Reinforcement Learning-based Optimization of Multiple Access in Wireless Networks

Eleni Nisioti

A thesis submitted for the degree of
Doctor of Philosophy

School of Computer Science and Electronic Engineering
University of Essex
United Kingdom
Date of submission for examination September 2020

Acknowledgements

I would like to acknowledge and thank the following people, who made this thesis possible.

First, I want to thank my supervisor, Dr. Nikolaos Thomos, who entrusted me with the pursuit of a PhD and supported me throughout all its steps. His expertise in communications and our discussions during our frequent meetings have motivated my works and attracted me towards studying problems that reinforcement learning faces in real-world settings. This thesis would not have been possible without his continuous support and guidance.

I would also like to thank my other academic collaborators and the institutions that funded the research visits I undertook during my PhD. As part of the YERUN mobility award, I had the opportunity of visiting UPF, Barcelona and collaborating with Dr. Boris Bellalta and Dr. Anders Jonsson on the study of multi-player bandits. Also, during my visit at CWI, Amsterdam, I worked with Dr. Daan Bloembergen and Dr. Michael Kaisers on robustness in reinforcement learning. I thank each of them for sharing their expertise and guiding my first academic steps.

Special thanks also go to the members of my PhD board at the University of Essex, Dr. Leila Musavian, Dr. Thakur Manoj and Dr. David Richerby, who supervised my PhD and provided valuable feedback.

My family and friends do not probably require a special mention, but not thanking them for their unconditional love and support would have been a large omission.

Abstract

In this thesis, we study the problem of Multiple Access (MA) in wireless networks and design adaptive solutions based on Reinforcement Learning (RL). We analyze the importance of MA in the current communications scenery, where bandwidth-hungry applications emerge due to the co-evolution of technological progress and societal needs, and explain that improvements brought by new standards cannot overcome the problem of resource scarcity. We focus on resource-constrained networks, where devices have restricted hardware-capabilities, there is no centralized point of control and coordination is prohibited or limited. The protocols that we optimize follow a Random Access (RA) approach, where sensing the common medium prior to transmission is not possible. We begin with the study of time access and provide two reinforcement learning algorithms for optimizing Irregular Repetition Slotted ALOHA (IRSA), a state-of-the-art RA protocol. First, we focus on ensuring low complexity and propose a Q-learning variant where learners act independently and converge quickly. We, then, design an algorithm in the area of coordinated learning and focus on deriving convergence guarantees for learning while minimizing the complexity of coordination. We provide simulations that showcase how coordination can help achieve a fine balance, in terms of complexity and performance, between fully decentralized and centralized solutions. In addition to time access, we study channel access, a problem that has recently attracted significant attention in cognitive radio. We design learning algorithms in the framework of Multi-player Multi-armed Bandits (MMABs), both for static and dynamic settings, where devices arrive at different time steps. Our focus is on deriving theoretical guarantees and ensuring that performance scales well with the size of the network. Our works constitute an important step towards addressing the challenges that the properties of decentralization and partial observability, inherent in resource-constrained networks, pose for RL algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation of this thesis

Our motivation lies in the pressure on contemporary society to advance and align its technological efforts with the needs of an overwhelmingly complex and uncertain environment. Recent problems, such as global warming, over-population and pandemics, have illuminated how important it is to consistently evaluate the relationship between human activities and the physical world. In the last 150 years, the field of wireless communications has provided the theory and technology behind making our world a network of inter-connected devices that sense, communicate and help regulate human activity. In a 1926 interview, Nicola Tesla said: *"When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole ... and the instruments through which we shall be able to do this will be amazingly simple compared with our present telephone. A man will be able to carry one in his vest pocket."* Simplicity in a wireless device is more than an aesthetic property: devices are becoming increasingly low-cost, leading to a reduction in computational capabilities, memory capacity and battery lifetime. Furthermore, technological innovation is turning towards solutions characterized as *smart*; devices need to perceive their environment and adapt their operation to an uncertain and changing world. Solutions should, thus, be complexity-aware and leverage this shift of focus from the hardware to the software infrastructure. While wireless devices are employed for a variety of tasks, there is one particular task that today's energy-constrained society cannot afford to ignore: the allocation of common resources. Networks where devices act in a selfish manner that leads to depleting common resources or a sub-optimal manner that leads to their underutilization are associated with important losses in efficiency and high operational costs. As the 5G and beyond standards are setting the tone for the anticipated renaissance in wireless communications, one may wonder: *"what form will resource allocation take in the era of smart devices?"*

This thesis advocates that recent advances in Reinforcement Learning (RL) will be a cornerstone in answering this question. Our work aims at making contemporary wireless resource-constrained networks adaptive by designing solutions inspired by the field of RL. We particularly focus on Multiple Access (MA) problems, where wireless devices attempt to access a common pool of resources. In these problems, interference, caused by the simul-

taneous access of resources, degrades the quality of communication. As we discuss in this chapter, such problems are ubiquitous in wireless communication and have attracted significant interest during the development of the 5G [1] and beyond communication standards [2]. Our solutions equip devices with the ability of decentralized decision making with low complexity taking into account characteristics of these networks, such as their ad hoc and varying topology, locality of interaction and restricted computational and memory capabilities. An underlying objective of our works is to develop a theoretical understanding of the ability of learning algorithms to provide optimal solutions in reasonable time.

A brief history of wireless networks Wireless communication has come a long way since the pre-industrial practises of transmitting information over line-of-sight distances using smoke signals and flashing mirrors. The first wireless transmission of analog signals was performed by Marconi in 1895 [3]. It wasn't until 1971, when the first transmission of digital signals took place at the University of Hawaii as part of the ALOHAnet [4], a system that paved the way for the contemporary Random Access (RA) communication protocols studied in this thesis.

Our work focuses on networks primarily consisting of resource-constrained devices accessing a common and limited pool of resources. Under this broad definition, one can encounter families such as Wireless Sensor Networks (WSNs), the Internet of Things (IoT) [5], the Internet of Vehicles [6] and Industry 4.0 [7]. Enabled by the co-evolution of wireless technology and digital electronics, these devices are mainly characterized by their low cost, ad hoc deployment and limited battery lifetime. Furthermore, we are interested in networks lacking a central coordinator, whose operation can be orchestrated by designing the behaviour of individual devices. As communication between devices needs to be limited, to elongate their battery lifetime or minimize their energy consumption, decisions need to be made in a decentralized way. We summarize the evolution that wireless resource-constrained networks have undergone in Figure 1.1: after the first wireless transmission of analog signals by Marconi [3] and digital signals by Abramson at the University of Hawaii [4], the first cellular network was born in 1979 [8]. Networks of resource-constrained devices were initially studied for the military by DARPA and termed as Wireless Sensor Networks [9]. The term Internet of Things (IoT) was coined in 1999 to describe networks of physical objects embedded with sensors, software and other technologies for exchanging information [10]. Today, new types of Internet of Things (IoT) are continuously arising [10]. For example, vehicular ad hoc networks [11] have evolved into the Internet of Vehicles [12], a result of the shift of focus to intelligent integration of vehicles into the IoT ecosystem, while the term Internet of Everything (IoE) extends the IoT paradigm to include people, data and processes [10].

The recent emergence of IoT is an example of how technological innovation can give birth to a variety of applications. Industrial and commercial IoT applications, such as smart grids, smart homes and smart healthcare are characterized as *green* and are entangled with the recent shift of our society towards achieving a compromise between the proliferation of human activity and the need for preserving environmental resources. To emphasize that human supervision or participation is not required, we characterize communication in IoT as *machine-to-machine* communication. Important traits of devices employing this type of communication are: (i) low device cost, (ii) low battery life, and, (iii) massive connectivity

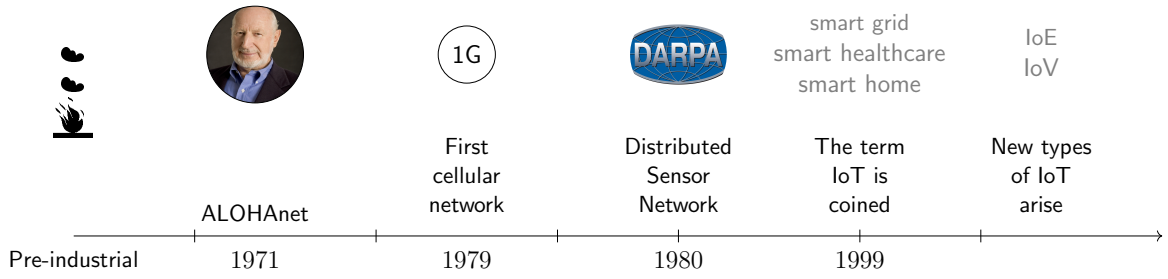


Figure 1.1: A timeline of the evolution of wireless networks

[13]. In order to accommodate these needs, we are also experiencing the emergence of new communication standards, such as the recent LoRaWAN standard [14]. This stage in the evolution of wireless technology is, thus, powered by our realization that ubiquitous connectivity is viable only if networks are low-cost and energy-efficient.

The family of wireless communication standards is also quickly expanding to cover the requirements of emerging applications and types of networks. The IEEE 802.15.4 [15] and IEEE 802.11 [8], known as wifi, technical standards, specify the operation of the physical and media access control layers and encapsulate a variety of standards for low-rate personal area networks. In general, standards are characterized by their coverage: (i) local area networks [16] are used in wearables, home and industrial automation applications. Popular standards in this category are the Bluetooth [17] and Zigbee [18] standards; (ii) wide area networks can be used to cover the needs exceeding the area of a single building and, thus, require different standards, such as the WiFi and LoRaWAN [14] standards; (iii) wireless communication that covers the area of a city commonly requires a WiFi or cellular infrastructure [19].

In this thesis, we only study communication at the MAC layer and focus in local area networks. In Figure 1.2, we present a high level representation of the considered communication framework. In our framework, there is an ad-hoc network of resource-constrained devices transmitting packets to a central node, commonly termed a *sink*, e.g., a base station, assuming a single-hop communication. Devices do not have the ability of sensing the channel prior to transmission. Thus, the occurrence of collisions can only be detected by the base station. We assume that, if interference succeeds at the MAC layer, then it is guaranteed to also succeed at the physical layer. We are not concerned with the multi-hop communication employed to transmit the collected packets from the base station to another infrastructure for further processing, such as the edge or the cloud, which as we discussed above can be performed using WiFi or cellular communication. The solutions proposed in this thesis are employed in two different types of networks: (i) in Chapters 7 and 8, the problem setting is a wireless network where devices are collecting measurements from their environment and transmitting them to the base station, which operates on a single frequency and performs collision resolution at the MAC layer; (ii) in Chapter 9, devices are accessing a base station where a certain number of channels, corresponding to different transmission frequencies of the available spectrum, are accessed simultaneously by all devices, which receive an acknowledgement (ACK) signal indicating the success of a transmission. These types of networks can be encountered in a variety of scenarios, where IoT devices are employed for monitoring and tracking in applications related to health-care, agriculture and the industry. Due to their

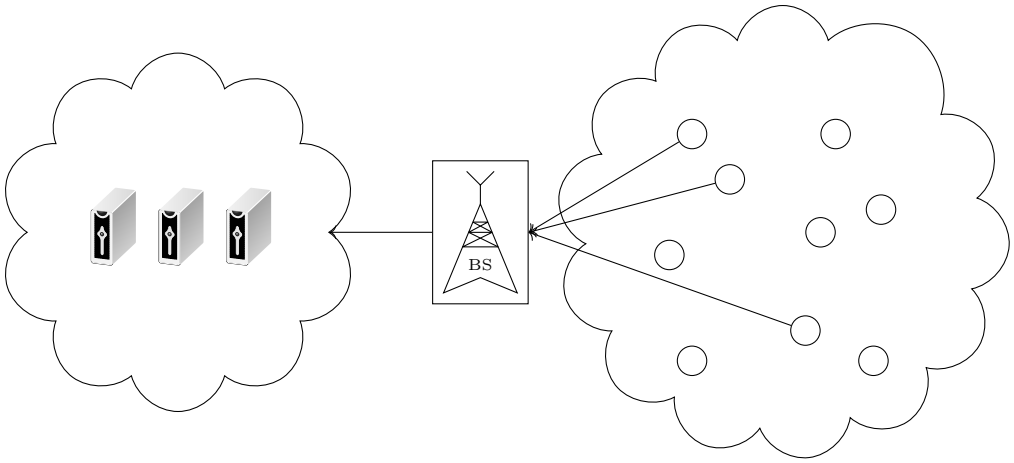


Figure 1.2: Schematic of the network architecture considered in this thesis.

wide applicability, their optimization plays an important part in establishing sustainable technological progress, which is a major underlying motivation in this thesis.

