Distributed Multi-agent Multi-armed Bandits

Peter Chal Landgren

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Mechanical and Aerospace Engineering
Adviser: Professor Naomi Ehrich Leonard

January 2019

Abstract

Social decision-making is a common feature of both natural and artificial systems. Humans, animals, and machines routinely communicate and observe each other to improve their understanding of a complex world. Additionally, many real-world tasks involve sequential decision-making under uncertainty. Such tasks are inherently subject to the explore-exploit tradeoff, where one must select between options with the highest expected payoffs based on current knowledge (exploitation) and options with less well-known but potentially better outcomes (exploration).

In this thesis, we consider distributed social decision-making under uncertainty. Specifically, we develop and utilize the multi-agent multi-armed bandit (MAB) problem to model and study how multiple interacting agents make decisions that balance the explore-exploit tradeoff. we consider several different communication protocols for sharing information between agents. We develop and analyze algorithms that address the multi-agent MAB problem under each protocol. We derive bounds on performance and use the bounds to analyze the influence of network structure, i.e., who is communicating with whom, on decision-making outcomes.

We first consider communication through consensus, and derive novel results concerning the performance of cooperative estimation of expected reward. We then use these results to develop, analyze, and prove performance bounds for several algorithms that address the multi-agent MAB problem both with and without constraints on concurrent sampling of arms by multiple agents. Furthermore, we develop a new graph centrality measure, which we call "explore-exploit" centrality, that can be used to predict performance of networked agents in an MAB problem with communication through consensus. We demonstrate the utility of this centrality measure, and the performance of the algorithms through numerical simulations and robotic experiments.

Next, we consider the multi-agent MAB problem with strictly local communication, and develop a novel partition-based algorithm that uses imitation to improve performance. We analyze this algorithm through performance bounds and simulation results.

Finally, we consider application to robotic search for radioactive material in a facility. The search for radioactive material is an inherently noisy process, and can be modeled as an MAB problem. We develop and test a MAB-based algorithmic solution and demonstrate that it enables a robot to find multiple radioactive sources efficiently.

Acknowledgements

First of all, I would like to thank and acknowledge my advisor, Naomi Leonard. It has been a privilege to work with someone who has the rare talent of being very encouraging and positive while also pushing students to do high quality work. Her enthusiasm for research is infectious, and her guidance and positive encouragement have made me a better researcher in many ways. I am also very grateful for the academic freedom Naomi has given me, and for letting me take on many tasks that helped me grow even though they did not contribute directly to this thesis, as well as allowing me to work with in the Princeton StudioLab.

I am also grateful for the opportunity to work with several outstanding postdoctoral researchers during my time at Princeton. Foremost among these is Vaibhav Srivastava, an exemplary collaborator and mentor, who patiently showed me the basics of the multi-armed bandit problem – and whose near-encyclopedic knowledge of references and mathematical techniques got me out of a pinch more times than I can count. I would also like to thank Biswa Dey and Kayhan Ozcimder for their guidance, mentorship, and friendship.

My fellow lab members have provided mentorship, friendship, and collaboration over the years, for which I am deeply grateful. In particular I would like to thank Paul Reverdy for his assistance in the Tank Lab at Forrestal, as well as Katie Fitch, Will Scott, Renato Pagliara, Bec Gray, Liz Davison, Desmond Zhong, Anthony Savas, Anastasia Bizyaeva, Udari Madhushani, and Cosmo.

My work on radiation detection would not have been possible without the leadership of Rob Goldston, Alex Glaser, and Moritz Kütt. I am very thankful they involved me on the project; and I enjoyed the opportunity to work across multiple disciplines. In particular I'd like to thank Moritz for his help getting the robot up and running and for being a great collaborator.

Many thanks to Luigi Martinelli and Simon Levin for serving on my Ph.D. committee and also for serving as readers of this thesis. Their diverse perspectives and valuable feedback have been an invaluable resource during the development of my research. Additionally, I would like to thank Clancy Rowley and Anirudha Majumdar for serving as Final Public Oral examiners.

There have also been several academic opportunities that, while tangential to my research, have been important to my development as a researcher. These include my work at the Princeton StudioLab as a teaching assistant and machine tool instructor. I'd like to thank Laura Sarubbi, Aatish Bhatia, Sharon De La Cruz, and Mike Galvin. Additionally, I thank Glenn Northey and Al Gaillard from the MAE machine shop for my time as a teaching assistant under them.

While at Princeton I had the opportunity to mentor many undergraduate students as part of summer internships, senior theses, and independent projects. It was wonderful to collaborate with so many talented undergraduate researchers on a wide variety of projects. Their curiosity and drive have helped me refine my own work.

One could not ask for a better cohort of fellow Ph.D. students than mine. Our conversations at the atrium table and trash can fires at Lawrence, and our time in MAE 501/502 will be fondly remembered. I imagine that interacting with this remarkable group of people will be the aspect of graduate school that I will miss the most. In particular, my friends Chuck Witt, Jon MacArt, and Chris Reuter have been a constant help and support (not to mention entertainment). Beyond my cohort many others have impacted my time here at Princeton, particularly those on the MAE softball team and the climbing group at Rockville.

I have also been blessed with a fantastic church community here in Princeton through Stone Hill Church. This family has been instrumental to my time here at Princeton, and I am so thankful for their support. It has been wonderful to be a part of a vibrant and growing church community during my time here and to be

involved with the associated small groups and Bible studies. In particular, I'd like to acknowledge David Keddie, Jamie Rankin, Logan Matthews, and Thomas Dixon for their friendship, scones, and whiskey. I will miss this church community dearly.

Outside of Princeton there are many colleagues and mentors who have been instrumental in my academic and personal journey. John Larkin, my undergraduate advisor at Whitworth University, was instrumental in preparing me to conduct research. Kamesh Sankaran, also a professor at Whitworth, is a major reason I came to Princeton in the first place, and his encouragement and advice were, and still are, deeply appreciated.

Finally, I would not be submitting this thesis without the unwavering support of my wife, Kaylee, and my family. Mom and Dad, thank you for your unremitting support of my education and for the many, many opportunities you have provided for me. Kristin and Will (and my niece Naomi), you have been a constant source of advice and encouragement. Kristin, thank you for being the best big sister and academic role model a little brother could ask for. Kaylee, thank you for your loving support. Your love, backing, and constant encouragement are evident on every page of this work.

My work at Princeton was partially funded by the National Defense Science and Engineering Graduate Fellowship and I thank them for their generous support.

Last but not least I would like to thank God for providing me with such amazing educational opportunities and supportive collaborators and friends. I feel beyond grateful for the abundant blessings He has provided along this journey and for His continual guidance in my life.

This dissertation carries T#3366 in the records of the Department of Mechanical and Aerospace Engineering.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Explore-Exploit Tradeoff

A persistent feature of real-world decision-making processes is the presence of uncertainty in the outcome of a future decision. Uncertainty in the result of a given action makes it difficult for a decision-maker to choose the best next action. Furthermore, real-world tasks often involve making many sequential decisions with the goal of maximizing cumulative outcomes.

Successful completion of such a task entails a fundamental tension: A decision-maker must continually choose between exploiting all options already known to be good, and exploring options not yet fully understood but potentially even better. This tension is known as the explore-exploit trade-off, and it lies at the heart of optimizing the decision-making process.

The explore-exploit tradeoff can be seen in many different types of systems, both natural and artificial. In the natural world, foraging animals seek to consume as much food as possible, while concurrently finding the most rewarding foraging areas [49, 51]. A similar paradigm arises in predator search behavior [43], where predators

must decide where to hunt on the basis of noisy information regarding prey location. Human search behavior has been shown to face similar challenges [20, 84].

In artificial systems or models the explore-exploit tradeoff arises quite frequently as well, as does decision-making under uncertainty. Reinforcement learning algorithms [101] frequently face this challenge, as algorithms must explore potentially rewarding solutions to problems while focusing computational power on the most promising options. Various robotic tasks, such as surveillance [96] and transmitter positioning [19], encounter similar challenges.

1.2 Motivation and Goals

In many practical scenarios decision-making is a *social* and *distributed* phenomenon: decision-makers interact with each other to share information regarding the world in order to make better choices as individuals. This is readily seen in humans and animals. In animals, social interaction is often a key component of foraging or mating behavior, and communicating efficiently is vital for survival. Humans routinely share or seek out information from a wide variety of sources, both real and virtual, to inform personal decisions in almost every domain from restaurants to romantic partners [105]. Distributed social decision-making allows decision-makers to leverage the experiences, data, and opinions of their peers to improve performance while still acting as an individual. Understanding, predicting, or engineering the behavior of such a group thereby requires one to understand the behavior of an individual and what drives individual decisions in light of external information.

Distributed social decision-making has been studied in a variety of contexts. The field of collective animal behavior is of particular importance to our work here, and has included studies of honeybees [89], monkeys [70], and many others [21, 24, 62]. The field of network science studies the role of network structure and communication

in distributed decision-making [39, 60, 68, 76]. Additional areas of research include social choice theory [4, 86, 90], which considers group ethics within voting theory, and social neuroeconomics [29, 61, 87], which considers psychology and neurosience in a game theoretic framework.

Recently, the study of the explore-exploit tradeoff has been extended to distributed social decision-making in various contexts. This work has included experiments based on simple multi-player games [68] and simulations [60]. The findings of several researchers, including preliminary versions of our work, have addressed this topic through model-based analysis [13, 45, 50].

This thesis investigates the explore-exploit tradeoff within distributed social decision-making from a model-based perspective. We rigorously examine how knowledge gleaned through communication affects performance outcomes for both the individual and the group, and also how individuals should behave in light of this additional information. We also investigate how the underlying communication structure and decision-making protocols affect performance in different settings.

1.3 Background and Related Work

1.3.1 The Multi-armed Bandit Problem

A canonical mathematical formulation of the explore-exploit tradeoff is the Multi-armed Bandit (MAB) problem [85]. In the MAB problem, a decision-maker faces a sequential series of decisions. In each decision, the decision-maker must choose between two or more options, also called "arms," each of which has an associated probability distribution that models its reward. After selecting an option the decision-maker then receives a noisy reward drawn from this option's associated probability distribution. The decision-maker's goal is to maximize their expected cumulative reward, which is equivalent to choosing the option with the highest mean as often as

possible. This goal is challenging because agents do not know the underlying mean associated with a given option; they can only estimate it through receiving a noisy reward. To perform well a decision-maker must deftly balance learning about these means through sampling (exploration), and maximizing their current expected reward (exploitation).

The MAB problem was first investigated by allied scientists during World War II, one of whom remarked that the problem "so sapped the energies and minds of Allied analysts that the suggestion was made that the problem be dropped over Germany, as the ultimate instrument of intellectual sabotage" [34]. The MAB problem was put into its modern form in 1952 by Robbins [85] and, in a seminal work, Lai and Robbins [54] established a lower bound on the expected number of times a sub-optimal option needs to be selected by an optimal policy. In another seminal work, Auer *et al.* [5] developed the upper confidence bound (UCB) algorithm for the stochastic MAB problem, which achieves this lower bound uniformly in time. We heavily utilize the UCB algorithm in this work and describe it in greater detail in Section 2.3.

MAB problems are pervasive across a variety of scientific communities, and have been used to model many systems that are characterized by the explore-exploit trade-off. The first major area of application was the design of clinical trials [6, 33], where a designer's goal is attempts to assign patients to experiments in order to both learn about the efficacy of a given treatment while maximizing the benefit to patients. Since then, the MAB problem has been applied to diverse areas such as control and robotics [19, 96], ecology [51, 95], human behavior and psychology [84], and communications [2, 53].

1.3.2 The Multi-agent Multi-armed Bandit Problem

Classically, the MAB problem involves only one decision-maker, or agent. This formulation has proven to be very useful in modeling in a wide variety of fields [11]. This

thesis principally considers the multi-agent MAB problem, in which multiple agents all choose sequentially from the same set of options.

Researchers have considered the multi-agent MAB problem in either a *centralized* or a *distributed* setting. In the centralized setting, agents' actions are controlled by a centralized decision-maker. In the distributed setting, each agent independently makes decisions regarding their own actions. As a modeling tool distributed decision-making is more reflective of many natural and large-scale artificial systems. Distributed algorithms also offer additional benefits in terms of robustness and flexibility as the failure of any component does not doom the whole. In this work we focus on distributed decision-making, but draw comparisons with the performance possible in a centralized setting.

In the multi-agent MAB problem researchers generally assume that agents can communicate in some fashion, and prior work can be roughly sorted according to two different models of communication between agents. The first is *indirect* communication, in which agents can detect when another user has selected the same option, or arm, as them but do not communicate any other information. The second is *direct* communication, where agents explicitly share information regarding the arms selected, rewards received, or estimates of arm means. These two models of communication, each with their associated features and constraints, allow researchers to model a wide variety of systems.

Foundational work on the multi-agent MAB problem was completed by Anantharam *et al.* [3], who studied single decision-makers with multiple plays, which is equivalent to multiple decision-makers acting under centralized control. Work in [3] established a lower bound on sub-optimal selections for this case (see Equation (2.8)). In related work, Shahrampour *et al.* [91] considered direct communication where rewards are player dependent and at each time all agents select the same arm as determined by majority vote.

The largest body of research utilizing the multi-agent MAB framework is focused on solving the cognitive radio network spectrum access problem. This problem considers a radio network, such as a wireless router, with multiple channels. A user can select a single channel and transmit data over it if the channel is unused, either by another user or some other higher-level task. The goal for a user is to choose a channel at each time such that they maximize the time spent in unused channels, thereby transmitting as much data as possible. Channel availability can be modeled by a probability distribution with a given mean, so users can learn which channels enable maximum data throughput by learning this mean.

Multiple researchers [2, 30, 45] have framed this problem mathematically as a multi-agent MAB problem, where an arm represents a channel, the probability distribution of the arm represents the channel availability, and a decision-making agent represents a network user. The constraint that only one user can use a channel at a time is modeled by *collisions*, where if two or more agents select the same arm at the same time they receive no reward. The presence of collisions implies that the optimal solution for the collective is for the M agents to select the M arms with the highest mean reward without any agents choosing the same arm.

Historically, researchers in the cognitive radio network problem have not considered the role of direct communication between network users. Specifically, users only interact indirectly through their experience of collisions with other users when accessing channels, and they use this information to sort themselves to achieve the optimal solution.

Another body of MAB literature that is closely related to the multi-agent MAB problem considers the case of *side-observations*. In this setting, when an agent chooses an arm they observe not only a noisy reward from the selected arm, but also from some set of other arms. These additional observations are the side-observations, and

which "side" arm(s) are observed is typically dependent on the selected arm. The agent will only receive reward from the arm selected, and not from side-observations.

This setting is similar to the multi-agent MAB problem because the additional information gained from side-observations is similar to that gained by directly communicating with others in the case when agents can communicate their rewards. Several authors have investigated the MAB problem with side observations [17, 66, 107], and Buccapatnam *et al.* [14] considered the case where the set of side arms observed is determined by a network. Inspired by social networks such as Facebook, Buccapatnam *et al.* [13] considered the case of an external agent selecting actions for multiple users in a network, where each action selection produces side-observations from the user's neighbors in the same network.

In a related vein, Kolla *et al.* [50] considered the case where agents can directly communicate by observing a neighbor's actions and also mimic the actions of a local decision-maker. Additionally, Kar *et al.* [46] considered the case of multiple agents, but where only one "major" agent can observe the reward values, and all other agents can only observe the actions of the major agent.

The prior work on the multi-agent MAB problem listed above that utilizes direct communication can be broadly categorized according to two different models of direct communication between agents. The first is *strictly local* communication, in which agents share the results of arm selections with their neighbors as determined by a given network graph that encodes who can communicate with whom. Depending on the problem formulation, side observations are a specific example or close analog of strictly local communication, and the work of Kolla *et al.* [50] falls neatly within this category. As the name implies, the spread of information when using strictly local communication between agents is limited because agents only share information regarding received rewards with neighbors, and this information generally cannot propagate through the entire network. The second model is direct communication

through *consensus*, in which an agent averages its own estimate of the means of the arms with the communicated estimates of its neighbors. Shahrampour *et al.* [91] used direct communication through consensus, and it has been used extensively in other fields. In terms of performance, consensus is a powerful and useful information sharing protocol because it allows information to propagate throughout the entire network of decision-makers. Also, as a model for how information is communicated and processed, information sharing through consensus can be applied to a wide variety of systems. We give a mathematical definition of both consensus and strictly local communication in Chapter 2.

1.4 Contributions and Research Overview

In this thesis we develop distributed algorithms for the multi-agent MAB problem in a variety of settings. We consider the *distributed*, *cooperative* multi-agent MAB problem, in which agents cooperate by communicating information regarding arm parameters, but make independent decisions regarding arm choice in order to maximize their own reward.

We principally consider the case of communication through consensus. Specifically, we use *running consensus*, in which agents average their opinions with those of their neighbors but also add in new observations as they become available. To our knowledge this is the first work to consider the distributed cooperative MAB problem using communication through consensus. We expect that this new formulation of the multi-agent MAB problem can be used in the future to answer important questions about the role of the communication network structure in social decision-making under the explore-exploit tradeoff.

We first develop several results regarding the performance of running consensus and bound the deviation of the estimates produced from those of a centralized esti-

mator. We expect that these results can be broadly useful in the study of multi-agent MAB problems in the future.

We then utilize these results to develop several novel distributed algorithms for the multi-agent MAB problem where agents can communicate over a network using running consensus. We prove an upper bound on the regret obtained using these algorithms, and relate it to the theoretical lower bound on regret from Anantharam *et al.* [3].

Additionally, we consider the case of multi-agent MAB with collisions, and investigate the case where agents can communicate through running consensus. We demonstrate that communication between agents improves performance over current algorithms that only utilize indirect communication in this setting.

We also study the multi-agent MAB problem with strictly local communication and develop a distributed algorithm that selects decision-makers and allows others to mimic those decision-makers. We show that this method enables significant performance improvements over methods that only allow restricted or no imitation.

Finally, we apply our results to the application area of nuclear facility inspection. We model the search for hazardous nuclear material in a facility using the MAB problem, and develop modified algorithms that enable robots to efficiently explore areas of interest. We show preliminary experimental results from this work.

1.5 Outline

This thesis is structured as follows. In Chapter 2 we define the MAB problem mathematically, along with several graph theory terms and running consensus. In Chapter 3 we give results on cooperative estimation of arm parameters using running consensus. We also develop and prove bounds for several novel heuristics that address the multi-agent MAB problem, including the coop-UCB, coop-UCB2, and coop-UCL al-

gorithms. In Chapter 3 we also discuss a proposed graph centrality measure, "explore-exploit centrality," and motivate its connections to networked processes facing the explore-exploit tradeoff. In Chapter 4 we expand on the heuristics of Chapter 3 and examine the case of the distributed multi-agent MAB problem with collisions. In Chapter 5, we study the multi-agent MAB problem with strictly local communication and derive and prove partition-based strategies. In Chapter 6 we utilize the lessons from our previous analyses to develop and apply MAB-based algorithms for multi-robot search tasks for radioactive material. Finally, we conclude in Chapter 7.

Chapter 2

The Single and Multi-agent Multi-armed Bandit Problems

In this chapter we review the multi-armed bandit (MAB) problem int he classical setting of a single decision0making agent and we introduce the new multi-agent MAB formulations. First, we specify relevant notation and then define both the single and multi-agent cases mathematically. Next, we review theoretical bounds on performance and discuss the classical Upper Confidence Bound policy for the single-agent MAB problem. We then discuss two different models for multi-agent communication and introduce relevant notation. Finally, we review the existing literature on multi-agent MAB policies.

2.1 The Multi-armed Bandit Problem

2.1.1 The Single-agent Multi-armed Bandit Problem

The single-agent MAB problem, first investigated in its modern form by Herbert Robbins [85], is a canonical formulation of the explore-exploit tradeoff. In the single-agent MAB problem, a single decision-making agent must choose an option, or arm,

from a finite set of alternatives at each timestep. At each time step, after choosing an arm, the agent receives a reward, which is drawn from a random distribution with a given mean that is unknown to the agent. The agent's goal is to maximize their cumulative reward over time. Doing this well requires choosing arms in such a way as to learn about unknown but potentially highly rewarding arms (exploration) while also accumulating reward from arms that are known to be good (exploitation).

We denote an arm in the set of alternative arms by $i \in \{1, \ldots, N\}$ with $N > 1$, and a timestep in the problem by $t \in \{1, \ldots, T\}$ with $T > 1$. Let $i(t)$ be the index of the arm selected at time t, $r_i(t)$ be the realized reward from arm i at time t, and $r(t) = r_i(t)\mathbb{1}\{i(t) = i\}$ be the corresponding received reward received from this selection where $\mathbb{1}\{\cdot\}$ is the indicator function. Let each arm i have mean m_i.

With this notation, the goal of the agent in the MAB problem to maximize the cumulative received reward up to time T becomes

$$\max \mathbb{E}\left[\sum_{t=1}^{T} r(t)\right] \tag{2.1}$$

where $\mathbb{E}[\cdot]$ denotes expectation. We can also express this goal in terms of arm selections as

$$\max \sum_{i=1}^{N} m_i \mathbb{E}\left[n_i(T)\right], \tag{2.2}$$

where $n_i(T)$ is the total number of times arm i has been selected up to and including time T.

In the above we have formulated the agent's goal as a maximization problem, but we can also conceive of it as a minimization problem. First, let us define the instantaneous expected regret at time t as $R(t) = m_{i^*} - m_{i(t)} = \Delta_{i(t)}$, where $i^* = \arg\max_{i \in \{1,\ldots,N\}} m_i$ is the arm with the highest mean. The instantaneous expected regret is therefore the expected difference in reward between the best arm and the arm chosen. Therefore, the goal of maximizing cumulative expected reward can be

equivalently expressed as minimizing the cumulative expected regret. Mathematically, the goal of the decision-making agent in the MAB problem is then

$$\min \sum_{t=1}^{T} \mathbb{E}\left[R(t)\right] = \min \sum_{i=1}^{N} \Delta_i \mathbb{E}\left[n_i(T)\right]. \tag{2.3}$$

In this thesis we use this second, regret-based formulation to investigate the efficiency of our algorithms. Using regret permits a cleaner interpretation of results and conforms to existing literature. Additionally, note that regret is purely an evaluative metric from the standpoint of an outside, all-knowing observer. The agent itself does not know their regret since they do not know the actual arm means m_i.

2.1.2 The Multi-agent Multi-armed Bandit Problem

The cooperative multi-agent MAB problem that we introduce in this thesis considers multiple decision-making agents acting over the same arm set. At each time each agent selects an arm and receives an independent and identically distributed reward associated withe the selected arm. Between rounds agents share information with each other in some fashion. In this work we focus on the impact, value, and consequences of this information exchange on decision-making strategy and performance.

We denote each agent $k \in \{1, \ldots, M\}$, with $M > 1$ and the arm selected by agent k at time t as $i^k(t)$. Analogously to the single-agent case, let $r_i^k(t)$ be realized reward for agent k selecting arm i at time t, and let $r^k(t) = r_i^k(t)\mathbb{1}\{i^k(t) = i\}$ be the corresponding received reward. We also define the expected instantaneous regret for agent k at time t as $R^k(t) = m_{i^*} - m_{i^k(t)} = \Delta_{i^k(t)}$.

The goal of each agent is to maximize their own expected cumulative reward. However, since they cooperate by exchanging information, we evaluate performance of the group of agents in the cooperative multi-agent MAB problem in terms of how

they maximize the group cumulative reward, expressed as

$$\max \mathbb{E}\left[\sum_{k=1}^{M}\sum_{t=1}^{T} r^k(t)\right] = \max \sum_{k=1}^{M}\sum_{i=1}^{N} m_i \mathbb{E}\left[n_i^k(T)\right]. \tag{2.4}$$

As in the single-agent case, this can also be equivalently expressed in terms of how they minimize the group cumulative regret, given as

$$\min \sum_{k=1}^{M}\sum_{t=1}^{T} \mathbb{E}\left[R^k(t)\right] = \min \sum_{k=1}^{M}\sum_{i=1}^{N} \Delta_i \mathbb{E}\left[n_i^k(T)\right]. \tag{2.5}$$

2.2 Lower Bound on Performance

Pioneering work by Lai and Robbins [54] established that there exists a lower bound on the regret of a decision-maker in the single-agent MAB problem in the frequentist setting. This lower bound effectively establishes the maximum expected achievable level of performance. For a general probability distribution p_i defining reward for each option i, the lower bound on the number of times a suboptimal arm is selected up to and including time T is

$$\mathbb{E}[n_i(T)] \geq \left(\frac{1}{\mathcal{D}(p_i||p_{i^*})} + o(1)\right) \ln T, \tag{2.6}$$

where $\mathcal{D}(p_i||p_{i^*})$ is the Kullback-Leibler divergence between distributions p_i and p_{i^*}. This simplifies to

$$\mathbb{E}[n_i(T)] \geq \left(\frac{2\sigma_s^2}{\Delta_i^2} + o(1)\right) \ln T \tag{2.7}$$

for Gaussian rewards with known variance. Note that as the difference between the mean of the best arm and a suboptimal arm i becomes small the bound on $n_i(T)$ becomes large. This encodes the intuitive result that a decision-maker will need

many samples to accurately distinguish between arms with highly similar probability distributions.

The results of Lai and Robbins [54] were extended by Anantharam *et al.*[3] to a cooperative setting with a centralized decision-maker with access to the realized rewards for every agent. In this setting the lower bound on the expected number of times a suboptimal arm i is selected by is

$$\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] \geq \left(\frac{1}{\mathcal{D}(p_i||p_{i^*})} + o(1) \right) \ln T, \tag{2.8}$$

and for Gaussian rewards with known variance this simplifies to

$$\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] \geq \left(\frac{2\sigma_s^2}{\Delta_i^2} + o(1) \right) \ln T. \tag{2.9}$$

In this thesis, we design several distributed algorithms whose expected cumulative regret is upper bounded by a logarithmic function. This implies that the performance of such algorithms is within a constant factor of the above bounds, and therefore is within a constant factor of the optimal performance possible.

Additionally, we show that, under the problem formulation using the consensus protocol for communication, the leading order of the upper bound on regret is independent of the number of agents in the network as $T \to \infty$. This result demonstrates the power and utility of cooperation and communication in the multi-agent MAB problem.

2.2.1 Probability Distribution of Rewards

Researchers have used a variety of assumptions regarding the probability distributions used to represent the noisy reward $r(t)$, or equivalently $r^k(t)$ in the multi-agent case. The particular distribution used is a function of modeling needs, as well as analytical tractability. The most common assumption is that rewards are bounded, without

loss of generality, in $[0, 1]$. The UCB algorithm [5] was originally developed under this assumption, and bounded rewards are applicable to a wide range of situations. Another common assumption is that rewards are normally distributed, with either known or unknown sample variance [47, 84, 95]. Gaussian, i.e. normally distributed, rewards are also applicable to a wide range of problems. In addition, some researchers have also considered rewards from heavy-tailed distributions [12, 64].

In this thesis, as in [63], we consider rewards drawn from some sub-Gaussian distribution, defined below for generalized variance σ_g. Sub-Gaussian random variables are a general class of distributions and include common distributions such as Gaussian, Bernoulli, and uniform, among others. We use the general sub-Gaussian class for tractability in the analysis of our algorithms, but we connect these results to the case of bounded rewards and Gaussian rewards with known variance to facilitate comparisons with the existing literature.

Definition 1 (***Sub-Gaussian random variables***)**.** *A random variable $X \in \mathbb{R}$ is sub-Gaussian [97] if $\mathbb{E}[X] = 0$ and*

$$\phi_X(\beta) \leq \frac{\sigma_g^2 \beta^2}{2}$$

where $\sigma_g > 0$, $\beta \in (-\infty, \infty)$ and $\phi_X(\beta) = \ln(\mathbb{E}[\exp(\beta X)])$ denotes the cumulant generating function of X.

For the case of Gaussian rewards the reward at arm i is drawn from a normal distribution with mean m_i and sample variance σ_s^2, denoted as $\mathcal{N}(m_i, \sigma_s^2)$. For bounded rewards we assume the reward at arm i is drawn from some bounded distribution with mean m_i, and the realized reward is assumed to be in $[0, 1]$ without loss of generality. With Gaussian rewards agents have access to σ_s^2, and with bounded rewards agents know rewards will fall in $[0, 1]$. Agents cannot access m_i in either case, so agents must estimate though sampling.

