of agent heterogeneity. Some works consider agents with access to only a subset of the arms [25], differing but known communication abilities [12], or varying user preferences [8, 22]. The latter is similar to our definition, but in their case the arm context is known and the user preferences are unknown, whereas we have unknown arm means but known sensitivities. Federated combinatorial bandits [18] also have heterogeneous agents, but their agents operate in a competitive environment and are subject to privacy constraints, whereas in our setting the agents are collaborating and there is no cost to their communication.

Our notion of sensitivity is inspired by past work that utilizes sensor models to capture imperfect observability. In Xu et al. [23], the agents do not observe the true state of each arm, though the more effort they exert, the more reliable their observations become. Their algorithm, however, assumes effort can be specified and distributed in each time step, whereas our agent sensitivities are fixed in advance. Rolf et al. [14] model a single sensor trying to pick out the environment point with the strongest signal on average given observations that include contributions from all points but are more sensitive to those nearby. Unlike this work, we have multiple sensors, and we assume the rewards from distinct arms are independent.

Past work has explored the consequences of agent communication in different settings. In Shi et al. [17], multiple agents can be assigned to the same arm of a MAB, which results in a collision that can be used to transmit information between agents. We assume each agent pulls a distinct arm, so our agents cannot communicate in this way. In Madhushani and Leonard [12], the agents observe the actions and rewards of their neighbors with some known probability, where the agents are heterogeneous in terms of their "sociability," so some may be more likely to observe their neighbors than others. Our agents' heterogeneity is unrelated to the way in which information is shared among them. Taylor et al. [19] consider a multi-agent explore-exploit optimization problem and demonstrate the uncertainty penalty phenomenon, wherein increased teamwork under uncertainty can degrade performance relative to the agents acting alone. Unlike their setting and the one in Cesa-Bianchi et al. [4], our problem has no intrinsic spatial nature.

4 Problem Statement

We introduce a sequential decision-making problem in which a set of heterogeneous agents are allocated among a set of arms that yield stochastic rewards. There are A agents $\mathcal{A} = \{a\}_{a=1}^{A}$, N arms $\mathcal{N} = \{n\}_{n=1}^{N}$, and T time steps $\mathcal{T} = \{t\}_{t=1}^{T}$. The state of arm n at time t is a random variable $X_{t,n} \sim \text{Bern}(\mu_n)$, so $X_{t,n} \in \{0,1\}$. The agents' heterogeneity is captured in their "sensitivity," a scalar value associated with each agent. We denote the set of agent sensitivities by $\mathcal{S} = \{s_a\}_{a=1}^{A}$, where s_a represents the probability agent a receives a reward of 1 conditional on the true state of the arm being 1. Thus, the reward $Y_{t,a,n}$ obtained when pulling arm n is a random variable that depends on both the agent a who pulls it and the arm's mean:

$$Y_{t,a,n} \sim \text{Bern}(s_a \mu_n). \tag{1}$$

By construction, $s_a = \mathbb{P}[Y_{t,a,n} = 1 | X_{t,n} = 1]$, and in our model, $\mathbb{P}[Y_{t,a,n} = 0 | X_{t,n} = 0] = 1$.

At each time step, the planner selects a super-arm assigning each agent to a distinct arm. The super-arm is chosen from the set $\mathcal{F} = \{f\}_{f=1}^{\frac{N!}{(N-A)!}}$, where $f : \mathcal{A} \to \mathcal{N}$ such that $f(a) \neq f(a')$ if $a \neq a'$. The super-arm selected at time t is denoted f_t, and so $f_t(a)$ is the arm to which agent a is assigned at time t. We keep track of the number of times each arm

has been pulled by each agent. Let

$$c_{t,a,n} = \sum_{\tau=1}^{t} \mathbb{1}_{f_\tau(a)=n} \tag{2}$$

be the number of times arm n has been pulled by agent a through time t and

$$c_{t,n} = \sum_{a=1}^{A} c_{t,a,n} \tag{3}$$

be the total number of times arm n has been pulled through time t by any agent. Based on these, let us also define

$$\mathcal{T}_{a,n} = \{t \in \mathcal{T} | c_{t,a,n} > 0\} \tag{4}$$

as the set of times agent a has pulled arm n at least once and

$$\mathcal{T}_{n} = \{t \in \mathcal{T} | c_{t,n} > 0\} \tag{5}$$

as the set of times for which arm n has been pulled at least once by any agent.

The total reward collected by pulling super-arm f is a sum over the individual agent rewards: $r_f = \sum_{a=1}^{A} Y_{t,a,f(a)}$. The expected reward is then

$$\overline{r}_f = \mathbb{E}[r_f] = \sum_{a=1}^{A} \mathbb{E}[Y_{t,a,f(a)}] = \sum_{a=1}^{A} s_a \mu_{f(a)}.$$

We define the optimal super-arm $f^\star$ to be the one that maximizes the expected reward:

$$f^\star = \operatorname{argmax}_{f \in \mathcal{F}} \overline{r}_f = \operatorname{argmax}_{f \in \mathcal{F}} \sum_{a=1}^{A} s_a \mu_{f(a)}. \tag{6}$$

The cumulative regret at time T captures how poor the super-arms selected at the time steps elapsed so far perform compared to the optimal super-arm in expectation. In other words, we measure how much worse the cumulative expected reward is given a sequence of agent-arm assignments relative to the best it could be given the agent sensitivities at hand. The objective is to minimize the cumulative regret at time T, given by

$$R_T = \sum_{t=1}^{T} \sum_{a=1}^{A} s_a \left(\mu_{f^\star(a)} - \mu_{f_t(a)}\right). \tag{7}$$

5 MIN-WIDTH Algorithm

5.1 Algorithm Structure

We introduce MIN-WIDTH, a UCB-style algorithm for assigning heterogeneous agents to stationary stochastic arms with Bernoulli rewards. We assume a centralized planner that knows the sensitivities of all the agents and coordinates their assignment to arms in each time step. MIN-WIDTH, outlined in Algorithm 1, revolves around an N-length vector of UCBs denoted by $\widetilde{\text{UCB}}$, where $\widetilde{\text{UCB}}_t$ represents the UCBs the planner uses to match each agent a with an arm n at time $t+1$. In each time step $1 \le t \le T$, the agents are assigned to arms sequentially in descending order by sensitivity (lines 6-10) so that the highest-sensitivity agent is assigned to the arm with the highest $\widetilde{\text{UCB}}_{t-1}$, the next-highest-sensitivity agent is then assigned to the arm with the highest $\widetilde{\text{UCB}}_{t-1}$ out of those remaining unselected, and so on until all the agents have been assigned to distinct arms. The super-arm corresponding to this assignment is then pulled and each agent gets some reward $Y_{t,a,f_t(a)}$ (line 11). Next, the UCBs are updated $\widetilde{\text{UCB}}_{t-1}[n] \to \widetilde{\text{UCB}}_t[n]$ for every arm n (line 13) according to Equation 11.