The paradox of spectrum scarcity A manifestation of the need of contemporary resource-constrained networks for adaptation is the problem of spectrum scarcity. Spectrum, the collection of frequency bands that communication devices access to transmit information, is an abundant resource. Nevertheless, it needs to be carefully distributed among different applications in order to ensure that communication is interference-free despite the coexistence of multiple transmitting devices. Early on in the brief history of wireless communications, authorities realised that the initial approach of fragmenting spectrum and determining a fixed assignment licensed to specific devices and applications, will lead towards quickly exhausting it. This fragmentation reached unanticipated levels, making the introduction of new categories of devices difficult. Simultaneously, monitoring the activity in different bands revealed that, most of the time, the spectrum remained underutilized [20]. This led to the realization that spectrum assignment cannot be both static and efficient in a world where applications arise frequently and users exhibit mobility and time-varying needs. In reality, licensed users need to coexist with unlicensed ones, who have not been assigned pre-determined frequency bands. Despite the increase in bandwidth brought about by the introduction of 5G and beyond standards, spectrum access is expected to remain an ongoing concern. Emerging applications, such as virtual and augmented reality, are becoming increasingly bandwidth hungry and have accelerated the realisation that the illusion of spectrum abundance should not urge us back to solutions of fixed assignment [21].

The need for unlicensed users that can reconfigure their transmission frequency initially lead to the birth of *software-defined* radio, whose evolution today is the *cognitive radio*. Both terms, introduced by Mitola in 1991 [22] and 1998 [23], respectively, signify a departure from expensive, rigid hardware-based solutions. In software-defined radios, switching between different frequency bands is made possible by combining digital radio technology with computer software. Cognitive radio subsumes this capability, but extends it to the ability of understanding the environment and intelligently adapting to it, with the two-fold objective of ensuring on-demand reliable communication and efficient utilization of the spectrum. In-

telligence in cognitive radio is primarily enabled by the use of advanced signal processing techniques and machine learning, and is the overall result of a harmonious collaboration of the hardware, operating system and algorithms embedded on the devices. While machine learning subsumes a variety of fields, such as supervised and unsupervised learning, the area that is of most interest to us is that of RL. Being concerned with decision making in uncertain environments, RL can provide the frameworks and solutions required to render real-world systems adaptive to their environment.

Having described the history and characteristics of the wireless networks considered in this thesis, we are now faced with the following challenges:

Challenge 1 How can a system designer balance optimality with complexity when designing solutions for resource allocation? We investigate this question by examining different learning approaches: in Chapter 7 we focus on low complexity by making the assumption of independent learning and in Chapter 8 propose to reach a balance between optimality and complexity by allowing learners to exchange messages and coordinate their transmission strategies.

Challenge 2 Can devices in a decentralized network operate optimally in terms of throughput maximization without imposing extensive communication between them? In Chapters 7 and 9 we prohibit communication between devices and in Chapter 8 we intorduce a technique for restricting communication.

Challenge 3 Can we ensure that the operation of a network remains optimal when the topology of the network and channel conditions change with time? In chapter 9, we consider resource allocation in dynamic networks, where devices arrive at different time steps in the network.

Challenge 4 How can we design solutions that can be applied to low-cost devices with restricted hardware capabilities? In all our works, we make minimum requirements on the hardware capabilities of devices: in Chapters 7 and 8 the memory requirements are kept small by keeping a small history of observations and computational complexity is kept low by avoiding centralized solutions. Furthermore, we assume that devices do not have the ability of sensing the channel prior to transmission.

1.2 Multiple access and reinforcement learning

In this thesis, we study the problem of MA in wireless networks and design adaptive solutions based on RL techniques. Let us now define these two concepts.

Multiple access In our work, the broad term MA encapsulates any problem setting where multiple wireless devices simultaneously access a common pool of network resources, termed as a *medium*. We abstractly define a resource as an object desirable to all devices that, if simultaneously accessed by more than one of them, becomes unavailable to all. We term this simultaneous access of a resource by more than one devices a *collision*. We consider two examples of resources that are often encountered in wireless applications: (i) in a network where multiple devices are sharing a single channel, we can divide time into frames and

frames into slots, which represent the resources. In this setting, typically termed as *framed* to distinguish it from *frameless* approaches, a device attempts to find a specific slot to transmit in, that is not used by other devices; (ii) in multi-channel communication, resources correspond to non-overlapping channels, each one associated with a different frequency band. In both these types of problems, an important property of a network from the perspective of MA is its *traffic* or *load*, defined as the ratio of actively transmitting devices to resources.

If we view MA as an end, then a Medium Access Control (MAC) protocol is the means that communication systems employ to achieve it. MAC protocols orchestrate the access of devices to resources with the aim of maximizing a designer's objective, associated with the quality of communication in a network. The family of MAC protocols is rich , as one would anticipate for an essential mechanism required in wireless networks [24, 25].

An important quality of access is whether it is *contention-free* or *contention-based*. In contention-free protocols, resources are distributed among devices in an orthogonal manner in order to ensure the absence of collisions. In this category, one can find Time Division Multiple Access (TDMA) [26] and Frequency Division Multiple Access (FDMA) protocols [27]. The original approach of dividing spectrum among users, that led to the problem of spectrum scarcity, is an example of an FDMA protocol. In contention-based protocols on the other hand, devices can initiate transmissions simultaneously, rendering collisions a possibility.

Another classification of protocols can be done on the basis of whether they are centralized or decentralized. In general, centralized protocols require extensive communication in the form of broadcasting or transmitting to a central node. Solutions for ad hoc networks cannot often afford the energy expenditure required by this communication and are, therefore, primarily decentralized. In addition, the additional message exchanges consume resources, such as time slots and channels, often defeating our purpose of optimizing MA. A concept related to the level of decentralization is that of *cooperation*: if devices communicate with each other in order to coordinate their access, the protocol is termed cooperative, while absence of communication is equivalent to a non-cooperative protocol. A cooperative protocol can, in the limit of unrestricted communication, become equivalent to a centralized protocol.

MAC protocols that are specifically designed for wireless resource-constrained networks need to take additional considerations into account. Arguably, their most distinctive property is that their objective is to optimize both throughput and energy efficiency. The former is measured as the probability of successful transmission of an information unit, typically termed as a *packet*, in a communication slot, while the latter commonly refers to the battery lifetime of devices. On the contrary, the majority of MAC protocols, concerned with applications where battery lifetime is not a constraint, are designed with just throughput in mind. S-MAC [28], a major protocol in this category, accommodates both these objectives by dividing time into *active* and *sleeping* phases of fixed duration. During the active phase devices operate normally to transmit information, while, during the sleeping phase, they deactivate to reduce energy consumption or recharge.

An important consideration when it comes to resource-constrained devices is whether they have the ability of sensing the medium prior to transmission. Protocols that exploit this ability follow a Carrier Sense Multiple Access (CSMA) approach, and are capable of establishing the availability of a resource with high certainty prior to transmisison. In the

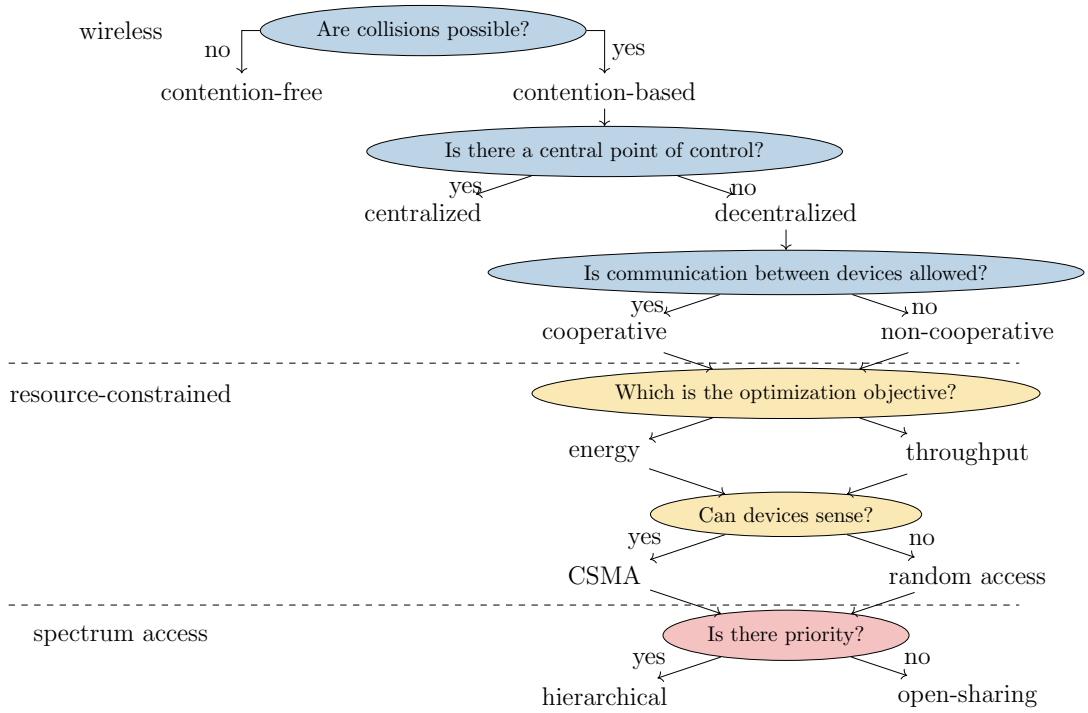


Figure 1.3: A map of the types of MAC protocols for wireless networks discussed in this thesis. Our categorization starts with general considerations and then progressively focuses on resource-constrained networks and spectrum access.

absence of this ability, collision-based protocols follow a Random Access (RA) approach, where devices transmit without explicitly avoiding collisions. In Chapter 3, we provide a review on RA protocols and describe the protocol that our work focuses on, Irregular Repetition Slotted ALOHA (IRSA) [29]. As we will explain, IRSA is primarily configured through the *degree distribution*, a probability distribution that each device samples to determine how many copies of a packet to transmit in order to maximize the probability of successfully transmitting it despite collisions.

Dynamic Spectrum Access (DSA) is an example of an MA setting formulated to address the problem of spectrum scarcity [30]. Solutions in this area follow two distinct approaches: (i) hierarchical DSA discriminates between licensed devices, traditionally termed as Primary Users (PUs) and unlicensed ones, termed as Secondary Users (SUs). MAC protocols orchestrate the access of SUs to channels, which have been previously assigned to PUs. Thus, the throughput of the ad hoc network of SUs is maximized and spectrum scarcity is minimized. An important concern here is to ensure that the activity of SUs does not interfere with the PUs. This requires that SUs sense a channel prior to transmission, a functionality that is not always embedded in resource-constrained devices. (ii) in open-sharing DSA, there is no distinction between SUs and PUs, so the problem becomes equivalent to the general MA problem. In the absence of sensing information, employing a CSMA protocol is not possible, so RA is the norm. For a comprehensive survey of DSA solutions we refer readers to [30].

We summarize the discussed classes of MAC protocols in the map depicted in Figure 1.3. The classification begins with considerations encountered generally in wireless networks and increasingly become specific for resource-constrained networks. Our survey of MAC protocols

is not exhaustive, but aims at analyzing considerations related to the contributions of this thesis in order to position them alongside the needs of resource-constrained networks.

Reinforcement Learning RL is a sub-field of machine learning that provides a purely mechanistic approach to describing the nature of learning. To quote the definition given by the fathers of RL, Sutton and Barto: *"Reinforcement learning is learning what to do - how to map situations to actions - so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them."* [31]

In general, a reinforcement learning *task* describes the interaction between an *agent* and its *environment*. The agent represents the intelligent entity that is performing *actions*, which alter the environment, and observing *rewards*, which quantify how useful a performed action was towards completing the task. Context related to characteristics of the agent and environment can be incorporated through the notion of a *state*. For example, a wireless device may be modelled as an agent whose state includes its battery level, an action corresponds to the selection of channel to transmit in, and the reward provided by the environment is the acknowledgement signal sent by the receiver. The objective of learning is to find the optimal *policy*, which indicates the action that will give the maximum reward in any given state. The formulation of RL tasks primarily follows two mathematical frameworks: (i) the framework of Markov Decision Processes (MDPs) [31]; (ii) the framework of Multi-armed Bandits (MABs) [32], also termed as bandits, which can be seen as a special case of MDPs.

The defining characteristic of an MDP is the *Markov property*, which states that knowing the current state of the process and the action performed by the agent is adequate for predicting the next state and reward. This property helps maintain low complexity, as learning algorithms can learn from experience that is at most one step in the future. Figure 1.4 presents the agent-environment interaction in an MDP process.