2.3 Single-agent Upper Confidence Bound Policy

Many different algorithms have been developed that address the single-agent MAB problem. MAB algorithms are typically analyzed along two dimensions. The first dimension is the expected cumulative regret. As seen in (2.6), the expected cumulative regret is lower bounded by a logarithmic function with a constant factor that is a function of the probability distribution of the arms. Therefore, an algorithm's theoretical performance can be judged by how close the upper bound on expected cumulative regret is to the lower bound.

The second dimension is the complexity of a given algorithm. It is desirable for an algorithm to require minimal computation for each decision-making step, and for the complexity of each calculation to be constant in T. This consideration is critical for practical implementation in robotic systems or for descriptive models of human or animal decision-making.

In this work we principally utilize the popular Upper Confidence Bound (UCB) algorithm, first developed by Auer *et al.* [5] for rewards bounded in $[0, 1]$. The UCB algorithm operates as follows:

After an initialization phase where the agent selects each arm once, the agent at time t computes

$$Q_i(t) = \mu_i(t-1) + C_i(t-1) \tag{2.10}$$

for each arm $i \in \{1, \ldots, N\}$, where

$$C_i(t-1) = \sqrt{\frac{2\ln(t-1)}{n_i(t-1)}} \tag{2.11}$$

and

$$\mu_i(t) = \frac{s_i(t)}{n_i(t)}. \tag{2.12}$$

The agent then selects the arm with the highest value of $Q_i(t)$. In the above $s_i(t)$ is the total sum of rewards from arm i and $n_i(t)$ is the total number of times arm i has been selected up to and including time t.

Auer *et al.* proved in [5] that the expected cumulative regret of UCB is upper bounded by a logarithmic function that is a constant factor of the lower bound (2.6). They also proved that UCB requires also requires minimal computation that is constant in T and linear in N, which is easily tractable for robotic systems and applicable to natural systems.

Furthermore, the structure of the UCB algorithm lends itself to simple qualitative analysis in terms of the explore-exploit tradeoff. As seen above, the Q term of UCB is composed of two quantities. The first, the estimated mean of the arm in question, can be understood as driving exploitation. This term pushes agents to select arms with a high estimated mean. The second term, the C value, can be seen as promoting exploration. This term pushes an agent to explore arms that have not been sampled much relative to t.

C can be interpreted as a bound on the uncertainty of the estimates of the mean of an arm. In this light one can consider Q to be an optimistic estimate of the value of an arm, and the UCB algorithm to being optimistic in the face of uncertainty.

2.4 Multi-agent Communication

In this thesis we consider two different communication protocols for inter-agent communication: consensus and strictly local communication. Here we define these two protocols mathematically as well some relevant graph theoretic terms.

2.4.1 Graph Terminology

We define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges between nodes. We model the communication network connecting agents as a graph, where each node corresponds to an agent and each edge between a pair of nodes as a communication path between those nodes. In this work we assume unweighted, bidirectional communication between agents, so if agent i can communicate with agent j, j can communicate with i. In graph notation, this means that $\mathcal{G}$ is an undirected graph, which requires that if $(i, j) \in \mathcal{E}$ then $(j, i) \in \mathcal{E}$ where $i, j \in \mathcal{V}$.

A graph can be encoded in matrix form easily using the Laplacian matrix $L \in \mathbb{R}^{M \times M}$ associated with $\mathcal{G}$, where $l_{ij} = -1$ if there is a edge between node i and node $j \neq i$ and $l_{ij} = 0$ otherwise. Additionally, diagonal element $l_{ii} = \deg(i)$, where $\deg(i)$ is the degree of node i and node i's degree is defined as the number neighbors of node i. We assume that the graph $\mathcal{G}$ is connected, i.e., there exists a path formed by edges for every pair of nodes. Additionally, since $\mathcal{G}$ is undirected L will be symmetric.

2.4.2 Consensus

In the discrete-time consensus algorithm [44, 104], agents average their opinion with their neighbors' opinions at each timestep. The objective of the consensus algorithm is to ensure agreement among agents on a common value. In this work we utilize consensus to model and enable communication between agents regarding estimates of arm means.

The discrete-time consensus algorithm can be expressed as

$$\mathbf{x}(t+1) = P\mathbf{x}(t), \tag{2.13}$$

where $\mathbf{x}(t) = (x^1(t), \ldots, x^M(t))^T$ is the vector of agent opinions at time t, and P is a row stochastic matrix given by

$$P = \mathcal{I}_M - \frac{\kappa}{d_{\max}} L. \tag{2.14}$$

Here $\mathcal{I}_M$ is the identity matrix of order M, $\kappa \in (0, 1]$ is a step size parameter [75], $d_{\max} = \max\{\deg(i) \mid i \in \{1, \ldots, M\}\}$.

In the context of social networks, the consensus algorithm (2.13) is referred to as the DeGroot model [27] and has been successfully used to describe evolution of opinions [36].

One drawback of the consensus algorithm (2.13) is that it does not allow for the incorporation of new external information. This drawback can be mitigated by adding a forcing term and the resulting algorithm is called *running consensus* [9] or dynamic consensus [93]. Similar to (2.13), running consensus updates the opinion at time t as

$$\mathbf{x}(t) = P\mathbf{x}(t-1) + P\boldsymbol{\upsilon}(t-1), \tag{2.15}$$

where $\boldsymbol{\upsilon}(t)(\upsilon^1(t), \ldots, \upsilon^k(t))^T$ is the vector of information received by the agents at time t. In the running consensus update (2.15), each agent k collects information $\upsilon^k(t)$ at time t, adds it to its current opinion, and then averages its updated opinion with the updated opinion of its neighbors.

2.4.3 Strictly Local Communication

In the strictly local communication protocol agents can access the choices and realized rewards of their neighbors in $\mathcal{G}$, but they do not share estimates.

Using a consensus formulation, strictly local communication is equivalent to setting

$$P = \mathcal{I}_M - D_i L \tag{2.16}$$

where D_i is an $M \times M$ matrix with $\deg(i)$ on the i'th diagonal entry and zeros elsewhere. Note that P is now not necessarily row stochastic and all agent's opinions will not necessarily converge to the centralized average.

Chapter 3

The Distributed Cooperative Upper-Confidence Bound Algorithm [1]

In this chapter we first define and then analyze a running consensus algorithm that is used for direct communication between agents for the purpose of cooperative estimation of arm means. We then prove an important theorem on the performance of this cooperative estimation algorithm.

Next, we describe the coop-UCB1, coop-UCB2, and coop-UCL algorithms for the multi-agent MAB problem with direct communication through consensus. For each of these algorithms we show that they achieve logarithmic regret. We also describe and motivate "explore-exploit" centrality, a new centrality measure that is predictive of performance in networked explore-exploit problems.

[1] This chapter is adapted from [57], [56], and [55]. Sections 3.1 and 3.2.2 are mostly taken verbatim from [55], with a preliminary version of Section 3.1 applying only to Gaussian rewards appearing in [57]. Section 3.2.1 is adapted from [57] and generalized to sub-Gaussian rewards. Section 3.3 is mostly taken verbatim from [56]. The numerical illustrations in Section 3.4 are partially adapted from [55], with some text taken verbatim.

We then analyze the performance of these three algorithms and demonstrate the utility of explore-exploit centrality through several numerical simulations. Finally, we utilize the coop-UCB2 algorithm to conduct a robotic search task.

3.1 Cooperative Estimation of Mean Rewards

In this section we investigate the cooperative estimation of mean rewards at each arm. To this end, we propose two running consensus algorithms for each arm and analyze their performance.

3.1.1 Cooperative Estimation Algorithm

For distributed cooperative estimation of the mean reward at each arm i, we employ two running consensus algorithms: (i) for estimation of total reward provided at the arm, and (ii) for estimation of the total number of times the arm has been sampled.

Let $\hat{s}_i^k(t)$ and $\hat{n}_i^k(t)$ be agent k's estimate of the total reward provided at arm i per unit agent and the total number of times arm i has been selected until time t per unit agent, respectively. Using $\hat{s}_i^k(t)$ and $\hat{n}_i^k(t)$ agent k can calculate $\hat{\mu}_i^k(t)$, the estimated empirical mean of arm i at time t defined by

$$\hat{\mu}_i^k(t) = \frac{\hat{s}_i^k(t)}{\hat{n}_i^k(t)}. \tag{3.1}$$

Let $i^k(t)$ be the arm sampled by agent k at time t and let $\xi_i^k(t) = \mathbb{1}(i^k(t) = i)$. $\mathbb{1}(\cdot)$ is the indicator function, here equal to 1 if $i^k(t) = i$ and 0 otherwise. For simplicity of notation we define $r_i^k(t)$ as the realized reward at arm i for agent k, which is a random variable sampled from a sub-Gaussian distribution, and the corresponding accumulated reward is $r^k(t) = r_i^k(t) \cdot \mathbb{1}(i^k(t) = i)$.

The estimates $\hat{s}_i^k(t)$ and $\hat{n}_i^k(t)$ are updated using running consensus as follows

$$\hat{\mathbf{n}}_i(t+1) = P\hat{\mathbf{n}}_i(t) + P\boldsymbol{\xi}_i(t), \tag{3.2}$$

$$\text{and} \quad \hat{\mathbf{s}}_i(t+1) = P\hat{\mathbf{s}}_i(t) + P\mathbf{r}_i(t), \tag{3.3}$$

where $\hat{\mathbf{n}}_i(t)$, $\hat{\mathbf{s}}_i(t)$, $\boldsymbol{\xi}_i(t)$, and $\mathbf{r}_i(t)$ are vectors of $\hat{n}_i^k(t)$, $\hat{s}_i^k(t)$, $\xi_i^k(t)$, and $r_i^k(t) \cdot \mathbb{1}(i^k(t) = i)$, $k \in \{1, \ldots, M\}$, respectively.

3.1.2 Analysis of the Cooperative Estimation Algorithm

We now analyze the performance of the estimation algorithm defined by (3.1), (3.2) and (3.3). Let $n_i^{\text{cent}}(t) \equiv \frac{1}{M}\sum_{\tau=1}^{t} \mathbf{1}_M^\top \boldsymbol{\xi}_i(\tau)$ be the total number of times arm i has been selected per unit agent up to and including time τ, and let $s_i^{\text{cent}}(t) \equiv \frac{1}{M}\sum_{\tau=1}^{t} \boldsymbol{\xi}_i^\top(t)\mathbf{r}_i(t)$ be the total reward provided at arm i per unit agent up to and including time t. Also, let λ_i denote the i-th largest eigenvalue of P, $\mathbf{u}_i$ the eigenvector corresponding to λ_i, u_i^d the d-th entry of $\mathbf{u}_i$, and

$$\epsilon_n = \sqrt{M}\sum_{p=2}^{M} \frac{|\lambda_p|}{1-|\lambda_p|}. \tag{3.4}$$

Note that $\lambda_1 = 1$ and $\mathbf{u}_1 = \mathbf{1}_M/\sqrt{M}$. Let us define

$$\nu_{pj}^{+\text{sum}} = \sum_{d=1}^{M} u_p^d u_j^d \mathbb{1}(u_p^k u_j^k \geq 0)$$

$$\text{and} \quad \nu_{pj}^{\text{-sum}} = \sum_{d=1}^{M} u_p^d u_j^d \mathbb{1}(u_p^k u_j^k \leq 0).$$

We also define

$$a_{pj}(k) = \begin{cases} \nu_{pj}^{+\text{sum}} u_p^k u_j^k, \text{ if } \lambda_p \lambda_j \geq 0 \ \& \ u_p^k u_j^k \geq 0, \\ \nu_{pj}^{\text{-sum}} u_p^k u_j^k, \text{ if } \lambda_p \lambda_j \geq 0 \ \& \ u_p^k u_j^k \leq 0, \\ \nu_{pj}^{\max} |u_p^k u_j^k|, \text{ if } \lambda_p \lambda_j < 0, \end{cases} \tag{3.5}$$

where $\nu_{pj}^{\max} = \max\{|\nu_{pj}^{\text{-sum}}|, \nu_{pj}^{+\text{sum}}\}$. Furthermore, let

$$\epsilon_c^k = M \sum_{p=1}^{M} \sum_{j=2}^{M} \frac{|\lambda_p \lambda_j|}{1 - |\lambda_p \lambda_j|} a_{pj}(k). \tag{3.6}$$

We note that both ϵ_n and ϵ_c^k depend only on the structure of the communication graph. These are measures of distributed cooperative estimation performance.

Proposition 1 (***Performance of cooperative estimation***)**.** *For the distributed estimation algorithm defined in* (3.1), (3.2) *and* (3.3)*, and a doubly stochastic matrix* P *defined in* (2.14)*, the following statements hold*

(i) the estimate $\hat{n}_i^k(t)$ *satisfies*

$$n_i^{cent}(t) - \epsilon_n \leq \hat{n}_i^k(t) \leq n_i^{cent}(t) + \epsilon_n;$$

(ii) the following inequality holds for the estimate $\hat{n}_i^k(t)$ *and the sequence* $\{\xi_i^j(\tau)\}_{\tau \in \{1,\ldots,t\}}, j \in \{1, \ldots, M\}$

$$\sum_{\tau=1}^{t} \sum_{j=1}^{M} \left(\sum_{p=1}^{M} \lambda_p^{t-\tau+1} u_p^k u_p^j \right)^2 \xi_i^j(\tau) \leq \frac{\hat{n}_i^k(t) + \epsilon_c^k}{M}.$$

Proof. We begin with the first statement. From (3.2) it follows that

$$\begin{aligned}\hat{\mathbf{n}}_i(t) &= P^t\hat{\mathbf{n}}_i(0) + \sum_{\tau=1}^{t} P^{t-\tau+1}\boldsymbol{\xi}_i(\tau) \\ &= \sum_{\tau=0}^{t}\Big[\frac{1}{M}\mathbf{1}_M\mathbf{1}_M^\top\boldsymbol{\xi}_i(\tau) + \sum_{p=2}^{M}\lambda_p^{t-\tau+1}\mathbf{u}_p{\mathbf{u}_p}^\top\boldsymbol{\xi}_i(\tau)\Big] \\ &= n_i^{\text{cent}}(t)\mathbf{1}_M + \sum_{\tau=1}^{t}\sum_{p=2}^{M}\lambda_p^{t-\tau+1}\mathbf{u}_p{\mathbf{u}_p}^\top\boldsymbol{\xi}_i(\tau). \end{aligned} \tag{3.7}$$

We now bound the k-th entry of the second term on the right hand side of (3.7):

$$\begin{aligned}\sum_{\tau=1}^{t}\sum_{p=2}^{M}\lambda_p^{t-\tau+1}\big(\mathbf{u}_p{\mathbf{u}_p}^\top\boldsymbol{\xi}_i(\tau)\big)_k &\le \sum_{\tau=1}^{t}\sum_{p=2}^{M}|\lambda_p^{t-\tau+1}|\|\mathbf{u}_p\|_2^2\|\boldsymbol{\xi}_i(\tau)\|_2 \\ &\le \sqrt{M}\sum_{\tau=1}^{t}\sum_{p=2}^{M}|\lambda_p^{t-\tau+1}| \le \epsilon_n.\end{aligned}$$

This establishes the first statement.

To prove the second statement, we note that

$$\begin{aligned}&\sum_{\tau=1}^{t}\sum_{j=1}^{M}\left(\sum_{p=1}^{M}\lambda_p^{t-\tau+1}u_p^k u_p^j\right)^2\xi_i^j(\tau) \\ &\quad= \sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{w=1}^{M}(\lambda_p\lambda_w)^{t-\tau+1}u_p^k u_w^k\sum_{j=1}^{M}u_p^j u_w^j\xi_i^j(\tau) \\ &\quad= \sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{w=2}^{M}(\lambda_p\lambda_w)^{t-\tau+1}u_p^k u_w^k\nu_{pwi}(\tau) + \frac{1}{M}\sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{j=1}^{M}\lambda_p^{t-\tau+1}u_p^k u_p^j\xi_i^j(\tau) \\ &\quad= \sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{w=2}^{M}(\lambda_p\lambda_w)^{t-\tau+1}u_p^k u_w^k\nu_{pwi}(\tau) + \frac{1}{M}\hat{n}_i^k(t),\end{aligned} \tag{3.8}$$

where $\nu_{pwi}(\tau) = \sum_{j=1}^{M} u_p^j u_w^j \xi_i^j(\tau)$.

We now analyze the first term of (3.8):

$$\begin{aligned}
&\sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{w=2}^{M}(\lambda_p\lambda_w)^{t-\tau+1}u_p^k u_w^k \nu_{pwi}(\tau) \\
&\qquad \leq \sum_{\tau=1}^{t}\sum_{p=1}^{M}\sum_{w=2}^{M}|(\lambda_p\lambda_w)^{t-\tau+1}||u_p^k u_w^k \nu_{pwi}(\tau)| \\
&\qquad \leq \sum_{\tau=0}^{t-1}\sum_{p=1}^{M}\sum_{w=2}^{M}|\lambda_p\lambda_w|^{t-\tau+1} a_{pw}(k) \\
&\qquad \leq \sum_{p=1}^{M}\sum_{w=2}^{M}\frac{|\lambda_p\lambda_w|}{1-|\lambda_p\lambda_w|} a_{pw}(k). \qquad (3.9)
\end{aligned}$$

Bounds in (3.9) establish the second statement. □

We now derive bounds on the deviation of the estimated mean when using the cooperative estimation algorithm using techniques from [31]. Recall that for $i \in \{1,\dots,N\}$ and $k \in \{1,\dots,M\}$ let $\{r_i^k(t)\}_{t\in\mathbb{N}}$ be the sequence of i.i.d. sub-Gaussian with mean $m_i \in \mathbb{R}$. Let $\mathcal{F}_t$ be the filtration defined by the sigma-algebra of all the measurements until time t. Let $\{\xi_i^k(t)\}_{t\in\mathbb{N}}$ be a sequence of Bernoulli variables such that $\xi_i^k(t)$ is deterministically known given $\mathcal{F}_{t-1}$, i.e., $\xi_i^k(t)$ is pre-visible w.r.t. $\mathcal{F}_{t-1}$. Additionally, let $\phi_i(\beta) = \ln\big(\mathbb{E}[\exp\big(\beta r_i^k(t)\big)]\big)$ denote the cumulant generating function of $r_i^k(t)$.

Theorem 1 (***Estimator Deviation Bounds***)**.** *For the estimates $\hat{s}_i^k(t)$ and $\hat{n}_i^k(t)$ obtained using equations* (3.2) *and* (3.3) *given rewards drawn from a sub-Gaussian distribution as defined in Definition 1, the following concentration inequality holds*

$$\mathbb{P}\left(\frac{\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{1/2}} > \delta\right) < \left\lceil\frac{\ln\left(t+\epsilon_n\right)}{\ln\left(1+\eta\right)}\right\rceil \exp\left(\frac{-\delta^2}{2\sigma_g^2}G(\eta)\right), \qquad (3.10)$$

where $\delta > 0$, $\eta > 0$, $G(\eta) = (1 - \frac{\eta^2}{16})$, and ϵ_c^k and ϵ_n are defined in (3.6) *and* (3.4)*, respectively.*

Proof. We begin by noting that $\hat{s}_i^k(t)$ can be decomposed as

$$\hat{s}_i^k(t) = \sum_{\tau=1}^{t} \sum_{p=1}^{M} \lambda_p^{t-\tau+1} \sum_{j=1}^{M} u_p^k u_p^j r_i^j(\tau) \xi_i^j(\tau). \tag{3.11}$$

Let $\hat{s}_i^{kp}(t) = \sum_{\tau=1}^{t} \lambda_p^{t-\tau+1} \sum_{j=1}^{M} u_p^k u_p^j r_i^j(\tau) \xi_i^j(\tau)$. Then,

$$\sum_{p=1}^{M} \hat{s}_i^{kp}(t) = \sum_{p=1}^{M} \sum_{j=1}^{M} \lambda_p u_p^k u_p^j r_i^j(t) \xi_i^j(t) + \sum_{p=1}^{M} \lambda_p \hat{s}_i^{kp}(t-1). \tag{3.12}$$

It follows from (3.11) and (3.12) that for any $\Theta > 0$

$$\begin{aligned}
\mathbb{E}\left[\exp\big(\Theta \hat{s}_i^k(t)\big) \big| \mathcal{F}_{t-1}\right] &= \mathbb{E}\left[\exp\left(\Theta \sum_{p=1}^{M} \hat{s}_i^{kp}(t)\right) \middle| \mathcal{F}_{t-1}\right] \\
&= \mathbb{E}\left[\exp\left(\Theta \sum_{p=1}^{M} \lambda_p \sum_{j=1}^{M} u_p^k u_p^j r_i^j(t) \xi_i^j(t)\right) \middle| \mathcal{F}_{t-1}\right] \times \exp\left(\Theta \sum_{p=1}^{M} \lambda_p \hat{s}_i^{kp}(t-1)\right) \\
&= \exp\left(\sum_{j=1}^{M} \phi_i \left(\Theta \sum_{p=1}^{M} \lambda_p u_p^k u_p^j r_i^j(t)\right) \xi_i^j(t)\right) \times \exp\left(\Theta \sum_{p=1}^{M} \lambda_p \hat{s}_i^{kp}(t-1)\right),
\end{aligned}$$

where both $r_i^j(t)$ and $\xi_i^j(t)$ are known deterministically and $r_i^j(t)$ are i.i.d. for each $j \in \{1, \ldots, M\}$. Therefore, it follows that

$$\begin{aligned}
\mathbb{E}\left[\exp\left(\Theta \sum_{p=1}^{M} \hat{s}_i^{kp}(t) - \sum_{j=1}^{M} \phi_i \left(\Theta \sum_{p=1}^{M} \lambda_p u_p^k u_p^j r_i^j(t)\right) \times \xi_i^j(t)\right) \middle| \mathcal{F}_{t-1}\right] \\
= \exp\left(\Theta \sum_{p=1}^{M} \lambda_p \hat{s}_i^{kp}(t-1)\right).
\end{aligned}$$

Using the above argument recursively with the fact that $s_i^k(0) = 0$, we obtain

$$\mathbb{E}\left[\exp\left(\Theta \hat{s}_i^k(t) - \sum_{\tau=1}^{t} \sum_{j=1}^{M} \phi_i \left(\Theta \sum_{p=1}^{M} \lambda_p^{t-\tau+1} u_p^k u_p^j r_i^j(\tau)\right) \xi_i^j(\tau)\right)\right] = 1.$$

Since for sub-Gaussian random variables $\phi_i(\beta) \leq \beta m_i + \frac{1}{2}\sigma_g^2\beta^2$, we have

$$1 = \mathbb{E}\left[\exp\left(\Theta\big(\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)\big) - \frac{\sigma_g^2}{2}\sum_{\tau=1}^{t}\sum_{j=1}^{M}\left(\Theta\sum_{p=1}^{M}\lambda_p^{t-\tau+1}u_p^k u_p^j\right)^2\xi_i^j(\tau)\right)\right] \tag{3.13}$$
$$\geq \mathbb{E}\left[\exp\left(\Theta\big(\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)\big) - \frac{\sigma_g^2\Theta^2}{2M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)\right)\right],$$

where the last inequality follows from the second statement of Proposition 1. Now using the Markov Inequality, we obtain

$$e^{-a} \geq \mathbb{P}\left(\exp\left(\Theta\big(\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)\big) - \frac{\sigma_g^2\Theta^2}{2M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)\right) \geq e^a\right)$$
$$= \mathbb{P}\left(\frac{\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)\right)^{\frac{1}{2}}} \geq \frac{a}{\Theta}\left(\frac{1}{M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)\right)^{-\frac{1}{2}} + \frac{\sigma_g^2\Theta}{2}\left(\frac{1}{M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)\right)^{\frac{1}{2}}\right).$$

The right hand side of the above equation contains a random variable $\hat{n}_i^k(t)$ which is dependent on the random variable on the left hand side. Therefore, we use union bounds on $\hat{n}_i^k(t)$ to obtain the desired concentration inequality. Towards this end, we consider an exponentially increasing sequence to time indices $\{(1+\eta)^{h-1} \mid h \in \{1, \ldots, D\}\}$, where $D = \left\lceil \frac{\ln(t+\epsilon_n)}{\ln(1+\eta)} \right\rceil$ and $\eta > 0$. For every $h \in \{1, \ldots, D\}$, define

$$\Theta_h = \frac{1}{\sigma_g}\sqrt{\frac{2aM}{(1+\eta)^{h-\frac{1}{2}} + \epsilon_c^k}}. \tag{3.14}$$

Thus, if $(1+\eta)^{h-1} \leq \hat{n}_i^k(t) \leq (1+\eta)^h$, then

$$\begin{aligned}
&\frac{a}{\Theta_h}\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{-\frac{1}{2}} + \frac{\sigma_g^2\Theta_h}{2}\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}} \\
&\quad = \sigma_g\sqrt{\frac{a}{2}}\left(\left(\frac{(1+\eta)^{h-\frac{1}{2}}+\epsilon_c^k}{\hat{n}_i^k(t)+\epsilon_c^k}\right)^{\frac{1}{2}} + \left(\frac{\hat{n}_i^k(t)+\epsilon_c^k}{(1+\eta)^{h-\frac{1}{2}}+\epsilon_c^k}\right)^{\frac{1}{2}}\right) \\
&\quad \leq \sigma_g\sqrt{\frac{a}{2}}\left(\left(\frac{(1+\eta)^{h-\frac{1}{2}}}{\hat{n}_i^k(t)}\right)^{\frac{1}{2}} + \left(\frac{\hat{n}_i^k(t)}{(1+\eta)^{h-\frac{1}{2}}}\right)^{\frac{1}{2}}\right) \\
&\quad \leq \sigma_g\sqrt{\frac{a}{2}}\left((1+\eta)^{\frac{1}{4}} + (1+\eta)^{-\frac{1}{4}}\right),
\end{aligned}$$

where the second-last inequality follows from the fact that for $a, b > 0$, the function $\epsilon \mapsto \sqrt{\frac{a+\epsilon}{b+\epsilon}} + \sqrt{\frac{b+\epsilon}{a+\epsilon}}$ with domain $\mathbb{R}_{\geq 0}$ is monotonically non-increasing, and the last inequality follows from the fact that for $\eta > 0$, the function $x \mapsto \sqrt{\frac{(1+\eta)^{h-\frac{1}{2}}}{x}} + \sqrt{\frac{x}{(1+\eta)^{h-\frac{1}{2}}}}$ with domain $[(1+\eta)^{h-1}, (1+\eta)^h]$ achieves its maximum at either of the boundaries. Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\frac{\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}}} > \sigma_g\sqrt{\frac{a}{2}}\left((1+\eta)^{\frac{1}{4}} + (1+\eta)^{-\frac{1}{4}}\right)\right) \\
&\leq \sum_{h=1}^{D}\mathbb{P}\Bigg(\frac{\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}}} > \frac{a}{\Theta_h}\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{-\frac{1}{2}} + \frac{\sigma_g^2\Theta_h}{2}\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}} \\
&\qquad \& \ (1+\eta)^{h-1} \leq \hat{n}_i^k(t)+\epsilon_c^k < (1+\eta)^h\Bigg) \leq De^{-a}.
\end{aligned}$$

Setting $\sigma_g\sqrt{\frac{a}{2}}\left((1+\eta)^{\frac{1}{4}} + (1+\eta)^{-\frac{1}{4}}\right) = \delta$, this yields

$$\mathbb{P}\left(\frac{\hat{s}_i^k(t) - m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}}} > \delta\right) \leq D\exp\left(\frac{-2\delta^2}{\sigma_g^2\left((1+\eta)^{\frac{1}{4}} + (1+\eta)^{-\frac{1}{4}}\right)^2}\right)$$

It can be verified that the first three terms in the Taylor series for $\frac{4}{\left((1+\eta)^{\frac{1}{4}}+(1+\eta)^{-\frac{1}{4}}\right)^2}$ provide a lower bound, i.e.,

$$\frac{4}{\left((1+\eta)^{\frac{1}{4}}+(1+\eta)^{-\frac{1}{4}}\right)^2} \geq 1-\frac{\eta^2}{16}.$$

Therefore, it holds that

$$\begin{aligned}\mathbb{P}\left(\frac{\hat{s}_i^k(t)-m_i\hat{n}_i^k(t)}{\left(\frac{1}{M}\left(\hat{n}_i^k(t)+\epsilon_c^k\right)\right)^{\frac{1}{2}}}>\delta\right) &\leq D\exp\left(\frac{-\delta^2}{2\sigma_g^2}\left(1-\frac{\eta^2}{16}\right)\right)\\ &= \left\lceil\frac{\ln\left(t+\epsilon_n\right)}{\ln\left(1+\eta\right)}\right\rceil\exp\left(\frac{-\delta^2}{2\sigma_g^2}\left(1-\frac{\eta^2}{16}\right)\right).\end{aligned}$$

□

3.2 Cooperative Decision-Making

In this section, we extend the UCB algorithm [5] to the distributed cooperative setting in which multiple agents can communicate with each other according to a given graph topology. We develop the coop-UCB1 and coop-UCB2 algorithms for the case of sub-Gaussian rewards and prove upper-bounds on the regret of both algorithms. Intuitively, compared to the single agent setting, in the cooperative setting each agent will be able to perform better due to communication with neighbors. However, the extent of an agent's performance advantage depends on the network structure. We compute bounds on the performance of the group in terms of the expected group cumulative regret.

