Online Learning for Multi-Agent Based Resource Allocation in Weakly Coupled Wireless Systems

Jianhan Song
The University of Texas at Austin
Austin, Texas, USA
jianhansong@utexas.edu

Gustavo de Veciana
The University of Texas at Austin
Austin, Texas, USA
deveciana@utexas.edu

Sanjay Shakkottai
The University of Texas at Austin
Austin, Texas, USA
sanjay.shakkottai@utexas.edu

ABSTRACT

We propose and evaluate a learning-based framework to address multi-agent resource allocation in coupled wireless systems. In particular we consider, multiple agents (e.g., base stations, access points, etc.) that choose amongst a set of resource allocation options towards achieving their own performance objective /requirements, and where the performance observed at each agent is further coupled with the actions chosen by the other agents, e.g., through interference, channel leakage, etc. The challenge is to find the best collective action. To that end we propose a Multi-Armed Bandit (MAB) framework wherein the best actions (aka arms) are adaptively learned through online reward feedback. Our focus is on systems which are "weakly-coupled" wherein the best arm of each agent is invariant to others' arm selection the *majority* of the time – this majority structure enables one to develop light weight efficient algorithms. This structure is commonly found in many wireless settings such as channel selection and power control. We develop a bandit algorithm based on the Track-and-Stop strategy, which shows a logarithmic regret with respect to a genie. Finally through simulation, we exhibit the potential use of our model and algorithm in several wireless application scenarios.

CCS CONCEPTS

• **Networks** → **Mobile networks**; *Network resources allocation*; *Packet scheduling*.

KEYWORDS

Scheduling, Wireless Networks, Online Learning, Bandit Algorithms

ACM Reference Format:
Jianhan Song, Gustavo de Veciana, and Sanjay Shakkottai. 2022. Online Learning for Multi-Agent Based Resource Allocation in Weakly Coupled Wireless Systems. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3492866.3549714

1 INTRODUCTION

Dynamic resource allocation, including the allocation of time slots, frequency sub-channels, power, etc., is a key part of the design of wireless systems. In a multi-cell setting, resource allocation is especially challenging due to the triad of: (i) heterogeneity and uncertainty of the network environment (e.g., time-varying loads, channel states, interference, etc.), (ii) distributed decision making (separate controller/agent in each base-station), and (iii) availability of only partial state information at each agent (e.g., only local channel states). In such settings, if each agent selects their own allocation strategy/action without consideration of other agents' decisions, the collective can suffer a significant loss in total utility.

We can view the multi-agent resource allocation problem through the following abstraction. Each agent is allowed an action from among a collection of actions (e.g., choice of frequency sub-band in the channel selection problem). Its choice of action has two consequences: (a) the agent accrues a *reward* for itself (e.g., average throughput/delay for users in its cell), and (b) the action induces an *environment* that affects all other agents (e.g., transmitting on a frequency sub-band generates strong interference to other agents in that frequency sub-band, and weaker interference in nearby frequency sub-bands). In a multi-agent setting, the goal then is to find an action for each of the agents (equivalently, a collective of actions across agents), which in-turn induces a collective of environments, such that the utility of the collective is maximized.

The immediate search-based solution to this problem – attempt every action at each agent for a sufficiently long duration, empirically estimate the collective reward, and choose the collective that has highest utility – can scale poorly due to the super-linear growth in search space. Indeed, even with two users, the number of environments scales as k^2 if each user has k possible actions, making it computationally impractical to learn the best actions within a reasonable time. In general, with no assumptions on the actions and the resulting environments, it is not hard to see that such complexity is unavoidable.

However, in many resource allocation settings that we are interested in, there are additional properties of the overall system that can be used to reduce this complexity. Specifically in this paper, we focus on systems that are *weakly coupled.* We say a system is weakly-coupled if it satisfies the *majority condition*: we suppose that the optimal action of an agent is also the best action in a majority of environments, where each environment corresponds to a distinct action tuple that can be chosen by the other agents. The intuition is that under moderate interference levels, *most of the time*, the performance of one agent's action does not fluctuate much when actions taken by nearby agents are changed. The majority condition holds in several wireless settings. For example, in the channel selection

MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9165-8/22/10…$15.00
https://doi.org/10.1145/3492866.3549714

problem, once an agent selects a frequency sub-band, only a small set of adjacent channels will be significantly interfered with due to channel leakage. Another example is one where each base-station can choose a *scheduler* from among a candidate set [22]. Different schedulers trade-off for different performance metrics within the cell (e.g. MaxWeight for stability, vs. round robin for jitter); however, they have different impact on neighboring cells. In this setting, good schedulers tend to incur low interference to nearby agents (cells) since they typically schedule opportunistically and use channels more efficiently (therefore, the majority condition holds provided that most of the schedulers are "good").

The majority condition is especially useful for algorithm design, because we show that it has the following three properties. *(1) Local greedy property:* For each agent, it suffices to learn its best action in each of the environment and choose the "majority best" as its overall best action; *(2) Avoiding hard environments:* Identifying each agent's best action can be cast as k separate multi-armed bandit best-arm identification instances, where k is the number of possible environments. Some of these environments might be especially hard, e.g., strong interference/poor channel quality, thus all actions of the agent in this "hard" environment have low reward, making this best arm identification instance difficult. Crucially however, the majority conditions enables one to avoid solving such hard environments, once the best actions from the easiest $k/2$ environments has been learned; *(3) Sub-sampling property:* When the number of environments k is large, it is possible to sample a subset of environments and still learn the best action (with high probability).

Building on these properties, we develop a decentralized algorithm for multi-agent resource allocation with bandit feedback. The algorithm proceeds episodically with each episode consisting of an exploration and an exploitation phase. During the exploration phase, one agent runs a collection of best-arm-identification subroutines to learn the optimal arm (aka action) in each environment based on local reward feedback, while the other agents cycle over actions from a randomly-chosen subset (of all the actions) in a round-robin fashion, until the first agent learns the "majority best arm" with a *fixed confidence* (crucially, not all environments have to be "solved"). Once each agent learns the best collective arm using the above procedure, it is applied in the exploitation phase. As the episode index grows, the confidence level is made increasingly tight as the increment of regret converges to zero. We build on *Track-and-Stop* [8], which is designed for best arm identification with fixed confidence, as the subroutines used in the main algorithm. Track-and-Stop focuses on exploring arms with good rewards, and is known to be asymptotically optimal in terms of the number of plays needed for determining the best arm. This further accelerates the exploration and improves the overall performance (in particular, compared to the vanilla Explore-Then-Commit (ETC) approach).

Our main contributions are summarized as follows:

1. Weakly Coupled Systems under the Majority Condition: We develop a multi-armed bandit framework to address the multi-agent resource allocation problem for weakly coupled systems. In these systems, the best arm of each agent is invariant to other agents' arm choices in the majority of scenarios. We believe this assumption is reasonable in many wireless applications, and allows the design of an algorithm with manageable computational and communication costs.

2. Track-and-Stop Based Decentralized Algorithm: We develop a decentralized bandit algorithm specifically designed for weakly coupled systems based on Track-and-Stop. For systems satisfying the majority condition, this algorithm has two main advantages over classical bandit algorithms: (1) Low communication cost: the decision making is decentralized as no reward/action information is exchanged and the only coordination needed is when one agent signaling others the end of a Track-and-Stop subroutine. Note that for centralized algorithms such as UCB or the vanilla Track-and-Stop (i.e., best arm identification among all the collective arms), a central controller who has access to all the reward feedback has to be introduced to determine the action for each agent. (2) Efficient with a logarithmic regret: it can be shown that with high probability the regret scales as $O((m-1)k \log k \log T)$ where T is the time horizon, m is the number of agents and k is the (max) number of arms of each agent — this is much improved compared to any classical algorithm which equally views all the k^m collective arms in implementation, with the regret scaling as $O(k^m \log T)$.

3. Empirical Evaluation: We simulate the algorithm in two wireless applications to show the potential usage of our model: (1) channel selection with power leakage and (2) best scheduler selection for wireless queueing systems. In both cases, we show the systems are indeed weakly-coupled such that our algorithm can be applied. Furthermore, our simulations show that the agents can correctly learn the best collective action in reasonable time with a sub-linear regret.

1.1 Related Work

Multi-Agent Resource Allocation in Wireless Settings. Many well-studied wireless applications are by nature multi-agent resource allocation problems, such as power control and cognitive spectrum access. A classical theoretic approach is to study the problems through a game theory perspective, e.g., [10, 11] on power control, [25, 26] on dynamic spectral access and cognitive radio, [5, 9, 21] on wireless sensor networks, [19] on edge computing, etc. Moreover, due to the complexity of the problem, machine learning/reinforcement learning techniques have recently be proposed to address related problems, see e.g., [1, 27, 28].

Decentralized Multi-Agent MABs. Multi-agent decision making has been formulated as decentralized multi-armed bandit problems, where multiple players simultaneously pull their arms at each round. In a collaborative setting, the agents learn the same stochastic bandit instance in a decentralized manner, and the goal is to minimize individual regret via information sharing, see, e.g., [6, 12, 14, 24]. Recent works [7, 13] further consider the tradeoffs between regret minimization and communication cost.