A MAB, on the other hand, does not employ states in its definition. In essence, a bandit is an elegant mathematical formulation of the *exploration versus exploitation dilemma*, which is a fundamental component in every online learning problem. A multi-armed bandit consists of a *player*, the equivalent of an agent, interacting with a set of arms, which form the environment, by pulling them. The rewards that arms provide upon being pulled are viewed as random variables following a probability distribution. The objective of the agent is to discover the arm with the highest mean reward as quickly as possible. Figure 1.5 presents the agent-environment interaction in a multi-armed bandit.

An important property of the algorithms proposed in this thesis is that of *decentralization*, which, as we will explain in Chapter 4, has a significant effect on the learning task. In the learning nomenclature, decentralized architectures are termed *multi-agent systems*. In the bandit nomenclature, a multi-agent system can also be termed a *multi-player game*. Throughout this thesis, depending on the context, we may employ the terms "devices", "users", "agents" and "players", when referring to the decision-making entity.

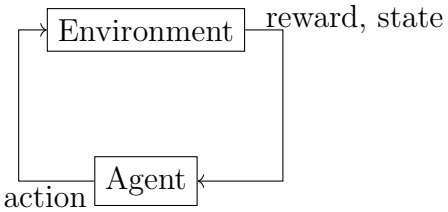


Figure 1.4: Interaction in a Markov Decision Process.

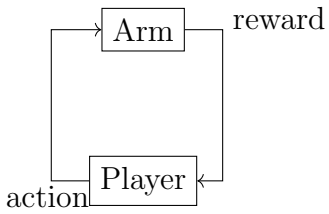


Figure 1.5: Interaction in a Multi-armed Bandit.

1.3 Our contributions

We now delve deeper into the objectives of our study and the contributions of this thesis. Our research can be summarized into the following question: *"can we design reinforcement learning algorithms that achieve a fine balance between optimality and complexity for contemporary wireless resource-constrained networks?"* We refine this question alongside presenting our contributions.

1.3.1 Random access using independent Q-learning

A large part of our work is concerned with designing an adaptive version of the IRSA protocol [29], where the degree distribution is optimized using a reinforcement learning algorithm [33, 34, 35]. As we briefly discussed above and will explain in detail in Chapter 3, IRSA is a state-of-the-art Random Access protocol that holds great promise in resource-constrained networks. Traditionally, its optimization concerns asymptotic settings, as this makes it possible to theoretically analyze its performance. However, the predicted theoretical performance deviates significantly from the real one when networks are small. In contrast, the performance of our protocol, RL-IRSA, does not depend on the size of the network. We focus on network settings with high traffic levels and relatively small frame sizes. The reason why assuming large networks in wireless networks is not always appropriate is that it implies large frame sizes. The latter introduces complexity at the receiver and delays, as devices need to wait for the current frame to finish before initiating a transmission.

An important research question that our work aims to tackle is: *"how can we ensure that the complexity of our learning-based solutions is low enough for resource-constrained networks?"* In our initial works [33, 34], each device attempts to learn an optimal transmission strategy using Q-learning without communicating with other devices. This approach, termed as independent Q-learning, has significantly lower complexity than centralized solutions. As devices do not need to process or store an amount of information that increases with the size of the network, they avoid the exponential complexity in terms of memory and learning time associated with centralized Q-learning. In addition, devices do not need to expend energy or bandwidth on exchanging information. Nevertheless, it is known that independent Q-learners often converge to sub-optimal strategies, due to not having access to information that is necessary for computing the globally optimal strategy [36].

In addition to decentralization, our work in [33, 34] lays focus on the property of partial observability. As sensors can observe only information that is local to them and is possibly inaccurate due to their limited hardware capabilities, it is possible that they lack informa-

tion essential for ensuring optimal collective behaviour. In order to capture this property, we model the network as a Dec-POMDP and adopt finite histories of observations to approximate the continuous beliefs of Belief MDPs, which significantly reduces the size of the state space. In Section 7.3.1, we will analyze the quality of solutions found using independent Q-learning in this framework.

In networks where the environmental dynamics, such as the number of devices and quality of channels vary with time, it is important to ensure that adaptation is quick. A designer may, thus, wonder: *"is it possible to leverage properties of the system in order to accelerate learning?"* A common property of real-world systems is that a part of the environmental dynamics may be known prior to their deployment. For example, a group of sensors monitoring a moving target may know that the target cannot accelerate above a physically imposed threshold. In these cases, the system does not need to learn the environmental dynamics from scratch, but can combine its existing knowledge with experience acquired through learning. This idea is leveraged by Virtual Experience (VE) [37], a technique introduced to improve the convergence rate of tabular Q-learning that we will describe in detail in Section 7.2.2. As the original introduction of VE was not accompanied by a theoretical analysis, we have provided a proof of the effect of VE on the convergence rate of Q-learning. In addition to its theoretical analysis, we have also empirically examined the effect of VE on the performance of our adaptive protocol and observed significant gains in learning time.

1.3.2 Random access using coordinated Q-learning

In our next work, we tackle the following question: *"is it possible to reach a compromise between the low complexity of independent Q-learning and the optimality of a centralized approach?"* We, therefore, shift our attention to the area of coordinated reinforcement learning [38], which exhibits both theoretical guarantees of optimality and low complexity. This is made possible by leveraging the observation that agents do not need to communicate consistently, but solely when they interact with each other.

Conceptually, coordinated learning is positioned between the two extremes of independent and centralized learning. Particularly in MA for resource-constrained networks, the need for coordination arises because: (i) decisions require information non-local to devices. For example, the two devices in Figure 1.6 must sense the target simultaneously in order to detect its speed and, therefore, need to coordinate their actions; (ii) limited computational and battery resources prohibit centralized solutions. Among the vast family of algorithms for coordinated learners, we have focused on Coordinated Q-learning. We will provide a comprehensive description of this algorithm in Section 8.2. To the best of our knowledge, our work is the first to propose the use of Coordinated Q-learning on the problem of MA. This choice was inspired by the observation that such a task can be naturally described by a bipartite graph that closely resembles the Coordination Graph (CG) used in Coordinated Q-learning.

We performed extensive simulations using both independent and coordinated Q-learners to render the optimization of the degree distribution of IRSA adaptive and observed that performance improved in both cases, when compared to degree distributions that were optimized in asymptotic settings. In particular, the performance of coordinated Q-learners was better than the performance of independent Q-learners. In Figure 1.7, we compare the

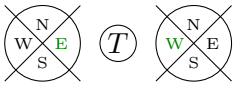


Figure 1.6: An example of the necessity of coordination in a wireless network: two devices with the ability of sensing north(N), east(E), south(S) and west(W) are monitoring different areas. In order to estimate the speed of the target, the devices need to sense it simultaneously. If the target is between them, they need to coordinate their actions and perform the actions highlighted in green. If only one of the two devices senses the target, detection fails and unnecessary energy is consumed.

performance of coordinated, centralized and independent Q-learners and the performance of IRSA optimized in an asymptotic setting [29]. As anticipated, the learning-based solutions surpass the performance of the non–learning based one, with the centralized learner achieving the highest throughput for high traffic levels. Nevertheless, as we will explain in Chapter 8, the complexity of centralized Q-learning is prohibitive in networks with more than a handful of devices.

In addition, we introduced the framework of Groupwise-Dependent Decentralized Partially Observable Markov Decision Processes (GGD-POMDPs), which is a generalisation of the framework of Dec-POMDPs appropriate for studying a variety of applications for resource-constrained networks. In particular, in Section 8.5, we prove that Coordinated Q-learning has convergence guarantees in Groupwise-Dependent Decentralized Partially Observable Markov Decision Processes (GGD-POMDPs).

Although coordinated Q-learning avoids the complexity of a centralized solution, it can still incur high cost if the required communication is extensive and frequent. In resource-constrained networks, it is important to keep communication to a minimum without negatively impacting the performance of the network. For this reason, we have also devised two techniques for reducing and bounding the complexity of coordination, which aim at (i) avoiding coordination when we are not certain that it will bring a benefit over independent learning. We achieve this by deriving a sufficient condition for the convergence of the algorithm used to determine the optimal actions; (ii) removing dependencies between agents when this does not affect the optimality of Q-learning by formulating the coordination task as a Multiple Knapsack Problem (MKP). We will review these techniques more closely in Section 8.4.

1.3.3 Dynamic spectrum access using multi-player bandits

In addition to adaptive protocols for slot access, we also design solutions for channel access in Chapter 9. In this work, we employ the framework of MMABs and design two online learning algorithms for finding the optimal allocation of channels to devices. Our first proposed algorithm, Collision Resolution-Upper Confidence Bound (CR-UCB), concerns a setting termed as *static*, where all devices start learning simultaneously. We, then, raise this assumption to design the Dynamic Collision Resolution-Upper Confidence Bound algorithm, which finds the optimal assignment in both static and *dynamic* settings, where devices start learning at different time steps. Motivated by the needs of contemporary resource-constrained networks we make minimum assumptions on the abilities of devices. In particular, our solutions: (i) are decentralized and do not require cooperation between devices; (ii) do not require that devices sense the medium prior to transmission; (iii) do not assume that a device knows how

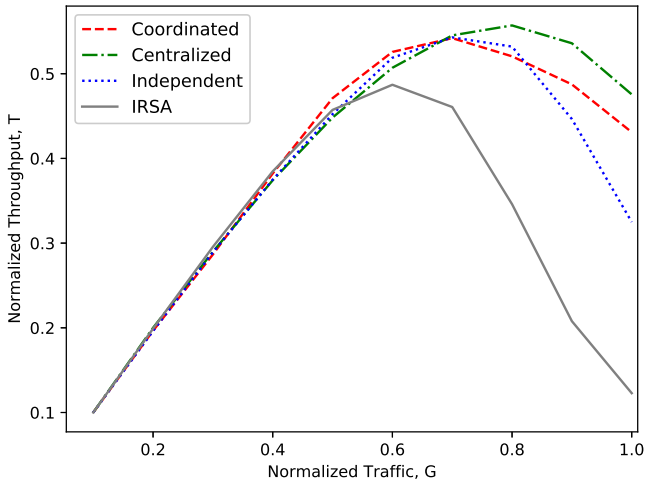


Figure 1.7: Comparison of independent, coordinated and centralized Q-learning on the task of optimizing the degree distribution of IRSA in terms of the throughput of the network for various levels of traffic, defined as the ratio of devices to time slots in a frame. We also present the performance of IRSA optimized using the technique proposed in [29], which exhibits state-of-the-art performance in asymptotic settings.

many devices are present in the network; (iv) scale well with the size of the network; (v) take into account that devices may arrive at the network at different time steps.. An important property of our bandit algorithms is that they are accompanied by a theoretical understanding of their optimality, which proves that they exhibit state-of-the-art performance for the considered setting.

The main novelty of CR-UCB is the Collision Resolution (CR) mechanism employed by devices to ensure that a collision-free channel is consistently found. This mechanism is inspired by SIC, the collision resolution mechanism introduced in [39], which it extends in two ways:(i) the mechanism in [39] was designed for optimizing slot access. Therefore, it assumes that resources are always available. When studying channel access, however, one needs to account for the fact that the availability of resources is probabilistic and unknown to devices; (ii) our mechanism does not require the presence of a controller observing transmissions in all slots. Collision resolution is achieved through the individual behavior of devices.

The regret bound characterizing CR-UCB indicates that it is most effective in the regime of large networks. Our work, thus, represents a paradigm shift in the study of multi-player bandits, which has focused on ensuring regret bounds that scale well with learning time, but has neglected the effect that the number of players has on regret bounds. Indeed, recent works that contain empirical evaluations of multi-player bandits [40, 41] have examined settings with a handful of players. In contrast, we examine settings where the number of players extends into the hundrends. We argue that, considering that the motivation of multi-player bandits lies in DSA, it is important to examine the scalability of learning solutions.

To accommodate the arrival of players during the operation of the network, we also designed DYN-CR-UCB, a dynamic version of CR-UCB. This algorithm differs significantly from its static counterpart, but is also largely based on the success of the CR mechanism. Its inspiration lies in the DYN-MMAB algorithm proposed in [42], which concerns a similar

setting to our own, but exhibits slower learning speed. In contrast to static algorithms which perform well only when all devices start learning simultaneously, our dynamic algorithm is appropriate for wireless devices that exhibit mobility.