3.2.1 The coop-UCB1 Algorithm

The cooperative UCB1 (coop-UCB1) algorithm is analogous to the UCB algorithm, and uses a modified decision-making heuristic that captures the effect of the additional

information an agent receives through communication with other agents as well as the rate of information propagation through the network.

The coop-UCB1 algorithm is initialized by each agent sampling each arm once and proceeds as follows. At each time t each agent k selects the arm with maximum $Q_i^k(t-1) = \hat{\mu}_i^k(t-1) + C_i^k(t-1)$, where

$$C_i^k(t-1) = \sigma_g \sqrt{\frac{2\gamma}{G(\eta)} \cdot \frac{\hat{n}_i^k(t-1) + \epsilon_c^k}{M\hat{n}_i^k(t-1)} \cdot \frac{\ln(t-1)}{\hat{n}_i^k(t-1)}}, \tag{3.15}$$

and receives realized reward $r_i^k(t)$ from a sub-Gaussian distribution, where $\gamma > 1$, $G(\eta) = (1 - \eta^2/16)$, and $\eta \in (0, 4)$. Each agent k updates its cooperative estimate of the mean reward at each arm using the distributed cooperative estimation algorithm described in (3.1), (3.2), and (3.3). Note that the heuristic Q_i^k requires the agent k to know ϵ_c^k, which depends on the global graph structure. This requirement can be relaxed by replacing ϵ_c^k with an increasing sub-logarithmic function of time, as seen in Section 3.2.2

Regret Analysis of the coop-UCB1 Algorithm

We now derive a bound on the expected cumulative group regret using the distributed coop-UCB1 algorithm. This bound recovers the upper bound given in (2.8) within a constant factor. The contribution of each agent to the group regret is a function of its location in the network as determined by ϵ_c^k (see Remark 4).

Theorem 2 (***Regret of the coop-UCB1 Algorithm***)**.** *For the coop-UCB1 algorithm and the MAB problem with sub-Gaussian rewards the number of times a*

suboptimal arm i is selected by all agents until time T satisfies

$$\sum_{k=1}^{M}\mathbb{E}[n_i^k(T)] \leq \left\lceil M\epsilon_n + \sum_{k=1}^{M}\frac{8\sigma_g^2\gamma(1+\epsilon_c^k)}{\Delta_i^2 M}\ln(T)\right\rceil + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2} + \frac{\ln\left((1+\epsilon_n)(1+\eta)\right)}{\gamma-1} + 2\right)$$

where $\eta > 0$ and $\gamma > 1$.

Proof. We proceed similarly to [5]. The number of times a suboptimal arm i is selected by all agents until time T is

$$\begin{aligned}\sum_{k=1}^{M} n_i^k(T) &= \sum_{k=1}^{M}\sum_{t=1}^{T}\mathbb{1}(i^k(t) = i^k) \\ &\leq \sum_{k=1}^{M}\sum_{t=1}^{T}\mathbb{1}(Q_i^k(t-1) \geq Q_{i^*}^k(t-1)) \\ &\leq A + \sum_{k=1}^{M}\sum_{t=1}^{T}\mathbb{1}(Q_i^k(t-1) \geq Q_{i^*}^k(t-1), Mn_i^{\text{cent}}(t-1) \geq A), \end{aligned} \tag{3.16}$$

where $A > 0$ is a constant that will be chosen later.

At a given time t an individual agent k will choose a suboptimal arm only if $Q_i^k(t-1) \geq Q_{i^*}^k(t-1)$. For this condition to be true at least one of the following three conditions must hold:

$$\hat{\mu}_{i^*}(t-1) \leq m_{i^*} - C_{i^*}^k(t-1) \tag{3.17}$$

$$\hat{\mu}_i(t-1) \geq m_i + C_i^k(t-1) \tag{3.18}$$

$$m_{i^*} < m_i + 2C_i^k(t-1). \tag{3.19}$$

We now bound the probability that (3.23) holds. Applying Theorem 1 and noting that $t \geq 1$ it follows that

$$\begin{aligned}
\mathbb{P}\big(\hat{\mu}_i^k \geq m_i + C_i^k(t)\big) &= \mathbb{P}\left(\frac{\hat{s}_i^k - m_i\hat{n}_i^k}{\sqrt{\frac{1}{M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)}} \geq \sigma_g\sqrt{\frac{2\gamma\ln(t)}{G(\eta)}}\right) \\
&\leq \left\lceil \frac{\ln(t+\epsilon_n)}{\ln(1+\eta)} \right\rceil \exp(-\gamma\ln(t)) \\
&\leq \left(\frac{\ln(t+\epsilon_n)}{\ln(1+\eta)} + 1\right)\exp(-\gamma\ln(t)) \\
&= \left(\frac{\ln\left(t\frac{t+\epsilon_n}{t}\right)}{\ln(1+\eta)} + 1\right)\frac{1}{t^\gamma} \\
&\leq \left(\frac{\ln(t(1+\epsilon_n))}{\ln(1+\eta)} + 1\right)\frac{1}{t^\gamma} \\
&= \left(\frac{\ln(t)}{\ln(1+\eta)} + \frac{\ln(1+\epsilon_n)}{\ln(1+\eta)} + 1\right)\frac{1}{t^\gamma}.
\end{aligned}$$

It follows analogously with a slight modification to Theorem 1 that

$$\mathbb{P}((3.23) \text{ holds}) \leq \left(\frac{\ln(t)}{\ln(1+\eta)} + \frac{\ln(1+\epsilon_n)}{\ln(1+\eta)} + 1\right)\frac{1}{t^\gamma}.$$

Finally, we examine the probability that (3.24) holds. It follows that

$$\begin{aligned}
& m_{i^*} < m_i + 2C_i^k(t) \\
\implies\ & n_i^{\text{cent}}(t) < \left\lceil \epsilon_n + \frac{8\sigma_g^2\gamma(\hat{n}_i^k(t) + \epsilon_c^k)\ln(t)}{M\Delta_i^2(\hat{n}_i^k(t))^2} \right\rceil \\
& \leq \left\lceil \epsilon_n + \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)\ln(t)}{M\Delta_i^2} \right\rceil.
\end{aligned}$$

From monotonicity of $\ln(t)$, it follows that (3.24) does not hold if $n_i^{\text{cent}}(t) \geq \lceil \epsilon_n + \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)\ln(T)}{M\Delta_i^2} \rceil$.

Now, setting $A = \lceil M\epsilon_n + \sum_{k=1}^{M} \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)\ln(T)}{M\Delta_i^2} \rceil$ we get from (3.21) that

$$\begin{aligned}
\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] &\leq \left\lceil M\epsilon_n + \sum_{k=1}^{M} \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)\ln(T)}{M\Delta_i^2} \right\rceil \\
&\quad + 2\sum_{k=1}^{M}\sum_{t=1}^{T} \left(\frac{\ln(t)}{\ln(1+\eta)} + \frac{\ln(1+\epsilon_n)}{\ln(1+\eta)} + 1 \right) \frac{1}{t^\gamma} \\
&\leq \left\lceil M\epsilon_n + \sum_{k=1}^{M} \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)}{M\Delta_i^2} \ln(T) \right\rceil \\
&\quad + \frac{2M}{\ln(1+\eta)} \left(\sum_{t=1}^{T} \frac{\ln((1+\epsilon_n)(1+\eta))}{t^\gamma} + \sum_{t=4}^{T} \frac{\ln(t)}{t^\gamma} + 1 \right) \\
&\leq \left\lceil M\epsilon_n + \sum_{k=1}^{M} \frac{8\sigma_g^2\gamma(1+\epsilon_c^k)}{M\Delta_i^2} \ln(T) \right\rceil \\
&\quad + \frac{2M}{\ln(1+\eta)} \left(\frac{1}{(\gamma-1)^2} + \frac{\ln((1+\epsilon_n)(1+\eta))}{\gamma-1} + 2 \right).
\end{aligned}$$

This establishes the proof.

□

3.2.2 The coop-UCB2 Algorithm

The coop-UCB2 algorithm is initialized by each agent sampling each arm once and proceeds as follows. At time t each agent k selects the arm with maximum $Q_i^k(t-1) = \hat{\mu}_i^k(t-1) + C_i^k(t-1)$, where

$$C_i^k(t-1) = \sigma_g \sqrt{\frac{2\gamma}{G(\eta)} \cdot \frac{\hat{n}_i^k(t-1) + f(t-1)}{M\hat{n}_i^k(t-1)} \cdot \frac{\ln(t-1)}{\hat{n}_i^k(t-1)}} \tag{3.20}$$

for sub-Gaussian rewards. In the above $f(t)$ is an increasing sublogarthmic function, $\gamma > 1$, $\eta \in (0, 4)$, and $G(\eta) = 1 - \eta^2/16$.

Then, at each time t, each agent k updates its cooperative estimate of the mean reward at each arm using the distributed cooperative estimation algorithm described in (3.1–3.3). Note that the heuristic Q_i^k requires the agent k to know the total number

of agents M, but not the global graph structure. That is, unlike coop-UCB1, agent k doesn't not need to know ϵ_c^k.

Theorem 3 (***Regret of the coop-UCB2 Algorithm***)**.** *For the coop-UCB2 algorithm and the cooperative MAB problem with sub-Gaussian rewards, the number of times a suboptimal arm i is selected by all agents until time T satisfies*

$$\sum_{k=1}^{M}\mathbb{E}[n_i^k(T)] \leq \left(\frac{4\sigma_g^2\gamma}{\Delta_i^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_i^2 M G(\eta)}{2\sigma_g^2\gamma}\frac{f(T)}{\ln T}}\right)\right)\ln T + M\epsilon_n + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right),$$

where $t_k^\dagger = f^{-1}(\epsilon_c^k)$.

Proof: We proceed similarly to [5]. The number of selections of a suboptimal arm i by all agents until time T is

$$\begin{aligned}\sum_{k=1}^{M} n_i^k(T) &\leq \sum_{k=1}^{M}\sum_{t=1}^{T}\mathbb{1}(Q_i^k(t-1) \geq Q_{i^*}^k(t-1)) \\ &\leq A + \sum_{k=1}^{M}\sum_{t=1}^{T}\mathbb{1}(Q_i^k(t-1) \geq Q_{i^*}^k(t-1), M n_i^{\text{cent}} \geq \eta),\end{aligned} \tag{3.21}$$

where $A > 0$ is a constant that will be chosen later.

At a given time t an individual agent k will choose a suboptimal arm only if $Q_i^k(t-1) \geq Q_{i^*}^k(t-1)$. For this condition to be true at least one of the following three conditions must hold:

$$\hat{\mu}_{i^*}(t-1) \leq m_{i^*} - C_{i^*}^k(t-1) \tag{3.22}$$

$$\hat{\mu}_i(t-1) \geq m_i + C_i^k(t-1) \tag{3.23}$$

$$m_{i^*} < m_i + 2C_i^k(t-1). \tag{3.24}$$

We now bound the probability that (3.22) holds using Theorem 1:

$$\begin{aligned}
&\mathbb{P}\left((3.22) \text{ holds } | \, t \geq t_k^\dagger\right) \\
&\quad = \mathbb{P}\left(\frac{\hat{s}_i^k - m_i \hat{n}_i^k}{\sqrt{\frac{1}{M}\left(\hat{n}_i^k(t) + f(t)\right)}} \geq \sigma_g \sqrt{\frac{2\gamma \ln(t)}{G(\eta)}} \,\middle|\, t \geq t_k^\dagger\right) \\
&\quad \leq \mathbb{P}\left(\frac{\hat{s}_i^k - m_i \hat{n}_i^k}{\sqrt{\frac{1}{M}\left(\hat{n}_i^k(t) + \epsilon_c^k\right)}} \geq \sigma_g \sqrt{\frac{2\gamma \ln(t)}{G(\eta)}} \,\middle|\, t \geq t_k^\dagger\right) \\
&\quad \leq \left(\frac{\ln(t)}{\ln(1+\eta)} + \frac{\ln(1+\epsilon_n)}{\ln(1+\eta)} + 1\right) \frac{1}{t^\gamma}.
\end{aligned}$$

It also follows analogously that

$$\mathbb{P}\left((3.22) \text{ holds } | t \geq t_k^\dagger\right) \leq \left(\frac{\ln(t)}{\ln(1+\eta)} + \frac{\ln(1+\epsilon_n)}{\ln(1+\eta)} + 1\right) \frac{1}{t^\gamma}.$$

We now examine the event (3.24).

$$\begin{aligned}
& m_{i^*} < m_i + 2C_i^k(t) \\
\implies \; & \hat{n}_i^k(t)^2 \frac{\Delta_i^2 M G(\eta)}{8\sigma_g^2} - \gamma \hat{n}_i^k(t) \ln(t) - \gamma f(t) \ln(t) < 0. \qquad (3.25)
\end{aligned}$$

The quadratic equation (3.25) can be solved to find the roots, and if $\hat{n}_i(t)$ is greater than the larger root the inequality will never hold. Solving the quadratic equation (3.25), we obtain that event (3.24) does not hold if

$$\begin{aligned}
\hat{n}_i^k(t) &\geq \frac{4\sigma_g^2 \gamma \ln(t)}{\Delta_i^2 M G(\eta)} + \sqrt{\left(\frac{4\gamma\sigma_g^2 \ln(t)}{\Delta_i^2 M G(\eta)}\right)^2 + \frac{8\sigma_g^2 f(t) \gamma \ln(t)}{\Delta_i^2 M G(\eta)}} \\
&= \frac{4\sigma_g^2 \gamma \ln t}{\Delta_i^2 M G(\eta)} \left(1 + \sqrt{1 + \frac{\Delta_i^2 M G(\eta)}{2\sigma_g^2 \gamma} \frac{f(t)}{\ln t}}\right).
\end{aligned}$$

Now, we set $A = \left\lceil M\epsilon_n + \frac{4\sigma_g^2\gamma \ln T}{\Delta_i^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_i^2 MG(\eta)}{2\gamma\sigma_g^2}\frac{f(T)}{\ln T}}\right)\right\rceil$. It follows from monotonicity of $f(t)$ and $\ln(t)$ and statement (i) of Proposition 1 that event (3.24) does not hold if $Mn_i^{\text{cent}}(t) > A$.

Therefore, from (3.21) we see that

$$\begin{aligned}
\sum_{k=1}^{M} n_i^k(T) &\leq A + 2\sum_{k=1}^{M}\sum_{t=1}^{t_k^\dagger - 1} 1 + \frac{2}{\ln(1+\eta)}\sum_{k=1}^{M}\sum_{t=t_k^\dagger}^{T}\left(\frac{\ln(t)}{t^\gamma} + \frac{\ln((1+\epsilon_n)(1+\eta))}{t^\gamma}\right) \\
&\leq A + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\sum_{t=1}^{T}\left(\frac{\ln(t)}{t^\gamma} + \frac{\ln((1+\epsilon_n)(1+\eta))}{t^\gamma} + 4\right) \\
&\leq A + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2} + \frac{1}{\gamma-1} + 5\right),
\end{aligned}$$

completing the theorem. □

Remark 1 (***Coop-UCB2 for Gaussian Rewards***)**.** For the case of Gaussian rewards with known variance σ^2 the $C_i^k(t)$ term of coop-UCB2 reduces to

$$C_i^k(t) = \sigma\sqrt{\frac{2\gamma}{G(\eta)}\cdot\frac{\hat{n}_i^k(t) + f(t)}{M\hat{n}_i^k(t)}\cdot\frac{\ln(t)}{\hat{n}_i^k(t)}}$$

as the cumulant generating function for a Gaussian random variable $X \sim \mathcal{N}(0,\sigma)$ can be bounded as $\phi_X(\beta) \leq \frac{\sigma^2\beta^2}{2}$. In this case, the result of Theorem 3 gives

$$\begin{aligned}
\sum_{k=1}^{M}\mathbb{E}[n_i^k(T)] \leq &\left(\frac{4\sigma^2\gamma}{\Delta_i^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_i^2 MG(\eta)}{2\sigma^2\gamma}\frac{f(T)}{\ln T}}\right)\right)\ln T \\
&+ M\epsilon_n + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2} + \frac{1}{\gamma-1} + 5\right).
\end{aligned}$$

Remark 2 (***Coop-UCB2 for Bounded Rewards***)**.** For the case of bounded rewards assumed without loss of generality to be in $[0,1]$ the $C_i^k(t)$ term of coop-UCB2

reduces to

$$C_i^k(t) = \sqrt{\frac{\gamma}{2G(\eta)} \cdot \frac{\hat{n}_i^k(t) + f(t)}{M\hat{n}_i^k(t)} \cdot \frac{\ln(t)}{\hat{n}_i^k(t)}}$$

as the cumulant generating function for a random variable X with range $[0,1]$ can be bounded as $\phi_X(\beta) \leq m_i + \frac{\beta^2}{8}$. In this case, the result of Theorem 3 gives

$$\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] \leq \left(\frac{\gamma}{\Delta_i^2 G(\eta)}\left(1+\sqrt{1+\frac{2\Delta_i^2 MG(\eta)}{\gamma}\frac{f(T)}{\ln T}}\right)\right)\ln T + M\epsilon_n + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right).$$

Remark 3 (***Asymptotic Regret for coop-UCB2***)**.** In the limit $t \to +\infty$, $\frac{f(t)}{\ln(t)} \to 0^+$, $\eta \to 0$, and

$$\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] \leq \left(\frac{8\sigma_g^2\gamma}{\Delta_i^2} + o(1)\right)\ln T$$

for sub-Gaussian rewards and

$$\sum_{k=1}^{M} \mathbb{E}[n_i^k(T)] \leq \left(\frac{8\sigma^2\gamma}{\Delta_i^2} + o(1)\right)\ln T$$

for Gaussian rewards.

We thus recover the upper bound on regret for a centralized agent as given in (2.8) within a constant factor.

□

Remark 4 (***Performance of Individual Agents***)**.** Theorem 3 provides bounds on the performance of the group when using coop-UCB2 as a function of the graph structure, and the logarithmic portion of the bound is independent of agent location. However, the constant factor is dependent on the agent's position in the network since it depends on ϵ_c^k. In this sense, ϵ_c^k can be thought of as a measure of "explore-exploit"

centrality, which indicates that agents with a higher ϵ_c^k will contribute more to the group's regret.

The observation that agents with higher ϵ_c^k contribute more to the group's regret when using coop-UCB2 can also be seen when agents employ coop-UCB1 as demonstrated in the performance bounds provided in Theorem 2. However, agents using the coop-UCB1 algorithm know their own value of ϵ_c^k and use it as part of the decision-making process. As ϵ_c^k is part of the coop-UCB1 algorithm, it is expected that the performance bounds are dependent upon ϵ_c^k. However, it is noteworthy that the same ordering of agent performance is predicted by ϵ_c^k in coop-UCB2, where each agent k does not know ϵ_c^k.

□

3.3 Bayesian Cooperative Decision-Making

In this section, we extend the coop-UCB2 algorithm to a Bayesian setting and develop the coop-UCL algorithm. The Bayesian setting allows us to model arms with correlated rewards and incorporate a priori knowledge about reward and correlation structure in the Bayesian prior. We first recall the UCL algorithm proposed in [47, 84] and extend it to the cooperative setting. We then analyze the performance of this algorithm for an uninformative prior.

3.3.1 The UCL Algorithm

The UCL algorithm developed in [84] applies the approach of Bayes-UCB [47] to MAB problems with correlated Gaussian rewards. The UCL algorithm at each time computes the posterior distribution of mean rewards at each option and then computes the $(1 - 1/Kt^a)$ upper-credible-limit for each arm, i.e., an upper bound that holds with probability $(1 - 1/Kt^a)$ where $K = \sqrt{2\pi e}$, $\gamma > 1$, and $a = 4/3\gamma$. The

algorithm chooses the arm with highest upper credible limit. For Gaussian rewards, the $(1 - 1/Kt^a)$ upper-credible-limit can be written as

$$Q_i(t) = \nu_i(t) + \sigma_i(t)\Phi^{-1}(1 - 1/Kt^a), \tag{3.26}$$

where $\nu_i(t)$ is the posterior mean and $\sigma_i(t)$ the posterior standard deviation of mean reward at time t. $\Phi^{-1}(\cdot)$ is the standard Gaussian inverse cumulative distribution function.

Let the prior on rewards from each arm be multivariate Gaussian with mean vector $\boldsymbol{\nu}_0 \in \mathbb{R}^N$ and covariance matrix $\Sigma_0 \in \mathbb{R}^{N\times N}$. Then, the posterior mean and covariance of mean reward at time t can be computed using the following recursive update rule [48]:

$$\begin{aligned}
\mathbf{q}(t) &= \frac{r(t)\boldsymbol{\phi}(t)}{\sigma_s^2} + \Lambda(t-1)\boldsymbol{\nu}(t-1) \\
\Lambda(t) &= \frac{\boldsymbol{\phi}(t)^\top \boldsymbol{\phi}(t)}{\sigma_s^2} + \Lambda(t-1), \quad \Sigma(t) = \Lambda(t)^{-1} \\
\boldsymbol{\nu}(t) &= \Sigma(t)\boldsymbol{q}(t),
\end{aligned} \tag{3.27}$$

where $\boldsymbol{\phi}(t)$ and $\boldsymbol{\nu(t)}$ are column vectors of $\phi_i(t)$ and $\nu_i(t)$, respectively, and $\phi_i(t)$ is the indicator function of selecting arm i at time t. The update equation (3.27) can be reduced to

$$\begin{aligned}
\boldsymbol{\nu}(t) &= (\Lambda_0 + \Gamma(t)^{-1})^{-1}(\Gamma(t)^{-1}\boldsymbol{\mu}(t) + \Lambda_0\boldsymbol{\nu}_0) \\
\Lambda(t) &= \Lambda_0 + \Gamma(t)^{-1}, \quad \Sigma(t) = (\Lambda(t))^{-1},
\end{aligned} \tag{3.28}$$

where $\Lambda_0 = \Sigma_0^{-1}$, $\Gamma(t)$ is a diagonal matrix with entries $\frac{\sigma_s^2}{n_i(t)}$, and $\boldsymbol{\mu}(t)$ is the vector of $\mu_i(t)$, which is the empirical mean of rewards from arm $i \in \{1, \ldots, N\}$ until time t. Note that diagonal entries of $\Sigma(t)$ are $(\sigma_i(t))^2$, $i \in \{1, \ldots, N\}$.

3.3.2 The coop-UCL Algorithm

We now extend the coop-UCB2 algorithm to the Bayesian setting with Gaussian rewards and propose the coop-UCL algorithm. In the coop-UCL algorithm, each agent first computes an approximate posterior distribution of mean rewards conditioned on rewards obtained by all the agents. To this end, each agent uses the approximate frequentist estimator $\hat{\mu}_i^k$ from Section 3.1 in update equation (3.28).

Let the prior of agent k be a multivariate Gaussian distribution with mean $\boldsymbol{\nu}_0^k$ and covariance Σ_0^k. Let $\hat{\Sigma}^k(t)$ and $\hat{\boldsymbol{\nu}}^k(t)$ be the estimated covariance matrix and posterior mean at time t, respectively. Then, the coop-UCL algorithm performs cooperative approximate Bayesian estimation:

$$\begin{aligned} \hat{\boldsymbol{\nu}}^k(t) &= (\Lambda_0^k + \Gamma^k(t)^{-1})^{-1}(\Gamma^k(t)^{-1}\hat{\boldsymbol{\mu}}^k(t) + \Lambda_0^k\boldsymbol{\nu}_0^k) \\ \hat{\Lambda}^k(t) &= \Lambda_0^k + \Gamma^k(t)^{-1}, \quad \hat{\Sigma}^k(t) = (\hat{\Lambda}^k(t))^{-1}, \end{aligned} \tag{3.29}$$

where $\Gamma^k(t)$ is a diagonal matrix with diagonal entries $\sigma_s^2/M\hat{n}_i^k(t),\ i \in \{1,\ldots,N\}$, and $\Lambda_0^k = (\Sigma_0^k)^{-1}$.

After computing $\hat{\boldsymbol{\nu}}^k(t-1)$ and $\hat{\Sigma}^k(t-1)$, the coop-UCL algorithm at time t requires each agent k to choose the option with maximum $(1-\alpha(t))$-upper-credible-limit given by

$$Q_i^k(t-1) = \hat{\nu}_i^k(t-1) + \hat{\sigma}_i^k(t-1)\Phi^{-1}(1-\alpha(t-1)), \tag{3.30}$$

where $\alpha(t)$ is defined such that

$$\Phi^{-1}(1-\alpha(t)) = \sqrt{\frac{\hat{n}_i^k(t)+f(t)}{G(\eta)\hat{n}_i^k(t)}}\Phi^{-1}\Big(1-\frac{1}{Kt^a}\Big),$$

where $\hat{\nu}_i^k(t)$ is the i-th entry of $\hat{\boldsymbol{\nu}}^k(t)$, $(\hat{\sigma}_i^k(t))^2$ is the i-th diagonal entry of $\hat{\Sigma}^k(t)$, $K = \sqrt{2\pi e}$, $\gamma > 1$, and $a = 4/3\gamma$.

Regret Analysis of the coop-UCL Algorithm

We now derive bounds on the expected cumulative regret for each agent using the coop-UCL algorithm with uninformative priors for each agent. For an uninformative prior, $\Lambda_0^k = 0$, for each $k \in \{1, \ldots, M\}$, and consequently, $\hat{\nu}^k(t) = \hat{\mu}^k(t)$ and $\hat{\Sigma}^k(t) = \Gamma^k(t)$. In addition, we first present a bound on $\Phi^{-1}(\cdot)$.

Lemma 1 (***Inverse Gaussian CDF Bounds***)**.** *For the standard normal random variable z and the associated inverse cumulative distribution function $\Phi^{-1}(\cdot)$, the following holds for any $\alpha \in [0, 0.5]$, $t \in \mathbb{N}$ and $a > 1$:*

$$\Phi^{-1}(1-\alpha) \leq \sqrt{-2\log(\alpha)}$$

$$\Phi^{-1}(1-\alpha) > \sqrt{-\log(2\pi\alpha^2(1-\log(2\pi\alpha^2)))}$$

$$\Phi^{-1}\Big(1 - \frac{1}{\sqrt{2\pi}et^a}\Big) > \sqrt{\nu \log t^a},$$

for $0 < \nu \leq 1.59$.

Proof: The first inequality can be found in [1]. The second inequality was established in [84]. To establish the last inequality, it suffices to show that

$$-\log\left(\frac{1}{et^2}\left(1-\log\left(\frac{1}{et^2}\right)\right)\right) - \nu\log t \geq 0,$$

for $0 < \nu \leq 1.59$. The left hand side of the above inequality is

$$g(t) := 1 - \log 2 + (2-\nu)\log t - \log(1+\log t).$$

It can be verified that g admits a unique minimum at $t = e^{(\nu-1)/(2-\nu)}$ and the minimum value is $\nu - \log 2 + \log(2-\nu)$, which is positive for $0 < \nu \leq 1.59$. □

Theorem 4 (***Regret of the coop-UCL Algorithm***). *For the Gaussian MAB problem and the coop-UCL algorithm with uninformative priors for each agent, the number of times a suboptimal arm i is selected by all agents until time T satisfies*

$$\sum_{k=1}^{M}\mathbb{E}[n_i^k(T)] \leq \left(\frac{4a\sigma_s^2}{\Delta_i^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_i^2 MG(\eta)}{2a\sigma_s^2}\frac{f(T)}{\ln KT}}\right)\right)\ln KT + 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{2M}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right)+M\epsilon_n$$

where $t_k^\dagger = f^{-1}(\epsilon_c^k)$ and $sigma_s$ is the sample standard deviation.