More aligned with this paper is the study on multi-agent MABs with *collision.* In those problems, agents receive normal reward feedback only if other agents do not choose the same arm ("collision") — otherwise zero rewards are observed by colliding agents. Several settings have been studied in this line of work, including [2, 3, 20] on the homogeneous reward setting (agents observe the same reward distributions on the same arm), and [4, 16, 18] on the heterogeneous reward setting. A recent work [17] further explores the scenario when agents observe non-zero rewards on collisions. Compared to these works, our model is more general regarding the impact of interference on rewards — we do not restrict to a collision-based model and the reward distribution of an arm may be different when nearby agents change to any arm (not necessarily the "colliding arm"). Instead, we explore a special arm-reward structure, i.e., weakly coupled systems under the majority condition, and develop efficient decentralized bandit algorithms.

1.2 Notation

Throughout the paper, we use $[n]$ to denote the set $\{1, 2, \cdots, n\}$, and $\mathbb{1}$ for the $\{0, 1\}$ indicator function. The symbols $\lceil a \rceil$ and $\lfloor a \rfloor$ represent the ceiling and floor function over the value a.

2 PROBLEM FORMULATION

2.1 Two-Agent Weakly Coupled Systems

For simplicity let us first focus on a 2-agent system and introduce the notion of weak coupling. Here on, we use the standard bandit terminology of 'arm' to denote an agent's action. In this system, Agent 1 and Agent 2 can choose one over k_1 and k_2 arms respectively for each play (round). We call any pair of arms (i_1, i_2) a collective arm. The joint rewards for two agents choosing $(i_1, i_2) \in [k_1] \times [k_2]$ are independently and identically distributed over multiple plays, and the average rewards are denoted as $(\mu^{(1)}_{i_1,i_2}, \mu^{(2)}_{i_1,i_2})$. Note that the rewards are "coupled" and changing either arm of (i_1, i_2) might affect both rewards $(\mu^{(1)}_{i_1,i_2}, \mu^{(2)}_{i_1,i_2})$.

As usual the goal of a bandit framework is to find the best (collective) arm. Let the arm pair (i_1^*, i_2^*) satisfy that $\mu^{(1)}_{i_1^*,i_2^*} + \mu^{(2)}_{i_1^*,i_2^*} > \mu^{(1)}_{i_1,i_2} + \mu^{(2)}_{i_1,i_2}$ for all $(i_1, i_2) \in [k_1] \times [k_2]$. Simply applying classical bandit algorithms (such as UCB) in this problem can be challenging and problematic, since it requires a centralized controller observing rewards from both agents and exploring all $k_1 k_2$ arms, leading to high communication and computational cost.

Therefore, in this paper, we consider *weakly coupled systems*, which have a special arm-reward structure such that only minimal communication between agents is needed — in particular, no reward/action information is required to be shared — and that fewer arm pairs are necessarily explored to locate the best collective arm. Before formally define the condition regarding weak coupling, we introduce several notations as follows: Denote $i_1^*(j)$ as the best arm for Agent 1 when Agent 2 plays arm j for any $j \in [k_2]$, i.e., $i_1^*(j) = \arg\max_{i'} \mu^{(1)}_{i',j}$. Similarly, let $i_2^*(j) = \arg\max_{i'} \mu^{(2)}_{j,i'}$ for any $j \in [k_1]$. Let $c_1(i) = \sum_{j\in[k_2]} \mathbb{1}\{i=i_1^*(j)\}$ (i.e., the number of Agent 2 choices, aka "environments", resulting in arm i being the best arm for Agent 1) and similarly, $c_2(i) = \sum_{j\in[k_1]} \mathbb{1}\{i=i_2^*(j)\}$.

We call a system weakly coupled if it satisfies the following *majority condition*:

Condition 1 (Majority Condition). *Suppose there exist an arm pair* $(i_1^M, i_2^M) \in [k_1] \times [k_2]$ *such that* $c_1(i_1^M) \geq (1+\gamma)k_2/2$, *and* $c_2(i_2^M) \geq (1+\gamma)k_1/2$ *for some* $0 < \gamma \leq 1$. *Furthermore, assume that* $\mu^{(1)}_{i_1^M,i_2^M} > \mu^{(1)}_{i_1,i_2}$ *and* $\mu^{(2)}_{i_1^M,i_2^M} > \mu^{(2)}_{i_1,i_2}$ *for any* $(i_1, i_2) \in [k_1] \times [k_2]$.

Arm i_1^M of Agent 1 is the best choice for him for majority of Agent 2's selections, and analogously for i_2^M (an illustration is given in Figure 1.). We call i_1^M, i_2^M the *majority arms* of both agents (hence

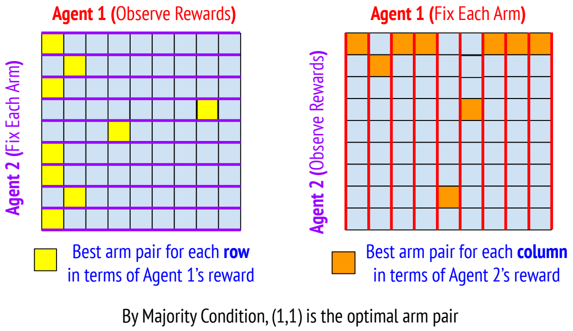

Agent 1 (Observe Rewards)
Agent 2 (Fix Each Arm)
Agent 1 (Fix Each Arm)
Agent 2 (Observe Rewards)
Best arm pair for each row in terms of Agent 1's reward
Best arm pair for each column in terms of Agent 2's reward
By Majority Condition, (1,1) is the optimal arm pair

Figure 1: An illustration of the majority condition.

the notation). Clearly, the majority arm pair is the optimal, i.e., (i_1^*, i_2^*), if the condition holds.

To understand the intuition of this condition, first consider the case when there is no coupling, i.e, $\mu^{(1)}_{i_1,i_2}$ is a constant for any $i_2 \in [k_2]$ when fixing i_1 (similar for $\mu^{(2)}_{i_1,i_2}$ when fixing i_2). The majority condition holds with $\gamma = 1$. In this case, each agent can locate the best arm solely based on the observed rewards itself.

With more coupling, the mean rewards observed by one agent are no longer constant as the other agent changes arms — however, in a weakly-coupled system, we assume the change of arm at the other agent will not affect the best arm *majority of time.* In other words, only a small number of actions by the other agent make a significantly negative impact on the best arm (actually a stronger condition would be that only a few arm pairs lead to significant reward degradation, but we focus on the best arm exclusively). As we will see, with more robustness this arm-reward structure still preserves the property that local reward feedback is sufficient for the best arm identification of each agent.

Remark 1 (Weakly Coupled Wireless Systems). *Weak coupling can be found in several wireless settings. Two examples that we consider in this paper are: (i) channel selection across multiple base stations, with coupling due to interference leakage across adjacent channels, and (ii) scheduler selection at multiple base stations, with coupling due to the out-of-cell interference resulting from the transmission patterns induced by the chosen scheduler. We study both these settings in Section 4, where we discuss the nature of coupling, as well as the efficiency benefits of our approach.*

2.2 An Alternative Condition, Regret

Condition 1 naturally captures the weak-coupling nature of some applications. In Condition 1, both agents are "symmetric". Here, we introduce a non-symmetric, weaker notion as follows.

Condition 2 (Weaker Majority Condition). *Suppose there exist an arm* i_1^M *such that* $c_1(i_1^M) \geq (1+\gamma)k_2/2$. *Furthermore, assume that* $\mu^{(1)}_{i_1^M,i_2^*(i_1^M)} + \mu^{(2)}_{i_1^M,i_2^*(i_1^M)} \geq \mu^{(1)}_{i_1^*,i_2^*} + \mu^{(2)}_{i_1^*,i_2^*}$.

Note that Condition 1 strictly implies Condition 2 since i_2^M must be $i_2^*(i_1^M)$ under Condition 1 — therefore, it is better to adopt a more general notion. Consider the channel selection example: with some small probability, the majority arms of both agents might happen to "collide" with each other (i.e., being adjacent channels). Then it is preferred to aim at finding the arm pair $(i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}}))$ rather than $(i_1^{\mathsf{M}}, i_2^{\mathsf{M}})$ when we design an algorithm so as to avoid the collision (when there is no collision, Condition 2 becomes Condition 1). In practice, even when Condition 2 is not held, the pair $(i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}}))$ still gives acceptable "near-optimal" rewards for similar settings which involve collision avoiding.