1.3.4 Summary of the contributions

Our contributions can be summarized in the following points:

- an adaptive version of the IRSA protocol, where the degree distribution is optimized using independent Q-learning [33, 34]. We examine the performance of the proposed scheme in simulations of various settings and observe that the derived distributions perform better than the ones proposed by classical IRSA for high channel traffic loads and small frame sizes;
- an adaptive version of the IRSA protocol, where the degree distribution is optimized using coordinated Q-learning [35]. We show that coordination achieves better performance than independent learning and exhibits lower complexity than centralized Q-learning;
- a theoretical analysis of the effect of Virtual Experience on the convergence rate of Q-learning;
- introduction of the framework of GGD-POMDPs, which is appropriate for modelling MA in resource-constrained networks;
- a proof that coordinated Q-learning has convergence guarantees in the GDD-POMDP framework;
- a sufficient condition for the convergence of the message-passing algorithm used to determine the optimal actions in coordinated Q-learning. We employ this condition to avoid coordination when it is not guaranteed to bring any benefit over independent learning;
- a technique for reducing the complexity of coordination by formulating it as an MKP that preserves optimality;
- CR-UCB, an algorithm in the framework of multi-player bandits for optimizing Dynamic Spectrum Access;
- DYN-CR-UCB, a variant of CR-UCB that can be used in both static and dynamic networks, where devices arrive at different, unknown time steps;

1.4 Organization of the thesis

The reading order of this manuscript can follow any one of the two independent paths between the current Chapter 1 and the conclusion in Chapter 10, presented in Figure 1.8. The colour-coded arrows indicate the two paths: (i) the path that includes Chapters 1,2,3,4,

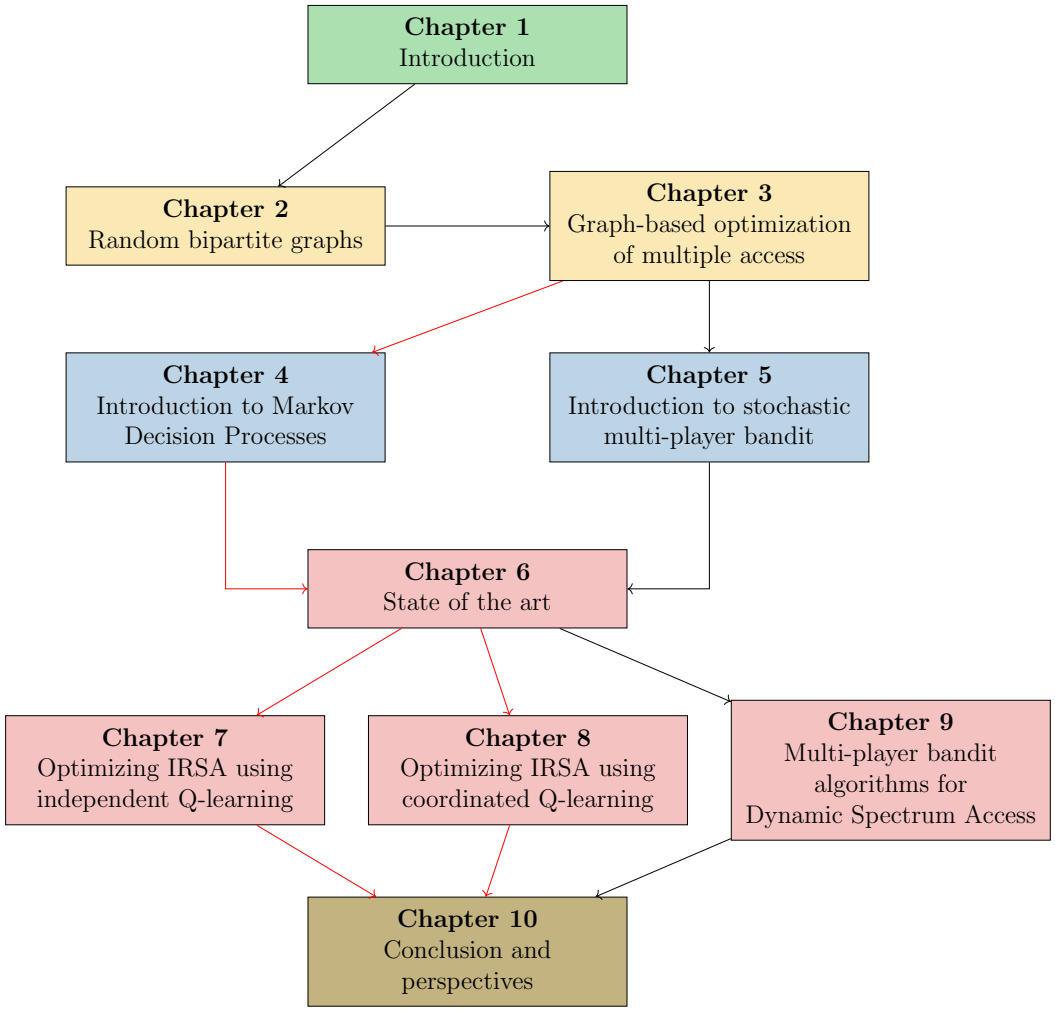


Figure 1.8: A reading map of the thesis: readers can (i) read all chapters sequentially; (ii) follow the path in red, which is concerned with optimizing IRSA using MDPs; or, (iii) follow the path in blue, which is concerned with optimizing DSA using multi-armed bandits. Arrows in black belong to both paths.

6 and then bifurcates into chapters 7 and 8, before reaching the final chapter, regards the use of MDPs for optimizing IRSA; (ii) the path that includes chapters 1, 2, 3, 5, 6 and 9 regards the optimization of channel access using multi-armed bandits. We have divided the main body of the thesis into three parts.

First, in Part 1 we introduce the main idea behind the random access protocols employed in our work. Reading these chapters is necessary for understanding the IRSA protocol, which is employed in chapters 7 and 8, and the collision resolution mechanism proposed by our CR-UCB algorithm, in Chapter 9. This part begins with an introductory discussion on bipartite graphs, which are extensively used throughout our thesis to represent a MA problem. We, then, describe the idea of collision resolution, and in particular the SIC mechanism, in order to consequently provide a full description of the IRSA protocol. We conclude this chapter with a model of resource-constrained networks that is used in our optimization of IRSA.

In Part 2, we provide required background on reinforcement learning. In Chapter 4, we take a deep dive into the framework of MDPs. As this framework is vast, we have chosen to

focus on sub-areas that are most relevant with our study of wireless networks. We therefore present frameworks designed to account for partial observability and decentralization, which are necessary for understanding Chapters 7 and 8. In Chapter 5, we offer information on the framework of multi-player multi-armed bandits, which is helpful for understanding CR-UCB and DYN-CR-UCB, our contributions presented in Chapter 9.

Finally, in Part 3 we present our contributions in three independent chapters. Each chapter corresponds to a different publication and can be read independently from the others. Prior to this, we review the state-of-the-art in RL-based MA.

The thesis concludes with Chapter 10, which attempts to summarize our contributions and position them among future developments in the field of RL for MA. We envision these developments based on our extensive review of this field and our expertise acquired by conducting this thesis.

1.5 List of publications

We conclude this chapter with a list of works published during my PhD. Publications are ordered historically, from the most recent to the oldest.

Publications in conferences and journals

- E. Nisioti and N. Thomos,"Fast Q-learning for Improved Finite Length Performance of Irregular Repetition Slotted ALOHA", in IEEE Transactions on Cognitive Communications and Networking 6.2 (2020), pp. 844-857
- E. Nisioti and N. Thomos,"Robust Coordinated Reinforcement Learning for MAC Design in Sensor Networks", in IEEE Journal on Selected Areas in Communications, vol. 37, no. 10, pp. 2211-2224, Oct. 2019.
- E. Nisioti and N. Thomos, "Decentralized Reinforcement Learning Based MAC Optimization", 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, 2018, pp. 1-5.

In progress work waiting for submission

- E. Nisioti and N. Thomos, "Design of Capacity-Approaching Low-Density Parity-Check Codes using Recurrent Neural Networks"
- E. Nisioti, N. Thomos, B. Bellalta, A. Jonnson, "Collision resolution in multi-player bandits without observing collision information"
- E. Nisioti, D. Bloembergen, M. Kaisers, "Robust multi-agent Q-learning in cooperative games with adversaries"

Part I

Multiple access on graphs

Chapter 2

Random bipartite graphs

As we briefly discussed in Chapter 1 and will become apparent in later chapters, bipartite graphs are an important concept in our work. They are the backbone of the architecture that we employ to describe a resource allocation task. Bipartite graphs differ from general graphs in that they consist of two disjoint sets of nodes, traditionally termed as variable nodes (VNs) and check nodes (CNs). In the problems that we are studying, check nodes are used to describe resources and variable nodes represent the entities attempting to access them. Depending on the context, resources may correspond to time slots or channels and the entities may be termed devices, agents or players.

We begin this chapter by presenting a high-level description of a bipartite graph, an example of which we present in Figure 2.1. We are particularly interested in *random bipartite graphs*, which are built based on a probabilistic description of their structure. This way of describing a graph is an alternative to an explicit enumeration of the connections between VNs and CNs. We will describe in detail the core of the random construction process, the degree distribution.

We, then, provide an account of areas that have employed bipartite graphs in the past. In this thesis, bipartite graphs are used for two distinct problem settings:(i) MA, a resource allocation task where resources can be either time slots or channels; (ii) finding the optimal action in a Coordination Graph. We also describe channel coding as an area where these graphs have found prominent use.

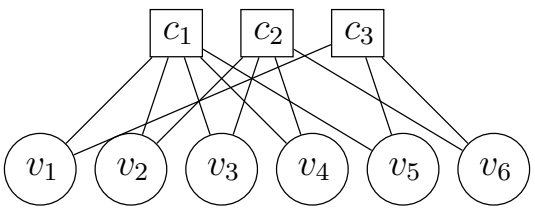


Figure 2.1: Example of a bipartite graph: this graph consists of 6 VNs and 3 CNs. The degree distribution used to built it is $\Lambda(x) = x^2$. On average, the CNs have a degree of $\bar{P} = 4$.

2.1 Modelling a random bipartite graph

We denote a bipartite graph describing a problem setting of M VNs and K CNs as $G(\mathcal{M}, \mathcal{K}, \mathcal{E})$, where $\mathcal{M}$ ($\mathcal{K}$) is the set of VNs (CNs) and $\mathcal{E}$ is the set of edges connecting the two sets of nodes. An edge between VN m and CN k is denoted as $\vec{e} = (m, k)$. We refer to the number of edges connected to a node as its degree and denote by L the random variable modelling the degree. For a VN, the probability distribution is a multinomial, described by the coefficients $\Lambda_l = \mathbb{P}(L = l)$ defining the probability that the node has l edges connected it. This distribution can be equivalently described by the following generating function:

$$\Lambda(x) = \sum_{l=1}^{\Lambda_{\max}} \Lambda_l x^l$$

Equivalently, the degree of a CN is determined by the multinomial distribution described by the coefficients $P_l = \mathbb{P}(L = l)$. The generating function in this case is denoted as:

$$P(x) = \sum_{l=1}^{P_{\max}} P_l x^l$$

The functions $\Lambda(x)$ and $P(x)$ are termed *degree distributions* while the constants Λ_l and P_l are termed to as their coefficients. As these coefficients represent probabilities, they should respect the following conditions:

$$\Lambda_l \in [0, 1],\ \forall l \in [1, \Lambda_{\max}], \quad \sum_l \Lambda_l = 1$$
$$P_l \in [0, 1],\ \forall l \in [1, P_{\max}], \quad \sum_l P_l = 1$$

where $\Lambda_{\max}(P_{\max})$ denotes the maximum degree allowed to a VN (CN).

An essential difference between $\Lambda(x)$ and $P(x)$ in our considered setting, is that only the former is in the control of the designer. A solution for random access will determine the number of connections of a VN, while $P(x)$ can be derived under knowledge of $\Lambda(x)$ and the size of the graph, while taking into account the randomness in determining connections. If we denote the average number of edges on a VN as $\bar{\Lambda}$ and the average number of edges connected to a CN as $\bar{P}$, then the probability that a VN is connected to a certain CN is $\bar{P}/M$. Thus, the probability that a CN has degree l is given by:

$$P_l = \binom{M}{l} \left(\frac{\bar{P}}{M}\right)^l \left(1 - \frac{\bar{P}}{M}\right)^{M-l}$$

The CN degree distribution has the form:

$$P(x) = \sum_l P_l x^l = \left(1 - \frac{\bar{P}}{M}(1 - x)\right)^M$$

In addition to the $\Lambda(x)$ and $P(x)$ degree distributions, which are traditionally called node-perspective, the edge-perspective degree distributions $\lambda(x) = \sum \lambda_l x^{l-1}$, $\rho(x) = \sum \rho_l x^{l-1}$ are also commonly encountered when analyzing bipartite graphs. Here, $\lambda_l(\rho_l)$ denotes the percentage of edges that are connected to a VN (CN) of degree l.