Proof: For uninformative priors, coop-UCL is analogous to coop-UCB2 with $C_i^k(t) = \hat{\sigma}_i^k(t)\Phi^{-1}(1-\alpha(t))$. Similar to the proof of Theorem 3, we first note that for (3.22) simple manipulations lead to

$$\begin{aligned}\frac{\hat{s}_{i^*}^k(t) - m_{i^*}\hat{n}_{i^*}^k(t)}{\sqrt{\hat{n}_{i^*}^k(t)}} &\geq \frac{\sigma_s}{\sqrt{G(\eta)}}\Phi^{-1}\left(1-\frac{1}{Kt^a}\right)\\ &> \frac{\sigma_s}{\sqrt{G(\eta)}}\sqrt{\frac{3a}{2}\ln t} \qquad (3.31)\\ &= \frac{\sigma_s}{\sqrt{G(\eta)}}\sqrt{2\ln t^\gamma}\end{aligned}$$

where (3.31) follows from Lemma 1 for $K = \sqrt{2\pi e}$.

We now use Theorem 1 adapted to Gaussian rewards, giving

$$\begin{aligned}&\mathbb{P}\left((3.22)\text{ holds } | \, t \geq t_k^\dagger\right)\\ &\quad\leq \mathbb{P}\left(\frac{\hat{s}_{i^*}^k(t) - m_i\hat{n}_{i^*}^k(t)}{\sqrt{\frac{1}{M}\left(\hat{n}_{i^*}^k(t)+f(t)\right)}} \geq \sigma_s\sqrt{\frac{2\gamma\ln(t)}{G(\eta)}}\;\middle|\; t\geq t_k^\dagger\right)\\ &\quad\leq \left(\frac{\ln(t)}{\ln(1+\eta)}+\frac{\ln(1+\epsilon_n)}{\ln(1+\eta)}+1\right)\frac{1}{t^\gamma}\end{aligned}$$

resulting in sub-logarithmic regret as in Theorem 3.

We now examine the event (3.24). Following the argument in the proof of Theorem 3 and using the upper bound on $\Phi^{-1}(\cdot)$ from Lemma 1, we obtain that the event (3.24) does not hold if

$$\hat{n}_i^k(t) \geq \frac{4\sigma_s^2 a \ln Kt}{\Delta_i^2 MG(\eta)} \left(1 + \sqrt{1 + \frac{\Delta_i^2 MG(\eta)}{2\sigma_s^2 a} \frac{f(t)}{\ln Kt}}\right).$$

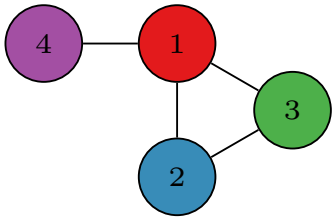
We set $A = \left\lceil M\epsilon_n + \frac{4\sigma_s^2 a \ln KT}{\Delta_i^2 MG(\eta)} \left(1 + \sqrt{1 + \frac{\Delta_i^2 MG(\eta)}{2\sigma_s^2 a} \frac{f(T)}{\ln KT}}\right)\right\rceil$ and the theorem follows by proceeding similarly to the proof of Theorem 3. □

3.4 Numerical Illustrations

In this section, we elucidate our theoretical analyses from the previous sections with numerical examples. We first compare the performance of the coop-UCB1, coop-UCB2, and coop-UCL algorithms. We then provide examples in which the ordering of the performance of nodes obtained through numerical simulations is equal to the ordering predicted by the explore-exploit centrality measure: the larger the ϵ_c^k the lower the performance. Finally, we provide examples in which the performance of a network of agents as a whole is equal to the network's value of ϵ_n.

Unless otherwise noted in the simulations we consider a 10-arm bandit problem with mean rewards drawn from a normal random distribution for each Monte-Carlo run with mean 0 and standard deviation 10. The sampling standard deviation is $\sigma_s = 30$ and the results displayed are the average of 10^6 Monte-Carlo runs. These parameters were selected to give illustrative results within the displayed time horizon, but the relevant conclusions hold across a wide variation of parameters. The simulations used $f(t) = \sqrt{\ln t}$, and consensus matrix P as in (2.14) with $\kappa = \frac{d_{\max}}{d_{\max}-1}$.

3.4.1 Comparing Multi-agent MAB Algorithms



Agent	Degree	ϵ_c^k
1	3	0
2	2	2.31
3	2	2.31
4	1	5.41

Table 3.1: Fixed network used in Examples 1 and 2.

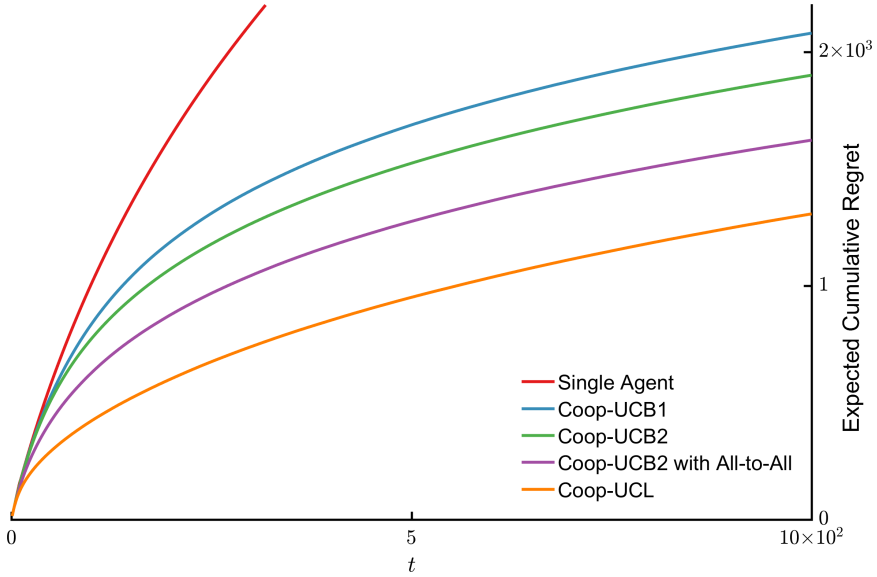


Figure 3.2: Simulation results of expected cumulative regret for several different MAB algorithms using the fixed network shown in Table 3.1.

Example 1. Figure 3.2 demonstrates the relative performance between coop-UCB1, coop-UCB2, and coop-UCL with the same run parameters and the communication graph depicted in Table 3.1, as well as coop-UCB2 with all-to-all communication and single-agent-UCB. For the multi-agent algorithms the regret shown is the average per agent. Here the coop-UCL algorithm is shown with an informative prior and no correlation structure. Each agent in the coop-UCL simulation shown here has $\Sigma_0 = 625 \cdot \mathcal{I}_M$ and $\boldsymbol{\nu}_0 = 0 \cdot \mathbf{1}_M$.

Note that coop-UCB2 outperforms coop-UCB1, as is predicted by our analytical results, and the cooperative algorithms significantly outperform the single-agent

case. Furthermore, the use of informative priors markedly improves performance for coop-UCL, implying that prior information, when available, is highly beneficial in cooperative explore-exploit tasks, just as it was shown to be in single-agent explore-exploit tasks [84].

3.4.2 Comparing Performance between Agents using ϵ_c^k (Explore-Exploit Centrality)

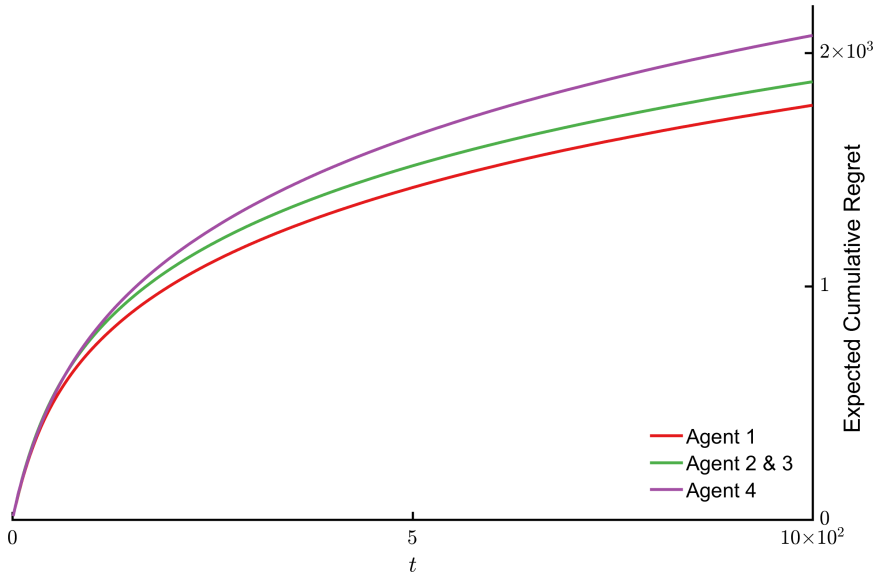
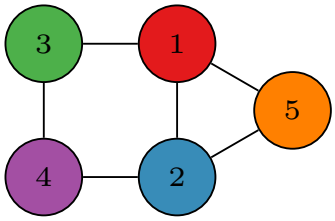


Figure 3.3: Simulation results comparing expected cumulative regret for different agents in the fixed network shown in Table 3.1. Note that agents 1 and 2 have nearly identical expected regret.

Example 2. Figure 3.3 demonstrates the relative performance between agents using coop-UCB2 with the underlying graph structure in Table 3.1. The values of ϵ_c^k for each node are also given in Table 3.1. As predicted by Theorem 3 (Remark 4), agent 1 should have the lowest regret, agents 2 and 3 should have equal and intermediate regret, and agent 4 should have the highest regret as this is their ordering with respect to ϵ_c^k. These predictions are validated in our simulations shown here.

Example 3. Figure 3.5 demonstrates the relative performance between agents using coop-UCB2 with the underlying graph structure in Table 3.2 and where rewards are



Agent	Degree	Info. Cent.	ϵ_c^k
1	3	.35	1.4
2	3	.35	1.4
3	2	.28	3.4
4	2	.28	3.4
5	2	.27	2.9

Table 3.2: Fixed network used in Example 3 and several centrality indices.

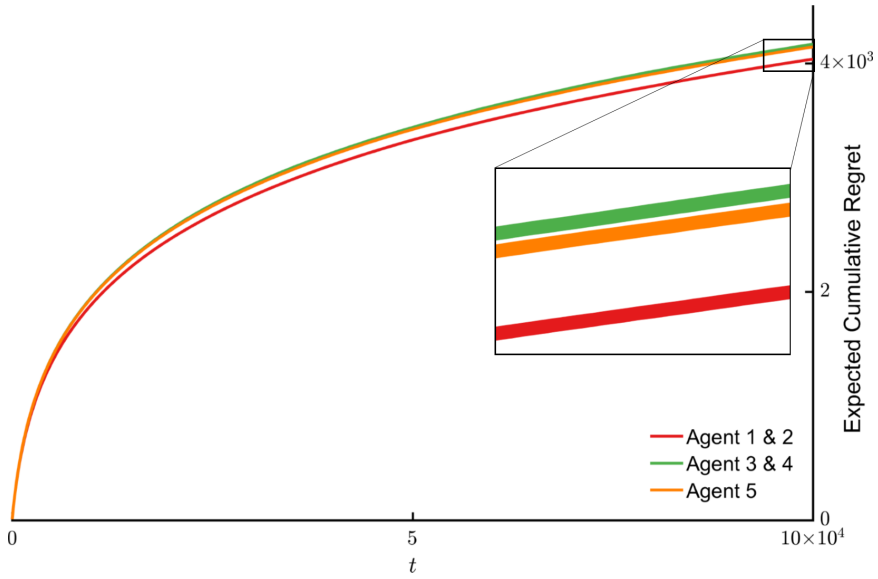


Figure 3.5: Simulation results of expected cumulative regret for each agent using coop-UCB2 in the house graph [78] shown in Table 3.3.

drawn from a normal distribution with mean 0 and standard deviation 5. The values of ϵ_c^k for each node are also given in Table 3.2, along with the values of degree and information centrality for each node [78], for comparison. Here degree centrality is defined as the number of neighbors. Information centrality is defined in Stephenson and Zelen [98] but in broad terms it is a measure of the "effective resistance" between nodes.

Note that, unlike in Figure 3.3, degree does not distinguish agent 5 from agents 3 and 4, whereas ϵ_c^k (and information centrality) does. Further, according to information centrality, which is larger the more central the node, node 5 is less information central than nodes 3 and 4. In contrast, according to ϵ_c^k, which is smaller the more central the node, node 5 is more explore-exploit central than nodes 3 and 4.

As in the prior example and predicted by Theorem 3 (Remark 4), relative agent performance is consistent with the relative values of ϵ_c^k, with lower values of ϵ_c^k corresponding to lower regret. That is, the ordering of nodes by performance is predicted by the ordering of nodes by our new notion of explore-exploit centrality, ϵ_c^k, and not by the ordering of nodes by degree or information centrality.

We note that we have found some parameter regimes, specifically for rewards that are far apart in mean value, where information centrality does give the correct ordering of performance, rather than ϵ_c^k. It is possible that this is due to sensitivity of performance to the Δ_i. However, we have observed that ϵ_c^k is broadly predictive of performance for a variety of regimes and graphs, as further demonstrated in Example 4.

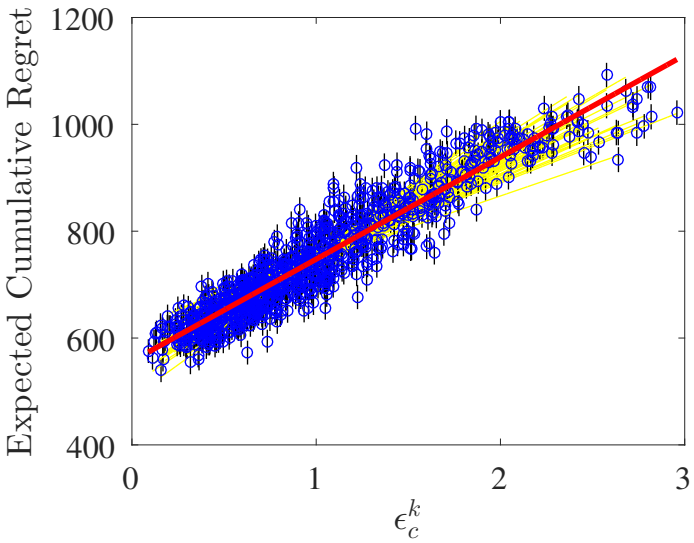


Figure 3.6: Simulation results of expected cumulative regret as a function of normalized ϵ_c^k for nodes in ER graphs at $T = 500$. Also shown in red is the best linear fit.

Example 4. We now explore the effect of ϵ_c^k on the performance of an agent in an Erdös-Réyni (ER) random graph. ER graphs are a widely used class of random graphs where any two agents are connected with a given probability ρ [7]. Consider a set of 10 agents communicating according to an ER graph and using the coop-UCB2 algorithm. In our simulations, we consider 100 connected ER graphs, and for each ER

graph we compute the expected cumulative regret of agents using 1000 Monte-Carlo simulations with $\rho = \ln(10)/10$. We show the behavior of the expected cumulative regret of each agent as a function of the normalized ϵ_c^k in Fig. 3.6. It is evident that increased ϵ_c^k results in a sharp decrease in performance. Conversely, low ϵ_c^k is indicative of better performance.

3.4.3 Comparing Performance between Graphs using ϵ_n

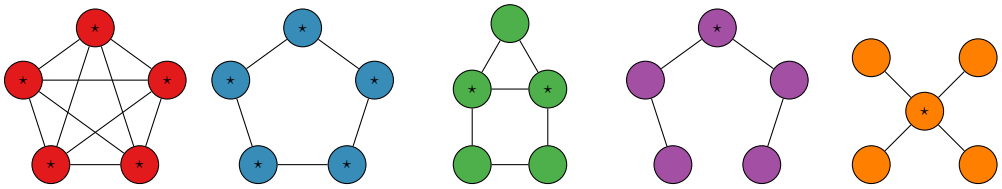


Table 3.3: Fixed networks used in Examples 5 and 6 arranged in order of increasing value of ϵ_n. Values of ϵ_n are calculated using P as in (2.14) and $\kappa = 0.02$. A $\star$ indicates that this is the best performing agent(s) in the graph as determined in the simulations described in Example 5, and the regret for this agent(s) is discussed in Example 6. Note that each agent uses the sample algorithm.

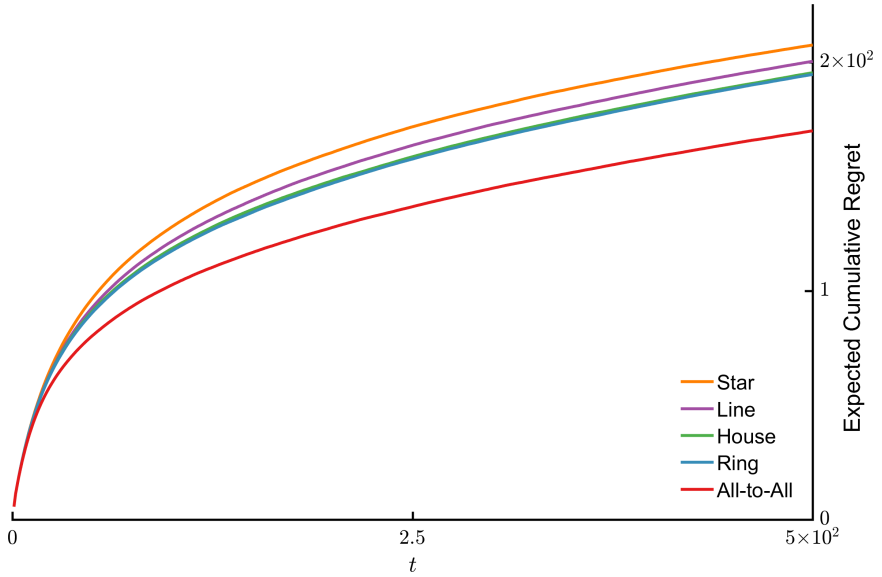


Figure 3.8: Simulation results of expected cumulative regret for the group using coop-UCB2 using each of the fixed graphs shown in Table 3.3.

Example 5. Figure 3.8 compares the expected cumulative regret averaged over all agents in each of the five graphs in Table 3.3, where agents use coop-UCB2. The value of ϵ_n is shown in Table 3.3 for each graph. Theorem 3 predicts that graphs with lower ϵ_n will have lower average expected cumulative regret. Here we use two arms and $\kappa = 0.02$. Figure 3.8 verifies this prediction, showing ordering of graphs by performance is equal to the order by ϵ_n.

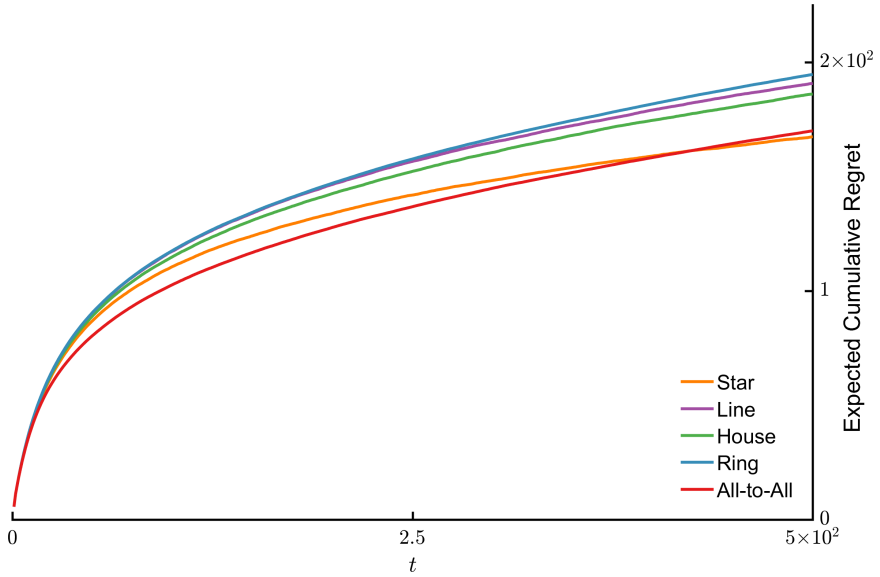


Figure 3.9: Simulation results of expected cumulative regret using coop-UCB2 for the agent with lowest regret in each of of the fixed graphs shown in Table 3.3.

Example 6. Figure 3.9 compares the expected cumulative regret for the best performing agent(s) in each of the five node graphs in Table 3.3. Here we use two arms and $\kappa = 0.02$. Note that the central agent in the star graph outperforms the best agent in the all-to-all graph despite the star graph's poor group performance. This indicates that the four peripheral agents are doing most of the exploration. The stark different in the propensity to explore or exploit between the central and peripheral agents in the star graph demonstrates that regret accumulation for certain agents could be controlled by designing the appropriate communication graph structure.

3.5 Robotic Implementation

In this section we build on our previous analytical results and numerical examples to conduct three experiments that demonstrate the utility of our new multi-agent MAB algorithms in robotic search tasks. We consider two wheeled robots that can traverse a space and sample from a virtual reward field that is represented by a visual light field.

3.5.1 Experimental Setup

Three experiments were conducted in the Council of Science and Technology's StudioLab located in the basement of Fine Hall at Princeton University. We utilized the VICON motion capture area in the StudioLab, which covers an open floor space measuring approximately 5×8 meters. The VICON camera system uses multiple cameras mounted on the ceiling to detect, in real time, the position of objects in the room. We use the positional information from this system to control two wheeled robots through feedback.

We used two iRobot Create robots, which are essentially Roomba robotic vacuum cleaners without the vacuum. We controlled the forward and turning velocity of each robot from a central computer through bluetooth and calculated control commands with MATLAB. Control commands were calculated using a proportional controller that turned the robot to face the target location and then drive forwards towards it. The controller also utilized a local collision avoidance algorithm to avoid collisions within short distances. This collision avoidance algorithm dictated that a robot would turn away and travel a short distance if the other robot entered a certain short zone in front of it.

In the VICON motion capture area we installed an array of lights on the ceiling which produced 20 colored spots on the floor in a 4×5 grid. Each light spot represents

an arm in the MAB problem, and the light color represents the arm mean. A frame from a video of one experiment is shown in Figure 3.10 which shows the light field and the light color key which matches light colors to arm means. Note that arm mean values range from 20 to 85. When a robot selects and visits a light spot, it receives a reward drawn from a Gaussian distribution with the associated mean and sample standard deviation 25.

Each robot's goal is to maximize the cumulative reward received, which corresponds to selecting arms with the greatest intensity of red light. The robots utilize the coop-UCB2 algorithm to accomplish this task. Specifically, after an initialization phase where each root visits each arm once, at each timestep t each robot selects an arm using the coop-UCB2 algorithm and traverses to that arm. Upon arriving at the selected arm, the robot receives a virtual reward corresponding to the arm mean and some noise. Depending on the experiment in question the robot may then share information with the other robot using consensus. In our three experiments we individually consider the three cases where the robots cannot communicate with each other, they both can communicate, and only one can communicate, respectively. Additionally, note that the reward received is virtual and is merely represented visually by the light color. We extend on our work here in Section 4.4.1 where we conduct an experiment where the robot actually measures the light intensity.

3.5.2 Robotic Experiments

Example 7. In this example two robots use the coop-UCB2 algorithm without any communication between agents, which is equivalent to the UCB algorithm from [5] for Gaussian rewards. This is demonstrated in Video 1, also available online at `youtu.be/INzy1zeGlis`. Figure 3.10 shows a frame from the end of the video where one can see the cumulative regret for each robot in the top panel in blue as the "No

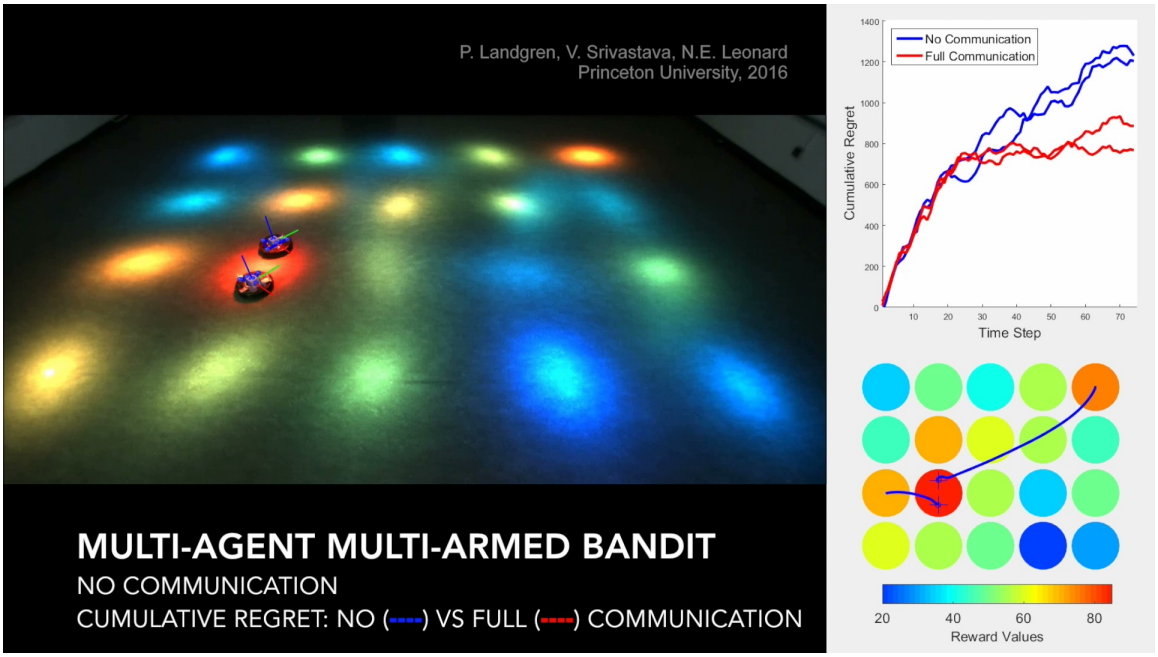


Figure 3.10: A frame from near the end of Video 1. The panel on the left shows a camera view of the experiment space with the 20 light spots, each representing an arm, arranged in a grid as well as the two robots. The robot's locations are also shown in the panel on the bottom right, along with a color key that shows how light colors correspond to reward values. The panel in the upper right shows the cumulative regret of each of the two robots in the video ("No Communication") as well as two robots from another run that could communicate their estimates.

Communication" case. Note that the case with "Full Communication," discussed next, significantly outperforms the "No Communication" case.

Example 8. In this example two robots use the coop-UCB2 algorithm with communication between both agents. As there are only two agents this results in both robots having the exact same estimates of the arm means at each time, and therefore they make the exact same decisions and go to the same arms at each time. This example is not fully depicted in a video, but the cumulative regret for each agent is shown in the top right panel of Video 1 and depicted in Figure 3.10 in red as the "Full Communication" case. As would be expected, communication significantly improves the performance of the robots.