5.2 Agent Allocation Strategy

Our agent allocation strategy (lines 6-10) is inspired by the definition of the optimal super-arm in Equation 6: $f^\star$ assigns the ith-highest-sensitivity agent to the ith-highest-mean arm. In practice, we do not know the true means $\{\mu_n\}_{n \in \mathcal{N}}$. Instead, we can estimate them with some $\{\hat{\mu}_n\}_{n \in \mathcal{N}}$

Algorithm 1 MIN-WIDTH Algorithm

1: $\widetilde{\text{UCB}}_0 \leftarrow [\infty,...,\infty]$ (length N)
2: ranked_agents $\leftarrow$ flip(argsort(sensitivities))
3: **for** t in range$(1, T+1)$ **do**
4: $\quad f \leftarrow [-1,...,-1]$ (A)
5: $\quad$ unassigned_arms $\leftarrow (0, \ldots, N-1)$
6: $\quad$ **for** a in ranked_agents **do**
7: $\quad\quad$ UCBs$\leftarrow \widetilde{\text{UCB}}_{t-1}$[unassigned_arms]
8: $\quad\quad n \leftarrow$ unassigned_arms[argmax(UCBs)]
9: $\quad\quad f[a] \leftarrow n$
10: $\quad\quad$ unassigned_arms.remove(n)
11: $\quad$ pull super-arm f and get $\{Y_{t,a,f_t(a)}\}_{a \in \mathcal{A}}$
12: $\quad$ **for** n in range(N) **do**
13: $\quad\quad \widetilde{\text{UCB}}_t[n] \leftarrow$ update rule given by Equation 11

and set some upper confidence bounds $\{\text{UCB}_n = \hat{\mu}_n + \epsilon_n\}_{n \in \mathcal{N}}$ on them, where ϵ_n is the width of the confidence interval around our estimate $\hat{\mu}_n$ of μ_n. In the standard UCB algorithm, the optimal action is to pull the arm with the highest mean, and since that is unknown, the algorithm instead pulls the arm with the highest UCB [2]. Analogously, in our setting, the optimal action is to match the ith-best agent with the arm with the ith-highest mean, but since the means are unknown, we instead match the ith-best agent with the arm with the ith-highest UCB. Another way to motivate this is to consider the infinite-data setting, in which the UCBs are equal to the true arm means. In that case, to maximize our expected reward we must match the highest-sensitivity agent to the highest-mean arm. In the finite-data setting, we optimistically estimate the means with the UCBs (which converge to the means with increasing amounts of data), which is why we match the highest-sensitivity agent with the highest-UCB arm.

5.3 Update Rule

At each time step t, the planner uses all the rewards collected so far to generate a new $\text{UCB}_{t,n}$ for each arm n, as derived in §6. In particular, the planner constructs an empirical estimator $\hat{\mu}_{t,n}$ for the mean of each arm that combines all the agents' rewards while accounting for their heterogeneity so as to minimize the width of the confidence interval $\epsilon_{t,n}$ around that estimator. The empirical estimator of the mean of arm n at time $t \in \mathcal{T}$ is given by

$$\hat{\mu}_{t,n} = \begin{cases} t \notin \mathcal{T}_n & 0.5 \\ t \in \mathcal{T}_n & \frac{\sum_{a=1}^{A} s_a \sum_{\tau=1}^{t} \mathbb{1}_{f_\tau(a)=n} Y_{\tau,a,n}}{\sum_{b=1}^{A} {s_b}^2 c_{t,b,n}} \end{cases}. \tag{8}$$

The width of the confidence interval on μ_n at time $t \in \mathcal{T}$ is

$$\epsilon_{t,n} = \sqrt{\frac{\ln(2NG(T,A)/\delta)}{2\sum_{a=1}^{A} {s_a}^2 c_{t,a,n}}} \tag{9}$$

for

$$G(T,A) = \sum_{t=1}^{T} \binom{t+A-1}{A-1} < (T+1)^A. \tag{10}$$

The UCB on the mean of arm n at time $t \in \mathcal{T}$ is

$$\text{UCB}_{t,n} = \hat{\mu}_{t,n} + \epsilon_{t,n} \tag{11}$$

and is used as the $\widetilde{\text{UCB}}_t[n]$ in line 13: $\widetilde{\text{UCB}}_t[n] = \text{UCB}_{t,n}$. Note that the algorithm returns a finite $\widetilde{\text{UCB}}_t[n]$ after a single pull of arm n by any agent. The distribution of the rewards can vary greatly depending on the sensitivity of the agent who collected them, but because the planner knows all the agent sensitivities, they can harness that information to appropriately weight the rewards from different agents in generating a shared UCB.

6 Theoretical Results

We provide analytical results for the MIN-WIDTH algorithm introduced in §5, with complete proofs in Appendix A [10]. Incorporating all the agents' rewards to generate a shared UCB for each arm is a complex problem due to the agents' heterogeneity. Naively, one may think we could simply divide each reward for a given arm by the sensitivity of the agent who collected it and apply the original UCB algorithm to this sequence. This is invalid, however, because while these rescaled rewards will have identical means, they will still have different variances. The UCB algorithm assumes the rewards from a given arm are i.i.d. and hence would not apply to these rescaled rewards. To resolve this, in Proposition 1 we take a more general approach by treating this as an optimization problem where we optimize over weights on the agents' rewards to get the tightest confidence interval around the arm mean estimator.