An important property of bipartite graphs is that of *concentration*, which states that the behaviour of randomly generated graphs built using the same $\Lambda(x)$ is similar. To denote the ensemble of bipartite graphs generated using the same random process we employ the notation $\mathcal{G}(M, K, \Lambda(x))$. Analyzing $\mathcal{G}$ using probabilistic arguments instead of the specific graph instance G significantly limits the complexity of solutions employing bipartite graphs. We should note that the concentration property is weak for small bipartite graphs and its strength increases proportionally to the number of nodes [43].

2.2 Application areas of bipartite graphs

A bipartite graph is a mathematical graphical model for describing the interaction between two distinct sets of entities, the VNs and CNs. It is not, therefore, surprising that it has been proven useful in analyzing a variety of applications, some of them seemingly unrelated to each other [44].

Arguably the most prominent application of bipartite graphs in the area of wireless communications is in analyzing MA. The problem of time access in slotted communication was first associated with bipartite graphs in the work that introduced IRSA [29]. This analysis has also been extended to coded communication schemes [45] and schemes where users have different levels of priority [46]. In this case, VNs represent devices, the CNs represent time slots and the degree distribution $\Lambda(x)$ determines how many copies each user will transmit in order to maximize their probability of finding a collision-free slot. We will delve deeper into this application in Chapter 3.

A MA application studied under the prism of bipartite graphs that is very close to time access is that of DSA. In this case, VNs are mapped to users and CNs to channels. The algorithms proposed in this thesis, namely CR-UCB and DYN-CR-UCB leverage the AND-OR tree analysis of random bipartite graphs [47] to prove that players, representing wireless devices, will find a collision-free assignment. In the past, finding the optimal channel assignment has been formulated as a matching problem in DSA in a variety of works [48, 49].

A different type of bipartite graph employed in our work is that of a Coordination Graph. As we briefly discussed in Chapter 1, CGs model the interactions between agents, who are mapped to VNs, as these are attempting to access the common resources and interfering with each other. In this case, a CN corresponds to the *payoff function*, which describes how the rewards that an agent receives are influenced by the actions of agents interfering with it. In this type of application, the main interest is in analyzing the message-passing algorithm that agents employ to coordinate their actions and ensuring that the agents converge to a globally optimal solution. These graphs have been used for ensuring collision avoidance in teams of robots [50], coordinating agents playing Starcraft [51] and solving combinatorial MMABs [52].

Random bipartite graphs owe a considerable part of their theoretical analysis to the

Table 2.1: The role of VNs and CNs in different application areas of bipartite graphs

Area	Variable Nodes	Check Nodes
DSA	users	channels
slot access	users	time slots
MMABs	players	arms
CGs	agents	payoff functions
channel codes	codeword bits	parity-check bits

area of channel coding [53], which aims at answering a fundamental question in the field of communications: *"how can a transmitter ensure that messages communicated through a noisy channel will be correctly received by a receiver?"* To address this issue Forward Error Correcting (FEC) channel codes transform an original message into a more robust representation, termed a *codeword*. Codewords contain redundant bits, which are formed by calculating the *parity-check bits* of the message bits. In linear channel codes, these bits are computed as the exclusive-or of the message bits connected to a parity-check bit. This facilitates the representation of these codes using bipartite graphs, where VNs correspond to codeword bits and CNs to parity-check bits. Examples of channel codes that employ bipartite graphs are Low-Density Parity-Check codes (LDPC) [54], Raptor codes [55] and LT codes [56].

We conclude this chapter with a summary of the different application areas of bipartite graphs in Table 2.1. For each area, we present the entities that the VNs and CNs are associated with.

Chapter 3

Graph-based optimization of multiple access

The MAC protocols that we are considering in this thesis are closely linked to bipartite graphs. In Chapters 7 and 8, we propose an adaptive version of IRSA, which is a MAC protocol whose theoretical throughput analysis and optimization is largely based on tools derived in the study of bipartite graphs. Also, the algorithm that we propose in Chapter 9 employs a mechanism that is similar to the Successive Interference Cancellation (SIC) mechanism used in IRSA, that we describe in detail later in this chapter.

The main idea behind graph-based optimization is that one can analyze the evolution of variables or signals traversing a graph with no explicit knowledge of its structure. This idea was first introduced by Moore and Shannon, in their study of *amplification*, which attempted to estimate the probability that a digital circuit will be reliable even if some of its components are known to be faulty with some probability [57]. Inspired by this idea, the AND-OR tree analysis, introduced in [47], provided the basis for the analysis of a large number of random processes, particularly in the field of channel codes [58, 59, 43] and MAC design [29, 46].

We begin this chapter with an account of the evolution of Random Access protocols that culminates into IRSA. We, then, describe the idea of contention resolution, which was originally introduced in the form of the SIC mechanism [39]. IRSA was the first RA protocol that observed that the SIC mechanism can be analyzed as a random process using the AND-OR tree analysis. In this chapter, we describe IRSA in detail, laying emphasis on the degree distribution $\Lambda(x)$. Finally, we introduce a model of resource-constrained networks, which is used in Chapters 7 and 8. The network model considered in Chapter 9 is a simplification of the one discussed here.

3.1 The evolution of random access protocols

As we briefly discussed in Chapter 1, contention-based protocols are divided into CSMA and RA protocols based on whether devices employ sensing of the common medium prior to transmission in order to establish its availability. Sensing is in general advantageous as it equips devices with knowledge that can be used to reduce collisions. When it comes to wireless resource-constrained networks, however, one needs to take into account that:

(i) sensing requires expending energy, which can decrease the battery of devices significantly, especially if it is not guided by an intelligent strategy for avoiding unnecessary sensing; (ii) low-cost devices may not be equipped with the hardware required for sensing; (iii) even if sensing is employed, collisions cannot be entirely avoided due to the *hidden terminal problem* [60]: if two wireless devices sense the medium simultaneously and find it free of other transmissions, then they will both attempt to transmit, and will therefore collide with each other. We should note that the hidden terminal problem differs from the partial observability problem that we examine in Chapter 9. Specifically, the problem of partial observability arises in the non-sensing setting and is due to the fact that the absence of an ACK signal does not contain enough information to discern an unsuccessful transmission due to an unavailable channel from an unsuccessful transmission due to a collision.

RA protocols have experienced an evolution in sophistication and effectiveness, but the main mechanism, proposed in the original ALOHA [61] protocol, remains the same. In its simplest form, a device following the ALOHA protocol will randomly attempt to access a resource and be informed about the success of its action through an acknowledgement signal transmitted by the receiver. This naïve approach is inappropriate for energy-constrained networks, as it is known that throughput cannot exceed a value of 0.37 [29]. Two subsequent modifications significantly improved the performance of the ALOHA protocol: (i) transmitting a number of copies of the original packet, introduced in [62], brought a slight improvement in throughput by increasing the probability of randomly finding a collision-free resource, provided that the number of devices is not too large; (ii) employing SIC to resolve collisions significantly improved throughput by ensuring that at least one of the copies will be successfully transmitted [39].

The IRSA protocol is the culmination of this series of RA protocols. In IRSA, for each transmission of a packet, a device copies it a variable number of times, decided by sampling a probability distribution termed as the *degree distribution*. Then, the device transmits the replicas in randomly selected slots. By combining this ability with the SIC mechanism, IRSA achieves near-optimal throughput in asymptotic settings [29], i.e., for large frame sizes and networks. The protocol can be configured through the degree distribution, the optimization of which is traditionally performed using evolutionary algorithms [63] and assuming that the number of slots is infinite [29]. We will provide a more in depth description of IRSA in Section 3.3

3.2 The successive interference cancellation mechanism

Let us consider a network of M devices collecting information from their environment and transmitting it to a core network for further process. The main bottleneck of the operation of this network is the transmission of the packets that devices possess through a common communication channel, as it is simultaneously used by neighbouring devices. Abiding to ALOHA and its variants, time is divided into frames of fixed duration, each one consisting of N time slots. At the beginning of each frame each device randomly chooses one of the N available slots to transmit its packet. The channel traffic can be calculated as $G = M/N$. A MAC protocol aims at maximizing the normalized throughput T, defined as the probability of successful packet transmission per slot.

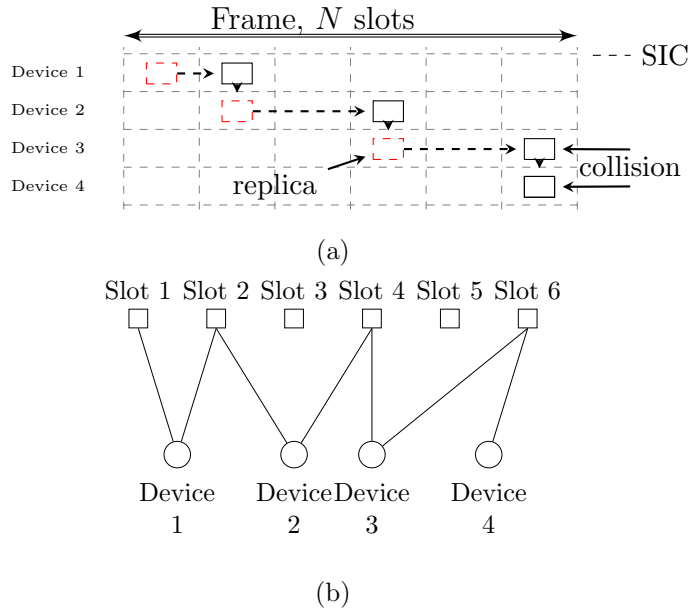


Figure 3.1: A wireless network where devices transmit packets to a common channel following the IRSA protocol. We present a single frame, consisting of $N = 6$ time slots. (a) Transmitted packets under the standard Slotted ALOHA (in black solid line) and replicas under the IRSA protocol (in red, dashed line) (b) the bipartite graph describing this frame.

As we described earlier, protocols that employ SIC allow devices to transmit multiple copies of their packets per frame, which are termed replicas. Packet replicas are indicated in Figure 3.1a with red, dashed outlines. If one of the replicas is transmitted in a collision-free slot, then the packet is successfully transmitted. If, however, two replicas collide, as highlighted in Figure 3.1a for Slot 6, they might still be recovered by removing the interference of a replica that has previously been successfully received. This leads to substantial improvements in the network's throughput. Figure 3.1a illustrates SIC by indicating with a blue, dashed line that the interference of a replica can be removed due to a replica of the same packet having been successfully received in another slot. In particular, the replica of Device 1 is successfully received in Slot 1 and its interference is removed from the replica of Device 2 in Slot 2. Then, the interference caused to the replica of Device 3 due to the replica of Device 2 in Slot 4 is removed. Finally, the interference caused by Device 3 to Device 4 in Slot 6 is removed.

3.3 The Irregular Repetition Slotted ALOHA protocol

In IRSA, a device has the capability of transmitting a variable number of replicas of its packets in the available time slots, decided by randomly sampling the node perspective degree distribution $\Lambda(x)$. The latter is a polynomial probability distribution describing the probability Λ_l that a device transmits l replicas of its message in a particular time frame. This degree distribution is an example of a general degree distribution that can be used to describe the structure of a bipartite graph, as this was described in Section 2.1. The packet transmissions depicted in Figure 3.1a can be mapped to a bipartite graph consisting

Table 3.1: System-related parameters

Symbol	Name
M	number of users
N	number of slots in frame
G	channel traffic
T	throughput
K	number of transmitted packets
PLR	probability loss rate
F	size of packet

Table 3.2: Device-related parameters

Symbol	Name
C_t	condition
l	number of replicas
F_t	number of arrivals in buffer
B	size of buffer
d	maximum number of replicas
b_t	current state of buffer
$\Lambda(x)$	degree distribution

of variable nodes, representing devices, and check nodes representing time slots, as shown in Figure 3.1b.