The goal is to develop an efficient and communication-light bandit algorithm to minimize regret. We define the regret as the loss of rewards with respect to the arm pair $(i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}}))$ in accordance with Condition 2.[1] Let $(I_1(t), I_2(t))$ denote the arm pair selected by the two users at time t. The regret with horizon T is defined as

$$\mathrm{Regret}_T = \mathbb{E}\left[\sum_{t=1}^{T}(\mu^{(1)}_{i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}})} - \mu^{(1)}_{I_1(t), I_2(t)}) \right.$$
$$\left. + \sum_{t=1}^{T}(\mu^{(2)}_{i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}})} - \mu^{(2)}_{I_1(t), I_2(t)})\right].$$

When Condition 2 holds, the regret expression above reduces to the normal definition (i.e., with respect to the best pair (i_1^*, i_2^*)). We use $i_1^{\mathsf{M}}, i_2^*(i_1^{\mathsf{M}})$ in the current definition to allow general comparisons.

2.3 Generalization to Multi-Agent Systems

The model described above can be generalized to systems with more than 2 agents. For notation simplicity we consider a 3-agent system in this subsection. Let $i_1^*(\cdot, i_2, i_3) \in [k_1]$ be the best arm for Agent 1 when Agent 2 and Agent 3 play $i_2 \in [k_2]$ and $i_3 \in [k_3]$ respectively. (Arm $i_2^*(i_1, \cdot, i_3)$ and arm $i_3^*(i_1, i_2, \cdot)$ are analogously defined.) The majority condition is stated as follows:

Condition 3 (Majority Condition: 3-Agent System). *Suppose there exist an arm $i_1^{\mathsf{M}} \in [k_1]$ such that*

$$\sum_{(i_2, i_3) \in [k_2] \times [k_3]} \mathbb{1}_{\{i_1^{\mathsf{M}} = i_1^*(\cdot, i_2, i_3)\}} \geq (1+\gamma)(k_2 k_3)/2,$$

and an arm $i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}} \in [k_2]$ such that

$$\sum_{i_3 \in [k_3]} \mathbb{1}_{\{i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}} = i_2^*(i_1^{\mathsf{M}}, \cdot, i_3)\}} \geq (1+\gamma) k_3/2.$$

Furthermore, assume that $(i_1^{\mathsf{M}}, i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}}, i_3^*(i_1^{\mathsf{M}}, i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}}, \cdot))$ is the best collective arm in terms of sum (mean) rewards.

Note that we follow the non-symmetric pattern of the alternative condition in Section 2.2. Accordingly, the regret is defined as the loss with respect to the triplet $(i_1^{\mathsf{M}}, i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}}, i_3^*(i_1^{\mathsf{M}}, i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}}, \cdot))$.

3 ALGORITHM DESIGN AND ANALYSIS

3.1 Building Block: Track-and-Stop

Our algorithm applies the *Track-and-Stop (T-a-S)* algorithm [8] as subroutines to locate the best arm in each environment. Track-and-Stop is a single-agent bandit algorithm for the purpose of *best arm identification* — the goal is to learn the best arm with a fixed confidence δ using the least number of plays. The T-a-S agent explores arms and collects feedback until a certain criterion is met, and outputs a "recommended" arm such that it is the best arm *w.p.* $1-\delta$. In each round, the agent computes the "optimal proportion" of arms needed for exploration based on observed mean rewards[2], and chooses the arm which better matches ("tracks") the proportion.

Compared to the "pure exploration" approach (i.e., exploring the arms in a round-robin fashion), a T-a-S agent spends more effort on exploring arms with better reward feedback, which is significantly more efficient. Indeed, for some structured bandit environments, it has been shown that Track-and-Stop is asymptotically optimal in terms of the number of explorations needed for the fixed-confidence best arm identification problem. We present the following result (taken from [15]) which will be used in our regret analysis.

Let $\mathcal{E}_k$ be the set of k-armed Gaussian bandit environments. For any $\nu \in \mathcal{E}_k$, denote ν_i as the reward distribution of arm $i \in [k]$ (which is normally distributed) and μ_i as its mean. We denote $\mathcal{E}_{k,\mathrm{alt}(\nu)}$ as the set of bandits whose best arms are different from the one in ν, i.e., $\mathcal{E}_{k,\mathrm{alt}(\nu)} = \{\nu' \in \mathcal{E}_k : i^*(\nu) \cap i^*(\nu') = \phi\}$ where $i^*(\nu) = \arg\max_{i \in [k]} \mu_i(\nu)$.

Lemma 3.1 ([15], Theorem 33.6). *For any bandit environment $\nu \in \mathcal{E}$, the stopping time of a Track-and-Stop instance with a confidence parameter δ, $\tau(\delta; \nu)$, satisfies that*

$$\lim_{\delta \to 0} \frac{\mathbb{E}[\tau(\delta; \nu)]}{\log(1/\delta)} = \rho^*(\nu) := \sup_{\alpha \in \mathcal{P}_{k-1}} \left(\inf_{\nu' \in \mathcal{E}_{k,\mathrm{alt}(\nu)}} \left(\sum_{i=1}^{k} \alpha_i d(\nu_i, \nu_i') \right) \right)$$

where $\mathcal{P}_{k-1}$ is the $(k-1)$-probability simplex and $d(\cdot, \cdot)$ denotes the Kullback-Leibler divergence of two distributions.[3]

Note that the value $\rho^*(\nu)$ is the asymptotic lower bound.

3.2 Algorithm for Weakly Coupled Systems

In this section, we introduce the main result — a decentralized bandit algorithm for weakly coupled systems using Track-and-Stop as a building block. As we will see, our algorithm exploits three properties: (i) *(1) Local greedy property,* where there is no sample sharing across agents and decision-making is based on local majority; *(2) Avoiding hard environments,* where the T-a-S algorithm is initially deployed on a larger set of environments, but is stopped early on those environments that are hard (meaning the gap between the means of the best and second-best arms is small), and *(3) Sub-sampling property,* where only a limited set of environments are ever explored by any agent. The complete algorithm is presented in Algorithm 1, and an illustrative figure is exhibited in Figure 2.

Let $\mathsf{TAS}^{(i,\cdot)}(\delta)$ denote a sub-routine as follows: Agent 1 plays arm i repeatedly; Agent 2 implements T-a-S with respect to the confidence parameter δ based on her own feedback. The sub-routine $\mathsf{TAS}^{(\cdot,i)}(\delta)$ is defined analogously. Before implementation, let Agent 2 randomly choose a sample set of arms $\mathcal{S}^{(2)} \subset [k_2]$ such that $|\mathcal{S}^{(2)}| = s(k_2)$, where $s(k_2)$ is a global constant which is known to both users. By choosing each of the arms in $\mathcal{S}^{(2)}$, Agent 2 will generate $s(k_2)$ environments for Agent 1 where Agent 1 can learn

[1] Indeed, with slight modification our algorithm can minimize a regret that is defined with respect to $(i_1^{\mathsf{M}}, i_2^{\mathsf{M}})$.

[2] For example, if an arm shows much worse reward feedback than others after some initial exploration, the proportion assigned to this arm should be lower.

[3] A similar result on exponential family bandits is given in [8] (Theorem 14).

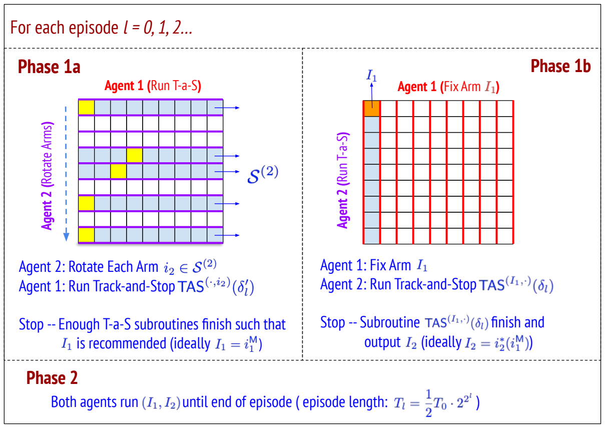

For each episode $l = 0, 1, 2...$
Phase 1a
Agent 1 (Run T-a-S)
Agent 2 (Rotate Arms)
$\mathcal{S}^{(2)}$
Agent 2: Rotate Each Arm $i_2 \in \mathcal{S}^{(2)}$
Agent 1: Run Track-and-Stop TAS$^{(\cdot,i_2)}(\delta_l')$
Stop -- Enough T-a-S subroutines finish such that I_1 is recommended (ideally $I_1 = i_1^{\mathsf{M}}$)
Phase 1b
I_1
Agent 1 (Fix Arm I_1)
Agent 2 (Run T-a-S)
Agent 1: Fix Arm I_1
Agent 2: Run Track-and-Stop TAS$^{(I_1,\cdot)}(\delta_l)$
Stop -- Subroutine TAS$^{(I_1,\cdot)}(\delta_l)$ finish and output I_2 (ideally $I_2 = i_2^*(i_1^{\mathsf{M}})$)
Phase 2
Both agents run (I_1, I_2) until end of episode (episode length: $T_l = \frac{1}{2}T_0 \cdot 2^{2^l}$)

Figure 2: An illustration of the main algorithm.