Proposition 1. *(MIN-WIDTH Weights Derivation). Suppose agent a pulls arm n a fixed number of times, a number we denote $c_{a,n}$, where the reward from each pull is $Y_{i,a,n} \sim Bern(s_a \mu_n)$. Let $\mathcal{C}_n = \{c_{a,n}\}_{a=1}^{A}$ contain the $c_{a,n}$ for every agent. Let $D_{\mathcal{C}_n,n}$ be the weighted sum of the independent rewards collected by all the agents from arm n, expressed in terms of weights $w_{\mathcal{C}_n,a,n}$:*

$$D_{\mathcal{C}_n,n} = \sum_{a=1}^{A} w_{\mathcal{C}_n,a,n} \sum_{i=1}^{c_{a,n}} Y_{i,a,n} = \sum_{a=1}^{A} \sum_{i=1}^{c_{a,n}} w_{\mathcal{C}_n,a,n} Y_{i,a,n}. \tag{12}$$

Then the weights $w_{\mathcal{C}_n,a,n}$ that minimize the width of the confidence interval on μ_n given by

$$\gamma_{\mathcal{C}_n,n} = \sqrt{\frac{\ln(2/\delta)}{2} \sum_{a=1}^{A} {w_{\mathcal{C}_n,a,n}}^2 c_{a,n}}$$

under the constraint that the empirical estimator $D_{\mathcal{C}_n,n}$ is unbiased are

$$w_{\mathcal{C}_n,a,n} = \mathbb{1}_{c_{a,n}>0} \times \frac{s_a}{\sum_{b=1}^{A} {s_b}^2 c_{b,n}}. \tag{13}$$

Proof. Since the rewards collected by a certain agent when pulling a certain arm are i.i.d., we consider weights on such sequences of i.i.d. rewards rather than on every single reward. If $c_{a,n}=0$, then agent a has collected no rewards for arm n, and so we set $w_{\mathcal{C}_n,a,n}=0$ for any such agent. Hence, we need to solve for $w_{\mathcal{C}_n,a,n}$ only for agents with $c_{a,n}>0$. If $D_{\mathcal{C}_n,n}$ is to be unbiased, then we need $\mathbb{E}[D_{\mathcal{C}_n,n}]=\mu_n$, which sets the constraint

$$\sum_{a=1}^{A} w_{\mathcal{C}_n,a,n} s_a c_{a,n} = 1.$$

Since $w_{\mathcal{C}_n,a,n} Y_{i,a,n}$ is bounded by $0 \le w_{\mathcal{C}_n,a,n} Y_{i,a,n} \le w_{\mathcal{C}_n,a,n}$, Hoeffding's inequality gives

$$\forall \delta \in (0,1), \mathbb{P}\left[\left| D_{\mathcal{C}_n,n} - \mu_n \right| < \sqrt{\frac{\ln(2/\delta)}{2} \sum_{a=1}^{A} {w_{\mathcal{C}_n,a,n}}^2 c_{a,n}} \right] > 1-\delta, \tag{14}$$

yielding $\gamma_{\mathcal{C}_n,n} = \sqrt{\frac{\ln(2/\delta)}{2} \sum_{a=1}^{A} {w_{\mathcal{C}_n,a,n}}^2 c_{a,n}}$ as the width of the confidence interval on the mean of arm n for some fixed number of pulls of each arm by each agent captured in $\mathcal{C}_n$. To make this confidence interval as tight as possible, we solve for the weights that minimize $\gamma_{\mathcal{C}_n,n}$ under the constraint that $D_{\mathcal{C}_n,n}$ is unbiased for any non-random $\mathcal{C}_n$. We solve this constrained optimization problem with the method of Lagrange multipliers, using Lagrangian

$$\mathcal{L}(w,\lambda) = \sqrt{\frac{\ln(2/\delta)}{2} \sum_{b=1}^{A} {w_{\mathcal{C}_n,b,n}}^2 c_{b,n}} + \lambda \left(\sum_{b=1}^{A} w_{\mathcal{C}_n,b,n} s_b c_{b,n} - 1 \right),$$

which results in $w_{\mathcal{C}_n,a,n} = \frac{s_a}{\sum_{b=1}^{A} {s_b}^2 c_{b,n}}$, which holds for any agent a with $c_{a,n}>0$. Since $w_{\mathcal{C}_n,a,n}=0$ for agents with $c_{a,n}=0$, we get Equation 13. □

Next, in Theorem 1, we show that we can use the weights derived in Proposition 1 to construct an empirical estimator for the mean of each arm, whose deviation from the true mean we bound with high probability. This bound involves the challenge of counting the number of possible pulls of each arm by each agent.

Theorem 1. *(MIN-WIDTH Concentration Bound). Suppose the empirical estimator of the mean of arm n at time $t \in \mathcal{T}$ is given by Equation 8. Then for $\epsilon_{t,n}$ given in Equation 9, $\hat{\mu}_{t,n}$ satisfies*

$$\forall \delta \in (0,1), \mathbb{P}[\forall n \in \mathcal{N}, t \in \mathcal{T}, |\hat{\mu}_{t,n} - \mu_n| < \epsilon_{t,n}] > 1-\delta. \tag{15}$$

Proof. Applying a union bound over the arms to Equation 14 gives $\forall \delta \in (0,1)$,

$$\mathbb{P}\left[\forall n \in \mathcal{N}, |D_{\mathcal{C}_n,n} - \mu_n| < \sqrt{\frac{\ln(\frac{2N}{\delta})}{2} \sum_{a=1}^{A} {w_{\mathcal{C}_n,a,n}}^2 c_{a,n}} \right] > 1-\delta.$$