The design of IRSA aims at selecting the values Λ_l so that the overall network throughput T is maximized. The dependence of throughput on the probability distribution chosen permits us to express T in terms of $\Lambda(x)$. This dependence becomes obvious if one considers the waterfall effect in IRSA [29], that indicates the existence of a threshold value G^* for the channel traffic G, above which transmission will fail with a probability bounded away from 0. It is observed in [29] that, in asymptotic settings ($N \to \infty$), the value of this threshold depends on the degree distribution, namely:

$$G^* < \frac{1}{\lambda_2 \overline{\Lambda}} \tag{3.1}$$

Formally, the optimization objective of IRSA can be cast as:

$$\begin{aligned} &\text{Find:} && (\Lambda^*(x)) : \arg\max_{\Lambda(x)} T(\Lambda(x)) \\ &\text{subject to} && \sum_{l=1}^{d} \Lambda_l = 1, \Lambda_l \in [0,1]. \end{aligned} \tag{3.2}$$

Commonly, IRSA is optimized in an asymptotic setting by iteratively alternating between choosing values for Λ_l and evaluating them using (3.1), in order to acquire a higher threshold G^*. This optimization procedure of $\Lambda(x)$ is computationally intensive, and is performed offline. For large frame sizes, the maximum allowable number of replicas $\Lambda_{\max}$ is also large, and, thus, the number of Λ_l coefficients in $\Lambda(x)$ increases, making optimization harder. Differential evolution, an evolutionary optimization algorithm, is usually employed to find the optimal values of Λ_l [29, 46] due to its ability to efficiently search in large search spaces [63].

3.4 A model for resource-constrained networks

This section presents our assumptions made regarding the physical layer. Tables 3.1 and 3.2 summarize the notation used for system-related and device-related parameters, respectively.

Similarly to the works in [45, 64], we consider frequency channels, characterized at the beginning of each time frame by the traffic G_t, which represents the average number of

attempted packet transmissions by all devices per time slot, with t indicating the time index at the beginning of a frame. We assume that traffic can be estimated perfectly in light of the number of devices and frame size, and that it remains constant during a frame.

Following the work in [37], we assume that the packet throughput T_t and number of transmitted packets K_t can be expressed as:

$$T_t = T(G_{t-1}, K_{t-1}, PLR_{t-1}) \tag{3.3}$$

$$K_t = K(G_t, T_t, C_t, PLR_t) \tag{3.4}$$

where PLR_t is the packet loss rate and C_t denotes a device's condition. The latter can include any information that could potentially affect the behaviour of devices, such as the buffer state and battery level. For the sake of simplicity, we hereafter consider only the buffer state (number of packets in the buffer) of devices. However, the proposed framework is generic, and, depending on the application of interest, can incorporate additional characteristics to C_t. From (3.3), we can also see that T_t is a non-deterministic function of its arguments, as devices randomly select the slots to transmit in. Note that our modelling of the physical layer is oblivious to the underlying modulation, coding schemes and channel noise.

We assume that the transmission buffer of a device is modeled as a first-in first-out queue. At the beginning of a frame, a source injects F_t packets into a finite-length buffer of capacity B. Therefore, the buffer state $b_t^i \in \mathcal{B} = \{0, 1, ..., B\}$ of a device i evolves recursively as follows:

$$\begin{aligned} b_0^i &= b_{init}^i \\ b_{t+1}^i &= \min\{b_t^i - T_t^i(PLR_t, K_t, G_t) + F_t^i, B\} \end{aligned} \tag{3.5}$$

where b_{init}^i denotes the initial buffer state and $T_t^i(PLR_t, K_t, G_t)$ is the throughput. We consider that the packets arriving after the beginning of frame t cannot be transmitted until frame $t+1$. Also, packets whose transmission fails stay in the buffer for future retransmission.

Part II

A deep dive into reinforcement learning

Chapter 4

Learning in a markov decision process

In this chapter, we explore Markov Decision Processes, a major reinforcement learning framework. Our exploration is targeted: we aim at explaining fundamental concepts in our quest to design a framework appropriate for wireless resource-constrained networks. Our proposed framework, Groupwise-Dependent Decentralized Partially Observable Markov Decision Processes [35], will be presented in Chapter 8. Understanding this framework and our learning algorithms requires the description of various concepts from this area.

We start by providing a complete definition of MDPs, which we briefly described in Chapter 1. We, then, present Q-learning, a major learning algorithm in the MDP framework that is the basis of both our proposed solutions in Chapters 7 and 8. The two sections that follow concern partial observability and decentralization. As we discussed in Chapter 1, these two properties possess particular significance for resource-constrained networks. In this chapter, we will examine how their introduction alters the MDP framework.

4.1 The markov decision process

In Section 1.2, we briefly discussed the major components of an MDP, i.e., the agent and the environment: an agent finds itself in a given state and executes actions that affect its environment and incur rewards. We now provide a more formal definition of MDPs.

As part of solving a task, the agent and the environment interact for a series of discrete time steps, $t = 0, 1, 2, 3, \cdots$. At each time step t, the agent receives some representation of the environment's state, $S_t \in \mathcal{S}$, where $\mathcal{S}$ is the set of possible states. Based on the observed state, the agent selects an action, $A_t \in \mathcal{A}(S_t)$, where $\mathcal{A}(S_t)$ is the set of actions available in state S_t. One time step later, in part as a consequence of its action, the agent receives a numerical reward, $R_{t+1} \in \mathbb{R}$, and finds itself in a new state S_{t+1}. [1]

An MDP can be fully described by its state and action sets $(\mathcal{S}, \mathcal{A})$ and by the one-step dynamics of the environment. Given any state s and action a, these dynamics determine the probability of occurrence of each possible pair of next state s' and reward, r as follows:

$$p(s', r|s, a) = \mathbb{P}\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\} \tag{4.1}$$

[1] Throughout this thesis, we denote a random variable with a small letter s, the set it takes values from with $\mathcal{S}$ and its realization in a specific time step t as S_t. We also denote vectors of random variables as $\vec{s}$. This convention follows the notation introduced in [31].

This function is often referred to as the transition model of the MDP.

4.2 Q-learning in a markov decision process

The objective of learning is that of finding the optimal policy $\pi^*(a|s)$, a mapping from states to actions, which, when executed in their corresponding state will lead to accruing the highest rewards. In this framework, we quantify how good a particular state is by estimating a value function. We define the value function under a policy $\pi(a|s)$ as the expected discounted reward starting from state s and then following that policy:

$$V_\pi(s) = \mathbb{E}_\pi[\rho_t|S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^t R_{t+k+1}|S_t = s\right] \tag{4.2}$$

where ρ_t is the expected reward, R_t is the immediate reward at time slot t and $\mathbb{E}_\pi$ denotes expectation under policy π. The parameter $0 \leq \gamma < 1$ is the discount factor that controls the effect of future rewards on the current state; a value of γ closer to zero makes the agent myopic, while when γ is close to 1 the agent is farsighted. Specifically in Q-learning, instead of the value function $V_\pi(s)$, the $Q_\pi(s, a)$ function, often termed as the Q-function, is employed. The Q-function is defined as the expected discounted reward starting from s, taking the action a, and thereafter following policy π.

Formally, the objective of a learner is to compute the optimal policy, defined as:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \big(Q^*(s, a)\big), \quad \text{with } s \in \mathcal{S} \tag{4.3}$$

where $Q^*(s, a)$ denotes the optimal Q-function. In order to compute $Q^*(s)$, the learner performs the following update at each learning time step t:

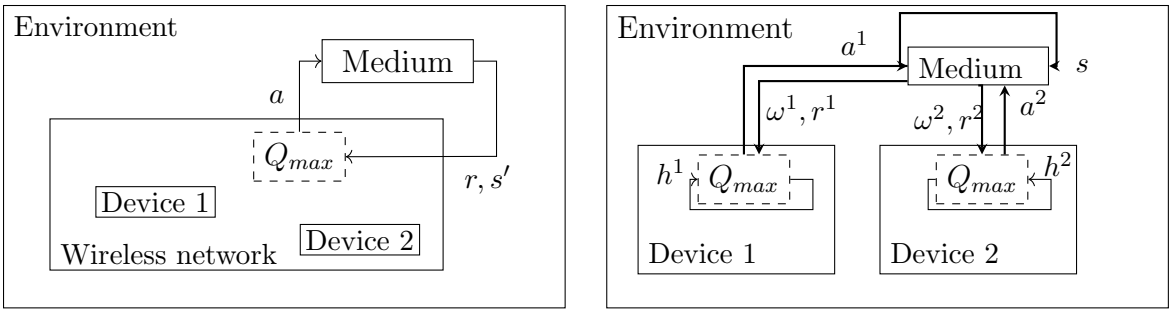
$$Q(S_t, A_t) = (1 - \alpha)Q(S_t, A_t) + \alpha[R_t + \gamma \max_a Q(S_{t+1}, a)] \tag{4.4}$$

It is known that the Q-function computed using this iterative procedure approximates the optimal value Q^* in any MDP under the assumption that the learning rate is lower than one and converges to zero by the end of learning [65].

4.3 Dealing with partial observability

By now, we have assumed that an agent is able to observe the state of the MDP at each time step of interacting with its environment. As we explained in Section 1.2, this assumption may be violated in resource-constrained networks for a variety of reasons. The information required to be present in the state, which depends on the application, may require full observability of the network, advanced hardware and software capabilities or extensive memory. These are properties that resource-constrained networks traditionally lack.

A Partially Observable Markov Decision Process (POMDP) [66] remedies the inability of an MDP to observe its state by introducing the notion of an observation, denoted as Ω_t. An observation contains information that is relevant but insufficient to describe the actual



(a) (b)

Figure 4.1: A wireless network consisting of two devices accessing a common medium modelled as an (a) MDP, where the RL agent includes both devices, and (b) a Dec-POMDP [67], where each device is an agent receiving partial observations ω, instead of observing the full state s.

state on their own. POMDPs can be optimally solved using the framework of Belief MDPs [66], but this renders learning intractable, as it is performed in continuous state spaces. One can instead adopt a fixed history window w and approximate beliefs with a finite-history of observations:

$$\vec{H}_t = \{\Omega_{t-w+1}, \cdots, \Omega_{t-1}, \Omega_t\} \tag{4.5}$$

For example, a device may be able to observe its own battery level, but not the battery level of other devices forming a network responsible for monitoring the temperature in a field. If we assume that the objective of the network is to prolong monitoring, then their decision on whether to hibernate will depend on the decision of neighboring devices. Thus, each device will need to form beliefs on whether another device in its neighbourhood is active and hibernate itself to save energy.

In order to apply Q-learning in a POMDP that approximates beliefs with finite histories of observations, we simply replace the states employed in (4.4) with histories:

$$Q(\vec{H}_t, A_t) = (1-\alpha)Q(\vec{H}_t, A_t) + \alpha[R_t + \gamma \max_a Q(\vec{H}_{t+1}, a)] \tag{4.6}$$

4.4 The decentralized partially observable MDP

A fundamental question that arises when solving a task using reinforcement learning is the following: *who is the agent and what comprises the environment?* While this question finds a straightforward answer in most tasks, it poses an important dilemma when designing solutions for communication networks, and multi-agent systems in general: (i) a centralized solution considers that the whole network represents a single agent and that the environment consists of whichever entity is related to the task and is not part of the network. This can be a communication channel, a base station or environmental factors, such as temperature and humidity; (ii) a decentralized solution considers that each device is an agent. In this case, the environment includes both the entities that are not part of the network and, from the perspective of each device, all other devices of the network. We present the modelling of a wireless network using these two distinct approaches in Figures 4.1(a) and 4.1(b).

A Decentralized Partially Observable Markov Decision Process (Dec-POMDP) offers a powerful framework for designing solutions that take into account partial observability and decentralization. It is formally defined as follows:

Definition 1. ([67]) A Dec-POMDP is a tuple $\langle \mathcal{M}, \mathcal{S}, \mathcal{A}, T, R, \Omega, O, w, I \rangle$, where $\mathcal{M}$ is the set of agents, $\mathcal{S}$ is the finite set of states, $\mathcal{A}$ is the finite set of joint actions, T is the transition probability function, R is the immediate reward function, Ω is the finite set of joint observations, O is the observation probability function, w is the history window and I is the initial state distribution of beliefs at time $t = 0$.

As we observe in this definition, similarly to an MDP, a Dec-POMDP requires the notion of a state space $\mathcal{S}$, an action space $\mathcal{A}$ and the transition probability and reward functions T and R. In addition, it introduces the notion of an observation Ω and an observation probability function O. The latter can viewed as a generalization of T from states to observations.

The introduction of partial observability and decentralization is known to lead to an explosion in the complexity of finding the optimal policy [68]. In the case of large decentralized systems, such as networks of numerous devices, this complexity makes the design of optimal solutions impractical. In general, in order to reduce the complexity of a framework, one can introduce additional assumptions in the transition model of the MDP. The GDD-POMDP framework, which we propose in Chapter 8 is our attempt to reduce complexity in wireless networks by leveraging the locality of interaction that often arises in MA.

Chapter 5

Learning in a multi-player bandit

In this chapter, we present Multi-player Multi-armed Bandits, a major framework in the study of online RL. Without loss of clarity, we also refer to this framework as multi-player bandits.