Example 9. In this example two robots use the coop-UCB2 algorithm with a directed, i.e. uni-directional, communication graph. The green robot labeled "Commu-

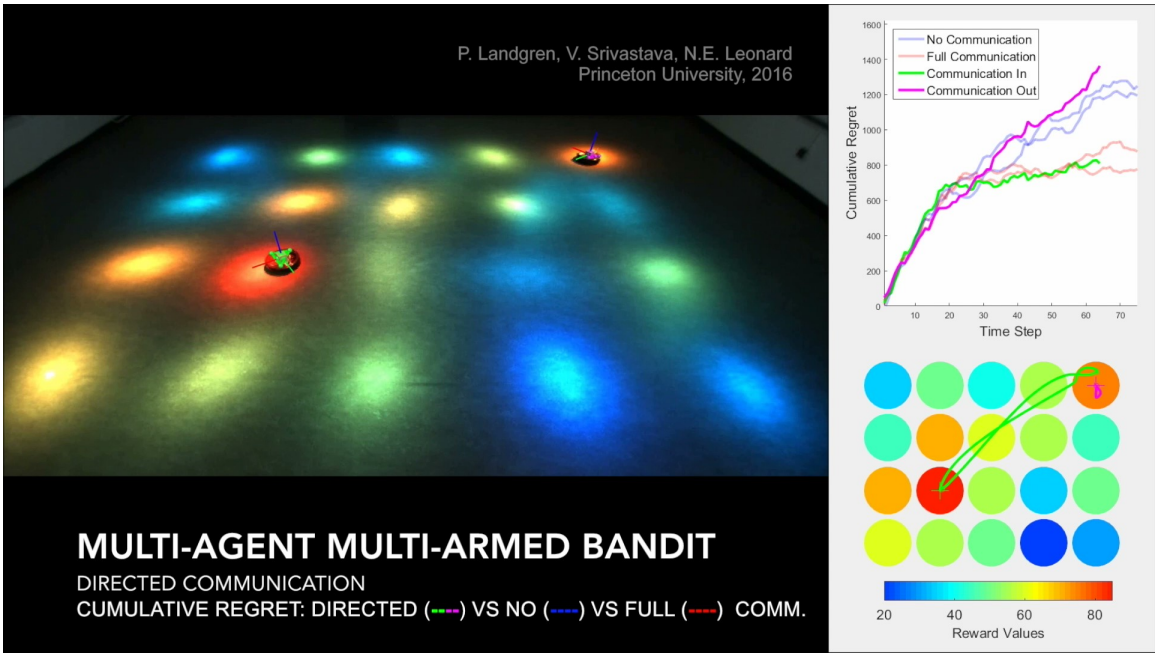


Figure 3.11: A frame from near the end of Video 2. The information corresponding to each panel is the same as in Figure 3.11.

nication In" can access the opinions of the pink robot labeled "Communication Out," but the pink robot can only access its own rewards to updates its estimates. The directed communication case goes beyond the undirected setting considered in this chapter where we proved bounds on performance for coop-UCB2. The directed communication setting is quite applicable to robotic applications where robot capabilities vary. For example, robots may carry different communication equipment that may only be able to broadcast or receive signals. This would establish a directed communication channel and this example shows that the benefits of cooperation extend to this scenario.

The results of this experiment are shown in Video 2, also available online at `youtu.be/ZZNn-ud8900`. Figure 3.11 also shows a frame from the end of the video that shows the cumulative regret for each robot in the panel at the top right. This panel also shows the cumulative regret from the robotic experiments described in Examples 7 and 8 in the background for reference.

The green robot with access to the estimates of both robots performs significantly better, and spends much more time visiting the options with the highest rewards. The pink robot with access to only its own information naturally performs much worse, and also expends much more effort moving around the space. In the end both robots settle on the best option, but the pink robot visits many more suboptimal arms. One such visit is shown in Figure 3.11.

3.6 Discussion

In this chapter we described two parallel running consensus algorithms that agents, who share estimates with neighbors defined by a communication graph, can use to estimate the mean reward at each arm. We analyzed these running consensus algorithms and gave bounds on the performance of the estimation algorithm. We also defined an "explore-exploit" centrality measure, ϵ_c^k, that is shown to be useful for predicting the performance of individuals in networked explore-exploit tasks.

We then described three algorithms for the multi-agent MAB problem with sub-Gaussian or Gaussian rewards that use the aforementioned two running consensus algorithms for estimation of mean rewards. The coop-UCB1 algorithm requires that each agent know the graph structure. In the coop-UCB2 algorithm, we relaxed this assumption and demonstrate that the coop-UCB2 algorithm improves upon coop-UCB1 also in terms of performance. The coop-UCL algorithm is a Bayesian algorithm that can utilize correlation structure in rewards or prior knowledge of reward means to improve performance. For each algorithm we proved upper bounds on the expected cumulative regret for the group and demonstrated that it is within a constant order of optimal.

Next, we elucidated our theoretical analyses with several numerical simulations. We considered fixed graphs of both four and five nodes and examined both the group

performance and performance of individual agents as predicted by their respective values of ϵ_n and ϵ_c^k. We also considered larger random graphs and examined relative performance among agents through ϵ_c^k. Finally, we demonstrated the utility of the coop-UCB2 algorithm in a multi-robot search task.

In the next chapter we adapt the coop-UCB2 algorithm to the case of multi-agent MAB with collisions.

Chapter 4

The Distributed Cooperative Upper-Confidence Bound Algorithm for MAB with Collisions [1]

In this chapter we consider the multi-agent MAB problem with collisions, where multiple agents sampling the same arm at the same time *collide*, meaning that neither agent receives a reward. The work in this chapter extends the analyses from Chapter 3. As described in Chapter 1, the MAB problem with collisions has been used by researchers to model the cognitive radio spectrum access problem and it can also be applied to many other real-world systems where agents interfere with each other through their actions. In contrast with previous work, we consider the case where agents can employ direct communication through consensus. Specifically, we assume agents can utilize the running consensus protocol given in Section 3.1. We also assume

[1] This chapter is adapted from Section V. of [55]. Sections 4.1 and 4.3 are mostly taken verbatim. Additionally, the experiment in Section 4.4 was originally presented in [58].

that agents make use of indirect communication, defined as the ability to sense when a collision has occurred.

We first define the coop-UCB2 collisions algorithm, and prove upper bounds on the expected cumulative regret. Using the bounds we demonstrate that the ability to communicate directly through consensus greatly improves performance in the MAB problem with collisions as compared to the state-of-the-art algorithms for MAB with collisions without direct communication.

We then implement a modified version of the coop-UCB2 collisions algorithm to guide decision-making for three robots in a multi-robot search task.

4.1 Problem Definition

The problem definition is identical to that in Chapter 3 except that if multiple agents sample the same arm at the same time none of the colliding agents receive a reward. Therefore, the expected cumulative regret of agent k at time T is

$$R^k(T) = \sum_{t=1}^{T} \left[m_{b^k} - \mathbb{E}\left[\sum_{i=1}^{N} r_i^k(t) \mathbb{1}\{i^k(t) = i\} \mathbb{I}_i^k(t) \right] \right] \tag{4.1}$$

where $\mathbb{I}_i^k(t) = 1$ if agent k is the only agent to sample arm i at time t, and 0 otherwise.

In this setting the optimal solution is for the M agents to each sample a different arm from among the M-best arms at each timestep. In the following we assume that each agent k has a preassigned unique rank $\omega^k \in [0, M]$ and will attempt to find the arm with the ω^k'th best reward. We also assume that no two arms have the same mean. We define agent k^i as the agent attempting to find arm i, and arm b^k as the arm with the ω^k'th best mean, which agent k is by definition trying to find. Additionally, we define $\mathcal{O}^*_{\omega^k}$ as the set of ω^k arms with the ω^k highest means and $\Delta_{\min} = \min_{i \neq j} |m_i - m_j|$.

Some authors [2, 45] have considered the case where agents that sample the same arm at the same time receive a split reward. The algorithm presented here is still appropriate for that scenario, and the regret as defined above will upper bound the regret in the case of split rewards.

Here we consider the case of sub-Gaussian rewards. We show how the coop-UCB2 can be extended to the MAB problem with collisions and we demonstrate that the performance of the group in the MAB problem with collisions is greatly enhanced through information sharing. Furthermore, we show that as in Chapter 3 ϵ_c^k is indicative of individual performance.

4.2 The coop-UCB2 Collisions Algorithm

Here we describe the coop-UCB2 collisions algorithm. This algorithm adapts SL(K) algorithm in [30] to the case where agents can directly communicate through consensus.

Let each agent k have an associated rank $\omega^k \in \{1, \dots, M\}$, and let no two agents have the same rank. The coop-UCB2 collisions algorithm for sub-Gaussian rewards is initialized by each agent sampling each arm once and proceeds as follows. At time t each agent k constructs the set $\mathcal{O}_{\omega^k}(t)$ such that $\mathcal{O}_{\omega^k}(t)$ contains the ω^k arms with maximum $Q_i^k(t-1) = \hat{\mu}_i^k(t-1) + C_i^k(t-1)$, where

$$C_i^k(t-1) = \sigma_g \sqrt{\frac{2\gamma}{G(\eta)} \cdot \frac{\hat{n}_i^k(t-1) + f(t-1)}{M\hat{n}_i^k(t-1)} \cdot \frac{\ln(t-1)}{\hat{n}_i^k(t-1)}}. \tag{4.2}$$

Each agent k then selects the arm in $\mathcal{O}_{\omega^k}(t)$ with minimum $W_i^k(t-1) = \hat{\mu}_i^k(t-1) - C_i^k(t-1)$, where in the above $f(t)$ is an increasing sublogarthmic function, $\gamma > 1$, $\eta \in (0, 4)$, and $G(\eta) = 1 - \eta^2/16$.

Then, at each time t, each agent k updates its cooperative estimate of the mean reward at each arm using the distributed cooperative estimation algorithm described in (3.1–3.3).

4.3 Regret of the coop-UCB2 Collisions Algorithm

In this section we prove upper bounds on the expected cumulative group regret of the coop-UCB2 collisions algorithm. To accomplish this we first prove an upper bound on the number of times an agent i chooses an arm that is not k^i in Theorem 5. We then use this result in Theorem 6 to prove an upper bound on the regret of the coop-UCB2 collisions algorithm. Our two theorems here take inspiration from [30].

Theorem 5 (***Upper Bound on Incorrect Selections***)**.** *For any arm i the following holds:*

$$\sum_{k\neq k^i}\mathbb{E}\left[n_i^k(T)\right] \leq \left\lceil M\epsilon_n + \frac{4\sigma_g^2\gamma}{\Delta_{\min}^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_{\min}^2 MG(\eta)}{2\sigma_g^2\gamma}\frac{f(T)}{\ln T}}\right)\ln T\right\rceil + 4\sum_{k=1}^{M}(t_k^\dagger - 1) + 2\frac{MN+M(M+1)}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right).$$

Proof. We begin by noting that

$$\begin{aligned}
\sum_{k\neq k^i} n_i^k(T) &= \sum_{k\neq k^i}\sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i\right\} \\
&= \sum_{k\neq k^i}\sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i, m_i < m_{b^k}\right\} + \sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i, m_i \geq m_{b^k}\right\} \\
&\leq A + \sum_{k\neq k^i}\sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i, m_i < m_{b^k}, Mn_i^{\text{cent}}(t)\geq A\right\} \\
&\quad + \sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i, m_i \geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A\right\} \qquad (4.3)
\end{aligned}$$

where A is a constant that will be chosen later. In the case where $m_i < m_{b^k}$, agent k picking arm i implies that there exists an arm $j \in \mathcal{O}^*_{\omega^k}$ such that $j \notin \mathcal{O}_{\omega^k}(t)$. Therefore, the following holds:

$$\begin{aligned}
&\sum_{k\neq k^i}\sum_{t=1}^{T-1} \mathbb{1}\left\{i^k(t)=i, m_i < m_{b^k}, Mn_i^{\text{cent}}(t) \geq A\right\} \\
&\quad \leq \sum_{k\neq k^i}\sum_{t=1}^{T-1} \mathbb{1}\left\{Q_i^k(t-1) \geq Q_j^k(t-1), m_i < m_{b^k}, Mn_i^{\text{cent}}(t) \geq A\right\} \\
&\quad \leq \sum_{k\neq k^i}\sum_{t=1}^{T-1}\sum_{j\in\mathcal{O}^*_{\omega^k}} \mathbb{1}\left\{Q_i^k(t-1) \geq Q_j^k(t-1), m_i < m_{b^k}, Mn_i^{\text{cent}}(t) \geq A\right\} \\
&\quad \leq \sum_{k\neq k^i}\sum_{j\in\mathcal{O}^*_{\omega^k}}\sum_{t=1}^{T-1} \mathbb{1}\left\{Q_i^k(t-1) \geq Q_j^k(t-1), m_i < m_{b^k}, Mn_i^{\text{cent}}(t) \geq A\right\}.
\end{aligned}$$

As in Theorem 3, $Q_i^k(t) \geq Q_j^k(t)$ implies that at least one of the following three conditions must hold:

$$\hat{\mu}_j(t-1) \leq m_j - C_j^k(t-1) \tag{4.4}$$

$$\hat{\mu}_i(t-1) \geq m_i + C_i^k(t-1) \tag{4.5}$$

$$m_j < m_i + 2C_i^k(t-1). \tag{4.6}$$

The first two equations are bounded using Theorem 1 as in 3. The third equation is equivalent to

$$2C_i^k(t) > \Delta_{j,i} > \Delta_{\min}$$

which, similarly to Theorem 3, does not hold if

$$n_i^k(t) > \frac{4\sigma_g^2\gamma}{\Delta_{\min}^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_{\min}^2 MG(\eta)}{2\sigma_g^2\gamma}\frac{f(T)}{\ln T}}\right)\ln t.$$

Therefore, for

$$A = \left\lceil M\epsilon_n + \frac{4\sigma_g^2\gamma}{\Delta_{\min}^2 G(\eta)}\left(1+\sqrt{1+\frac{\Delta_{\min}^2 MG(\eta)}{2\sigma_g^2\gamma}\frac{f(T)}{\ln T}}\right)\ln t\right\rceil$$

(4.6) does not hold.

This results in

$$\begin{aligned}
&\sum_{k\neq k^i}\sum_{j\in\mathcal{O}^*_{\omega^k}}\sum_{t=1}^{T-1}\mathbb{1}\left\{Q_i^k(t-1)\geq Q_j^k(t-1), m_i < m_{b^k}, Mn_i^{\text{cent}}(t)\geq A\right\}\\
&\leq 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \sum_{k\neq k^i}\sum_{j\in\mathcal{O}^*_{\omega^k}}\frac{2}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right)\\
&= 2\sum_{k=1}^{M}(t_k^\dagger - 1) + \frac{M(M+1)}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right). \qquad (4.7)
\end{aligned}$$

We now examine the second part of (4.3) when $m_i \geq m_{b^k}$ and split the conditional as

$$\begin{aligned}
&\mathbb{1}\left\{i^k(t)=i, m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A\right\}\\
&\quad=\mathbb{1}\left\{i^k(t)=i, m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, \mathcal{O}_{\omega^k}(t)=\mathcal{O}^*_{\omega^k}\right\}\\
&\qquad+\mathbb{1}\left\{i^k(t)=i, m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, \mathcal{O}_{\omega^k}(t)\neq\mathcal{O}^*_{\omega^k}\right\}\\
&\quad\leq\mathbb{1}\left\{m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, W_i^k(t-1)\leq W_{b^k}^k(t-1)\right\}\\
&\qquad+\mathbb{1}\left\{m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, W_i^k(t-1)\leq W_h^k(t-1)\right\} \qquad (4.8)
\end{aligned}$$

for any arm $h\notin\mathcal{O}^*_{\omega^k}$. We also define $j\notin\{\mathcal{O}^*_{\omega^k}\setminus b^k\}$ and note that

$$(4.8)=\mathbb{1}\left\{m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, W_i^k(t-1)\leq W_j^k(t-1)\right\}.$$

This results in

$$\sum_{k\neq k^i}\sum_{t=1}^{T}\mathbb{1}\left\{i^k(t)=i, m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A\right\}$$
$$\leq \sum_{k\neq k^i}\sum_{j\notin\{\mathcal{O}^*_{\omega^k}\setminus b^k\}}\sum_{t=1}^{T}\mathbb{1}\left\{m_i\geq m_{b^k}, Mn_i^{\text{cent}}(t)\geq A, W_i^k(t-1)\leq W_j^k(t-1)\right\}. \quad (4.9)$$

For $W_i^k(t-1)\leq W_j^k(t-1)$ to be true at least one of the following must hold:

$$\hat{\mu}_i(t-1)\leq m_i - C_i^k(t-1) \quad (4.10)$$
$$\hat{\mu}_j(t-1)\geq m_j + C_j^k(t-1) \quad (4.11)$$
$$m_i < m_j + 2C_j^k(t-1). \quad (4.12)$$

(4.10) and (4.11) can be bounded using Theorem 1. (4.12) never holds due to our previous choice of A. Similarly to (4.7) this gives

$$(4.9)\leq 2\sum_{k=1}^{M}(t_k^\dagger-1)+\sum_{k\neq k^i}\sum_{j\notin\{\mathcal{O}^*_{\omega^k}\setminus b^k\}}\frac{2}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right).$$
$$\leq 2\sum_{k=1}^{M}(t_k^\dagger-1)+\frac{2NM+M(M+1)}{\ln(1+\eta)}\left(\frac{1}{(\gamma-1)^2}+\frac{1}{\gamma-1}+5\right)$$

which completes the proof.

□

Theorem 6 (***Regret of the Coop-UCB2 Collisions Algorithm***)**.** *For the coop-UCB2 collisions algorithm with sub-Gaussian rewards, the expected cumulative regret of the group satisfies*

$$\sum_{k=1}^{M}R^k(T)\leq m_{i^*}NB+\sum_{k=1}^{M}m_{b^k}B$$

where

$$B = \left\lceil M\epsilon_n + \frac{4\sigma_g^2\gamma}{\Delta_{\min}^2 G(\eta)} \left(1 + \sqrt{1 + \frac{\Delta_{\min}^2 M G(\eta)}{2\sigma_g^2\gamma} \frac{f(T)}{\ln T}}\right) \ln T \right\rceil$$
$$+ 4\sum_{k=1}^{M}(t_k^\dagger - 1) + 2\frac{MN + M(M+1)}{\ln(1+\eta)} \left(\frac{1}{(\gamma-1)^2} + \frac{1}{\gamma - 1} + 5\right). \tag{4.13}$$

Proof. As in [30], an agent k can incur regret either by selecting an arm $i \neq b^k$ or when another user $j \neq k$ selects arm b^k at the same time. Therefore,

$$\begin{aligned}
\sum_{k=1}^{M} R^k(T) &\leq \sum_{k=1}^{M} \sum_{i \neq b^k} \mathbb{E}\left[n_i^k(T)\right] m_{b^k} + \sum_{k=1}^{M} \sum_{j \neq k} \mathbb{E}\left[n_{b^k}^j(T)\right] m_{b^k} \\
&\leq m_{i^*} \sum_{i=1}^{N} \sum_{k \neq k^i} \mathbb{E}\left[n_i^k(T)\right] + \sum_{k=1}^{M} \sum_{j \neq k} \mathbb{E}\left[n_{b^k}^j(T)\right] m_{b^k} \\
&\leq m_{i^*} \sum_{i=1}^{N} B + \sum_{k=1}^{M} m_{b^k} B
\end{aligned}$$

completing the proof.

Remark 5 (***Concise Upper Bound on Regret and the Benefits of Communication***)**.** The upper bound on expected cumulative group regret in Theorem 6 can be expressed more concisely at the expense of some tightness as

$$\sum_{k=1}^{M} R^k(T) \leq m_{i^*} B(M+N).$$

This bound is a factor of $4M$ times tighter than the bounds for the state-of-the-art algorithm presented in [30] when considering bounded rewards, demonstrating the benefits of communication between agents.

□

4.4 Robotic Implementation

In this section we build on our previous analytical results and numerical examples to conduct an experiment that demonstrates the utility of multi-agent MAB algorithms with collisions in robotic search tasks. We consider three wheeled robots that can traverse a space and carry sensors to sample from a real light field, with the goal of finding the location where light of a certain color has its maximum intensity.

4.4.1 Experimental Setup

This experiment was conducted in the Leonard Lab in room H121 in Princeton's Engineering Quadrangle. This room is equipped with a VICON system similar to that described in Section 3.5. The room is also equipped with an array of lights on the ceiling, but unlike in Section 3.5 we installed filters over each light to spread the light and create a diffuse light field.

The diffuse light field produced is used as the reward field for the MAB experiment. In this experiment we consider as arms 100 discrete points on the lab floor arranged in a 10×10 grid. Furthermore, due to the small grid size of 0.5 meters, only one robot can occupy a grid point, or arm, at each time. This small grid spacing, combined with the diffuse light field, creates a reward surface that has correlation structure, which we utilize.

We used three Turtlebot2 robots made by Robotis. Each robot has a Raspberry Pi 3 for communications and motor control, and is controlled using Robotics Operating System (ROS). Each robot also has a Adafruit TCS34725 color sensor, which has three sensors that measure the red, green, and blue components of the impinging light. In this experiment we use the green sensor, and it is notable that the green sensor is still somewhat sensitive to red and particularly blue light.

4.4.2 Robotic Experiments

Example 10. In this experiment we consider three robots searching the space described above for areas of high intensity green light. This search task is difficult for the robot because the green color sensor is somewhat sensitive to blue light and will register blue light as low intensity green light. Therefore, while we as observers can easily see the differences between green and blue light in the video, the robot cannot. Furthermore, the actual intensity of the light field is controlled by a computer and changes randomly between timesteps, with the intensity sampled from a normal distribution.

Each robot uses a modified version of the coop-UCB2 collisions algorithm to make a choice of arm as described below. Only one robot is allowed to occupy a grid space at a given time. If a "collision" occurs, defined as when multiple robots attempt to sample the same arm, the one that arrives first is given precedence and the others must sample adjacent arms. Additionally, robots communicate using consensus with P as in (2.14), $\kappa = \frac{d_{\max}}{d_{\max}-1}$, and an all-to-all communication graph. Each robot uses the estimation update procedure from coop-UCL given in Section 3.3.2 to update their estimates at each timestep. This allows agents to take advantage of the inherent correlation structure present in the light field.

When conducting MAB-inspired search tasks with robots it is desirable to both maximize the reward received and to reduce the number of transitions between options. Each transition requires the robot to move and use energy, and we wish to limit energy usage over time. To this end we take inspiration from Reverdy *et al.* [84] and employ a *block allocation* strategy, which limits each robot to selecting a new target arm only at the start of a block of time. We assign all times $t \in \{1, \dots, T\}$ into blocks of time, with each block of time increasing in length from the previous, which ensures that the robot will make fewer transitions over time.

At the start of each block of time each robot selects a target arm using the coop-UCB2 collisions algorithm. The robot then travels to this target arm by traveling from their current arm to the nearest arm in the direction of the target arm. The robot takes a sample from this arm before moving on to the next, until reaching the target arm. This transit procedure effectively uses the transit time to take additional samples by stopping to take samples from arms that are along the path to the target arm. The robots use the same motion planning controller as in Section 3.5.

The experiment is shown in Video 3, also available online at `youtu.be/c3ev-wKAEZA`. In the video the three robots explore the space, and eventually settle on the three locations with the highest intensity of green light. The experiment demonstrates that the robots are able to successfully distribute themselves to avoid collisions while finding and settling on the three best grid points.

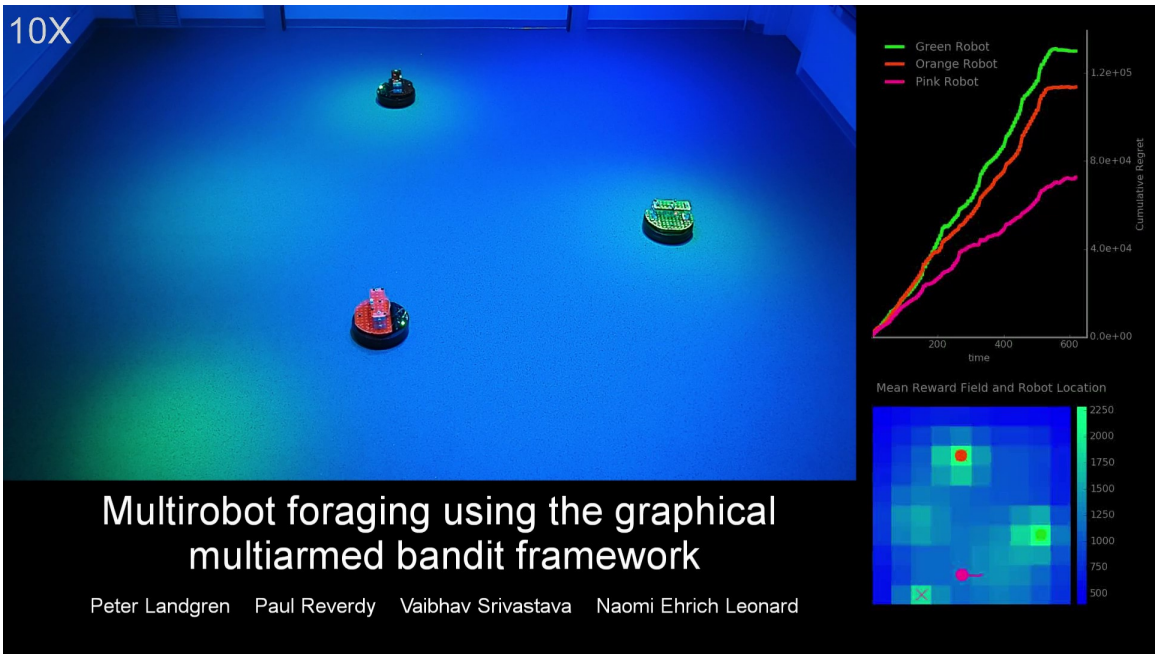


Figure 4.1: A screenshot from near the end of Video 3. The panel on the left shows a camera view of the experiment space as well as the three robots. The robot's locations are also shown in the panel on the bottom right, along with a color key that shows how light colors correspond to reward values. The panel in the upper right shows the cumulative regret of each of the three robots in the video.

4.5 Discussion

In this chapter we defined and analyzed the coop-UCB2 collisions algorithm. This algorithm addresses the multi-agent MAB problem with collisions and considers the case of direct communication through consensus. We demonstrated, through the derivation of an upper bound on regret, that the coop-UCB2 algorithm can be extended to the MAB problem with collisions where agents that pick the same arm at the same time get no reward. We also demonstrated that the use of direct communication and the coop-UCB2 collisions algorithm greatly improves the performance of the group compared to state-of-the-art algorithms for the MAB problem with collisions that only utilize indirect communication.

Additionally, we used the coop-UCB2 collisions algorithm in an multi-robot search experiment.

In the next chapter we consider direct communication through strictly local communication rather than consensus.

Chapter 5

Social Imitation in Multi-armed Bandits with Strictly Local Communication [1]

In this chapter we study the distributed cooperative MAB problem with strictly local communication. We consider the case where agents have the ability to imitate their neighbors in a communication graph. Our problem setup is motivated by the phenomenon of social imitation, which is often encountered in natural systems [82, 88, 103].

In this context strictly local communication means that agents can access the rewards and choices of their neighbors in a communication graph. This stands in contrast to communication through consensus, where agents share their estimates of these values. The strictly local communication model therefore does not require agents to broadcast their estimates, but only access the rewards and choices of neighbors. This setting is applicable to multi-agent scenarios where agents can only observe others.

[1] This chapter is adapted from [59]. Sections 5.1 to 5.4 are mostly taken verbatim.

We first define the cooperative MAB problem with strictly local communication mathematically and review prior work on imitation in this setting. We then introduce and analyze the UCB-Partition algorithm, a partition-based distributed decision-making algorithm, where only one agent in each partition, a so-called leader, makes independent decisions based on its local information. The other agents in the partition, the so-called followers, imitate the decisions of the leader in the partition, either directly if the leader is a neighbor, or, otherwise, indirectly by imitating a neighbor along a path to the leader. Finally, we study the UCB-Partition algorithm using Monte-Carlo simulations.

5.1 The Cooperative MAB Problem with Local Communication

Consider an MAB problem with N arms and M agents. The reward associated with arm $i \in \{1, \ldots, N\}$ is a bounded random variable in $[0, 1]$ with unknown mean m_i. The communication among agents is modeled by a connected, unweighted, undirected network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = M$. Let $\mathcal{N}(k)$ denote the set of neighbors of each agent $k \in \mathcal{V}$.

We assume that the agents can be classified into leaders and followers. We assume that every follower is connected to at least one leader through a path in $\mathcal{G}$, and it imitates one such leader, either directly or indirectly through a chain of followers. The set of leaders induces a partitioning of the graph in which every agent in a leader's partition ultimately imitates it. We assume that at each time t each leader has access to the arms chosen and rewards received by its neighbors, while each follower only has access to the arms chosen by its neighbors. We also assume that each agent k knows the degree of its neighbors: $|\mathcal{N}(j)|$ for each $j \in \mathcal{N}(k)$.

Let each agent k choose arm $i^k(t)$ at time $t \in \{1, \ldots, T\}$ and receive i.i.d reward $r^k(t)$. The total number of times up to time t that agent k has selected arm i is $n_i^k(t)$ and that agent k and its neighbors have selected arm i is $\bar{n}_i^k(t) = n_i^k(t) + \sum_{j \in \mathcal{N}(k)} n_i^j(t)$. The sequence of rewards received by agent k and its neighbors from arm i is $\{r_{i,s}^k\}_{s \in \{1,\ldots,\bar{n}_i^k(t)\}}$. The estimated mean of arm i at time t by agent k given its own and its neighbors' realized rewards is $\bar{\mu}_{i,\bar{n}_i^k(t)}^k = \frac{1}{\bar{n}_i^k(t)} \sum_{s=1}^{\bar{n}_i^k(t)} r_{i,s}^k$. Each leader ℓ can compute $\bar{n}_i^\ell(t)$ and $\bar{\mu}_i^\ell(t)$.