Algorithm 1 Decentralized Bandit for Weakly Coupled Systems

```
**Initialization**: Agent 2 randomly select $\mathcal{S}^{(2)} \subset [k_2]$, such that $|\mathcal{S}^{(2)}| = s(k_2)$.
for $l = 0, 1, 2, \cdots$ do
    Global clock $t \leftarrow 1$
    $T_l \leftarrow \frac{1}{2}T_0 \cdot 2^{2^l}, \delta_l \leftarrow 2\delta_0 \cdot (\frac{1}{2})^{2^l}, \delta_l' \leftarrow h(\delta_l)$
    [Phase 1a]
    [Agent 1] Set local variables: $D^{(\cdot,j)} \leftarrow \phi$ for all $j \in \mathcal{S}^{(2)}$
    [Agent 2] Set local variable: $i_2 \leftarrow$ lowest index in $\mathcal{S}^{(2)}$
    while NOT phase_1a_stop AND $t \leq T_l := \frac{1}{2}T_0 \cdot 2^{2^l}$ do
        Proceed TAS$^{(\cdot,i_2)}(\delta_l')$ by one time slot
        if TAS$^{(\cdot,i_2)}(\delta_l')$ stops (observed by Agent 1) then
            [Agent 1] $D^{(\cdot,i_2)} \leftarrow$ Output of TAS$^{(\cdot,i_2)}(\delta_l')$
            Agent 1 informs Agent 2 that $D^{(\cdot,i_2)} \neq \phi$
        end if
        [Agent 2] $i_2 \leftarrow$ the next arm (in a round-robin fashion)
                        in $\mathcal{S}^{(2)}$ where $D^{(\cdot,i_2)} = \phi$
        $t \leftarrow t + 1$
    end while
    [Agent 1] $I_1 \leftarrow$ Mode$((D^{(\cdot,i_2)})_{i_2 \in \mathcal{S}^{(2)}})$
    [Phase 1b]
    [Agent 2] Set local variable: $D^{(I_1,\cdot)}$
    while NOT phase_1b_stop AND $t \leq T_l := \frac{1}{2}T_0 \cdot 2^{2^l}$ do
        Proceed TAS$^{(I_1,\cdot)}(\delta_l)$ by one time slot
        if TAS$^{(I_1,\cdot)}(\delta_l)$ stops (Observed by Agent 2) then
            [Agent 2] $D^{(I_1,\cdot)} \leftarrow$ Output of TAS$^{(I_1,\cdot)}(\delta_l)$
            Agent 2 informs Agent 1 that $D^{(I_1,\cdot)} \neq \phi$
        end if
        $t \leftarrow t + 1$
    end while
    [Agent 2] $I_2 \leftarrow D^{(I_1,\cdot)}$
    [Phase 2]
    Agent 1 and Agent 2 choose $(I_1, I_2)$ repeatedly until $t = T_l$
**Definition (Phase Stopping Criteria)**:
phase_1a_stop = $\{\exists j \in \mathcal{S}^{(1)}$ such that $\sum_{i_2 \in \mathcal{S}^{(2)}} \mathbb{1}\{D^{(\cdot,i_2)}=j\} >$
        $(1-\gamma)\frac{|\mathcal{S}^{(2)}|}{2}\}$ or $\{D^{(\cdot,i_2)} \neq \phi, \forall i_2 \in \mathcal{S}^{(2)}\}$.
phase_1b_stop = $\{D^{(I_1,\cdot)} \neq \phi\}$.
```

its "majority best" arm i_1^{M}, i.e., which maximizes its local rewards in a majority of the environments. Sampling is important when k_2 is large — we will discuss its impact in the analysis section later.

The algorithm proceeds by episodes. Each episode l lasts $T_l := \frac{1}{2}T_0 \cdot 2^{2^l}$ rounds (arm pulls), and is split into two phases: the exploration phase which consists of phase (1a) and (1b), and the exploitation phase (phase (2)).

In phase (1a), Agent 2 selects $i_2 \in \mathcal{S}^{(2)}$ in a round-robin fashion while Agent 1 runs Track-and-Stop instances (with respect to corresponding Agent 2's arms) under a confidence parameter $\delta_l' = h(\delta_l)$ where $\delta_l := 2\delta_0 \cdot (\frac{1}{2})^{2^l}$. The definition of h will be discussed later in Lemma 3.2. Once TAS$^{(\cdot,j)}(\delta_l')$ stops (i.e., Agent 1 outputs an arm recommendation $D^{(\cdot,j)}$), Agent 1 will inform Agent 2 to skip choosing j in the following rounds. Phase (1a) stops when 1) Agent 1 learns $D^{(\cdot,j)}$ for all $j \in \mathcal{S}^{(2)}$ or when 2) more than $(1-\gamma)|\mathcal{S}^{(1)}|/2$ Track-and-Stop instances output the same (non-ϕ) arm recommendation. Note that this latter step corresponds to *avoiding hard environments* that we discussed earlier. Phase (1a) ends with Agent 1 choosing an arm I_1 which is most frequently recommended (and ideally i_1^{M}). In Phase (1b), Agent 1 chooses I_1 while Agent 2 runs subroutine TAS$^{(I_1,\cdot)}(\delta_l)$ until Agent 2 outputs a recommended arm I_2 (ideally $i_2^*(i_1^{\mathsf{M}})$).

Finally, in phase (2), Agent 1 and Agent 2 select I_1 and I_2 respectively for the remaining time slots in episode l. Note that there is possibility that phase (1a) or (1b) is not finished by the end of episode l — in this case, we start a new episode nevertheless. In practice, this scenario could be avoided by properly choosing parameters T_0 and δ_0.

Note that the constant γ is pre-selected as a hyper-parameter to reflect the degree of coupling of the system — the less coupling there is (as one assumes), the larger γ can be set, and the less exploration is needed. In an extreme case, when $\gamma = 1$ (i.e., i_1^{M} is the best with respect to any arm choice of Agent 2), only one sample is needed in $[k_2]$ for the exploration of arm I_1 in phase (1a).

Remark 2 (Communication Cost). *In this algorithm, communication occurs when one agent signals the other the end of a Track-and-Stop instance or the end of phase (1a) or (1b), and no other information requires exchange. Furthermore, in phase (1a) and (1b), the agent who implements the Track-and-Stop instance does not need to know which arm the other agent selects since the other agent chooses arms in a round-robin manner — the only knowledge needed is $s(k_2)$ for Agent 1. (In Algorithm 1, for notation simplicity we use $D^{(\cdot,j)}$ as Agent 1's local variables to denote the outputs of Track-and-Stop subroutines, although the exact indices j are not needed.)*

Remark 3 (Non-Weakly Coupled Systems). *When the system is not weakly-coupled, the recommended arms (I_1, I_2) can be suboptimal — in some settings, "greedy choices" may have a negative impact on each other. When this happens (e.g., the rewards observed in phase (2) are much smaller than the best rewards observed in phase (1)), one solution is for both agents to switch to a pre-agreed arm pair or a* *centralized bandit algorithm. For instance, an Explore-Then-Commit (ETC) approach is a reasonable centralized algorithm — all of the $k_1 \times k_2$ arm pairs are selected in a round-robin fashion for a fixed length of time, and the best arm pair (after exchanging the information regarding mean rewards) is used in the exploitation phase.*

Remark 4 (Extension to 3-Agent Systems). *This algorithm can be easily extended to systems with more than 2 agents. For example, when there are are 3 agents, phase (1) is split into 3 sub-phases: in phase (1a), Agent 2 and Agent 3 first select a subset of arms in $[k_2] \times [k_3]$ and rotate arms accordingly, while Agent 1 runs Track-and-Stop subroutines to identify the majority arm I_1 (ideally i_1^{M}); in phase (1b), Agent 1 fixes his arm choice I_1, while Agent 2 and Agent 3 follow the procedure as in the original phase (1a) of Algorithm 1 such that Agent 2 learns I_2 (ideally $i_2^{\mathsf{M}, i_1 = i_1^{\mathsf{M}}}$ as defined in Section 2.3); finally, in phase (1c), Agent 3 learns the recommended arm I_3 when Agent 1 and Agent 2 play (I_1, I_2). The exploitation phase remains the same.*

3.3 Regret Analysis

In this section, we present the regret analysis of our main algorithm. For simplicity, we assume the distribution of rewards (for each arm) observed by each agent is Gaussian, such that the theoretical guarantee of Track-and-Stop can be applied.

3.3.1 Soundness of Phase (1). The first result states the soundness of the exploration phase, i.e., the best collective arm is identified with high probability. We will focus on the soundness of phase (1a) since the result for phase (1b) is straightforward (as only one Track-and-Stop instance is involved).