Let $\mathcal{H}$ be the set of all possible instantiations of the set $\mathcal{C}_n$ assuming that arm n has been pulled at least once within a time horizon of T, so $\mathcal{H}$ is a set of sets. To apply a union bound over these sets, we determine the cardinality of $\mathcal{H}$ using the constraints that each element of $\mathcal{C}_n$ is between 0 and T and the sum of the elements in $\mathcal{C}_n$ is between 1 and T. We denote the resulting cardinality $G(T,A)$, which would simply be T, as for CUCB, if all the agents were identical. We perform the union bound, use that $\mathcal{C}_{t,n} \in \mathcal{H} \forall t \in \mathcal{T}_n$, and plug in for $D_{\mathcal{C}_{t,n},n}$ and $w_{\mathcal{C}_{t,n},a,n}$, resulting in

$$\forall \delta \in (0,1), \mathbb{P}[\forall n \in \mathcal{N}, t \in \mathcal{T}_n, |\hat{\mu}_{t,n} - \mu_n| < \epsilon_{t,n}] > 1-\delta.$$

For $t \notin \mathcal{T}_n$, Equation 8 gives $\hat{\mu}_{t,n}=0.5$, and $\epsilon_{t,n}=\infty$ since $c_{t,n}=0$. Equation 15 follows directly since the difference between the true mean μ_n and 0.5 must be $<\infty$. □

Finally, in Theorem 2 we use the concentration bound on the shared empirical mean from Theorem 1 to upper bound the cumulative regret of the MIN-WIDTH algorithm.

Theorem 2. *(MIN-WIDTH Regret Bound). Suppose we act according to the MIN-WIDTH algorithm. Then $\forall \delta \in (0,1)$, the cumulative regret at time T is bounded by*

$$\mathbb{P}\left[R_T < A(N-1) + 2\sqrt{2ANT \ln\left(\frac{2NG(T,A)}{\delta}\right)} \frac{\max \mathcal{S}}{\min \mathcal{S}} \right] > 1-\delta. \tag{16}$$

Proof. We bound $\mu_n - \hat{\mu}_{t,n}$ by $\mu_n - \hat{\mu}_{t,n} \le |\hat{\mu}_{t,n} - \mu_n|$. Using the bound on $|\hat{\mu}_{t,n} - \mu_n|$ from Equation 15 and the UCB on the mean of arm n at time $t \in \mathcal{T}$ from Equation 11 gives

$$\forall \delta \in (0,1), \mathbb{P}[\forall n \in \mathcal{N}, t \in \mathcal{T}, \mu_n < \mathrm{UCB}_{t,n}] > 1-\delta. \tag{17}$$

We split Equation 7 into terms with $t<N$ and $t \ge N$:

$$R_T = \sum_{t=1}^{N-1} \sum_{a=1}^{A} s_a \left(\mu_{f^\star(a)} - \mu_{f_t(a)} \right) + \sum_{t=N}^{T} \sum_{a=1}^{A} s_a \left(\mu_{f^\star(a)} - \mu_{f_t(a)} \right). \tag{18}$$

Because $0<\mu_n<1\ \forall n\in\mathcal{N}$, the difference between the means of any two arms is bounded by $\mu_n-\mu_{n'}<1\,\forall n,n'\in\mathcal{N}$. We apply this bound to the first term in Equation 18 with $n=f^\star(a)$ and $n'=f_t(a)$ and also use the fact that $s_a\le 1\ \forall a\in\mathcal{A}$, yielding

$$R_T<A(N-1)+\sum_{t=N}^{T}\sum_{a=1}^{A}s_a\left(\mu_{f^\star(a)}-\mu_{f_t(a)}\right). \tag{19}$$

Let $R_{N:T}$ be the second term in Equation 19. Using Equation 17 with $n=f^\star(a)$ to bound $\mu_{f^\star(a)}$ gives $\forall\delta\in(0,1)$,

$$\mathbb{P}\left[R_{N:T}<\sum_{t=N}^{T}\left(\sum_{a=1}^{A}s_a\mathrm{UCB}_{t,f^\star(a)}-\sum_{a=1}^{A}s_a\mu_{f_t(a)}\right)\right]>1-\delta. \tag{20}$$

By construction, for all t the MIN-WIDTH algorithm selects a configuration f that maximizes $\sum_{a=1}^{A}s_a\mathrm{UCB}_{t,f_t(a)}$:

$$\forall t\in\mathcal{T},\ \sum_{a=1}^{A}s_a\mathrm{UCB}_{t,f^\star(a)}\le\sum_{a=1}^{A}s_a\mathrm{UCB}_{t,f_t(a)}. \tag{21}$$

We use Equation 21 in Equation 20 and plug in Equation 11 with $n=f_t(a)$, then use Equation 15 and plug in for $\epsilon_{t,f_t(a)}$, yielding

$$\forall\delta\in(0,1),\ \mathbb{P}\left[R_{N:T}<\sqrt{2\ln(2NG(T,A)/\delta)}\times\sum_{t=N}^{T}\sum_{a=1}^{A}\frac{s_a}{\sqrt{\sum_{b=1}^{A}s_b{}^2c_{t,b,f_t(a)}}}\right]>1-\delta.$$

Note that $\forall a\in\mathcal{A}, s_a\le\max\mathcal{S}$ and $\forall b\in\mathcal{A}, s_b\ge\min\mathcal{S}$. Using Equation 3 for $n=f_t(a)$ along with Lemma A.6 [10] yields

$$\forall\delta\in(0,1),\mathbb{P}\left[R_{N:T}<2\sqrt{2ANT\ln\left(\frac{2NG(T,A)}{\delta}\right)}\frac{\max\mathcal{S}}{\min\mathcal{S}}\right]>1-\delta.$$

Plugging this bound on $R_{N:T}$ into Equation 19 gives Equation 16, completing the proof. By definition of $G(T,A)$, the time dependence in $R_{N:T}$ is bounded above by $\mathcal{O}(\sqrt{T\ln(T)})$. □

7 Experimental Setup

We perform experiments to compare the efficacy of five algorithms: the one we design for this setting, MIN-WIDTH; two sensitivity-aware baselines we introduce, NO-SHARING and MIN-UCB; and two canonical baselines, CUCB and UCB.[1]