The objective of this chapter is to design a distributed algorithm for partitioning the graph $\mathcal{G}$, assigning a leader to each partition such that every other agent in the partition imitates it, and determining a sequential decision-making policy for the leaders and the followers such that efficient group performance is achieved. Alternatively, a set of leaders may be assigned and the distributed algorithm should select the set of followers and, consequently, the graph partitioning.

The regret of agent k at each time t conditioned on the choice $i^k(t)$ is defined by $R^k(t) = m_{i^*} - m_{i^k(t)} \equiv \Delta_{i^k(t)}$, where $m_{i^*} = \max_{i \in \{1,\ldots,N\}} m_i$. We characterize group performance in terms of the total expected cumulative regret defined by $\sum_{k=1}^M \sum_{t=1}^T \mathbb{E}[R^k(t)] = \sum_{k=1}^M \sum_{i=1}^N \Delta_i \mathbb{E}[n_i^k(T)]$, where T is the horizon length.

In this chapter, we restrict our attention to policies in which the leaders follow the UCB algorithm with the estimates of the mean rewards that are computed using the rewards received by the leader and its neighbors and the followers imitate one of their neighbors.

5.2 Partition Based Multi-player MAB

In this section we describe and prove upper bounds on the performance of partition-based multi-player MAB. We introduce several definitions, describe the problem and

the UCB-Partition algorithm.We then establish bounds on performance of the UCB-Partition algorithm.

5.2.1 Definitions and Notation

We now introduce several definitions that formalize the leader/follower relationships inherent in the UCB-Partition algorithm. We will use these formal definitions to prove an upper bound on the cumulative expected regret of the algorithm. Fig. 5.1 illustrates these definitions with an example.

Let $\mathcal{G}_{\text{ldr}} = (\mathcal{V}_{\text{ldr}}, \mathcal{E}_{\text{ldr}})$ be a directed graph such that $\mathcal{V}_{\text{ldr}} = \mathcal{V}$ and

$$\mathcal{E}_{\text{ldr}} = \{(k, j) \in \mathcal{E} \mid k \ \text{ can } \ \text{imitate } \ j\}.$$

$\mathcal{G}_{\text{ldr}}$ encodes all possible variations of followers in the UCB-Partition algorithm: a directed edge in $\mathcal{G}_{\text{ldr}}$ indicates that the agent at the tail may follow the agent at the head. $\mathcal{G}_{\text{ldr}}$ can therefore be used to enforce operation constraints on who can or cannot follow others.

We now define the set of all leaders by $\mathcal{L}$ and, in the following, we will denote the i-th element of $\mathcal{L}$ by ℓ_i. We also define $\mathcal{G}_{\text{ldr}}^{\text{rlz}} = (\mathcal{V}_{\text{ldr}}^{\text{rlz}}, \mathcal{E}_{\text{ldr}}^{\text{rlz}})$ such that $V_{\text{ldr}}^{\text{rlz}} = \mathcal{V}$ and

$$\mathcal{E}_{\text{ldr}}^{\text{rlz}} = \{(k, j) \subset \mathcal{E}_{\text{ldr}} \mid \nexists\, m \neq j \in \mathcal{V}_{\text{ldr}}, (k, m) \in \mathcal{E}_{\text{ldr}}^{\text{rlz}}, k \notin \mathcal{L}\}.$$

Note that $\mathcal{E}_{\text{ldr}}^{\text{rlz}}$ is defined recursively and restricts follower agent k to imitate at most one leader or follower. $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ thus encodes a possible realization of follower and leader combinations when using the UCB-Partition algorithm: a directed edge in $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ indicates that the agent at the tail will follow the agent at the head, and agents with no outgoing edges are leaders.

For a given realization $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$, we define the set of followers of leader ℓ_j as

$$\mathcal{F}_j^{\text{rlz}} = \{\ell_j\} \cup \{k \in \mathcal{V}_{\text{ldr}}^{\text{rlz}} \mid \exists \text{ directed path from } k \text{ to } \ell_j \text{ in } \mathcal{G}_{\text{ldr}}^{\text{rlz}}\}.$$

and the set of direct followers of leader ℓ_j as

$$\mathcal{F}_{j\text{-direct}}^{\text{rlz}} = \{\ell_j\} \cup \{k \in \mathcal{V}_{\text{ldr}}^{\text{rlz}} \mid (k, \ell_j) \in \mathcal{E}_{\text{ldr}}^{\text{rlz}}\}.$$

The sets $\mathcal{F}_j^{\text{rlz}}$, $j \in \{1, \ldots, |\mathcal{L}|\}$, define a partitioning of $\mathcal{G}$, where each partition contains one leader that every follower in the partition ultimately imitates, and $\mathcal{F}_{j\text{-direct}}^{\text{rlz}} \subseteq \mathcal{F}_j^{\text{rlz}}$. Fig. 5.1 illustrates these subgraphs for a given $\mathcal{G}$ and three example realizations $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$. We denote the length of the longest path present in $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ within the partition defined by $\mathcal{F}_j^{\text{rlz}}$ as $\text{diam}_j^{\text{rlz}}$.

Every realization of $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ induces a partitioning of the graph $\mathcal{G}$. Equivalently, for any partitioning of the graph $\mathcal{G}$, we can choose a leader in each partition and construct $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$. The following analysis holds for any realization $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ and is oblivious to how $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ is constructed, i.e., whether it is induced by a given set of leaders or if it is induced by a given partitioning.

The set $\mathcal{F}_{j\text{-direct}}^{\text{rlz}}$ is used later in this chapter to bound the expected cumulative regret of the UCB-Partition algorithm for both a single partition and multiple partitions of $\mathcal{G}$.

5.2.2 UCB-Network and Follow Your Leader Algorithms

In this chapter we draw inspiration from Kolla *et al.* [50], which also considers the multi-agent MAB problem with strictly local communication and the ability to imitate. The Follow Your Leader (FYL) algorithm of [50] partitions the network into "leaders," which use the UCB1 algorithm, and "copiers," which imitate the actions of an adjacent leader. The FYL algorithm selects how many and which agents will

be leaders using a dominating set[2] of the graph; these must be computed prior to runtime.

The ability to imitate a neighbor who is itself imitating another neighbor is the key difference in the problem formulation between our work and [50]. In [50] agents can only imitate a neighbor that is a leader, and our relaxation of this constraint leads to a richer set of possible strategies and analysis.

The UCB-Network and Follow Your Leader (FYL) algorithms are defined in [50]. The UCB-network algorithm is equivalent to setting $\mathcal{L} = \mathcal{V}$, making every agent a leader that can access rewards of its neighbors. The UCB-Network algorithm is thus easily distributed, but it does not allow for any agent to imitate.

In the FYL algorithm the leaders $\mathcal{L}$ are defined as a dominating set of $\mathcal{G}$, and the followers of ℓ_j are composed of a subset of the neighbors of ℓ_j. In the FYL algorithm the best performance is achieved when $\mathcal{L}$ is defined as the minimal dominating set. An example of leader selection corresponding to the minimal dominating set is shown in Panel C in Fig. 5.1.

5.2.3 UCB-Partition Algorithm

First, we define

$$Q_i^k(t, \bar{n}_i^k(t)) = \bar{\mu}^k_{i,\bar{n}_i^k(t)} + \sqrt{\frac{2\ln(t)}{\bar{n}_i^k(t)}}. \tag{5.1}$$

The UCB-Partition algorithm is as follows:

(i) Initialization phase: Every leader $j \in \mathcal{L}$ chooses each arm once, and each follower $k \in \mathcal{F}_j^{\text{rlz}}$ chooses randomly for the first timestep.

[2] A subset of nodes of a graph is called a *dominating set* if for every node not in the dominating set, there exists an adjacent node that belongs to the dominating set. The smallest dominating set is called the minimal dominating set.

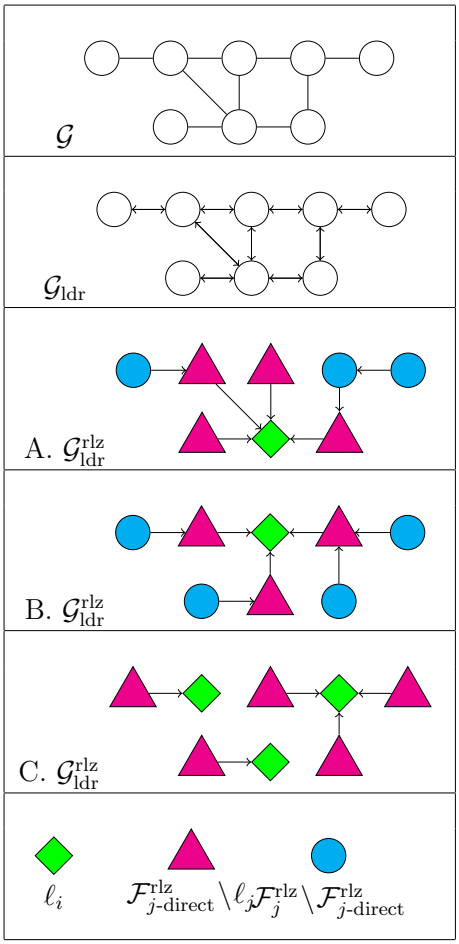


Figure 5.1: Example of a communication graph $\mathcal{G}$ and a $\mathcal{G}_{\text{ldr}}$ that allows for any agent to imitate any neighbor in $\mathcal{G}$. Panels A and B show two possible realizations of $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ for the case of one leader. Panel C demonstrates a realization for three leaders. The selection of each agent's role, which defines $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$, can be driven by design constraints or optimized to minimize the upper bounds on performance. Note that even if two agents are not connected in $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ they can still share sample information if connected in $\mathcal{G}$.

(ii) Each leader $j \in \mathcal{L}$ selects the arm with the highest $Q_i^j(t, \bar{n}_i^j(t))$, and each follower k selects the arm selected by agent $\{m \in \mathcal{V}_{\text{ldr}}^{\text{rlz}} \mid (k,m) \in \mathcal{E}_{\text{ldr}}^{\text{rlz}}\}$ at the previous timestep.

5.2.4 Expected Cumulative Regret of UCB-Partition

Here we establish an upper bound on the cumulative expected regret of the UCB-Partition algorithm.

Theorem 7. *For the UCB-Partition algorithm with definitions given in Section 5.2.1 the following bounds hold for $i \neq i^*$ and given a $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ with $|\mathcal{L}| = 1$:*

$$\sum_{k=1}^{M} \mathbb{E}\left[n_i^k(T)\right] \leq \frac{8\ln(T)}{\Delta_i} \cdot \frac{|\mathcal{F}_1^{\text{rlz}}|}{|\mathcal{F}_{\text{1-direct}}^{\text{rlz}}|} + M^3(1+\frac{\pi^2}{3}) + (|\mathcal{F}_1^{\text{rlz}}| - 1)\text{diam}_1^{\text{rlz}},$$

where $|\mathcal{F}_1^{\text{rlz}}| = M$ and $|\mathcal{F}_{\text{1-direct}}^{\text{rlz}}| = |\mathcal{N}(\ell_1)| + 1$.

Proof. We start by noticing that

$$\begin{aligned}
\sum_{k=1}^{M} n_i^k(T) &\leq |\mathcal{F}_1^{\text{rlz}}|\, n_i^{\ell_1}(T) + (|\mathcal{F}_1^{\text{rlz}}| - 1)\text{diam}_1^{\text{rlz}} \qquad (5.2)\\
&= |\mathcal{F}_1^{\text{rlz}}| \sum_{t=1}^{T} \mathbb{1}\left\{i^{\ell_1}(t) = i\right\} + (|\mathcal{F}_1^{\text{rlz}}| - 1)\text{diam}_1^{\text{rlz}}\\
&\leq (|\mathcal{F}_1^{\text{rlz}}| - 1)\text{diam}_1^{\text{rlz}} + |\mathcal{F}_1^{\text{rlz}}| \Bigg[\frac{A}{|\mathcal{F}_{\text{1-direct}}^{\text{rlz}}|}\\
&\qquad + \sum_{t=1}^{T} \mathbb{1}\left\{i^{\ell_1}(t) = i,\ n_i^{\ell_1}(t) > \frac{A}{|\mathcal{F}_{\text{1-direct}}^{\text{rlz}}|}\right\}\Bigg], \qquad (5.3)
\end{aligned}$$

where A is a constant that will be chosen later and the $(|\mathcal{F}_1^{\text{rlz}}| - 1)\text{diam}_1^{\text{rlz}}$ term in (5.2) follows because every follower will not necessarily be copying their leader until the leader's choices propagate through the network. We now bound the second part

of (5.3) using techniques from [5].

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbb{1}\left\{ i^{\ell_1}(t) = i,\ n_i^{\ell_1}(t) > \frac{A}{|\mathcal{F}^{\text{rlz}}_{\text{1-direct}}|} \right\} \\
&\qquad \leq \sum_{t=1}^{T} \mathbb{1}\left\{ Q_{i^*}^{\ell_1}(t, \bar{n}_{i^*}^{\ell_1}(t)) < Q_i^{\ell_1}(t, \bar{n}_i^{\ell_1}(t)), n_i^{\ell_1}(t) > \frac{A}{|\mathcal{F}^{\text{rlz}}_{\text{1-direct}}|} \right\} \\
&\qquad \leq \sum_{t=1}^{T} \mathbb{1}\left\{ Q_{i^*}^{\ell_1}(t, \bar{n}_{i^*}^{\ell_1}(t)) < Q_i^{\ell_1}(t, \bar{n}_i^{\ell_1}(t)), \bar{n}_i^{\ell_1}(t) > A - M \right\} \qquad (5.4) \\
&\qquad \leq \sum_{t=1}^{T} \mathbb{1}\left\{ \min_{1<a<(|\mathcal{N}(\ell_1)|+1)t} Q_{i^*}^{\ell_1}(t,a) < \max_{A-M<b<(|\mathcal{N}(\ell_1)|+1)t} Q_i^{\ell_1}(t,b) \right\} \\
&\qquad \leq \sum_{t=1}^{T} \sum_{a=1}^{(|\mathcal{N}(\ell_1)|+1)t} \sum_{b=A-M}^{(|\mathcal{N}(\ell_1)|+1)t} \mathbb{1}\left\{ Q_{i^*}^{\ell_1}(t,a) < Q_i^{\ell_1}(t,b) \right\}
\end{aligned}$$

where (5.4) follows because the direct followers of the leader choose $i^{\ell_1}(t)$ at time $t+1$. In the spirit of [5], if $\mathbb{1}\left\{ Q_{i^*}^{\ell_1}(t,a) < Q_i^{\ell_1}(t,b) \right\}$ holds then at least one of the following must hold:

$$\bar{\mu}_{i^*,a} \leq m_{i^*} - \sqrt{\frac{2\ln(t)}{a}} \qquad (5.5)$$

$$\bar{\mu}_{i,b} \geq m_i + \sqrt{\frac{2\ln(t)}{b}} \qquad (5.6)$$

$$m_{i^*} < m_i + \sqrt{\frac{8\ln(t)}{b}} \qquad (5.7)$$

As in [5], we bound (5.5) and (5.6) using Chernoff-Hoeffding bounds as

$$\mathbb{P}\left(\bar{\mu}_{i^*,a} \leq m_{i^*} - \sqrt{\frac{2\ln(t)}{a}} \right) \leq t^{-4}, \text{ and}$$
$$\mathbb{P}\left(\bar{\mu}_{i,b} \geq m_i + \sqrt{\frac{2\ln(t)}{b}} \right) \leq t^{-4}.$$

Setting $A = M + \frac{8\ln(t)}{\Delta_i^2}$, we see that (5.7) never holds. Thus,

$$
\begin{aligned}
|\mathcal{F}_1^{\text{rlz}}| \sum_{t=1}^{T} \mathbb{1}\left\{ i^{\ell_1}(t) = i, n_i^{\ell_1}(t) > \frac{A}{|\mathcal{F}_{\text{1-direct}}^{\text{rlz}}|} \right\} \\
&\leq |\mathcal{F}_1^{\text{rlz}}| \sum_{t=1}^{T} \sum_{a=0}^{(|\mathcal{N}(\ell_1)|+1)t} \sum_{b=A-M}^{(|\mathcal{N}(\ell_1)|+1)t} \frac{2}{t^4} \\
&\leq |\mathcal{F}_1^{\text{rlz}}| \sum_{t=1}^{T} \frac{2}{t^2} (|\mathcal{N}(\ell_1)| + 1)^2 \\
&\leq |\mathcal{F}_1^{\text{rlz}}| (|\mathcal{N}(\ell_1)| + 1)^2 (1 + \frac{\pi^2}{3}) \leq M^3 (1 + \frac{\pi^2}{3}),
\end{aligned}
$$

which completes the proof. □

Corollary 1. *For the UCB-Partition algorithm with definitions given in Section 5.2.1 the following bounds hold for $i \neq i^*$ and any given $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ with a generic set of leaders $\mathcal{L}$:*

$$
\sum_{k=1}^{M} \mathbb{E}\left[n_i^k(T)\right] \leq \frac{8 \ln(T)}{\Delta_i} \sum_{j \in \mathcal{L}} \frac{|\mathcal{F}_j^{\text{rlz}}|}{|\mathcal{F}_{j\text{-direct}}^{\text{rlz}}|} + M^3 (1 + \frac{\pi^2}{3}) + \sum_{j \in \mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1) \text{diam}_j^{\text{rlz}}.
$$

Proof. Similar to the proof of Theorem 1, we note that

$$\begin{aligned}
\sum_{k=1}^{M} n_i^k(T) &\leq \sum_{j\in\mathcal{L}} |\mathcal{F}_j^{\text{rlz}}|\, n_i^{\ell_j}(T) + \sum_{j\in\mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1)\text{diam}_j^{\text{rlz}} \\
&= \sum_{j\in\mathcal{L}} |\mathcal{F}_j^{\text{rlz}}| \sum_{t=1}^{T} \mathbb{1}\left\{i^{\ell_j}(t) = i\right\} + \sum_{j\in\mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1)\text{diam}_j^{\text{rlz}} \\
&\leq \sum_{j\in\mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1)\text{diam}_j^{\text{rlz}} + \sum_{j\in\mathcal{L}} |\mathcal{F}_j^{\text{rlz}}| \Bigg[\frac{A}{|\mathcal{F}_{j\text{-direct}}^{\text{rlz}}|} \\
&\qquad + \sum_{t=1}^{T} \mathbb{1}\left\{ i^{\ell_j}(t) = i,\ n_i^{\ell_j}(t) > \frac{A}{|\mathcal{F}_{j\text{-direct}}^{\text{rlz}}|} \right\} \Bigg] \\
&\leq \sum_{j\in\mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1)\text{diam}_j^{\text{rlz}} + A \sum_{j\in\mathcal{L}} \frac{|\mathcal{F}_j^{\text{rlz}}|}{|\mathcal{F}_{j\text{-direct}}^{\text{rlz}}|} + M^2(1 + \frac{\pi^2}{3}) \sum_{j\in\mathcal{L}} |\mathcal{F}_j^{\text{rlz}}| \quad (5.8) \\
&= \sum_{j\in\mathcal{L}} (|\mathcal{F}_j^{\text{rlz}}| - 1)\text{diam}_j^{\text{rlz}} + A \sum_{j\in\mathcal{L}} \frac{|\mathcal{F}_j^{\text{rlz}}|}{|\mathcal{F}_{j\text{-direct}}^{\text{rlz}}|} + M^3(1 + \frac{\pi^2}{3}),
\end{aligned}$$

where (5.8) follows from Theorem 7, completing the proof. □

5.2.5 Distributed Partition-Based Multi-agent MAB using Token Passing

The UCB-Partition algorithm and the associated bounds provide performance guarantees for a given $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$, which by definition defines $|\mathcal{L}|$ partitions of $\mathcal{G}$ and the leader-follower assignments. In this section we present a distributed method for choosing $|\mathcal{L}|$ leaders and partitions, which in turn, with follower assignments, gives $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$.

This method is comprised of two parts: leader identification and partition generation. The goal of the leader identification step is for each agent to construct, in a distributed fashion, a list of tuples of size $|\mathcal{L}|$, where each tuple contains the identify and degree of agents with top $|\mathcal{L}|$ degree. Let each agent k have a unique identity number v, and let each agent know their own degree, $|\mathcal{N}(k)|$, in $\mathcal{G}$, and identity of their neighbors. Each agent initially constructs a list of size $|\mathcal{L}|$ with only one entry:

the agent's identity and degree, and the other entries are empty. Then, each agent exchanges this list with each of their neighbors and combines their own list with those received to create a new list of agents with the top $|\mathcal{L}|$ degrees in the lists (in case of ties, the agent with lower identity is selected). Each agent then repeats this process with their new list, and the procedure converges in number of timesteps equal to at most two times the diameter of graph $\mathcal{G}$ plus one.

To accomplish partition generation each agent represented in the final list identifies itself as a leader. Followers then recursively choose an agent to imitate. First, the agents that are adjacent to leader(s) commit to imitating a leader, and transmit a committed signal to their neighbors. Subsequently the uncommitted neighbors may choose to imitate one of the committed agents, until all agents are committed.

This procedure defines $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$, and the performance bounds established in Section 5.2.4 hold. In future work will we rigorously show that this strategy converges to a valid partition for a connected communication graph $\mathcal{G}$.

Fig. 5.2 demonstrates three examples of leader selection and follower assignment using this method for one, three, and five leaders. Note that $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ is not unique for the three and five leader cases as some followers must choose arbitrarily between two or more options. The identity number v is omitted for clarity, but it is used to break ties when choosing five leaders.

5.3 Numerical Illustrations

In this section we compare the behavior and performance of the UCB-Partition algorithm with the algorithms in [50]. We show that the UCB-Partition algorithm performs well over a variety of graph structures and offers performance advantages over related algorithms.

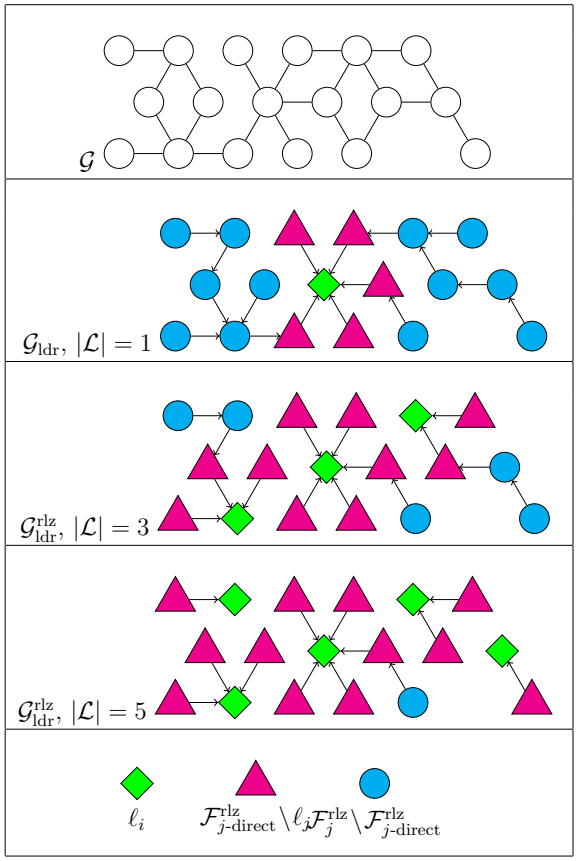


Figure 5.2: Example of a large communication graph $\mathcal{G}$ and a $\mathcal{G}_{\text{ldr}}$ (not shown) that allows for any agent to follow any neighbor in $\mathcal{G}$. Three panels show three possible realizations of $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ with one, three, and five leaders, respectively, where the leaders and followers are selected using the token passing method described in 5.2.5.

All simulations are conducted with a 2-armed bandit using rewards drawn from a Bernoulli distribution with $m = [0.5,\ 0.7]$ and $T = 10^3$ or $T = 10^4$. In Figs. 5.3 and 5.4 we show cumulative regret of the group over time for different graph structures as given in Figs. 5.1 and 5.2, respectively. The cumulative regret in our simulations are computed by averaging over 8000 Monte-Carlo runs using the UCB-Partition and UCB-Network algorithms, as well as for the case with no communication between agents.

Example 11 (***Regret for Small Graphs***)**.** Fig. 5.3 shows group cumulative expected regret for $\mathcal{G}$ and the three versions of $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ as given in Fig. 5.1. The UCB-Partition greatly improves performance over the UCB-Network algorithm, demon-

strating the benefits of imitation. Additionally, version C of $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ is a minimal dominating set partition for the FYL algorithm, so the better performance of UCB-Partition A over C here shows the advantage of the UCB-Partition algorithm over the FYL algorithm when used with suitable leaders. Finally, the the better performance of UCB-Partition A over B demonstrates the benefit of selecting agents with higher degree to be leaders.

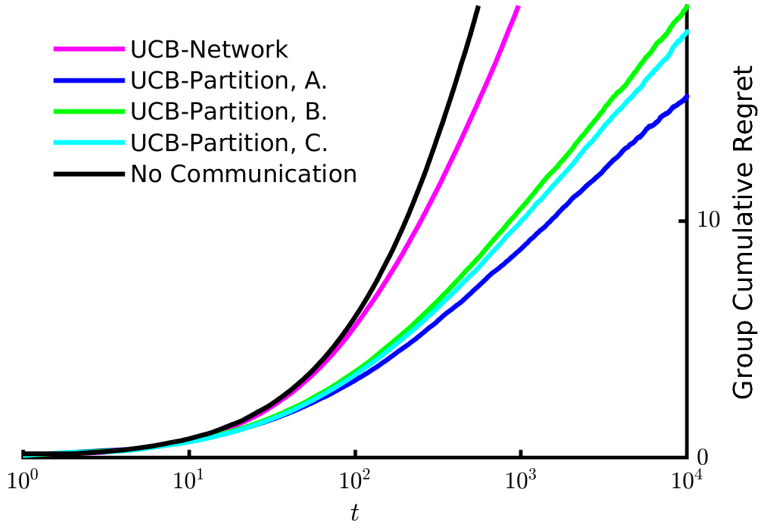


Figure 5.3: Simulation results of expected cumulative regret for the UCB-Network and UCB-Partition algorithms using $\mathcal{G}$ and $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ as given in Fig. 5.1.

Example 12 (***Regret for Large Graphs using Token Passing***)**.** Fig. 5.4 shows group cumulative expected regret for $\mathcal{G}$ and the three versions of $\mathcal{G}^{\text{rlz}}_{\text{ldr}}$ corresponding to one, three, and five leaders as given in Fig. 5.2. Here, increasing the number of leaders results in a small increase in group cumulative regret for large T, which is also reflected in the performance bounds, a phenomenon we discuss in Example 13. As in Example 1, the UCB-Partition significantly improves performance over the UCB-Network algorithm.

Example 13 (***Time Dependency of Optimal Leader Selection***)**.** Fig. 5.5 compares the relative performance of the UCB-Partition algorithm using the one,

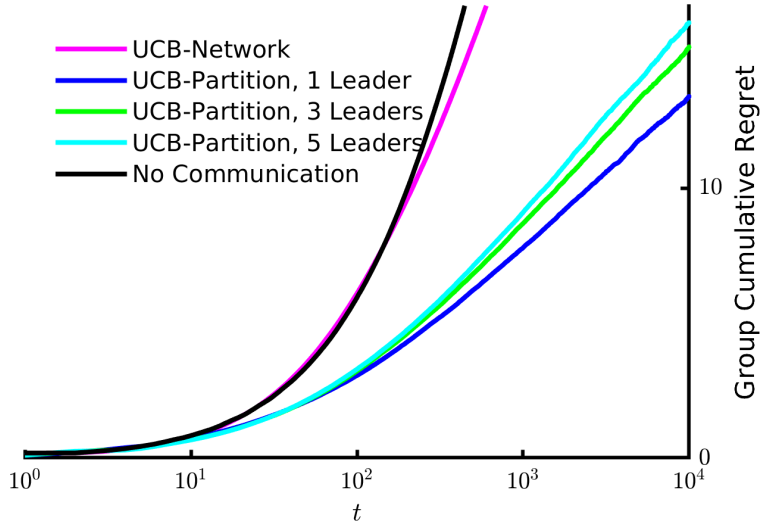


Figure 5.4: Simulation results of expected cumulative regret for the UCB-Network and UCB-Partition algorithms using $\mathcal{G}$ and $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ as given in Fig. 5.2.

three, and five leader realizations $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$ in Fig. 5.2 at each timestep t for $T = 10^3$. Early on, the three leader network outperforms the one leader network, but as t grows the one leader network begins to perform the best, a trend which can be seen continuing in Fig. 5.4.