Lemma 3.2. *Assume that $\sum_{j \in \mathcal{S}^{(2)}} \mathbb{1}\{i_1^{\mathsf{M}} = i_1^*(j)\} \geq \frac{s(k_2)}{2}$, i.e., the sample set $\mathcal{S}^{(2)}$ preserves the majority condition. Let $\tau_l^{(1a)}$ denote the stopping time of phase (1a) in episode l (and suppose episode l can run indefinitely). It satisfies that*

$$\mathbb{P}(\{\tau_l^{(1a)} < \infty\} \cap \{I_1 = i_1^{\mathsf{M}}\}) \geq 1 - \delta_l$$

provided that

$$h^{-1}(\delta) = 1 - \sum_{n = \lfloor (1-\gamma) s(k_1)/2 \rfloor + 1}^{\lceil s(k_2)/2 \rceil} \binom{\lceil s(k_2)/2 \rceil}{n} (1-\delta)^n \delta^{\lceil s(k_2)/2 \rceil - n}.$$

Proof. The intuition is as follows: Let Event A be "more than $(1-\gamma)s(k_2)/2$ Track-and-Stop instances will *eventually* choose Arm i_1^{M} if running indefinitely". Let Event B be "the majority-stopping criterion `phase_1a_stop` is reached, and Arm i_1^{M} is chosen when `phase_1a_stop` is reached". Clearly we have that A implies B. Thus, it suffices to compute the error probability of event A to get the error bound for Event B.

Let $D_j[t]$ be the random variable denoting the decision ("output") for instance $\mathsf{TAS}^{(\cdot,j)}(\delta_l')$ at time t. Thus, $D_j[t] \in \{\phi\} \cup [k_1]$. Let $\tilde{\mathcal{S}}^{(1)} \subset \mathcal{S}^{(1)}$ be the set of "good" arms for Agent 2 such that $i_1^{\mathsf{M}} = i_1^*(j)$ for all $j \in \tilde{\mathcal{S}}^{(2)}$ (by assumption, $|\tilde{\mathcal{S}}^{(2)}| \geq \lceil s(k_2)/2 \rceil \geq \lfloor (1-\gamma)s(k_2)/2 \rfloor + 1$). Therefore, by the soundness of Track-and-Stop, for any $j \in \tilde{\mathcal{S}}^{(2)}$, $\lim_{t\to\infty} \mathbb{1}\{D_j[t] = i_1^{\mathsf{M}}\} = Y_j$, *a.s.* where Y_j is a $\mathsf{Bern}(1 - \delta_l')$ random variable. Furthermore, since $\{D_i[t]\}_{t \geq 0}$ and $\{D_j[t]\}_{t \geq 0}$ are independent from each other for any $i \neq j$, we have that Y_i are independent for all $i \in \tilde{\mathcal{S}}^{(2)}$.

Let $N[t] = \sum_{i \in \mathcal{S}^{(2)}} \mathbb{1}\{D_i[t] = i_1^{\mathsf{M}}\}$ and $\tilde{N}[t] = \sum_{i \in \tilde{\mathcal{S}}^{(2)}} \mathbb{1}\{D_i[t] = i_1^{\mathsf{M}}\}$. Note that phase (1a) is good if and only if there exists $t > 0$ such that $N[t] > (1-\gamma)s(k_2)/2$. Now observe that,

$$\begin{aligned}\{\forall t, N[t] \leq (1-\gamma)s(k_2)/2\} &\implies \{\lim_{t\to\infty} N[t] \leq (1-\gamma)s(k_2)/2\} \\ &\implies \{\lim_{t\to\infty} \tilde{N}[t] \leq (1-\gamma)s(k_2)/2\}.\end{aligned}$$

Note that $\lim_{t\to\infty} N[t]$ and $\lim_{t\to\infty} \tilde{N}[t]$ both exist due to monotonicity and $N[t] \geq \tilde{N}[t]$ for all $t \geq 0$.

Now observe that $\lim_{t\to\infty} \tilde{N}[t] = \sum_{i \in \tilde{\mathcal{S}}^{(2)}} Y_i$. Therefore, we have that

$$\begin{aligned}\mathbb{P}(\{\tau_l^{(1a)} = \infty\} \cup \{I_1 \neq i_1^{\mathsf{M}}\}) &\leq \mathbb{P}(\lim_{t\to\infty} \tilde{N}[t] \leq (1-\gamma)s(k_2)/2) \\ &\leq 1 - \sum_{n=\lfloor (1-\gamma)s(k_2)/2 \rfloor + 1}^{|\tilde{\mathcal{S}}^{(2)}|} \binom{|\tilde{\mathcal{S}}^{(2)}|}{n} (1-\delta_l')^n \delta_l'^{|\tilde{\mathcal{S}}^{(2)}| - n} \\ &\leq h^{-1}(\delta_l') = \delta_l\end{aligned}$$

The last inequality holds true for any possible $\tilde{\mathcal{S}}^{(2)}$, considering $|\tilde{\mathcal{S}}^{(2)}| \geq s(k_2)/2$ by assumption. □

3.3.2 Regret. To compute the cumulative regret, let us first give a bound on the expected length of the exploration phase. Note that for any fixed arm $j \in \mathcal{S}^{(2)}$, Agent 1 operates a k_1-armed bandit in instance $\mathsf{TAS}^{(\cdot,j)}(\delta_l')$. Let $\nu^{(\cdot,j)} \in \mathcal{E}_{k_1}$ denote the corresponding environment in which Agent 1 plays. We have the following result.

Lemma 3.3. *Rank the elements in $\mathcal{S}^{(2)}$ as $(j_1, j_2, \cdots, j_{|\mathcal{S}^{(2)}|})$, such that $\rho^*(\nu^{(\cdot,j_1)}) \leq \rho^*(\nu^{(\cdot,j_2)}) \leq \cdots \leq \rho^*(\nu^{(\cdot,j_{|\mathcal{S}^{(2)}|})})$. For any $\epsilon > 0$, there exists δ_0 such that for all $l \geq 0$,*

$$\mathbb{E}[\tau_l^{(1a)}] \leq (1+\epsilon)(\log \frac{1}{\delta_0} + 2^l \log 2)\mathrm{C}^{(1a)}$$

where

$$\mathrm{C}^{(1a)} = \left(\sum_{m=1}^{\tilde{s}} \rho^*(\nu^{(\cdot,j_m)}) + (s(k_2) - \tilde{s}) \cdot \rho^*(\nu^{(\cdot,j_{\tilde{s}})}) \right)$$

and $\tilde{s} = \lfloor (1-\gamma) \frac{s(k_2)}{2} \rfloor + 1$.

Remind that $\delta_l = 2\delta_0 \cdot (\frac{1}{2})^{2^l}$. The above is a direct result from Lemma 3.1 and the phase stopping criterion. Analogously, we can derive a similar (and simpler) result for phase (1b) using another constant $\mathrm{C}^{(1b)}$, which can be defined as $\mathrm{C}^{(1b)} = \max_{j \in [k_1]} \rho^*(\nu^{(j,\cdot)})$.

We have the following regret bound for the main algorithm.

Theorem 3.4. *Provided that $\sum_{j \in \mathcal{S}^{(2)}} \mathbb{1}\{i_1^{\mathsf{M}} = i_1^*(j)\} \geq s(k_2)/2$, we have that for any $\epsilon > 0$, there exists δ_0 such that*

$$\begin{aligned}\mathrm{Regret}_T \leq 4(1+\epsilon)\Delta_{\max}(\mathrm{C}^{(1a)} + \mathrm{C}^{(1b)}) \cdot \max(\log((T/T_0) \cdot 2), 1) \\ + o(\log T)\end{aligned}$$

where $\Delta_{\max} = \max_{(i_1,i_2)} \left((\mu^{(1)}_{i_1^*, i_2^*} + \mu^{(2)}_{i_1^*, i_2^*}) - (\mu^{(1)}_{i_1,i_2} + \mu^{(2)}_{i_1,i_2}) \right)$, and the constants $\mathrm{C}^{(1a)}$ and $\mathrm{C}^{(1b)}$ are defined as in Lemma 3.3.

Proof. Let $l(T)$ be the index of the episode at the horizon T. By observation we have that

$$l(T) = 0 \text{ or } (T_0/2) \cdot 2^{2^{l(T)-1}} \leq T$$

$$\implies l(T) \le \max(0, \log_2 \log_2((T/T_0)\cdot 2)) + 1.$$

The regret can be split into two parts: the loss of rewards due to the exploration in phase (1a) and (1b), and the loss of rewards due to the "wrong" recommendation in phase (2). Therefore, using the bound on $l(T)$, we have that

$$\begin{aligned}
\text{Regret}_T &\le \sum_{l=0}^{l(T)} (\mathbb{E}[\tau_l^{(1a)} + \tau_l^{(1b)}]\Delta_{\max} + (2\delta_l)\cdot T_l) \\
&\le \sum_{l=0}^{l(T)} \left((1+\epsilon)(\mathrm{C}^{(1a)} + \mathrm{C}^{(1b)})\Delta_{\max}(\log\frac{1}{\delta_0} + 2^l \log 2) + 2\delta_0 T_0 \right) \\
&\le 4(1+\epsilon)\Delta_{\max}(\mathrm{C}^{(1a)} + \mathrm{C}^{(1b)}) \cdot \max(\log(\frac{2T}{T_0}), 1) + o(\log T).
\end{aligned}$$

Note that the first inequality applies the soundness guarantee given in Lemma 3.2. □

3.3.3 Sampling. Note that the constant $\mathrm{C}^{(1a)}$ in the above theorem scales as $O(s(k_2))$. Thus, when $\mathcal{S}^{(2)} = [k_2]$ (no sampling), we have that $\text{Regret}_T = O(k_1 k_2 \log T)$. When the number of arms is large, sampling is typically needed to reduce the computational cost. Using standard concentration techniques (Bernstein inequality) which bound the probability of the event $\{\sum_{j\in\mathcal{S}^{(2)}} \mathbb{1}\{i_1^{\mathrm{M}} = i_1^*(j)\} \ge s(k_2)/2\}$, we have the following corollary.