7.1 NO-SHARING

The simplest information sharing setting is not to combine rewards across agents at all. In this NO-SHARING strategy, each agent keeps track of their own UCB for each arm relying solely on their own rewards. The empirical estimator of the mean of arm n according to agent a with sensitivity s_a at time $t\in\mathcal{T}$ is

$$\hat{\mu}_{t,a,n}=\begin{cases}t\notin\mathcal{T}_{a,n} & 0.5\\ t\in\mathcal{T}_{a,n} & \frac{1}{s_ac_{t,a,n}}\sum_{\tau=1}^{t}\mathbb{1}_{f_\tau(a)=n}Y_{\tau,a,n}.\end{cases} \tag{22}$$

Let the width of agent a's confidence interval on the mean of arm n at time $t\in\mathcal{T}$ for $\delta\in(0,1)$ be

$$\epsilon_{t,a,n}=\frac{1}{s_a}\sqrt{\frac{\ln(2ANT/\delta)}{2c_{t,a,n}}}. \tag{23}$$

[1] The code and data are available on GitHub [9].

Agent a's UCB on the mean of arm n at time $t\in\mathcal{T}$ is then

$$\mathrm{UCB}_{t,a,n}=\hat{\mu}_{t,a,n}+\epsilon_{t,a,n}. \tag{24}$$

Here, each agent is almost operating in the standard UCB setting except for the assignment hierarchy, which has more sensitive agents pick which arms they want to pull before less sensitive agents. Consequently, less sensitive agents may have to pull arms that do not have the maximum $\mathrm{UCB}_{t,a,n}$.

7.2 MIN-UCB

The MIN-UCB algorithm directly improves on the naive NO-SHARING strategy. Each agent still keeps track of their own UCB for each arm, but since all the agent UCBs on the mean of a given arm hold simultaneously by Proposition A.2 [10], we can take the minimum of these UCBs to get a tighter bound. The shared UCB for arm n at time $t\in\mathcal{T}$ is then

$$\mathrm{UCB}_{t,n}=\min_{a\in\mathcal{A}}\mathrm{UCB}_{t,a,n}. \tag{25}$$

In contrast to the NO-SHARING algorithm, now agents effectively get information about arms they have not yet pulled since $\mathrm{UCB}_{t,n}<\infty$ if *any* agent has pulled arm n even if agent a has not. The algorithm will match the ith-highest-sensitivity agent with the arm with the ith-highest $\mathrm{UCB}_{t,n}$, still giving higher-sensitivity agents priority.

While MIN-UCB yields a tighter UCB than NO-SHARING, it still ignores potentially valuable information by always using the UCB of one of the agents, which accounts for the rewards collected by that agent alone. If we want to tightly bound the mean on a given arm, intuitively it makes sense to use *all* the rewards from the arm, not just those of whichever agent happens to have the lowest UCB for the arm. This is most evident in a setting where there are two agents of high sensitivity, such as 0.9 and 0.8. Perhaps the 0.9-agent has a lower UCB for an arm, but the pulls by the 0.8-agent represent additional rewards that could be used to further shrink the UCB that MIN-UCB ignores, motivating our MIN-WIDTH algorithm that combines all the agents' rewards.

Note, however, that both NO-SHARING and MIN-UCB have AT in the logarithm, which is smaller than MIN-WIDTH's $G(T,A)$ factor. This may cause MIN-WIDTH's UCBs to be higher than those of NO-SHARING and MIN-UCB in some cases. As a result, we anticipate that MIN-WIDTH may not always outperform NO-SHARING and MIN-UCB.

7.3 CUCB

CUCB [5] combines all the rewards collected from each arm to generate a UCB for the arm. The algorithm is designed for sequences of i.i.d. rewards, which is not the case in our setting because of the agent heterogeneity. This algorithm has no way of accounting for heterogeneous agents, so we naively combine observations across agents in our implementation, ignoring the fact that the i.i.d. assumption does not hold. Consequently, it may never be able to learn the optimal agent-arm assignments. For CUCB we use Equation 26 for the UCB of arm n at time t. Since this algorithm was not designed for heterogeneous agents, we randomly assign agents to the top-UCB arms at each time step.

$$\mathrm{UCB}_{t,n}=\frac{\sum_{\tau=1}^{t}\sum_{a=1}^{A}\mathbb{1}_{f_\tau(a)=n}Y_{\tau,a,n}}{\sum_{\tau=1}^{t}\sum_{a=1}^{A}\mathbb{1}_{f_\tau(a)=n}}+\sqrt{\frac{\ln(2Nt/\delta)}{2\sum_{\tau=1}^{t}\sum_{a=1}^{A}\mathbb{1}_{f_\tau(a)=n}}} \tag{26}$$

7.4 UCB

The standard UCB algorithm [2] maintains a UCB on the mean reward of every action that can be taken at each time step. In our implementation, we treat every super-arm as an arm and apply the UCB algorithm to every super-arm. We use Equation 27 for the UCB of super-arm f at time t and pull the super-arm with the highest UCB at each time step. By treating each super-arm as an arm, the UCB algorithm is implicitly able to account for the heterogeneity among the agents. However, it does not combine any information across agents or arms, making it increasingly unsuitable as the number of agents or arms increases.

$$\mathrm{UCB}_{t,f} = \frac{\sum_{\tau=1}^{t}\sum_{a=1}^{A}\mathbb{1}_{f_\tau=f}Y_{\tau,a,f(a)}}{\sum_{\tau=1}^{t}\mathbb{1}_{f_\tau=f}} + \sqrt{\frac{\ln(2N!t/\delta(N-A)!)}{2\sum_{\tau=1}^{t}\mathbb{1}_{f_\tau=f}}} \quad (27)$$

7.5 Implementation

We implement MIN-WIDTH, NO-SHARING, and MIN-UCB as described in §5, §7.1, and §7.2, respectively, setting $T \to t$ in Equations 9 and 23. While using T facilitates the regret analysis, using t in our experiments allows us to assess the performance at different times with a single run and also compare across simulations. All graphs display the cumulative regret averaged over 90 trials with two standard errors and use $\delta = 0.05$.