MMABs are the primary framework for studying dynamic spectrum access (DSA), an important resource allocation task in wireless networks. Under this view, wireless devices are mapped to players and channels to arms, which players pull to receive rewards. MMABs have a long-standing history; an extensive survey of models, algorithms and techniques is outside of the scope of this thesis and chapter. Instead, we take a deep dive into MMABs: we begin with an introductory description of the basic components of the framework in Section 5.1 and the optimization objective of algorithms in Section 5.2. For a comprehensive review of bandits, we refer readers to recent books in this area [**lattimore˙szepesv“IeC −“’a˝ri˙2020**, 32].

5.1 Stochastic multi-player bandits

The thread of bandit history begins in single-player settings, and, in particular, the study of improving the efficiency of clinical trials. In 1933, Thompson provided the first description of a bandit problem as a way to monitor the effectiveness of a drug during the course of a trial in order to abort the trial as soon as the efficacy of the drug was proven or disproven, and thus avoid unnecessary delays and costs [69]. It was soon realized that this problem, which we today call online learning, is ubiquitous in human activities. Areas associated with decision making and profit, such as finance and advertising, have provided a fertile application ground for bandit algorithms. In multi-player settings, the main motivating application can be found in communication networks and, in particular, the study of DSA.

In a single-player MAB, a player sequentially selects an arm to pull among a set $\mathcal{K} = \{1, ..., K\}$ of arms. Pulling an arm at a given time step t corresponds to the action a_t of the player and incurs a reward r_t, which depends on the choice of arm. A player pulls arms and accrues rewards until the end of the problem horizon, denoted with T. Multi-player Multi-armed Bandits are a generalization of this framework to the case where $M >= 2$ players compete for the same set of arms. Each player is employing a learning algorithm, which is designed with the objective of ensuring that the player will accrue the maximum

possible reward within the problem horizon. The output of an algorithm is a mapping from the experience of the player, i.e., the set of observed actions and rewards, to the action to be performed at the current time step. In general, we refer to this mapping as a *strategy*.

Bandit models are primarily categorized based on the assumptions they impose on rewards. As rewards define the nature of the learning problem, different bandit models are optimally solved using algorithms that vary significantly in their objectives. A common assumption, which is also adopted in our work, is that each arm is associated with a probability distribution that remains fixed throughout learning. This distribution is traditionally characterized by its expected value, μ_k. We denote with $Y_{k,t}$ the i.i.d. random variable associated with the rewards of an arm, that satisfies $P(Y_{k,t} = 1) = \mu_k$ and refer to it as the availability of the arm. Bandits employing this assumption are termed as *stochastic*.

While the reward observed by a player is identical to the availability of the arm pulled by that player in a single-player setting, multi-player bandits exhibit a more complex reward model. Here, a player observes a reward of zero, even if the arm that they pulled was available, when at least one of the other players pulled it simultaneously. The reward model is formally:

$$r_{a_m}(t) = y_{a_m}(t)(1 - n_{a_m}(t)) \tag{5.1}$$

where $n_{a_m}(t)$ indicates whether player m collided with another player in their attempt to pull arm a_m at time step t and $r_{a_m}(t), y_{a_m}(t), n_{a_m}(t)$ are assumed to be binary. As we will see in Chapter 6, different bandit algorithms make different assumptions about which of these three variables are directly observed by a player. This gives rise to two major settings for studying DSA: (i) in the sensing setting, a player observes both $n_{a_m}(t)$ and one of the two variables $y_{a_m}(t), r_{a_m}(t)$. Thus, the player knows whether a reward of zero was due to an available channel or a collision and can appropriately adjust its strategy. (ii) in the no-sensing setting, a player can only observe $r_{a_m}(t)$. This renders the environment partially observable, making the task tougher to solve.

There exist two important alternatives to stochastic bandits: (i) adversarial bandits [32] make a worst-case assumption about the reward a player will observe at any given time step. By considering that arms are controlled by an adversary who has full knowledge of the strategy of a player, a player solving an adversarial bandit expects to observe the worst-possible reward. This type of bandit is useful in safety-critical applications, where robustness to attacks or random failures is important. Otherwise, it is not preferred as it leads to strategies that are too conservative and accrue small rewards in non-adversarial settings. (ii) non-stationary bandits [70] are a generalization of stochastic bandits to the case where the mean μ_k characterizing an arm varies with time. This is a relatively recent addition to the family of bandit models, which is primarily motivated by the study of real-world dynamic environments where the availability of resources may vary with time.

While in this thesis we focus on stochastic bandits, there are a number of ways in which these alternative models can become intertwined with our study of multi-player bandits. First, modelling the bandit as an adversary is a common way to derive a lower bound on the performance of any algorithm [**lattimore˙szepesv"IeC −"'a"ri˙2020**]. This bound is useful for comparing different algorithms and acquiring an understanding of the difficulty of solving a bandit problem. Second, in some applications the availability of channels may

be assumed to be stochastic, but other players may be viewed as adversaries. This setting arises in competitive scenarios, while the scenarios that we consider here regard *cooperative* networks, where the objective is to improve the performance of the whole network, and not just that of individual devices. Finally, non-stationary bandits are often considered in multi-player bandits in the no-sensing setting, because the presence of multiple players who are simultaneously updating their strategies effectively renders the environment non-stationary. One should however take into account that the non-stationarity attributed to the actions of players is different from the non-stationarity of arm availabilities: the former can be controlled by the designer and will disappear when all players have converged to their optimal strategy, while the latter is part of the external environment and requires adaptation.

5.2 Evaluating a bandit algorithm

The objective of a multi-player bandit algorithm is to discover the optimal assignment of players to arms as quickly as possible, so that the rewards accrued by the team of players within the problem horizon is the largest possible. An alternative to measuring rewards, that is particularly prominent in bandits, is that of measuring *regret*. The regret at a given time step is calculated as the difference between the reward that a player observes and the one that they would have observed, had they performed the optimal action. Thus, this alternative measure quantifies how much a player regrets their actions in hindsight.

The objective of the team of players is to minimize their expected cumulative regret at the end of the horizon, defined as:

$$R(T) = T \sum_{k=1}^{M} \mu_k^*(t) - \sum_{t=1}^{T} \sum_{m=1}^{M} r_m(t) \tag{5.2}$$

where μ_k^* is the mean availability of an arm belonging to the set of arms with the M highest means, which we denote as $\mathcal{M}^*$. We refer to such an arm as an M-best arm.

How to compare bandit algorithms Comparisons among bandit algorithms are primarily performed by deriving upper regret bounds. An upper bound on regret is satisfied with high probability and is valid among all possible reward distributions satisfying the stochastic assumption. An important target when designing a bandit algorithm is to prove that the regret bound scales sub-linearly with the problem horizon T. This suggests that, in order for an algorithm to be optimal, it needs to stop accruing regret before learning completes. When comparing two algorithms that both exhibit a sub-linear regret bound, we can prove that one of them is superior by comparing the constants appearing in the bound. These constants depend on parameters of the problem, such as the number of arms, the number of players and the hardness of the bandit setting. In general, hardness depends on how distinguishable the availability of the optimal arm is from the availabilities of all other arms.

Empirical simulations are not useful in the comparison of different algorithms, as they cannot offer an exhaustive evaluation for all possible distributions. Nevertheless, they can be used to disprove a sub-linear regret bound. In particular, if an algorithm exhibits linear regret in at least one bandit instance, then we can confirm that no upper bound exists.

In order to prove that an algorithm is optimal for a considered bandit model, one needs to also derive a lower regret bound. A worst-case lower bound L states the following: *"For any algorithm you give me, I will give you an instance of a bandit problem on which the regret is at least L"* [**lattimore˙szepesv"IeC −"'a"ri¨2020**]. Thus, lower bounds capture the difficulty of a bandit model, regardless of the algorithm used to solve it. By showing that the upper and lower bounds of an algorithm are equal, one can prove optimality. It is customary to characterise an algorithm as *order-optimal*: this means that the lower and upper bounds of the algorithm match up to a constant factor, which can be improved if one introduces additional assumptions about the distributions and designs an algorithm that takes them into account.

5.3 The principle of optimism in the face of uncertainty

Stochastic bandits are the oldest in the family of bandit models. It should not, thus, be a surprise that they are the most extensively analyzed bandit model. We today possess both a clear understanding of their nature and optimal algorithms for a single-player setting. In what follows, we will analyze the principle of Optimisim in the Face of Uncertainty (OFU), which has given rise to many algorithms that address the exploration versus exploitation dilemma optimally in stochastic settings. As our work regards multi-player settings, we will then explain why the OFU principle can fail to offer an optimal solution in a simple multi-player setting.

The OFU principle states that, in the face of uncertainty, one should act as if the environment is as nice as possible. This attitude favours exploration at the early stages of learning, where uncertainty is high. As the player acquires more experience, they become more certain on which arms give high rewards and, thus, reduce exploration in order to exploit the best arms. In addition to having premises in human psychology [71], this principle is known to lead to optimal strategies in any single-player stochastic environment [72].

The OFU principle gave rise to the family of Upper Confidence Bound (UCB) algorithms. In this section, we will review the most fundamental form of a UCB algorithm, UCB1 [72]. By introducing additional assumptions, different algorithms achieve better regret bounds in specific problem settings. For example, klUCB is another algorithm that leverages the assumption that arms follow Bernoulli distributions in order to compute tighter regret bounds than UCB1 [73].

UCB algorithms belong to the family of index-based policies, which attempt to find an optimal allocation by minimizing the number of times a sub-optimal arm is pulled. It is known that, under the assumption that the rewards are bounded, the regret achieved by index-based policies is asymptotically ($T \to \infty$) optimal [74]. All players compute an index i_k for each arm, which quantifies the empirical mean of an arm k under uncertainty about its actual expected value, and pull the arm with the highest index, k^*. To compute the index, UCB algorithms traditionally update an empirical estimate of the mean of expected rewards $\hat{\mu}_k = \frac{\sum_{t=1}^{T} r_{m,t}}{T_{k,t}}$, where $T_{k,t}$ denotes the number of times arm k has been pulled by time step t. In addition, players compute a bound $B_{k,t}$, which defines the size of the one-sided confidence interval for the average reward within which the true expected reward falls with high probability. In essense, this bound quantifies the uncertainty of the player in its

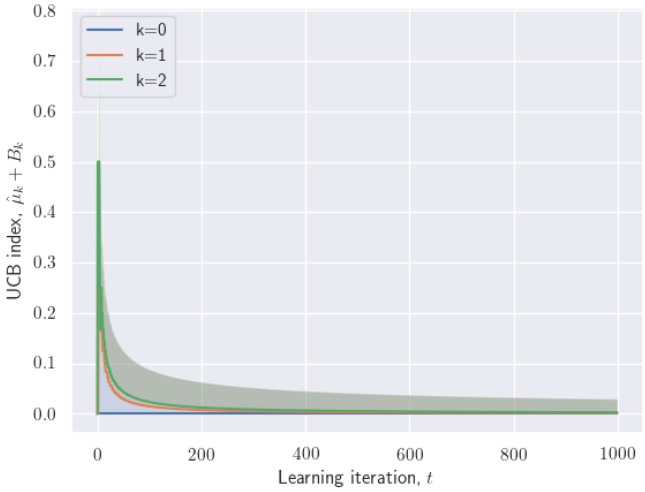


Figure 5.1: Example of failure of UCB in the no-sensing multi-player setting: a player in a network with $M = 2$ players and $K = 3$ channels that are always available($\mu_1 = \mu_2 = \mu_3 = 1$) wrongly estimate that all arms have 0 availability.

estimation of $\hat{\mu}_k$.

Formally, a UCB-based algorithm solves the following optimization problem:

$$\begin{aligned} k^* &= \arg\max_{k \in K} i_k \\ \text{where } i_k &= \hat{\mu}_{k,T_k} + B_{k,T_k} \end{aligned} \tag{5.3}$$

UCB1 calculates the confidence interval by leveraging the Chernoff-Hoeffding bound of sub-gaussian random variables [72]. Thus, no assumptions are made about the reward distributions, except that they have a bounded support $[0, b]$, where b can be take any positive finite value. The bound for the index of arm k is calculated as:

$$B_{k,t} = \sqrt{\frac{2b^2 \log t}{T_{k,t-1}}} \tag{5.4}$$

where t is the current time step. The expected cumulative regret of UCB1 is bounded as:

$$\mathbb{E}[\hat{R}_k] \leq 8\Big(\sum_{k:\mu_k < \mu^*} \frac{b^2}{\Delta_k} \Big) \log t + O(1) \tag{5.5}$$

where Δ_k is the difference between the mean availability of arm k, μ_k and that of the optimal arm, μ^*. As it is known that stochastic bandits are characterized by a lower bound that is logarithmic in T, one can conclude from this upper bound that UCB1 is order-optimal.