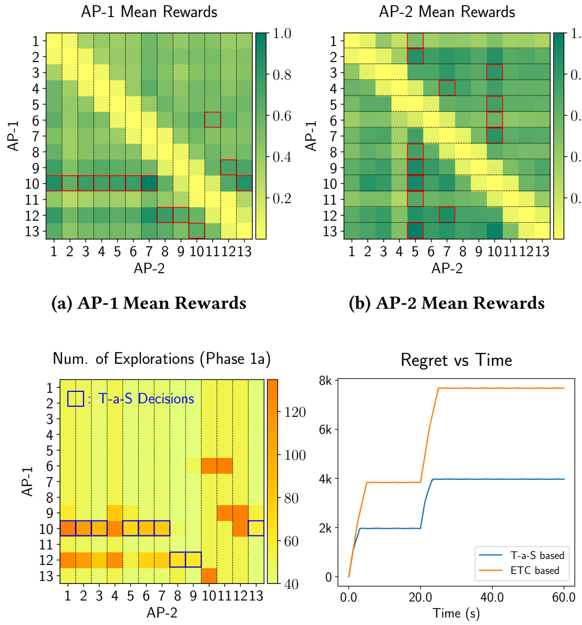
Corollary 3.5. *When Condition 2 holds, there exist $\beta_2 > 0$ such that when the sample size $s(k_2) = \beta_2 \log k_2$, the regret satisfies that $\text{Regret}_T = O(k_1 \log k_2 \log T)$ with probability $1/k_2$ for sufficiently large k_2.*

Corollary 3.5 can be extended to systems with more than two agents. Suppose there are m agents, each with k arms. Using the procedure discussed in Remark 4, and letting the size of sample sets in phase (1) scaling as $O(\log k)$, we have that the regret scales as $O((m-1)k \log k \log T)$ *w.p.* $O(1/k)$. Note that the term $(m-1)$ stems from the number of sub-phases needed in phase (1).

4 PERFORMANCE EVALUATION

4.1 Multi-Channel Selection

4.1.1 A Two-AP Example. We first consider an application example in the wireless channel selection problem. Suppose two access points (APs) are located within a close range, each serving a nearby mobile user. The APs decide amongst a set of channels (frequency bands) which one to use to serve the respective users. Due to some environment effects (such as shadowing or exogenous interference), the best channels are unknown to the APs. Furthermore, the decisions of each AP will interfere with the rewards received by the other due to *channel leakage*. However, since the leakage mainly affects a small number of adjacent channels, it is reasonable to believe that the majority condition holds, and one can apply our algorithm to locate the best channel at each AP.



(a) AP-1 Mean Rewards

(b) AP-2 Mean Rewards

(c) Number of Explorations for Each Arm Pair in Algorithm Phase 1a

(d) Regret vs Time

Figure 3: Simulation results on experiments in Section 4.1.1.

Experiment Settings: Suppose each AP chooses among (the same) n frequency bands which are indexed $1, 2, \cdots, n$ for each time slot (which lasts 0.5 ms). We set $n = 13$. The Signal-to-Interference ratio (SIR) at time slot t is modeled as $\mathtt{SIR}[t] = P_a g[t]/\mathtt{I}[t]$ where P_a is the transmit power of the AP, $g[t]$ denotes the channel gain at the user and $\mathtt{I}[t]$ denotes the interference level. We set $P_a = 23$ dBm for both APs. The channel gain is determined through the path loss, fading properties (Rayleigh fast fading) and a channel-dependent shadowing gain. We assume the channel-dependent gains (in dB) are drawn from Gaussian distribution $\mathcal{N}(0, 6)$ and constant through the simulation.

Assume that the interference $\mathtt{I}[t]$ is exclusively caused by channel leakage from the nearby AP. We adopt the following simplified power leakage model for AP s ($s = 1, 2$): when channel $i \in [n]$ is chosen, the relative power leakage (in dB) in channel $j \in [n]$ equals $\min(0, \max(-\beta_s|i-j|, -\varphi_s) + \gamma_{i,j})$. This reflects the nature of common channel leakage, i.e., adjacent channels experience significantly higher interference (subject to channel-dependent "noise" $\gamma_{i,j}$). In our simulation, we set $\beta_s = 33$ dB, $\varphi_s = 90$ dB and $\gamma_{i,j}$ is randomly chosen from Gaussian distribution $\mathcal{N}(0, 4)$. Note that we do not assume the agents (APs) have any prior knowledge of the leakage model, and our algorithm can easily address more complicated models (e.g., with APs using different sets of channels or abnormal non-adjacent channel leakage). Furthermore, we assume each mobile user is closer to its corresponding AP, and the relative gain due to path loss is 20 dB for both users.

Suppose the users have infinitely backlogged queues and the rewards received by each user equal the number of packets transmitted, i.e., the instantaneous service rate. For any time slot t, the rate at any user i is given by

$$S_i[t] = \mathtt{BW} \times \log_2(1 + 10^{0.1(\mathtt{SIR}_i[t] - \mathtt{L})}) \quad \text{bps} \qquad (1)$$

where $\mathtt{BW}$ is set to be 20 MHz and the parameter $\mathtt{L} = 3$dB describes a loss to Shannon capacity.

Results: We first run Monte-Carlo simulations to compute mean rewards received by both APs under different (i_1, i_2) pairs. Figure 3a and 3b show a typical realization of channel rewards (note that the results vary by simulations due to randomness), where the adjacent channel leakage negatively affects the rewards in the diagonal squares. In Figure 3a, we use red boxes to denote the best AP-1 arm under each AP-2 arm choice i_2, and AP-1 arm 10 is the majority arm i_1^{M} (under this simulation). Similarly, AP-2 arm 5 is shown as the majority arm i_2^{M} by Figure 3b (which is also $i_2^*(i_1^{\mathsf{M}})$).

We then run the main algorithm to find the best collective arm. For simplicity, we apply a computationally-efficient version of Track-and-Stop, assuming the rewards are normally distributed (see [8], Section 2). In particular, we are interested in the TAS subroutines in phase (1a) of the algorithm. Figure 3c shows the number of explorations for each arm pair in phase (1a) in the first episode. As expected, for each sub-routine TAS$^{(\cdot,i_2)}(\delta'_l)$ (corresponding to each column), typically only the top two arms are heavily explored to determine the best one — this shows a major advantage of Track-and-Stop compared to a naive round-robin exploration. In addition, we use blue boxes to denote decisions made by each sub-routine. The larger is the reward gap between the top two arms, the less exploration is required (e.g., when $i_2 = 13$). Note that phase (1a) stops once the stopping criterion has been met, and not all of the sub-routines are needed to output a recommendation.

Finally, in Figure 3d, we exhibit the cumulative regret over time, which grows logarithmically with an episodic behavior. We compare our algorithm to the classical *explore-and-commit (ETC)* algorithm, which utilizes round-robin exploration. Our algorithm exhibits a much-improved regret in the exploration phase, suggesting that the majority-based algorithm with Track-and-Stop subroutines better exploits the structure of the system.

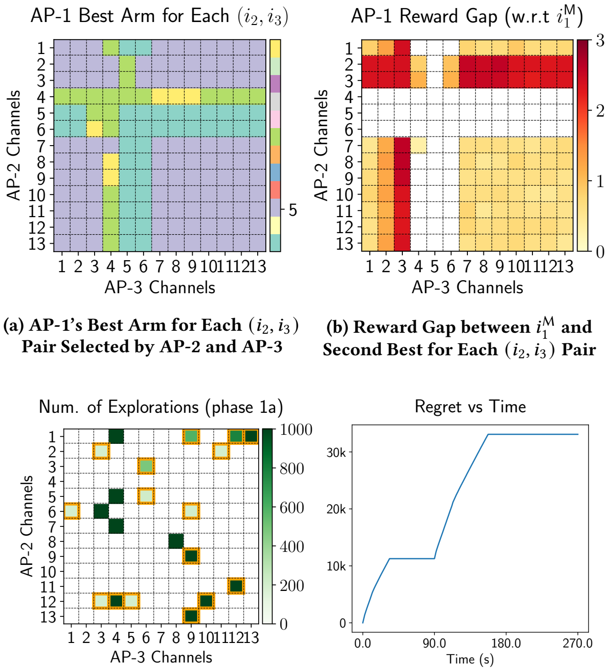
4.1.2 A Three-AP Example. In this experiment, we extend our multi-channel selection example to a 3-AP setting. We follow the channel leakage model introduced in the previous section, and set $\beta_s = 33, 39, 45$ dB for $s = 1, 2, 3$ respectively. The relative path loss gain ranges from 20 to 40 dB among different pairs of users. Other parameters remain the same.