8 Results

We perform simulations in four domains: three pseudo-synthetic domains with parameter values inspired by real data—COVID test allocation, hotel recommendation, and poaching prevention—and one fully synthetic domain, in which we vary the parameters of the simulations to study trends in the algorithms' behavior across problem settings. Finally, we test the robustness of the algorithms to estimating the agent sensitivities.

8.1 COVID Test Allocation

For our COVID simulation, suppose we have capacity to allocate COVID tests to 5 out of 6 floors of a college dorm with the following (assumed to be unknown) prevalence rates: $\mu = \{0.05,0.1,0.12,0.15,0.25,0.3\}$. Each pull of a super-arm corresponds to distributing 5 tests among 6 floors, where the choice of who to test on each floor is random. We have 3 antigen tests and 2 PCR tests to distribute each day, with sensitivities to COVID of 80% and 95%, respectively [6]: $\mathcal{S} = \{0.8,0.8,0.8,0.95,0.95\}$. We note that we are not making a prescriptive claim as to how schools should operate COVID testing but rather how our algorithm could be used to allocate tests accounting for the different sensitivities of the test types.

We simulate sequential super-arm pulls and show the results in Figure 1a, where we see all the algorithms perform similarly for the initial time steps, after which MIN-WIDTH performs the best. CUCB performs similarly to MIN-UCB and NO-SHARING, which is reasonable since the test sensitivities are not very different, so ignoring their heterogeneity and combining all the results is not a terrible strategy.

8.2 Hotel Recommendation

For our hotel simulation, we extract cleanliness rates for four Punta Cana hotels with more than 500 reviews on TripAdvisor [21]. The probability that each hotel is clean is the mean of these ratings: $\mu = \{0.72,0.74,0.93,0.61\}$. Sensitivity is the probability that a customer is satisfied with their stay given that the hotel is clean. We consider a set of four customer types $\mathcal{S} = \{0.3,0.5,0.7,0.9\}$ and match each type with one of the four hotels, assuming that their relative cleanliness likelihoods are initially unknown. We aim to maximize overall satisfaction by matching the customers that care most about cleanliness with the cleanest hotels. Assuming that space at the hotels is limited and someone will need to be matched with a hotel that tends to be less clean, we incur the lowest cost to our customers' satisfaction if we match the customer whose satisfaction is least correlated with the hotel's cleanliness to a less clean hotel. That way, we keep spots open at the cleanest hotels for the customers who are very likely to be dissatisfied if their hotel is not clean.

In Figure 1b, MIN-WIDTH outperforms the other four algorithms at all times, and the relative ordering of MIN-WIDTH, MIN-UCB, and NO-SHARING is consistent with the amount of information shared across agents. Both canonical baselines perform poorly; UCB has $4! = 24$ super-arms to learn about, and the agent sensitivities vary widely, making CUCB's assumption that the rewards from a given arm are i.i.d. even less appropriate than if the agent sensitivities were more similar.

8.3 Poaching Prevention

Consider a hypothetical park with five areas, each with a different probability of containing a snare: $\mu = \{0.1,0.3,0.5,0.7,0.9\}$. If rangers find a snare, they get a reward of 1, and if not, they receive 0 reward. Since even the best ranger teams probably cannot find more than $1/3$ of the snares present [11], we consider teams of two, three, and five rangers with sensitivities $\mathcal{S} = \{0.2,0.3\}$, $\mathcal{S} = \{0.1,0.2,0.3\}$, and $\mathcal{S} = \{0.1,0.1,0.1,0.2,0.3\}$, respectively.

Comparing Figures 2a-c shows that when there are fewer rangers, it takes much more time for the algorithms' performance to diverge, which makes sense because less information is collected during each round of patrols and the rangers' detection rates are similar. As we add rangers, even though they have very low sensitivity, the relative benefit of using MIN-WIDTH increases, as the low-sensitivity rangers assist with exploration and allow the higher-sensitivity rangers to exploit the best areas. The difference in performance between the two canonical baselines and the three sensitivity-aware algorithms also increases as we add rangers.

8.4 Fully Synthetic

We perform additional simulations for a variety of arm means and agent sensitivities to explore the long-range (high T) performance of the five algorithms, with results presented in Appendix B [10]. The results, summarized in Table B1 [10], demonstrate that this problem setting benefits from an approach distinct from a more general stochastic bandit algorithm. In the 2×2 (2 agents, 2 arms) and 3×3 experiments MIN-WIDTH has the best long-range performance, which is consistent with Figures 1b and 2c, in which MIN-WIDTH performed the best when the numbers of agents and arms were the same. When there are fewer agents than arms, MIN-UCB sometimes outperforms MIN-WIDTH. This suggests that sharing rewards across agents is most useful when there are enough agents to continue exploring so that the best agents can exploit the seemingly best arms sooner.

8.5 Sensitivity Robustness

We explore how robust the sensitivity-aware algorithms' performance is to imperfect knowledge of the agent sensitivities, which we may not always know exactly. When computing the UCBs and assigning agents to arms,

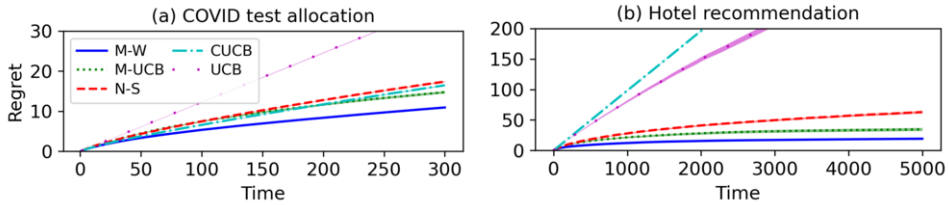

Figure 1: Regret plotted over time for the COVID test allocation (left) and hotel recommendation (right) domains.