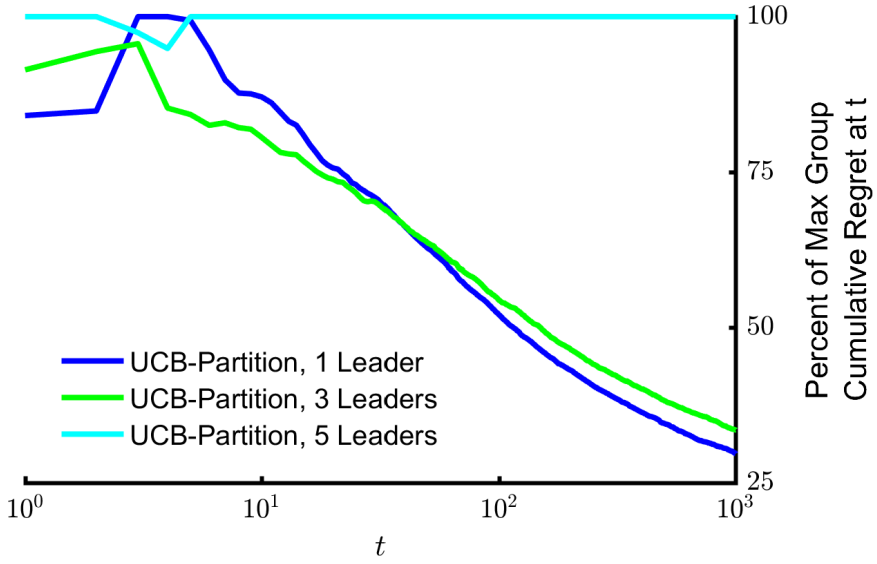


Figure 5.5: Simulation results of the expected cumulative group regret of the one, three, and five leader $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$'s in Fig. 5.2 as a percentage of the algorithm with highest regret at each time t. Lower percentage values indicate lower regret.

This is expected as bounds expressed in Theorem 7 and Corollary 1 indicate that as $T \to \infty$ the lowest regret will be obtained when the agent or agents with the highest overall degree in $\mathcal{G}$ are the only leaders. This result is indicated by the domination of the logarithmic term in the bound and is intuitive, as over large timescales it is beneficial to wait and imitate, through one's neighbors, a leader with the highest possible number of available samples.

However, for $T < \infty$ the $\mathcal{G}_{\text{ldr}}^{\text{rlz}}$-dependent constant terms in Theorem 7 and Corollary 1 can be significant relative to the logarithmic term, and having more leaders may be advantageous as this tends to reduce $|\mathcal{F}_j^{\text{rlz}}|$ and $\text{diam}_j^{\text{rlz}}$. Additionally, this factor would be particularly important for non-stationary MAB problems in social settings, where the mean rewards from the arms can change in time.

These results suggest that selecting the optimal leaders or optimal number of leaders is a function not only of $\mathcal{G}$ but also of the time horizon T. We intend to explore this trade-off in future work.

5.4 Discussion

In this chapter we investigated cooperative decision-making in networks using the cooperative multi-agent MAB problem with strictly local communication. We developed the UCB-Partition algorithm and proved bounds on its performance. Additionally, we developed a distributed policy that utilizes token-passing, does not require knowledge of the full communication graph, and can select an arbitrary number of leaders for use with the UCB-Partition algorithm. We demonstrated the utility of the UCB-Partition algorithm using several different examples of communication graphs and explored the time dependency of selecting the optimal number of leaders.

Future research directions include tightening the performance bounds of the UCB-Partition algorithm and constructing algorithms for leader selection as a function of

time. Additionally, alternative metrics for choosing when to imitate or lead may offer performance benefits, and a tight lower bound on expected regret as a function of the local communication graph remains an open problem. It would also be interesting to compare our results to studies of human or animal networks facing problems described by the MAB problem.

Chapter 6

Multi-armed Bandit based Algorithms for Localization of Radioactive Material [1]

In this chapter we utilize the MAB framework to develop algorithms for the localization of radiation sources in various environments. We build upon our earlier theoretical work on the MAB problem to inform the design of a modified UCB heuristic for radioactive material localization. Additionally, we describe a wheeled robotic platform that we designed and built for testing this algorithm in a real-world setting and present the results of several experiments.

6.1 Motivation and Background

6.1.1 Application Scenarios

Robots are well-suited to tasks that are dull, dirty, and dangerous. This description applies to many search tasks involving radioactive material, for which robots offer

[1] This chapter is partially adapted from [35]

distinct benefits. Robots can be deployed and left to investigate areas for long periods of time and using robots to investigate a space instead of humans reduces radiation exposure of personnel. Robots are also capable of making computationally heavily Bayesian inference calculations, which lend precise estimates of phenomena of interest, thereby enabling better decision-making capabilities.

We envision the role of the robot in nuclear search tasks as that of an assistant to a human operator. Our goal is to offload the laborious, dangerous tasks to the machine, while leaving the human operator to make more refined judgments about the objectives under consideration.

Our work is motivated by several application scenarios. The first is as a monitoring aid in facilities that enrich Uranium to produce fuel for reactors, or more nefariously, nuclear weapons. Uranium enrichment plants are employed world-wide to enrich the U-235 content in naturally mined Uranium for use in civilian nuclear power plants. Such facilities are often governed by treaties that permit human inspectors a limited amount of time to inspect the facility. We envision that inspector robots could accomplish this inspection task at more regular intervals and with fewer confidentiality concerns. Robots could be used to ensure that the facility is producing fuel within the guidelines of a governing treaty and detect if a portion of the facility is being used to produce fuel that is above a set enrichment level.

The second application area is locating undeclared radioactive material in a warehouse setting. A robot enabled with our algorithm could be used to efficiently find containers or items emitting radiation and provide feedback to security personnel regarding what items to target for further inspection. Such a robot could also be used to confirm the absence of such items. The algorithm developed here is particularly well-suited to such tasks as it makes no assumptions regarding facility layout.

The third application area is the mapping of contaminated areas such as the regions surrounding the Fukushima power plant in Japan. We envision that ground

robots equipped with radiation sensors and our algorithm could be used to map areas that were affected during the Fukushima disaster, thereby providing valuable information regarding contamination levels. The Safecast project [10] has effectively addressed this challenge for large areas, and we hope the solutions presented here can be used for detailed mapping of smaller areas such as houses. Our algorithm fits this task well because it allows for detection of distributed, non-point radiation sources, such as contaminated soil, and does not require a prior map of the space. Furthermore, it balances exploring the whole area with focusing on highly radioactive areas of potential interest.

6.1.2 Related Work and Goals

Multiple researchers have investigated the problem of radioactive material localization. In general, the applicability of a given solution or method is highly dependent on the assumptions made in the formation of the problem. Assumptions regarding the number of radioactive sources, source strength, the presence of shielding, and background radiation levels have a profound effect on the efficacy of a given solution.

Researchers have also focused on several different application areas. One major area is that of radiation monitoring at ports of entry, and several publications have considered the challenge of determining if a moving target is radioactive. Sun *et al.* [99, 100] considered the case of a moving target with known source strength and location, and developed evaluation strategies for a network of sensors. Others have developed strategies for the same situation but with unknown target location or source strength [65, 74, 80].

Another active area of research is the search for stationary sources of radiation using mobile robots, which is the problem we address in this chapter. Cortez *et al.* [22] formulated control laws for wheeled robots where robot velocity is based upon radiation sensor measurements from single point sources and compared this method

to simple coverage-based algorithms. Kumar *et al.* [52] followed a similar approach to localize multiple small sources, and Yadav *et al.* [108] expanded upon this to consider robot positioning error. Others have focused purely on building an accurate radiation map of a known space [23, 40, 69]. Recent work has also focused on fast localization of strong point sources using aerial robots [67].

In contrast to the above work, the algorithm developed here is designed to efficiently identify areas that experience radiation levels above a set threshold, rather than localizing a specific source. This output will give a human operator a map corresponding to interesting or dangerous locations to inspect or avoid, which is highly useful in the applications described in Section 6.1.1.

The solution developed in this chapter makes very few assumptions regarding the localization task at hand. We assume that there may be any number of sources of any strength, and that there may be shielding or reflective surfaces in the environment. Additionally, we do not assume that sources are point sources, but may be distributed, such as in contaminated soil. We have avoided making restrictive assumptions on the search environment in order to maximize the algorithm's efficacy in real-world scenarios and increase robustness.

6.1.3 Search as an MAB Problem

The MAB problem has previously been used as a model for search in several contexts. Srivastava *et al.* [96] used the MAB problem to model surveillance, and focused on scenarios where the rewards can change over time. Reverdy *et al.* [84] used the MAB problem to model human search tasks in a simplified computer game where users explored a smooth stationary reward surface, and Srivastava *et al.* [95] used the MAB problem to model search and foraging. The authors demonstrated that simple UCB type heuristics can closely approximate the behavior of skilled human operators in performing search tasks.

The MAB problem is well-suited as a modeling framework for real-world search problems for several reasons. The first reason is the ubiquity of noise. Noisy information is a persistent feature of real-world search tasks, and a searcher will often have to make sequential decisions under uncertainty. For example, in a search-and-rescue mission for a missing hiker, a rescuer will need to decide where to search next on the basis of noisy inputs such as eye-witness reports, cell phone records, and tracks. Noise is an fundamental feature of MAB problems, and therefore such problems serve as useful models for many real search tasks.

The second reason is the clear analog between reward in an MAB task and the goal of a search task. Returning to the example of the missing hiker, one could formulate reward as the recent presence of the missing person, with the highest reward obtained for finding them. In the context of the MAB problem this reward formulation will push searchers to hone in on where they guess the hiker is located while also making balanced judgments about exploring potentially rewarding areas.

The third reason is the easy adaptability of MAB tasks to various real-world conditions. Researchers have considered many variants of the classical MAB problem, such as time-varying rewards, correlated rewards, and different reward distributions, that can be used to model different real-world scenarios. Many MAB heuristics, such as UCB, are also computationally efficient, which is a necessary requirement for practical application.

In this chapter we consider a variant of the MAB problem that is well-suited to the radiation localization task. In particular we consider correlated, non-time varying Gaussian rewards under a satisficing formulation.

6.1.4 The Satisficing Problem

In many practical decision-making scenarios a decision-maker is not interested in necessarily finding the best option, but only reliably finding a sufficient option. For

example, a foraging animal may desire to find a reliable food location that can provide a day's meal, but not care if it is the most plentiful location available. This encodes the intuitive result that some decision-makers only care about reliably meeting a need. This decision-making objective has been formally investigated as the *satisficing* problem [92], a portmanteau combining satisfying and sufficing. It has been used to model decision-making in a wide variety of fields, including economics [8], management science [71], design optimization [73, 109], and control theory [37].

The satisficing objective has also been investigated in MAB problems [83]. In this context a decision-making agent seeks to obtain a reward value that is *satisfying*, defined as being above a user-defined threshold $\mathcal{M}$. The decision-maker also seeks to sample arms that are *sufficient*, defined as reliably yielding satisfying rewards with probability $1-\delta$ with $\delta \in [0, 1]$. The goal of a decision-maker is therefore to maximize the objective

$$\sum_{t=1}^{T} \mathbb{1}\left\{\mathbb{P}\left(m_{i(t)} \geq \mathcal{M}\right)\right\} \tag{6.1}$$

where m_i is the mean of arm i and $i(t)$ is the arm sampled at time t.

Reverdy *et al.* [83] also introduced the concept of *expected satisficing regret*, given by

$$R(t) = \max\{\mathcal{M} - m_{i(t)}, 0\} \cdot \mathbb{1}\left\{\mathbb{P}\left(m_{i(t)} \leq \mathcal{M}\right)\right\} \tag{6.2}$$

and defined the *Satisficing-In-Mean-Reward* MAB Problem, which is to minimize the cumulative satisficing regret.

This problem can take several different forms depending on the values of the arm means, $\mathcal{M}$, and δ. In this chapter we are concerned with one particular form, termed $(\mathcal{M}, \delta)$-satisficing in [83], where multiple arms have means above $\mathcal{M}$ and $\delta > 0$, implying that the decision-maker accepts less than complete certainty that an arm is satisfying. $(\mathcal{M}, \delta)$-satisficing corresponds well to the application scenario where there are multiple locations with high (above $\mathcal{M}$) radiation measurements in

an environment and the operator aims to discover their location with a given error rate.

6.2 Search using Satisficing

In this section we describe the Radiation Upper Confidence Limit (rad-UCL) algorithm and Gaussian process regression.

6.2.1 Gaussian Process Regression

A Gaussian process (GP) is defined in [81] as a "collection of random variables, any finite number of which have a joint Gaussian distribution." Intuitively, if one envisions a GP with each random variable representing a physical point in space, the expected values of a set of points would therefore describe a surface that is somewhat smooth, provided that the covariance function of the joint Gaussian distribution assumes nearby points are highly correlated.

Here we utilize the notation of [81] and define $K(X_1, X_2)$ as the covariance matrix between each of the points in the sets X_1 and X_2. This covariance matrix is produced by a covariance function, or kernel, which defines a relationship between any two input points. There are several different covariance functions that are typically used by researchers, and each one lends itself to different physical interpretations. Here we utilize the most common covariance function, exponential-squared, defined as

$$\exp\left(-\frac{1}{L}|\mathbf{x}_1 - \mathbf{x}_2|^2\right), \tag{6.3}$$

where $\mathbf{x}_i$ is the location of point i and L is a length scale parameter. We use 6.3 to calculate each of the values in $K(X_1, X_2)$ and it encodes the intuitive result that the correlation between points falls off exponentially with distance.

In this work we are concerned with Gaussian process regression, which aims to fit underlying data using a GP to enable predictions of data values at other points. GP regression has a long history in spatial measurement tasks, such as oil field characterization and measurements of radioactive contamination [28, 106]. GP regression is widely used in practice for two reasons. The first is the relative ease of incorporating modeling features such as prior information, assumptions on correlation, and noisy data. The second is the computational tractability of solving the regression problem. Both of these features are highly attractive for our current task.

The GP regression process [81] to calculate posterior expected value and variance is

$$\boldsymbol{\mu}(t) = K(X_*, X)[K(X, X) + V(t)]^{-1}\boldsymbol{y}(t) \tag{6.4}$$

$$\Sigma(t) = K(X_*, X_*) - K(X_*, X)[K(X, X) + V(t)]^{-1}K(X, X_*). \tag{6.5}$$

In the above $\Sigma(t)$ is the posterior variance at time t, X is the set of all points in the space, X_* is the set of all points for which measurement, or training, data exists, $\boldsymbol{y}(t)$ is a vector of training data for points in X_*, and $\boldsymbol{\mu}(t)$ is the posterior expected value of all points in X. Furthermore $V(t)$ is a matrix with zeros except for the diagonal entries, with the i'th diagonal entry equal to the sample variance of the i'th data point. In practice the posteriors defined in (6.4) and (6.5) are typically computed using Algorithm 2.1 in [81], which is the method we employ using the numpy package in python. This method uses a Cholesky decomposition to achieve faster computation and numerical stability, but accomplishes the same basic steps as (6.4) and (6.5).

Gaussian Process Regression and the MAB Problem

The GP-UCB algorithm, defined and analyzed in [94], considers an MAB problem where correlations between arm means can be modeled as a GP. This is highly ap-

plicable to MAB problems where arms represent locations in physical or parameter space with some correlation between arms.

The GP-UCB algorithm uses GP regression to estimate the arm means and variances at given points in a space, where each point represents an arm with an underlying reward mean and variance. The algorithm then uses a UCB type heuristic that combines the mean estimates with an exploration term composed of the posterior variance and a function of time to make a decision about which arm to sample next. In practice GP-UCB is very similar to the classical UCB algorithm [5] or UCL algorithm [84], but employs GP regression to estimate the posterior mean and variance of arms.

In the search task presented here we do not use GP-UCB directly, but rather take inspiration from GP-UCB in our use of GP regression.

6.2.2 The Rad-UCL Algorithm

The rad-UCL algorithm takes inspiration from the GP-UCB algorithm in [94] and the $(\mathcal{M}, \delta)$-satisficing UCL algorithm in [83]. The rad-UCL algorithm executes the following steps for each time $t \in \{1, \ldots, T\}$ in the search problem:

(i) Radiation Field Estimation: All space that is accessible to the robot is gridded into measurement cells. Each measurement cell is associated with a cumulative counter of the time in seconds $d_i(t)$ spent in cell i by any detector at time t, and the radiation counts $c_i(t)$ received by a detector while in that cell. Using these two values, the robot calculates the count rate in counts per second as $y_i(t) = \frac{c_i(t)}{d_i(t)}$, which is the i'th entry of $\boldsymbol{y}(t)$, and variance $\frac{c_i(t)}{d_i(t)^2}$, which is the i'th diagonal entry of $V(t)$. The robot calculates $K(\cdot, \cdot)$ using (6.3). The robot then uses these values to perform GP regression over all accessible measurement cells, producing a posterior estimate of count rate and variance.

(ii) Robot Decision-Making: For each accessible measurement cell the robot calculates

$$Q_i(t) = \mu_i(t) + V_{ii}(t)\,\Phi^{-1}\left(1 - \frac{\delta_1}{3}\right) \tag{6.6}$$

and defines cells for which $Q_i(t) > \mathcal{M}$ as "above threshold." Each robot also calculates

$$W_i(t) = \mu_i(t) - V_{ii}(t)\,\Phi^{-1}\left(1 - \frac{\delta_2}{3}\right) \tag{6.7}$$

and defines cells for which $W_i > \mathcal{M}$ as "alarmed." $\Phi^{-1}(\cdot)$ is the standard Gaussian inverse cumulative distribution function.

The robot then selects the nearest cell that is above threshold and not alarmed, where distance is defined as the length of the path to reach that cell given robot constraints. The robot then travels to the selected cell.

The $Q_i(t)$ term of the rad-UCL algorithm functions in the same manner as the $Q_i(t)$ terms presented in prior chapters, and is designed to drive the robot to explore new areas while also biasing motion toward areas with high radiation. Furthermore, we wish for the robot to move on after determining with high certainty that a cell has a high count rate. The $W_i(t)$ based alarmed condition accomplishes this goal, as it effectively excludes from consideration any cell that the robot can confidently determine has $m_i > \mathcal{M}$.

6.2.3 Practical Considerations

Here we list several practical considerations for implementing the rad-UCL algorithm.

(i) The GP regression algorithm in Algorithm 2.1 in [81] involves a Cholesky decomposition, for which computation time scales with the cube of the size of the input. This Cholesky decomposition thus becomes intractable as the number of accessible measurement cells grows and the robot collects more measurements.

Cubic scaling can be avoided by limiting the regression to a local area surrounding the robot while simply copying the estimates from grid cells outside this area. This modification results in a significant performance increase and introduces only very small errors in the overall regression.

(ii) The rad-UCL algorithm involves computation of multiple values at each accessible measurement cell, with a subsequent comparison. It is important to optimize these computations within the coding language used. Additionally, the computation time will scale linearly with the number of accessible measurement cells, which can become intractable for large maps. In these cases the algorithm can be restricted to evaluate accessible measurement cells within a local area first, expanding later if necessary.

6.3 Robot Hardware

In this section we describe the robot hardware purchased or built for the purposes of algorithm testing and validation.

6.3.1 The Turtlebot3 Burger Platform

We conducted our experiments using a modified Turtlebot3 Burger robot purchased from Robotis for $550. The Burger, pictured in Figure 6.1, is a small, modular low-cost robot that is commonly used for robotics classes. It has two drive wheels and a back caster, with a maximum translational velocity of $0.22\frac{m}{s}$. The Burger comes with a motor controller board, Raspberry Pi 3, and a small LiDAR unit, which is discussed in greater detail in Section 6.3.3.

The Burger is designed to be used with Robotics Operating System (ROS). ROS provides reliable control and message passing protocols for use with various components, such as LiDAR and odometers. The message passing protocols in ROS are

Figure 6.1: Turtlebot3 Burger robot. Photo credit: Robotis

controlled through a ROS "roscore" program which ensures consistent timing between devices. ROS is also widely used in the robotics research community. In this work we used ROS Kinetic.

We selected the Turtlebot3 Burger platform for several reasons. Most importantly, the low cost of the Burger will allow other researchers to utilize our designs to further test radiation localization or mapping algorithms. There is also a large community supporting the platform, and it utilizes common architectures in robotics.

6.3.2 Radiation Detection

In order to enable radiation sensing capabilities on-board the Turtlebot 3 Burger we designed and constructed an additional "layer" for the robot that fits in between the Raspberry Pi 3 and the LiDAR unit, as seen in Figure 6.3. This layer consists of a 3D-printed ring, on which are three mounting points for Geiger detectors. We also

3D-printed cases for the Geiger detectors and their associated electronics, shown in Figure 6.2. We utilize the same LND 7314 2in Pancake Geiger detector that is used by the Safecast project [10] in their bGeigie Nano detector unit. Inside the mounting ring we placed an Arduino Nano, which, when connected to the Geiger detectors, processes the incoming detections and relays the information to the primary ROS control system.

The entire radiation detection system is designed to be cheap, modular, and simple to modify. The system can easily accommodate more detectors, and can be mounted onto other robots with minimal modification. We have published all our designs, available along with a parts list and assembly instructions at `https://hackaday.io/project/158327-geigerros`.

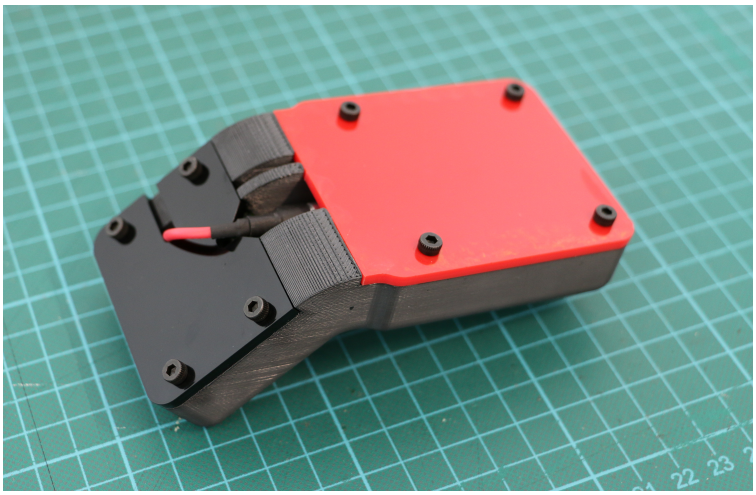


Figure 6.2: The Geiger radiation sensor in custom 3D-printed case used in Section 6.4. Note that the Geiger tube is under the red cover, and the associated electronics are under the black cover.

6.3.3 LiDAR and SLAM

We use a LiDAR (Light Detection And Ranging) based SLAM (Simultaneous Location And Mapping) algorithm to perform mapping for both the purposes of obstacle avoidance and measurement localization. LiDAR based SLAM is the current de-facto standard for mapping in robotics. Many different SLAM algorithms exist in the lit-

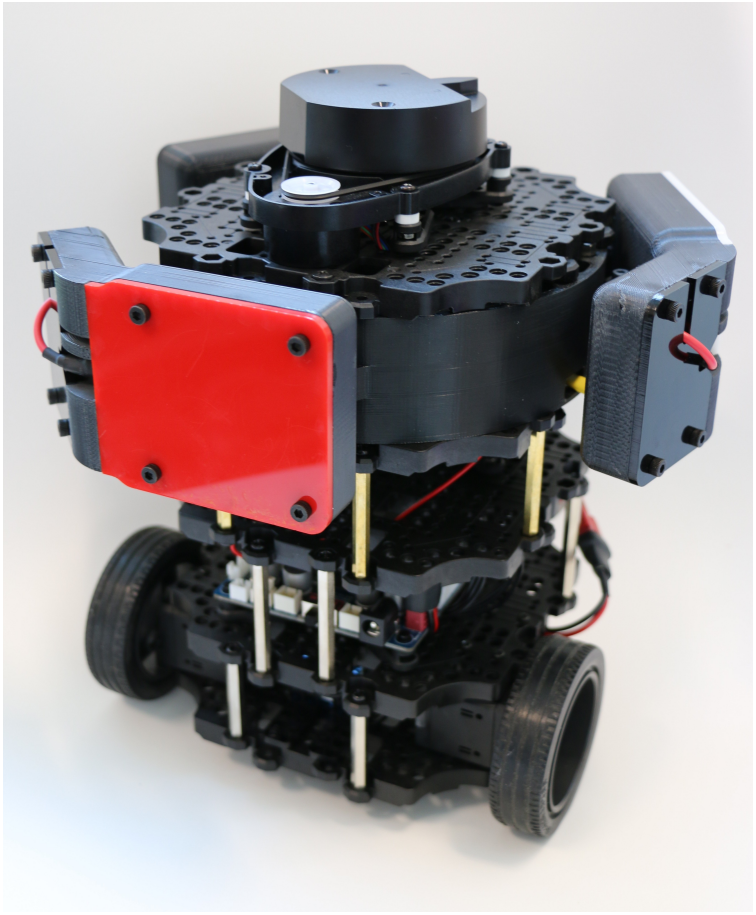


Figure 6.3: Turtlebot3 Burger robot with three Geiger radiation detectors used in Section 6.4.

erature, see [15] for a review. In general, a SLAM algorithm will use the range data from the LiDAR sensor in addition to other sensors such as odometers to map a space while also determining the robot's location. In this context, mapping a space means that the robot produces an accurate map encoding distances and obstacles in an area. In practice, this map is encoded as an occupancy grid, where the algorithm determines if a given grid cell corresponding to a small real-world area is free or occupied by another object such as a wall or other vehicle.

On the Burger robot we used the default Robotis LDS-01 laser scanner with a 360 degree viewing angle, 1 degree angular resolution, and 3.5 m maximum range. Compared to other LiDAR units this model is fairly weak and inexpensive, and we aim to demonstrate that our algorithm works well even with such a limited sensor.

We employ the ROS gmapping SLAM package to perform SLAM with a 5 cm grid cell resolution, and extend it using the frontier-exploration package for waypoint navigation. These packages are both commonly used within the ROS community.

6.4 Experiments

We conducted our experiments in Professor Alex Glaser's lab in room J207 in the Engineering Quadrangle at Princeton University. The lab covers approximately 30 square meters, split into two rooms separated by a short corridor. The floor plan, as generated by the SLAM algorithm, is shown in Figure 6.4. The floor is level and covered in synthetic wood flooring. For our experiments in Section 6.4.1 we placed two radiation sources in the room. We placed Source A on the rim of the mock warhead in the first room and it is comprised of Na-22, Cs-137, and Co-60 check sources with a total activity of 11.6×10^4 Becquerel. We placed Source B in the yellow case in the second room and it is comprised of Co-60 and Ba-133 check sources along with 201 g of thorated welding rods with a total activity of 5.9×10^4 Becquerel. When placed directly on the Geiger detector Source A and Source B produce 536 and 278 counts per second, respectively. The low detection rates are reasonable given the low detection efficiency of Geiger detectors. Detection efficiency also differs for different check sources because it is related to the energy of the emitted gamma radiation. Both sources are located at the height of the detectors and the source locations are indicated in Figures 6.4 and 6.5.

As per the recommendation of Robotis, we are using a 5 cm SLAM occupancy map resolution, and use this same resolution in all our maps of the space and the rad-UCL algorithm. Additionally, we used $L = 0.2$ and $\mathcal{M} = 1.75$, which provide good mapping and detection performance given our radiation environment.