Figure 4a shows the best arm for AP-1 when each (i_2, i_3) pair is selected (for one realization of the channel model). For the simulation we present here, the majority arm $i_1^{\mathsf{M}} = 5$. Figure 4b further exhibits the reward gap between the majority arm and the second-best in each environment (we set the gap as 0 when the majority arm 5 is not the best arm in that environment).

When implementing phase (1a) of the algorithm, we sample 20 (i_2, i_3) pairs (otherwise, the number of sub-routines needed significantly grows with more APs). The total number of explorations of each sampled sub-routine in the first episode is shown in Figure 4c, with orange boxes denoting the sub-routines that output arm recommendations before phase (1a) finishes — as expected, "easy" sub-routines with larger reward gap complete faster. A regret plot is presented in Figure 4d for completeness. The regret is computed with respect to the best collective action $(5, 11, 2)$.

4.2 Best Scheduler Selection

In this section, we explore a potential application of our algorithm for best scheduler selection in wireless queueing systems, which is first proposed as the "meta-scheduling" problem in [22, 23]. Wireless scheduling with queues is a challenging task — many schedulers are developed (e.g., MaxWeight, Log Rule, Exp Rule, etc.) for specific settings/goals, however, there lacks a systematic approach to find a good scheduler across diverse performance metrics and deployment scenarios. The authors in [22] thereby proposed a multi-armed bandit framework ("meta-scheduler"), which selects the best scheduler from a set of predefined policies through users' feedback evaluating the performance. This is a flexible model which allows complicated and user-customized reward schemes to be considered.



(a) AP-1's Best Arm for Each (i_2, i_3) Pair Selected by AP-2 and AP-3

(b) Reward Gap between i_1^{M} and Second Best for Each (i_2, i_3) Pair

(c) Phase (1a) Exploration Heatmap

(d) Regret vs Time

Figure 4: Simulation results on experiments in Section 4.1.2.

The algorithm proposed in [22] is designed for single-agent scenarios. When there are multiple nearby base stations, reward feedback at each agent is coupled with decisions from other agents due to signal interference. Furthermore, different scheduler combinations might lead to heterogeneous interfering behaviors. Therefore, it can be problematic to run a single-agent bandit algorithm individually at each station without effective coordination.

Under reasonable interference levels and typical reward schemes, we believe the systems are weakly coupled. The intuition is that effective scheduling policies tend to schedule users opportunistically (i.e., making use of good channels to improve transmission efficiency), and as a byproduct incur less interference to other agents (since less power/time is needed to transmit the same users' packet flows). Therefore, the majority condition should hold if the candidate set consists of mostly "good" schedulers. In the following, we will set up a simple downlink scheduling system and showcase the usage of our algorithm.

Experiment Settings: Suppose there exists 2 base stations (BS-1 and BS-2), each serving 4 downlink users. For each base station, we follow a packet transmission model used in [22]: The instantaneous

Table 1: Mean rewards observed by BS-1 and BS-2 in Scenario (S1) of Section 4.2. The best policy in each environment is highlighted in bold font. The best collective arm is (C, C).

			BS-1 Arms (Policies)					
			A)	B)	C)	D)	E)	F)
BS-1 Mean Rewards	BS-2	A)	0.773	0.549	**0.785**	0.311	0.768	0.760
		B)	0.769	0.508	**0.781**	0.264	0.764	0.755
		C)	0.776	0.571	**0.787**	0.337	0.771	0.763
		D)	0.745	0.046	**0.757**	0.019	0.739	0.724
		E)	0.773	0.541	**0.784**	0.303	0.768	0.759
		F)	0.749	0.117	**0.761**	0.027	0.744	0.730
BS-2 Mean Rewards			BS-1 Arms (Policies)					
			A)	B)	C)	D)	E)	F)
	BS-2	A)	0.945	0.945	0.944	0.945	0.941	0.943
		B)	0.937	0.936	0.933	0.932	0.932	0.930
		C)	**0.948**	**0.952**	**0.953**	**0.952**	**0.952**	**0.951**
		D)	0.632	0.682	0.633	0.636	0.640	0.634
		E)	0.940	0.936	0.941	0.940	0.938	0.941
		F)	0.878	0.861	0.871	0.872	0.862	0.875

Table 2: Mean rewards observed by BS-1 and BS-2 in Scenario (S2) of Section 3.2. The best policy in each environment is highlighted in bold font. The best collective arm is (C, C).

			BS-1 Arms (Policies)					
			A)	B)	C)	D)	E)	F)
BS-1 Mean Rewards	BS-2	A)	0.569	0.183	**0.612**	0.051	0.596	0.505
		B)	0.544	0.125	**0.587**	0.030	0.575	0.474
		C)	0.588	0.221	**0.630**	0.074	0.611	0.530
		D)	0.265	0.010	0.267	0.009	**0.372**	0.050
		E)	0.560	0.168	**0.604**	0.046	0.588	0.493
		F)	0.339	0.010	0.376	0.009	**0.417**	0.158
BS-2 Mean Rewards			BS-1 Arms (Policies)					
			A)	B)	C)	D)	E)	F)
	BS-2	A)	0.936	0.941	0.945	0.942	0.942	0.939
		B)	0.934	0.933	0.936	0.932	0.932	0.931
		C)	**0.955**	**0.953**	**0.951**	**0.951**	**0.952**	**0.952**
		D)	0.709	0.737	0.700	0.696	0.737	0.671
		E)	0.932	0.930	0.938	0.932	0.939	0.932
		F)	0.873	0.878	0.876	0.866	0.874	0.871

SINR of user i at time t equals $P_b g_i[t]/(\sigma^2 + I_i[t])$, where the transmit power of BS P_b is set to be 47 dBm and the noise level $\sigma^2 = -104$. The channel gain $g_i[t]$ is a combination of path loss and Rayleigh fast fading, and the path loss (in dB) is computed as $39.1 \log_{10}(\texttt{dist}) + 13.5 + 20 \log_{10}(f_c)$ where $f_c = 2.0$ GHz and $\texttt{dist}$ denotes the user distance. The interference level $I_i[t]$ is a result of packet transmission of the nearby base station, and $I_i[t] = 0$ when the other base station is idle. The instantaneous service rate of each user is computed according to (1) with $\texttt{BW} = 10$ MHz. Each time slot lasts 0.5 ms and each packet has a fixed size 5 kb.

Let the 4 users served by BS-2 close to their base station (subject to small interference) with a light load — the arrival rate for each user is set as 0.3 packets/slot. For the 4 users served by BS-1, we set the arrival rate as 0.6 packets/slot and focus on two scenarios: (S1) For each user, the distance to BS-1 ($\texttt{dist}_1$) equals 150 m and the distance to BS-1 ($\texttt{dist}_2$) ranges from 300 ± 10 m. (S2) For each user, $\texttt{dist}_1 = 150$ m and $\texttt{dist}_2 \in 250 \pm 10$ m. The second scenario sees a higher interference level. Each agent chooses over 6 scheduling policies: A) MaxWeight, B) Max-Queue, C) Max-Rate, D) Round-Robin, E) Log-Rule, F) Exp-Rule, and collect reward feedback every 200 time slots (aka one "round"). Packets not transmitted at the end of each round are dropped to ensure the reward feedback are conditionally independent.[4] We define the reward of each packet as $1 - \tanh(0.04 * \texttt{delay})$ and the reward feedback of one round is the sum of all packet rewards.

Results: We first compute the mean rewards observed by both base stations under different policy pairs using Monte-Carlo simulations, which is presented in Table 1 and 2 (the rewards are normalized by the episode length and packet loads). In both Scenario (S1) and Scenario (S2), the best arm for BS-2 is Max-Rate and the mean rewards do not vary much when BS-1 changes policies due to the low load and negligible interference.

Now let us focus on the rewards observed by BS-1. In Scenario (S1), it turns out Max-Rate is the best arm of BS-1 no matter what policy BS-2 selects (due to the relatively low interference level compared to Scenario (S2)). This can be expected since in our simulation settings, all the users are almost symmetric (in terms of load and service rates) — the Max-Rate policy, which greedily serves the user with the best service rate, is proved to be efficient in minimizing packet delays for symmetric moderate-load scenarios. By contrast, in Scenario (S2), as the interference level increases (with service rates degrading), for some choices of BS-2 (Round-Robin and Exp-Rule), Max-Rate performs badly — instead, the Log-Rule policy which has a better queue-stabilizing property prevails in these cases.[5] However, the majority condition still holds for Scenario (S2), showing the robustness of our model, and our algorithm can indeed be applied to find the best collective policy.