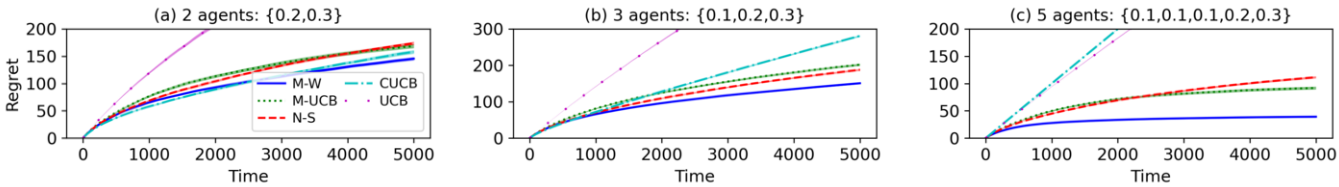

Figure 2: Regret plotted over time for the poaching prevention domain with varying agent sensitivities.

the planner uses the estimated sensitivities rather than the true, assumed-to-be-unknown sensitivities used to compute the regret. This change would have no effect on the canonical baselines, as they never model sensitivities explicitly. We rerun the COVID experiment with three sets of estimated sensitivities: all overestimated ($\tilde{\mathcal{S}} = \{0.85, 0.85, 0.85, 0.98, 0.98\}$), all underestimated ($\tilde{\mathcal{S}} = \{0.75, 0.75, 0.75, 0.9, 0.9\}$), and a mix ($\tilde{\mathcal{S}} = \{0.75, 0.75, 0.75, 0.98, 0.98\}$). The percent change in regret is shown in Table 1 and the cumulative regret is shown in Table 2 at $t = 300$ across 500 trials.

For all algorithms, overestimating the sensitivities on average hurts performance less than underestimating them in this case. For MIN-WIDTH and MIN-UCB, underestimating the antigen test sensitivities and overestimating the PCR test sensitivities hurts performance the most, possibly because the PCR tests are more frequently allocated to lower-COVID floors to perform exploration that the antigen tests are not considered accurate enough to reliably cover. Among the three sensitivity-aware algorithms, NO-SHARING is least affected by these sensitivity approximations because the relative ordering of the agents is unchanged and the agents' UCBs are not affected by the estimation error in the sensitivities of the other agents. Overall, MIN-WIDTH had the lowest regret in the original experiment and is sufficiently robust for it to continue outperforming the other algorithms in the three robustness experiments (Figures B2-B4 [10]).

Table 1: Percent change in regret (mean ± one SE) when using estimated vs. true sensitivities for the COVID experiment.

Algorithm	Overestimated	Underestimated	Mix
M-W	0.7±0.9	2.5±1.0	9.5±1.0
M-UCB	0.7±1.6	9.1±1.8	28.9±1.9
N-S	0.9±0.5	1.5±0.5	1.0±0.5

Table 2: Cumulative regret (mean ± one SE) when using estimated vs. true sensitivities for the COVID experiment.

Algorithm	Overestimated	Underestimated	Mix
M-W	10.8±0.1	11.0±0.1	11.7±0.1
M-UCB	14.0±0.1	15.2±0.2	18.0±0.2
N-S	17.5±0.1	17.6±0.1	17.5±0.1

9 Discussion

Overall, either MIN-WIDTH or MIN-UCB has the best long-range performance when the agents are heterogeneous, with CUCB performing best for identical agents. By combining observations across agents, its confidence intervals will shrink faster than those of MIN-UCB and there is no agent heterogeneity it fails to account for in that case. When there are more arms than agents, MIN-UCB sometimes achieves lower cumulative regret than MIN-WIDTH. As remarked in §7.2, the confidence intervals for MIN-WIDTH may be wider because of the $G(T,A)$ factor from the union bound in Theorem 1 over the possible instantiations of C_n, the set of times every agent has pulled arm n. This set depends on the number of agents but not their sensitivities, so it does not capture the fact that some agents are more similar than others, which we believe yields a resulting cardinality $G(T,A)$ that essentially overcounts.

For a clear example, consider two possible instantiations of the set C_n: {4,5,6} and {5,4,6}. In the first instantiation, agent 1 has pulled arm n four times, agent 2 five times, and agent 3 six times, while in the second instantiation, agent 1 has pulled it five times and agent 2 four times. Now suppose that agents 1 and 2 have the same sensitivity, making them interchangeable. In this case, these two instantiations of the set are not actually distinct and should only be counted as one instantiation rather than two (as is being done now). Future work could potentially resolve this by replacing the union bound with a different bound that is a function of the agent sensitivities.

Introducing the form of heterogeneous agent sensitivities that we study here gives rise to more realistic bandit models for myriad domains as discussed in §2. To model certain real-world applications with even more fidelity, future work could consider settings where the arm means change with time (e.g., COVID prevalence rates), possibly in response to arm pulls. One could also model agent sensitivities that vary across multiple dimensions (e.g., customer preferences for cleanliness, service, etc.). The constraint preventing multiple agents from pulling the same arm could also be relaxed, which would necessitate a model for interaction effects. Theoretically analyzing the estimated-sensitivity setting is another direction for future work that our robustness experiment opens up.

10 Conclusion

We introduce a stochastic multi-armed bandit problem with heterogeneous agents distinguished by a sensitivity parameter that uniquely characterizes their reward function for a given arm. We develop a method for assigning agents to arms that decomposes the combinatorial problem into one of learning the arm means while prioritizing the highest-sensitivity agents during arm assignment. Our MIN-WIDTH algorithm combines all of the rewards with a heterogeneity-aware weighting strategy. We provide a regret bound for MIN-WIDTH and evaluate it in simulations inspired by COVID test allocation, hotel recommendation, and poaching prevention, as well as a fully synthetic domain. Our results show that modeling agent heterogeneity tends to be most useful when the sensitivities are more diverse across a collection of agents and that sharing more information does not always improve performance.

Ethics Statement

In our paper we introduce a model that may be able to more accurately capture certain elements of real-world systems than past work, especially those with varying agent sensitivities. Having a more complete model can lead to better performance in appropriate domains, which we demonstrate with our COVID test allocation, hotel recommendation, and poaching prevention experiments. Nonetheless, any automatic decision-making tool has inherent risks and should be used in appropriate contexts by people with knowledge of its capabilities and limitations. Any application of our methods to real-world settings would require much more perspective and sociotechnical analysis beyond the algorithmic contribution we offer here.