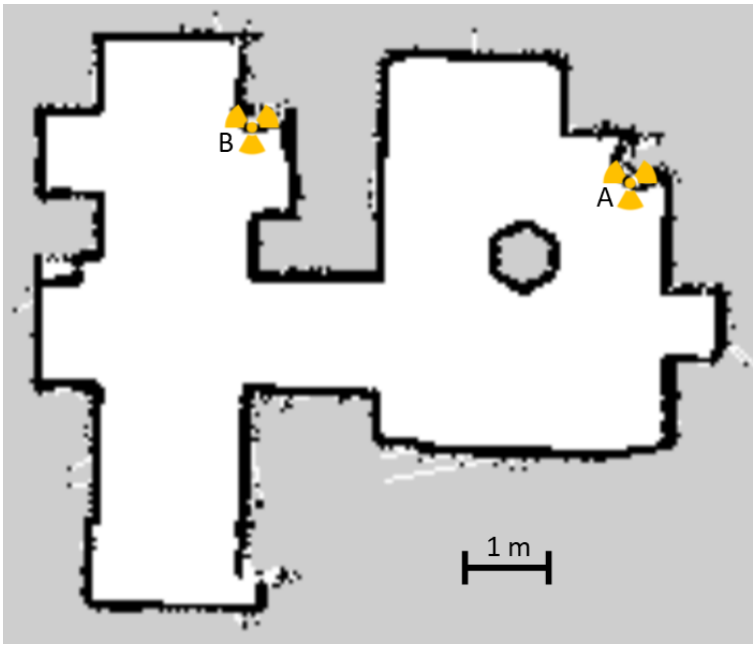


Figure 6.4: SLAM generated occupancy map of the room used in Section 6.4. In this map black, white, and gray cells indicate occupied, unoccupied, and unclassified cells, respectively. Also marked on the map in yellow are the locations of the two radioactive sources used in the experiments discussed in Section 6.4.1.

For ease of control, we use a remote ROS master on a laptop connected to the local network. This laptop runs a roscore, the SLAM algorithm, the frontier exploration waypoint navigation algorithm, and the principal decision-making control loop. The decision-making control loop uses rad-UCL at 10 Hz to select target locations to visit based on the SLAM generated occupancy map. This target location is then passed to the frontier exploration waypoint navigation algorithm, which computes a path to the target location and outputs appropriate motor commands.

The Raspberry Pi 3 on the Burger processes the data from the on-board sensors, including the radiation detectors. The processes running on the laptop could easily be run on the Raspberry Pi 3 for full on-board control.

6.4.1 Results and Discussion

Videos 4, 5, 6, and 7 show several runs of the Rad-UCL algorithm, also available online at `youtu.be/wcB-jaEfyEc`, `youtu.be/IvjDl6jDCNc`, `youtu.be/NEO0q6uZC2E`,

Figure 6.5: Composite overhead camera view of the room used in 6.4. Also marked on the map are the locations of the two radioactive sources used in the experiments discussed in Section 6.4.1.

and `youtu.be/hQ48ajOEawc`. Each video depicts a ceiling view of the test environment, along with the occupancy map, posterior mean, robot location, and alarmed measurement cells. Figure 6.6 depicts a screenshot from the middle of one run for reference.

The robot successfully finds both radiation sources in all videos, and successfully traverses and maps the entire space. Given the stochastic nature of the problem the time required to find the sources varies between runs, and we have included examples of both long and short detection times in the above videos.

Overall, the robot conducts a fairly efficient search for the sources, and moves on once a source has been identified with satisfactory certainty. The method is also adaptive, and lingers in areas with relatively high radiation count rates in order to gain confidence in its estimate. In all the videos one can see points where the robot detects a spurious higher than average count rate, depicted in the posterior mean

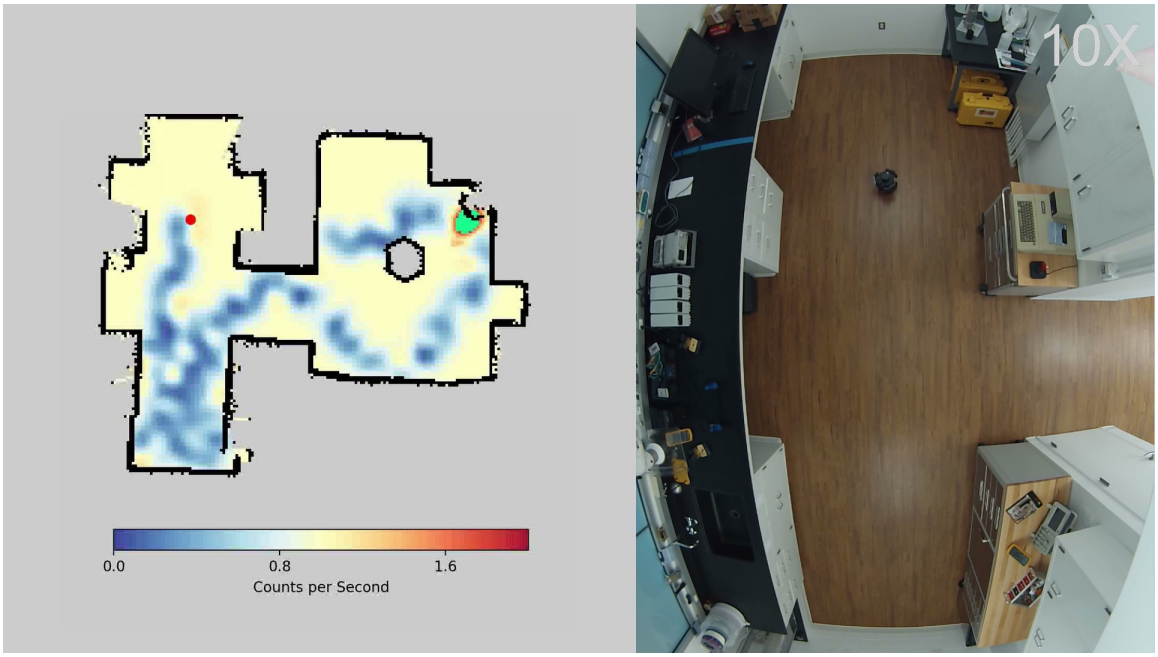


Figure 6.6: Screenshot from a video of an experiment in Section 6.4.1 demonstrating the video layout. The panel on the right shows an overhead camera view of the room. The panel on the left shows the SLAM generated occupancy map, posterior mean count rate, robot location (the red dot), and alarmed measurement cells (green cells).

map, which are then corrected once the robot lingers or reevaluates the space. The radiation environment for these experiments is relatively simple as it consists of two point sources. However, we believe that the rad-UCL algorithm will extend seamlessly to more complex environments, such as those with distributed radiation sources. For safety reasons we were not able to utilize such sources in our testing, but hope to in the future.

One significant drawback of the method used here is that it lends itself to trajectory paths that are not optimal in terms of coverage time or energy expenditure. One possible extension to this work is therefore to employ hybrid methods which would utilize another method, such as an optimal coverage algorithm or gradient descent, in conjunction with a satisficing control law. The satisficing component could be used for control in certain parameter regimes, or for selecting the parameters of some other algorithm (such as the distance between paths). These algorithms would need to be compatible to diverse radiation environments and map sizes.

Chapter 7

Final Remarks

In this chapter we summarize the work presented in this thesis and propose several future directions of study.

7.1 Summary

In this thesis we have considered the multi-agent MAB problem in several different permutations. In Chapter 3 we first considered the multi-agent MAB problem with communication through consensus, and developed the coop-UCB1, coop-UCB2, and coop-UCL algorithms. We proved upper bounds on expected cumulative regret for each algorithm. Furthermore, we considered numerical simulations and robotic experiments that demonstrated the utility of these algorithms and the new explore-exploit centrality measure.

In Chapter 4 analyzed the case of multi-agent MAB with collisions with applications to the cognitive radio spectrum access problem. We developed and proved bounds on expected cumulative regret for the coop-UCB2 collisions algorithm, and performed a robotic search experiment.

Next, in Chapter 5 we considered the multi-agent MAB problem with strictly local communication and developed a partition-based algorithm that allows agents to

imitate leaders in the network. This algorithm can markedly improve performance over algorithms that cannot imitate or can only imitate in a limited manner, and we also developed a distributed algorithm for leader selection.

Finally, we considered the problem of robotic search for nuclear material in a facility. We leveraged our previous analyses of the multi-agent MAB problem to develop and apply the rad-UCL algorithm. We built and tested a small radiation detecting robot using the rad-UCL algorithm and demonstrated its performance in a real-world setting.

7.2 Future Directions

The most promising avenues of future study are those that will enable the algorithmic solutions presented here to move from the pages of a thesis to use in practical, real-world applications. While networked sequential decision-making under uncertainty is a common feature of real-world challenges, several other features of practical problems are crucial as well. Here we present several research avenues that would be impactful and for which there is hope of gaining analytical tractability.

7.2.1 Multi-agent MAB

The principal avenue of improvement regarding the multi-agent MAB problem is to more throughly consider time-varying problem parameters. In practical scenarios virtually every aspect of a sequential decision-making problem can change on some time-scale. Capturing the changing and dynamic nature of practical problems will lead to a much richer set of solutions that are more broadly applicable. Here we give two examples of how a deeper consideration of time-varying parameters could lend new insights.

The first example is time-varying rewards. There is a significant volume of work on this problem in the single-agent case [26, 32, 38, 77, 96, 102], but the extension to multiple agents presents unique challenges and potential solutions. A detailed study of time-varying rewards will allow an investigation of how new knowledge propagates through networks and how networked decision-makers respond.

The second example is time-varying communication graphs. The communication graphs that connect decision-makers in real networks are often time-varying or even dynamic [18, 76]. Social-networks and mobile robot communications in particular face this challenge. One could consider deterministically [72] or randomly changing graphs [41, 42, 79], or link the graph to the arm choice of individual agents.

Beyond incorporating time-varying parameters, there are several other potential areas of future research. Obtaining asymptotically optimal upper bounds for a decision-making algorithm could give deeper insight into how the graph affects decision-making performance and also create new opportunities to design algorithms that improve performance. There is some work on asymptotically optimal bounds for single-agent algorithms [16, 25], but those methods are not currently amenable to the probabilistic methods used here. Additionally, obtaining tight lower bounds on performance that depend on graph structure would yield similar benefits.

We hope that future researchers will take up these challenges.

7.2.2 Search Algorithms

There are many potential avenues for improvement when using MAB algorithms for locating nuclear material and for other similar research problems. One principle area is to employ *hybrid* methods, which would use rad-UCL to make decisions in concert with another algorithm that could promote alternate goals like optimal coverage. One could also incorporate gradient information, as well as a noisy gradient ascent algorithm that would be useful for locating stronger sources.

UCB type solutions to multi-agent MAB problems have the potential to be widely applicable in robotic search problems. They are analytically tractable, easy to understand, and elegantly avoid the curse of dimensionality. We hope that others will utilize the methods presented here and any future extensions to enable real-world robotic search tasks.

Appendix A

Supplementary Material

Here we provide a list of the supplementary files associated with this thesis. All files are videos that show robot implementations of the relevant examples.

A.1 Supplemental Videos

Video 1.

Creator: Peter Landgren

Description: See Example 7 in Section 3.5.2.

Filename: CoopUCB2_Undirected.mp4

Link to online copy: `https://youtu.be/INzy1zeGlis`

Video 2.

Creator: Peter Landgren

Description: See Example 9 in Section 3.5.2.

Filename: CoopUCB2_Directed.mp4

Link to online copy: `https://youtu.be/ZZNn-ud8900`

Video 3.

Creator: Peter Landgren, Paul Reverdy, Vaibahv Srivastava, and Naomi E. Leonard.

Description: See Example 9 in Section 3.5.2.

Filename: CoopUCB2_Collisions.mp4

Link to online copy: `https://youtu.be/c3ev-wKAEZA`

Video 4.

Creator: Peter Landgren and Moritz Kuett

Description: See Section 6.4.1.

Filename: radUCL_Run1.mp4

Link to online copy: `https://youtu.be/wcB-jaEfyEc`

Video 5.

Creator: Peter Landgren and Moritz Kuett

Description: See Section 6.4.1.

Filename: radUCL_Run2.mp4

Link to online copy: `https://youtu.be/IvjDl6jDCNc`

Video 6.

Creator: Peter Landgren and Moritz Kuett

Description: See Section 6.4.1.

Filename: radUCL_Run3.mp4

Link to online copy: `https://youtu.be/NEO0q6uZC2E`

Video 7.

Creator: Peter Landgren and Moritz Kuett

Description: See Section 6.4.1.

Filename: radUCL_Run4.mp4

Link to online copy: `https://youtu.be/hQ48ajOEawc`

Bibliography

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1964.

[2] A. Anandkumar, N. Michael, A. K. Tang, and A. Swami. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications*, 29(4):731–745, 2011.

[3] V. Anantharam, P. Varaiya, and J. Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part I: I.I.D. rewards. *IEEE Transactions on Automatic Control*, 32(11):968–976, Nov 1987.

[4] K. J. Arrow. *Review of Social Economy*, 11(1):94–96, 1953.

[5] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[6] D. A. Berry and B. Fristedt. *Bandit problems*. Chapman and Hall Ltd., London, UK, 1985. ISBN 0-412-24810-7.

[7] B. Bollobás. *Random Graphs*. Springer, 1998.

[8] R. Bordley and M. LiCalzi. Decision analysis using targets instead of utility functions. *Decisions in Economics and Finance*, 23(1):53–74, May 2000.

[9] P. Braca, S. Marano, and V. Matta. Enforcing consensus while monitoring the environment in wireless sensor networks. *IEEE Transactions on Signal Processing*, 56(7):3375–3380, 2008.

[10] A. Brown, P. Franken, S. Bonner, N. Dolezal, and J. Moross. Safecast: successful citizen-science for radiation measurement and communication after fukushima. *Journal of radiological protection*, 36 2:S82–S101, 2016.

[11] S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[12] S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59:7711–7717, 2013.

[13] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff. Multi-armed bandits in the presence of side observations in social networks. *52nd IEEE Conference on Decision and Control*, pages 7309–7314, 2013.

[14] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff. Stochastic bandits with side observations on networks. In *SIGMETRICS*, 2014.

[15] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[16] O. Cappé, A. Garivier, O.-A. Maillard, R. Munos, G. Stoltz, et al. Kullback–leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

[17] S. Caron, B. Kveton, M. Lelarge, and S. Bhagat. Leveraging side observations in stochastic bandits. In *Conference on Uncertainty in Artificial Intelligence*, 2012.

[18] S. H. Cen, V. Srivastava, and N. E. Leonard. On robustness and leadership in markov switching consensus networks. pages 1701–1706, 2017.

[19] M. Y. Cheung, J. Leighton, and F. S. Hover. Autonomous mobile acoustic relay positioning as a multi-armed bandit with switching costs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3368–3373, Tokyo, Japan, November 2013.

[20] J. D. Cohen, S. M. McClure, and A. J. Yu. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481):933–942, 2007.

[21] L. Conradt and T. J Roper. Group decision-making in animals. *Nature*, 421: 155–8, 02 2003.

[22] R. A. Cortez, X. Papageorgiou, H. G. Tanner, A. V. Klimenko, K. N. Borozdin, and W. C. Priedhorsk. Experimental implementation of robotic sequential nuclear search. In *Mediterranean Conference on Control Automation*, pages 1–6, June 2007.

[23] R. A. Cortez, H. G. Tanner, and R. Lumia. Distributed robotic radiation mapping. In O. Khatib, V. Kumar, and G. J. Pappas, editors, *Experimental Robotics*, pages 147–156, Berlin, Heidelberg, 2009. Springer. ISBN 978-3-642-00196-3.

[24] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.

[25] W. Cowan, J. Honda, and M. N. Katehakis. Normal bandits of unknown means and variances: Asymptotic optimality, finite horizon regret bounds, and a solution to an open problem. *Journal of Machine Learning Research*, 2015.

[26] L. DaCosta, A. Fialho, M. Schoenauer, and M. Sebag. Adaptive operator selection with dynamic multi-armed bandits. In *Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 913–920, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-130-9.

[27] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[28] P. Diggle and P. J. Ribeiro. *Model-based geostatistics*. Springer Science Business Media, LLC, 2010.

[29] E. Fehr and C. Camerer. Social neuroeconomics: The neural circuitry of social preferences. trends in cognitive sciences. *Trends in Cognitive Sciences*, 11:419–27, 11 2007.

[30] Y. Gai and B. Krishnamachari. Distributed stochastic online learning policies for opportunistic spectrum access. *IEEE Transactions on Signal Processing*, 62 (23):6184–6193, 2014.

[31] A. Garivier and E. Moulines. On upper-confidence bound policies for non-stationary bandit problems. *arXiv preprint arXiv:0805.3415*, 2008.

[32] A. Garivier and E. Moulines. On upper-confidence bound policies for switching bandit problems. In J. Kivinen, C. Szepesvári, E. Ukkonen, and T. Zeugmann, editors, *Algorithmic Learning Theory*, pages 174–188, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-24412-4.

[33] J. Gittings, K. D. Glazebrook, and R. Weber. *Multi-armed bandit allocation indices*. Wiley, 2011.

[34] J. C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):148–177, 1979.

[35] R. J. Goldston, A. Glaser, M. Keutt, P. Landgren, and N. E. Leonard. Autonomous mobile directionally and spectrally sensitive neutron detectors. In *Internation Atomic Energy Agency Symposium on International Safeguards*, Vienna, Austria, Nov 2018.

[36] B. Golub and M. O. Jackson. Naïve learning in social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2(1):112–149, 2010.

[37] M. A. Goodrich, W. C. Stirling, and R. L. Frost. A theory of satisficing decisions and control. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(6):763–779, Nov 1998.

[38] O.-C. Granmo and S. Berg. Solving non-stationary bandit problems by random sampling from sibling kalman filters. In N. García-Pedrajas, F. Herrera, C. Fyfe, J. M. Benítez, and M. Ali, editors, *Trends in Applied Intelligent Systems*, pages 199–208, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-13033-5.

[39] B. R. Greening, N. Pinter-Wollman, and N. H. Fefferman. Higher-order interactions: Understanding the knowledge capacity of social groups using simplicial sets. *Current Zoology*, 61(1):114–127, Jan 2015.

[40] A. Hafiz Zakaria, Y. Mohd Mustafah, J. Abdullah, N. Khair, and T. Abdullah. Development of autonomous radiation mapping robot. *Procedia Computer Science*, 105:81–86, 12 2017.

[41] M. T. Hale and M. Egerstedt. Convergence rate estimates for consensus over random graphs. pages 1024–1029, Seattle, USA, 2017.

[42] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, 50(11):1867–1872, Nov 2005.

[43] A. M. Hein and S. A. McKinley. Sensing and decision-making in random search. *Proceedings of the National Academy of Sciences*, 109(30):12070–12074, 2012.

[44] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[45] D. Kalathil, N. Nayyar, and R. Jain. Decentralized learning for multiplayer multiarmed bandits. *IEEE Transactions on Information Theory*, 60(4):2331–2345, 2014.

[46] S. Kar, H. V. Poor, and S. Cui. Bandit problems in networks: Asymptotically efficient distributed allocation rules. In *IEEE Conference on Decision and Control and European Control Conference*, pages 1771–1778, Orlando, FL, Dec. 2011.

[47] E. Kaufmann, O. Cappé, and A. Garivier. On Bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600, La Palma, Canary Islands, Spain, Apr. 2012.

[48] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I : Estimation Theory*. Prentice Hall, 1993.

[49] T. Keasar, E. Rashkovich, D. Cohen, and A. Shmida. Bees in two-armed bandit situations: foraging choices and possible decision mechanisms. *Behavioral Ecology*, 13(6):757–765, 2002.

[50] R. K. Kolla, K. Jagannathan, and A. Gopalan. Collaborative learning of stochastic bandits over a social network. In *Allerton Conference on Communication, Control, and Computing*, pages 1228–1235, Sept 2016.

[51] J. R. Krebs, A. Kacelnik, and P. Taylor. Test of optimal sampling by foraging great tits. *Nature*, 275(5675):27–31, 1978.

[52] A. Kumar, H. G. Tanner, A. V. Klimenko, K. Borozdin, and W. C. Priedhorsky. Automated sequential search for weak radiation sources. In *Mediterranean Conference on Control and Automation*, pages 1–6, June 2006.

[53] L. Lai, H. Jiang, and H. V. Poor. Medium access in cognitive radio networks: A competitive multi-armed bandit framework. In *Asilomar Conference on Signals, Systems and Computers*, pages 98–102, Pacific Grove, CA, Oct. 2008. IEEE.

[54] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[55] P. Landgren, V. Srivastava, and N. E. Leonard. Distributed multi-agent multi-armed bandits. In preparation.

[56] P. Landgren, V. Srivastava, and N. E. Leonard. Distributed cooperative decision-making in multiarmed bandits: Frequentist and bayesian algorithms. In *IEEE Conference on Decision and Control*, 2016.

[57] P. Landgren, V. Srivastava, and N. E. Leonard. On distributed cooperative decision-making in multiarmed bandits. In *European Control Conference*, Aalborg, Denmark, June 2016. arXiv preprint arXiv:1512.06888.

[58] P. Landgren, P. Reverdy, V. Srivastava, and N. E. Leonard. Multi-robot foraging using the graphical multiarmed-bandit framework. In *ICRA Workshop on Informative Path Planning and Adaptive Sampling*, Brisbane, Australia, May 2018.

[59] P. Landgren, V. Srivastava, and N. E. Leonard. Social imitation in cooperative multiarmed bandits: Partition-based algorithms with strictly local information. In *IEEE Conference on Decision and Control*, Miami, Florida, 2018.

[60] D. Lazer and A. Friedman. The network structure of exploration and exploitation. *Administrative Science Quarterly*, Dec 2007.

[61] D. Lee. Game theory and neural basis of social decision making. *Nature Neuroscience*, 11:404–409, 2008.

[62] N. E. Leonard, T. Shen, B. Nabet, L. Scardovi, I. D. Couzin, and S. A. Levin. Decision versus compromise for animal groups in motion. *Proceedings of the National Academy of Sciences*, 109(1):227–232, 2011.

[63] K. Liu and Q. Zhao. Extended ucb policy for multi-armed bandit with light-tailed reward distributions. *CoRR*, abs/1112.1768, 2011.

[64] K. Liu and Q. Zhao. Multi-armed bandit problems with heavy-tailed reward distributions. In *Allerton Conference on Communication, Control, and Computing*, pages 485–492, Sept 2011.

[65] S. M. Brennan, A. M. Mielke, and D. C. Torney. Radiation source detection by sensor networks. *IEEE Transactions on Nuclear Science*, 52:813 – 819, 07 2005.

[66] S. Mannor and O. Shamir. From bandits to experts: On the value of side-observations. In *Conference on Neural Information Processing Systems*, 2011.

[67] F. Mascarich, T. Wilson, C. Papachristos, and K. Alexis. Radiation source localization in gps–denied environments using aerial robots. In *International Conference on Robotics and Automation*, Brisbane, Australia, May 2018.

[68] W. Mason and D. J. Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, 2012.

[69] R. McDougall, S. Nokleby, and E. Waller. Probabilistic-based robotic radiation mapping using sparse data. *Journal of Nuclear Engineering and Radiation Science*, 4, 10 2017.

[70] H. Meunier, J.-B. Leca, J.-L. Deneubourg, and O. Petit. Group movement decisions in capuchin monkeys: The utility of an experimental study and a mathematical model to explore the relationship between individual and collective behaviours. *Behaviour*, 143, 03 2007.

[71] T. M. Moe. The new economics of organization. *American Journal of Political Science*, 28(4):739–777, 1984.

[72] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50:169–182, 2005.

[73] H. Nakayama and Y. Sawaragi. Satisficing trade-off method for multiobjective programming and its applications. volume 17, pages 1345 – 1350, 1984.

[74] R. J. Nemzek, J. S. Dreicer, and D. C. Torney. Distributed sensor networks for detection of mobile radioactive sources. In *IEEE Nuclear Science Symposium*, volume 3, pages 1463–1467, Oct 2003.

[75] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[76] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan 2007.

[77] V. Perchet and P. Rigollet. The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721, 2013.

[78] I. Poulakakis, G. F. Young, L. Scardovi, and N. Ehrich Leonard. Information centrality and ordering of nodes for accuracy in noisy decision-making networks. 61:1–1, 01 2015.

[79] V. M. Preciado and G. C. Verghese. Synchronization in generalized erd ö s-r é nyi networks of nonlinear oscillators. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4628–4633, Dec 2005.

[80] N. S. V. Rao, M. Shankar, J. Chin, D. K. Y. Yau, S. Srivathsan, S. S. Iyengar, Y. Yang, and J. C. Hou. Identification of low-level point radiation sources using a sensor network. In *International Conference on Information Processing in Sensor Networks*, pages 493–504, April 2008.

[81] C. E. Rasmussen and C. Williams. *Gaussian processes for machine learning*. MIT Press, 2008.

[82] L. Rendell, R. Boyd, D. Cownden, M. Enquist, K. Eriksson, M. W. Feldman, L. Fogarty, S. Ghirlanda, T. Lillicrap, and K. N. Laland. Why copy others? insights from the social learning strategies tournament. *Science*, 328(5975): 208–213, 2010.

[83] P. Reverdy, V. Srivastava, and N. E. Leonard. Satisficing in multi-armed bandit problems. *IEEE Transactions on Automatic Control*, 62(8):3788–3803, 2017.

[84] P. B. Reverdy, V. Srivastava, and N. E. Leonard. Modeling human decision making in generalized Gaussian multiarmed bandits. *Proceedings of the IEEE*, 102(4):544–571, 2014.

[85] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[86] D. Saari. *Decisions and elections: explaining the unexpected*. Cambridge Univ. Press, 2001.

[87] A. G. Sanfey. Social decision-making: Insights from game theory and neuroscience. *Science*, 318(5850):598–602, 2007.

[88] K. H. Schlag. Why imitate, and if so, how?: A boundedly rational approach to multi-armed bandits. *Journal of Economic Theory*, 78(1):130–156, 1998.

[89] T. D. Seeley and S. C. Buhrman. Group decision making in swarms of honey bees. *Behavioral Ecology and Sociobiology*, 45(1):19–31, 1999.

[90] A. K. Sen. *Collective Choice and Social Welfare*. Harvard University Press, 2018.

[91] S. Shahrampour, A. Rakhlin, and A. Jadbabaie. Multi-armed bandits in multi-agent networks. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[92] H. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63:129–38, 04 1956.

[93] D. Spanos and R. Olfati-Saber. Dynamic consensus for mobile networks. In *International Federation of Automatic Control*, 2005.

[94] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[95] V. Srivastava, P. Reverdy, and N. E. Leonard. On optimal foraging and multi-armed bandits. In *Allerton Conference on Communication, Control, and Computing*, pages 494–499, Monticello, IL, USA, Oct. 2013.

[96] V. Srivastava, P. Reverdy, and N. E. Leonard. Surveillance in an abruptly changing world via multiarmed bandits. In *IEEE Conference on Decision and Control*, pages 692–697, Los Angeles, CA, Dec. 2014.

[97] B. Stephane, L. Gabor, and P. Massart. *Concentration inequalities a nonasymptotic theory of independence*. Oxford University Press, 2016.

[98] K. Stephenson and M. Zelen. Rethinking centrality: Methods and examples. *Social Networks*, 11(1):1 – 37, 1989.

[99] J. Sun and H. G. Tanner. Constrained decision-making for low-count radiation detection by mobile sensors. *Autonomous Robots*, 39(4):519–536, Dec 2015.

[100] J. Sun, H. G. Tanner, and I. Poulakakis. Active sensor networks for nuclear detection. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3549–3554, May 2015.

[101] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, volume 1. MIT press Cambridge, 1998.

[102] C. Tekin and M. Liu. Online algorithms for the multi-armed bandit problem with markovian rewards. *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1675–1682, 2010.

[103] U. Toelch, M. J. Bruce, M. T. H. Meeus, and S. M. Reader. Humans copy rapidly increasing choices in a multiarmed bandit problem. *Evolution and Human Behavior*, 31(5):326–333, 2010.

[104] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Massachusetts Institute of Technology, Nov. 1984.

[105] S. Utz and C. J. Beukeboom. The role of social network sites in romantic relationships: Effects on jealousy and relationship happiness. *Journal of Computer-Mediated Communication*, 16(4):511–527, 2011.

[106] H. M. Wainwright, A. Seki, J. Chen, and K. Saito. A multiscale Bayesian data integration approach for mapping air dose rates around the fukushima daiichi nuclear power plant. *Journal of Environmental Radioactivity*, 167:62–69, 2017.

[107] C.-C. Wang, S. R. Kulkarni, and H. V. Poor. Bandit problems with side observations. *IEEE Transactions on Automatic Control*, 50(3):338–355, March 2005.

[108] I. Yadav and H. G. Tanner. Controlled mobile radiation detection under stochastic uncertainty. *IEEE Control Systems Letters*, 1(1):194–199, July 2017.

[109] B. Yin, Z. Jin, W. Zhang, H. Zhao, and B. Wei. Finding optimal solution for satisficing non-functional requirements via 0-1 programming. In *2013 IEEE 37th Annual Computer Software and Applications Conference*, pages 415–424, July 2013.