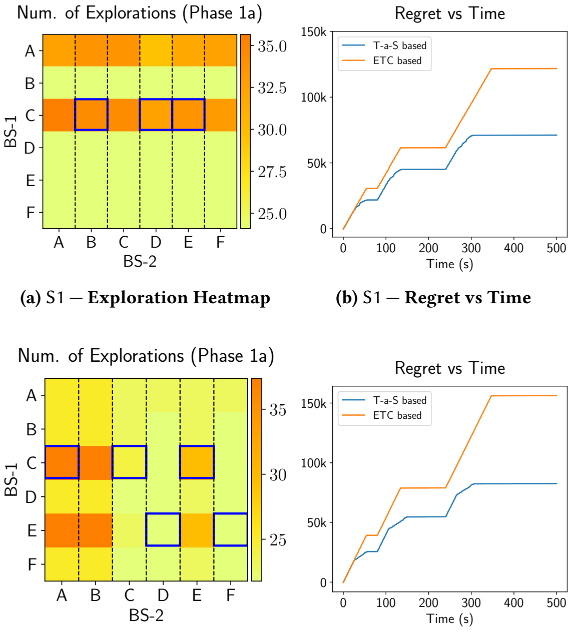
Finally, we run the main algorithm for both scenarios. The simulation results are exhibited in Figure 5. For each scenario, we show the exploration heatmap for the algorithm phase (1a) in the first episode — the best policy is identified with most of the explorations focusing on good performing policies. Moreover, not all Track-and-Stop sub-routines are needed to complete, and we use blue boxes to denote finished sub-routines. As a result, our algorithm has a lower regret than ETC as shown in Figure 5b and Figure 5d.

5 CONCLUSION

We study an online learning framework for the multi-agent resource allocation problem. In particular, we focus on so-called weakly coupled systems with a special arm-reward structure — the majority condition, which states that most of the time the best arm of each

[4] Note that if there is no packet drop, then a "bad" non-stable policy resulting in long queues will skew the reward feedback for the next round, even if a "good" queue-stabilizing policy is chosen. Ideally, only good policies are selected after some initial exploration, and thus the impact of packet drop is minimal. A detailed discussion on this issue is given in [22] , which introduces a queueing cycle-based algorithm to avoid packet drop; adapting it to our multi-agent setting is of future interest.

[5] To be precise, the Max-Rate policy, unlike Log-Rule or MaxWeight, is not *throughput-optimal* and has a smaller capacity region. In this setting, when BS-2 chooses Round-Robin or Exp-Rule which turns out incurring more interference, Max-Rate no longer stabilizes the load and tends to result in long queues, thus worsening the delay metric.



(a) S1 — Exploration Heatmap **(b) S1 — Regret vs Time**

(c) S2 — Exploration Heatmap **(d) S2 — Regret vs Time**

Figure 5: Simulation results on experiments in Section 4.2.

agent is invariant to other agents' arm selection. When this condition holds, the optimal arms can be learned with local signals (reward feedback) with proper coordination from other agents, therefore allowing design of less demanding algorithms compared to classical methods which simply examine all the collective actions. Furthermore, we develop an efficient decentralized bandit algorithm with minimal communication overheads. Through simulation, we validate the usefulness of our model and algorithm in two wireless settings: channel selection among nearby APs, and best scheduling policy selection by interfering base stations. We believe weak coupling is a reasonable abstraction for several wireless applications, and it is of great interest to explore its benefits in other related settings.

ACKNOWLEDGMENT

This work was supported by NSF grants CNS-1910112, CNS-2107037 and IIS-2112471, Army Futures Command Grant W911NF-19-2-0333, and the Wireless Networking and Communications Group Industrial Affiliates Program.

REFERENCES

[1] Ibrahim Althamary, Chih-Wei Huang, and Phone Lin. 2019. A survey on multi-agent reinforcement learning methods for vehicular networks. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 1154–1159.

[2] Orly Avner and Shie Mannor. 2014. Concurrent bandits and cognitive radio networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 66–81.

[3] Lilian Besson and Emilie Kaufmann. 2018. Multi-player bandits revisited. In *Algorithmic Learning Theory*. PMLR, 56–92.

[4] Ilai Bistritz and Amir Leshem. 2021. Game of thrones: Fully distributed learning for multiplayer bandits. *Mathematics of Operations Research* 46, 1 (2021), 159–178.

[5] Roberto Casado-Vara, Francisco Prieto-Castrillo, and Juan M Corchado. 2018. A game theory approach for cooperative control to improve data quality and false data detection in WSN. *International Journal of Robust and Nonlinear Control* 28, 16 (2018), 5087–5102.

[6] Mithun Chakraborty, Kai Yee Phoebe Chua, Sanmay Das, and Brendan Juba. 2017. Coordinated Versus Decentralized Exploration In Multi-Agent Multi-Armed Bandits.. In *IJCAI*. 164–170.

[7] Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. 2020. The gossiping insert-eliminate algorithm for multi-agent bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3471–3481.

[8] Aurélien Garivier and Emilie Kaufmann. 2016. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory*. PMLR, 998–1027.

[9] Wen Zhong Guo, Jia Ye Chen, Guo Long Chen, and Hai Feng Zheng. 2015. Trust dynamic task allocation algorithm with Nash equilibrium for heterogeneous wireless sensor network. *Security and Communication Networks* 8, 10 (2015), 1865–1877.

[10] Eun Jin Hong, Se Young Yun, and Dong-Ho Cho. 2009. Decentralized power control scheme in femtocell networks: A game theoretic approach. In *2009 IEEE 20th international symposium on personal, indoor and mobile radio communications*. IEEE, 415–419.

[11] Jianwei Huang, Randall A Berry, and Michael L Honig. 2005. A game theoretic analysis of distributed power control for spread spectrum ad hoc networks. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.* IEEE, 685–689.

[12] Ravi Kumar Kolla, Krishna Jagannathan, and Aditya Gopalan. 2018. Collaborative learning of stochastic bandits over a social network. *IEEE/ACM Transactions on Networking* 26, 4 (2018), 1782–1795.

[13] Nathan Korda, Balazs Szorenyi, and Shuai Li. 2016. Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*. PMLR, 1301–1309.

[14] Peter Landgren, Vaibhav Srivastava, and Naomi Ehrich Leonard. 2016. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 167–172.

[15] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[16] Akshayaa Magesh and Venugopal V Veeravalli. 2019. Multi-user MABs with user dependent rewards for uncoordinated spectrum access. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 969–972.

[17] Akshayaa Magesh and Venugopal V Veeravalli. 2021. Decentralized Heterogeneous Multi-Player Multi-Armed Bandits with Non-Zero Rewards on Collisions. *IEEE Transactions on Information Theory* (2021).

[18] Abbas Mehrabian, Etienne Boursier, Emilie Kaufmann, and Vianney Perchet. 2020. A practical algorithm for multiplayer bandits when arm means vary among players. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1211–1221.

[19] José Moura and David Hutchison. 2018. Game theory for multi-access edge computing: Survey, use cases, and future trends. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 260–288.

[20] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. 2016. Multi-player bandits–a musical chairs approach. In *International Conference on Machine Learning*. PMLR, 155–163.

[21] Hai-Yan Shi, Wan-Liang Wang, Ngai-Ming Kwok, and Sheng-Yong Chen. 2012. Game theory for wireless sensor networks: a survey. *Sensors* 12, 7 (2012), 9055–9097.

[22] Jianhan Song, Gustavo de Veciana, and Sanjay Shakkottai. 2021. Meta-scheduling for the wireless downlink through learning with bandit feedback. *IEEE/ACM Transactions on Networking* (2021).

[23] Jianhan Song, Gustavo de Veciana, and Sanjay Shakkottai. 2021. Online learning for hierarchical scheduling to support network slicing in cellular networks. *Performance Evaluation* 152 (2021), 102237.

[24] Chao Tao, Qin Zhang, and Yuan Zhou. 2019. Collaborative learning with limited interaction: Tight bounds for distributed exploration in multi-armed bandits. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 126–146.

[25] Beibei Wang, Yongle Wu, and KJ Ray Liu. 2010. Game theory for cognitive radio networks: An overview. *Computer networks* 54, 14 (2010), 2537–2561.

[26] Wenbo Wang, Andres Kwasinski, Dusit Niyato, and Zhu Han. 2016. A survey on applications of model-free strategy learning in cognitive wireless networks. *IEEE Communications Surveys & Tutorials* 18, 3 (2016), 1717–1757.

[27] Zhiyang Wang, Mark Eisen, and Alejandro Ribeiro. 2020. Decentralized Wireless Resource Allocation with Graph Neural Networks. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 299–303.

[28] Francesc Wilhelmi, Boris Bellalta, Cristina Cano, and Anders Jonsson. 2017. Implications of decentralized Q-learning resource allocation in wireless networks. In *2017 ieee 28th annual international symposium on personal, indoor, and mobile radio communications (pimrc)*. IEEE, 1–5.