Acknowledgements

We would like to thank Lucas Janson for helpful discussions. L.G. was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE2140743. E.R. was supported by the Harvard Data Science Initiative and the Harvard Center for Research on Computation and Society.

References

[1] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2009.01.016. URL https://www.sciencedirect.com/science/article/pii/S030439750900067X. Algorithmic Learning Theory.

[2] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

[3] H. Bastani, K. Drakopoulos, V. Gupta, I. Vlachogiannis, C. Hadjichristodoulou, P. Lagiou, G. Magiorkinis, D. Paraskevis, and S. Tsiodras. Efficient and targeted COVID-19 border testing via reinforcement learning. *Nature*, 599(7883):108–113, Sep 2021. doi: 10.1038/s41586-021-04014-z.

[4] N. Cesa-Bianchi, T. R. Cesari, and R. D. Vecchia. Cooperative online learning with feedback graphs, 2022.

[5] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework, results and applications. *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[6] FDA. At-Home COVID-19 Antigen Tests-Take Steps to Reduce Your Risk of False Negative Results: FDA Safety Communication, Nov 2022. URL https://www.fda.gov/medical-devices/safety-communications/home-covid-19-antigen-tests-take-steps-reduce-your-risk-false-negative-results-fda-safety.

[7] Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial Network Optimization With Unknown Variables: Multi-Armed Bandits With Linear Rewards and Individual Observations. *IEEE/ACM Transactions on Networking*, 20(5): 1466–1478, 2012. doi: 10.1109/TNET.2011.2181864.

[8] A. Ghosh, A. Sankararaman, and K. Ramchandran. Multi-agent heterogeneous stochastic linear bandits. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part IV*, page 300–316, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-26411-5. doi: 10.1007/978-3-031-26412-2_19. URL https://doi.org/10.1007/978-3-031-26412-2_19.

[9] L. Gordon, E. Rolf, and M. Tambe. Code and data for "Combining Diverse Information for Coordinated Action: Stochastic Bandit Algorithms for Heterogeneous Agents". *GitHub*, 2024. URL https://github.com/lgordon99/heterogeneous-stochastic-bandits.

[10] L. Gordon, E. Rolf, and M. Tambe. Combining Diverse Information for Coordinated Action: Stochastic Bandit Algorithms for Heterogeneous Agents, 2024. URL https://arxiv.org/abs/2408.03405. Full version of this paper.

[11] J. Hance. Rangers find 109,217 snares in a single park in cambodia. The Guardian, May 2018. URL https://www.theguardian.com/environment/radical-conservation/2018/may/22/snares-southeast-asia-cambodia-vietnam-laos-tigers-elephants-saola.

[12] U. Madhushani and N. E. Leonard. Heterogeneous stochastic interactions for multiple agents in a multi-armed bandit problem. In *2019 18th European Control Conference (ECC)*, pages 3502–3507, 2019. doi: 10.23919/ECC.2019.8796036.

[13] I. Rejwan and Y. Mansour. Top-k combinatorial bandits with full-bandit feedback, 2019.

[14] E. Rolf, D. Fridovich-Keil, M. Simchowitz, B. Recht, and C. Tomlin. A Successive-Elimination Approach to Adaptive Robotic Source Seeking. *IEEE Transactions on Robotics*, 37(1):34–47, 2021. doi: 10.1109/TRO.2020.3005537.

[15] A. Sankararaman, A. Ganesh, and S. Shakkottai. Social Learning in Multi Agent Multi Armed Bandits. *Proc. ACM Meas. Anal. Comput. Syst.*, 3(3), dec 2019. doi: 10.1145/3366701. URL https://doi.org/10.1145/3366701.

[16] S. Shahrampour, A. Rakhlin, and A. Jadbabaie. Multi-armed bandits in multi-agent networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2786–2790, 2017. doi: 10.1109/ICASSP.2017.7952664.

[17] C. Shi, W. Xiong, C. Shen, and J. Yang. Heterogeneous multi-player multi-armed bandits: Closing the gap and generalization. In *35th Conference on Neural Information Processing Systems*, pages 1–6, 2014. doi: 10.1109/CISS.2014.6814096.

[18] S. Solanki, S. Kanaparthy, S. Damle, and S. Gujar. Differentially private federated combinatorial bandits with constraints, 2023.

[19] M. E. Taylor, M. Jain, Y. Jin, M. Yokoo, and M. Tambe. When should there be a "me" in "team"? distributed multi-agent optimization under uncertainty. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, page 109–116, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9780982657119.

[20] D. Vial, S. Shakkottai, and R. Srikant. Robust Multi-Agent Multi-Armed Bandits. In *Proceedings of the Twenty-Second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, MobiHoc '21, page 161–170, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385589. doi: 10.1145/3466772.3467045. URL https://doi.org/10.1145/3466772.3467045.

[21] H. Wang. Trip Advisor Data (00040). PrefLib: A Library for Preferences, Aug. 2013. URL https://www.preflib.org/dataset/00040.

[22] N. Wanigasekara, Y. Liang, S. T. Goh, Y. Liu, J. J. Williams, and D. S. Rosenblum. Learning Multi-Objective Rewards and User Utility Function in Contextual Bandits for Personalized Ranking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3835–3841. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/532. URL https://doi.org/10.24963/ijcai.2019/532.

[23] L. Xu, E. Bondi, F. Fang, A. Perrault, K. Wang, and M. Tambe. Dual-mandate patrols: Multi-armed bandits for green security. *The Thirty-Fifth AAAI Conference on Artificial Intelligence Proceedings*, 2021.

[24] L. Xu, A. Biswas, F. Fang, and M. Tambe. Ranked Prioritization of Groups in Combinatorial Bandit Allocation. In *Proc. 31st International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

[25] L. Yang, Y.-Z. J. Chen, M. H. Hajiemaili, J. C. S. Lui, and D. Towsley. Distributed bandits with heterogeneous agents. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 200–209, 2022. doi: 10.1109/INFOCOM48880.2022.9796901.