5.4 Optimism and partial observability

As we discussed in Section 5.1, the reward in multi-player bandits is characterized by partial observability when players cannot directly observe the availability of a channel or collisions

occurring with other players. This property, captured in Eq. (9.1), suggests that the mean estimates a player computes using UCB algorithms can be biased under collisions and, therefore, do not approximate the true mean of arms. Thus, in the no-sensing setting, the upper regret bound of UCB1 is not valid. This failure of UCB1 can be observed through the following simple problem: consider a bandit setting with $K = 3$ arms with well-separated availabilities ($\mu_1 = 0.1, \mu_2 = 0.55, \mu_3 = 0.9$) and $M = 2$ players employing the UCB1 algorithm described above. As the players initially explore arms randomly, they inevitably collide. If collisions are frequent at this early stage, which occurs with some probability, the two players may conclude that all arms have low availability. In this case, the players will keep colliding until the end of the problem horizon. We present this behaviour in Figure 5.1, where we view the mean estimate (in solid lines) and the upper confidence bound (shaded area) that one of the two players has computed for the three arms. Despite the fact that all true means are positive, the player estimates that they are all close to zero with high certainty.

Part III

Reinforcement learning-based multiple access

Chapter 6

State of the art

Previous chapters provided a discussion on the history, recent scenery and needs of wireless resource-constrained networks, as well as the theoretical background required to understand the main algorithmic tools employed in this thesis, namely bipartite graphs and RL algorithms. Now, readers can appreciate a review on works related to our contributions, which will be presented in Chapters 7, 8 and 9.

We begin our review of related works by examining state-of-the-art graph-based RA protocols. The common property of such protocols is that they leverage theoretical tools from the analysis of graphs to optimize the process of collision resolution [1]. Different protocols employ different techniques when it comes to collision resolution; in Chapter 3, we presented the SIC mechanism, introduced in [39] and employed by the IRSA protocol, the protocol that we improve upon in Chapters 7 and 8. We will now review alternatives protocols that also employ SIC and describe how they fit in the landscape of resource-constrained networks.

We, then, focus on the state of the art in employing RL for optimizing MA. We are not aiming at an exhaustive review of these vast fields, but attempt to organize the related literature based on design considerations frequently encountered in wireless resource-constrained networks. We begin with general RL considerations and, then, delve deeper into recent works in multi-player bandits.

6.1 A review of of graph-based random access protocols

The elegant simplicity of the original ALOHA protocol has significantly influenced the contention-based RA protocols that followed. Today, descendants of ALOHA are widely used both in terrestrial and satellite communications [75, 76]. The driving factor behind the evolution of ALOHA-based protocols is improving throughput without significantly increasing the complexity of MA, in terms of required hardware on devices and operational cost of the communication system. For a comprehensive review of contemporary RA protocols following the ALOHA paradigm, we refer readers to [77].

Coded Slotted ALOHA (CS-ALOHA) [45, 78], is a state-of-the-art protocol that can be

[1] Commonly, collision resolution is also termed as contention resolution in the area of MA access.

viewed as a generalization of IRSA. Instead of transmitting identical copies of their packets, devices under this protocol employ FEC codes, which we briefly described in Section 2.2. Specifically, a device breaks its packet into segments and encodes them using a FEC code prior to transmission. By performing maximum-a-posteriori decoding [59], a receiver can recover the encoded segments despite potential collisions. An important property of this protocol is that the receiver does not need to provide feedback to the devices, i.e. no acknowledgement signal is transmitted, as error correction ensures that a packet is successfully transmitted. By combining SIC with coding, CS-ALOHA achieves higher throughput for networks with medium to high loads. When it comes to low loads however, where IRSA performs equally well, the additional complexity introduced by encoding and decoding makes CS-ALOHA inappropriate. Our work in Chapters 7 and 8 improves the performance of IRSA for medium and high loads and, therefore, bridges the distance in terms of performance between these two protocols. Simultaneously, our solutions maintain the attractively low complexity of IRSA.

The CS-ALOHA protocol was further improved in [79], with the introduction of *spatial coupling*. The idea of spatial coupling transcends the field of Multiple Access; it originated in the study of LDPC codes and is today used in many other areas of communications and signal processing, such as compressive sensing [80] and source coding [81]. In spatially coupled ALOHA, devices become active in different frames with a certain probability and frames are organized in *megaframes*. In order to transmit a packet, devices randomly pick one of the slots of the frame that they were activated in and, then, randomly select slots in the subsequent frames to transmit copies of the original packet with uniform probability. Although this protocol improves upon the performance of CS-ALOHA, it introduces additional complexity and delays in the transmission of packets.

An alternative to assuming that time is divided into frames that devices enter in a synchronous manner, is to assume that communication is frameless. While the concept of a frame still exists in the frameless slotted ALOHA protocol [82, 83], its length is not determined a priori. Instead, new slots are added until a sufficiently high fraction of devices has found a collision-free arm. Devices attempt transmission with a predefined probability, termed as the slot access probability, which is broadcasted to them by the receiver. By monitoring the success of transmissions, the receiver therefore decides when to initiate the next frame and broadcast the slot access probability for it. As was observed in [82], frameless slotted ALOHA achieves very close to optimal throughput for large frame sizes and improves upon the performance of IRSA for medium frame sizes. However, it comes with two limitations, compared to IRSA: (i) the broadcasting of information at the end of each frame requires additional communication. In addition to the fact that broadcasting increases complexity, one should take into account that there are applications, as the one that we will analyze in Chapter 9, where broadcasting is not possible; (ii) in order to compute the slot access probability the receives needs to know the number of devices in the network. This requirement is particularly limiting in ad hoc networks where devices exhibit mobility or hibernate to reduce energy consumption. While there exist techniques for estimating the number of devices [84, 85], they are associated with extensive exchange of information and assume that a centralized point of control exists. In contrast, the estimation technique that we introduce in Chapter 9 is fully decentralized and leverages the existing acknowledgement signals received by devices..

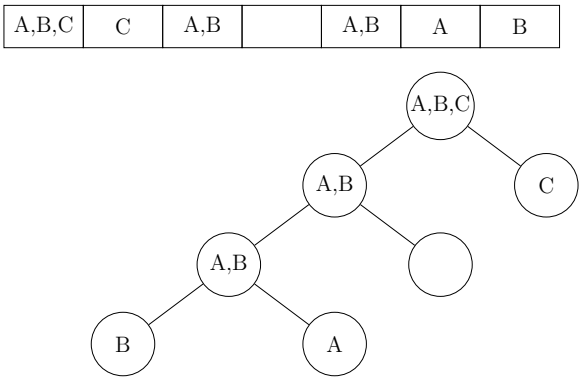


Figure 6.1: A frame under transmission employing the Standard Tree Algorithm and the corresponding tree describing collisions, successes and idle slots. The three packets transmitted by three devices are denoted as A,B and C. We observe that 7 slots are required to successfully receive them all despite collisions. The binary tree describes the process of collision resolution, where circles with a single packet correspond to successfully received packets and empty circles correspond to idle slots.

In the family of protocols following a Tree Algorithm (TA) approach, which arose independently from the ALOHA protocol, RA is contention-based and can be described by graphs that have a tree structure [86]. Although the throughput achieved by the Standard Tree Algorithm is inferior to the one achieved by Slotted ALOHA, this protocol does improve upon the *stability* of the network [86]. In general, we characterize a network as stable when delays in transmitting packets are bounded. In [87], the standard algorithm was combined with SIC to form the SICTA protocol, which significantly improved upon its throughput. However, its performance remains inferior to the one achieved by protocols based on bipartite graphs, such as IRSA and CS-ALOHA. To get a clearer picture of the difference between bipartite representations, which we have thoroughly reviewed in Chapters 2 and 3, and tree graphs we present the tree describing a given frame under TA in Figure 6.1.

As we discussed in Chapter 2, there has recently been an exchange of ideas between the research communities of MA and channel coding [44]. Although seemingly disparate, these two fields find a common ground on the use of bipartite graphs for analyzing the success of resolving collisions. The graph-based protocols that we have reviewed so far exemplify this relationship: (i) CS-ALOHA directly uses channel codes [43] to encode packets prior to transmission; (ii) frameless ALOHA is inspired from rateless codes [56]; (iii) IRSA leverages the analysis of irregular LDPC codes [54, 88]; and (iv) spatially coupled Random Access (RA) is inspired by the convolutional LDPC code construction [89, 90].

6.2 A review of reinforcement learning for multiple access

The efficiency of RA protocols can be significantly improved if devices access resources intelligently in order to reduce collisions. Reinforcement learning makes this possible, as a device can learn about the availability of resources and access patterns of other devices by interacting with them and adapting its behaviour to improve its performance. The RL-MAC protocol [91] is an adaptive variant of S-MAC [28], which first introduced the division of

the operation of WSNs into active and sleeping phases. Devices using RL-MAC adapt their schedule to their own traffic and traffic produced by neighbouring devices using Q-learning in order to reduce their energy consumption while maintaining high throughput. The ALOHA-QIR protocol [92] is a variant of ALOHA that improves energy efficiency by minimizing collisions between devices, which employ Q-learning to detect the most desirable slot to transmit in. The performance of IRSA has also been recently improved in [93], where the use of Multi-armed Bandits (MABs) was introduced as a remedy for inaccurate asymptotic analysis in non-asymptotic settings and as an alternative to computationally expensive finite length block analysis. This work has been proposed for an IRSA variant that incorporates devices' prioritization [46]. A disadvantage of the bandit algorithm proposed in [46] is that it is theoretically shown to exhibit sub-optimal performance.

In the following, we review considerations that have particular interest from the perspective of a system designer aiming to equip wireless network with RL abilities. We should note that the ecosystem of RL algorithms exhibits large variability that is outside of the scope of this thesis. Instead, we focus on concerns particularly relevant in wireless resource-constrained networks.

Which reinforcement learning framework to use? In Chapter 1, we presented bandits and MDPs, two major RL frameworks that can be used to model learning in a wireless network. One of the first questions a system designer will arguably encounter is: *"which are the traits that make one framework preferable to others in a particular application?"* While the two discussed frameworks only differ in their choice of whether to employ the notion of a state, they have been, up until now, primarily employed by disjoint communities with distinct objectives. This often creates the illusion that they differ significantly. In essence, the scientific community of MABs has focused on discovering efficient approaches to exploration, while the community of MDPs, concerned with the study of a more generic framework, primarily researches on policies and value functions. Recently, a large part of MDP research has also been devoted to approximating policies using artificial neural networks [94], a sub-area termed as deep reinforcement learning. However, the research paths of MDPs and bandits have recently merged, as the complexity of deep reinforcement learning algorithms has made exploration an essential part of any RL solution.

On the contrary, alternative RL frameworks, such as learning automata [95], have remained disconnected from the bulk of RL research. Learning automata can offer significant advantages in terms of efficiency and have been extensively used in the optimization of MA in wireless networks [95, 96]. Arguably, their absence from the deep learning scene can be attributed to the fact that they are not employing value functions; combined with artificial neural networks for function approximation, value functions are the driving force of most contemporary RL applications.

Model-based or model-free? In this thesis, we are specifically interested in *model-free* settings, where the learner does not have an understanding of the dynamical behaviour of its environment. The orthogonal approach of *model-based* learning can lead to more efficient solutions, but makes the unrealistic assumption that a model of the environment is available or can be learned quickly. Model-free learning solutions are *online*; the learner processes

experience as it comes along and attempts to learn how to behave in order to accrue rewards as quickly as possible. This comes in contrast to model-based settings, where the learner devotes some initial time to exploring its environment and modelling its dynamics prior to learning how to solve a task.

The model-based paradigm has served the needs of wireless communication until recently, but in the era of 5G and beyond standards, the community is shifting more towards data-driven solutions [97]. The reason for this is that networks are becoming increasingly complex, making their modelling computationally intensive and, often, impossible due to their ad hoc nature. At the same time, devices are generating huge volumes of data, which often contain information useful for their optimization. Arguably, the future of communications lies in the elegant combination of the efficiency of model-based and the wide applicability of model-free approaches [98]. The technique of Virtual Experience, described in detail in Section 7.2.2, is an example of how partial knowledge of the model can help in improving the convergence rate of model-free algorithms.

Is deep always better? The recent introduction of function approximation in classical RL algorithms, which signified the birth of deep reinforcement learning, was an important step towards bringing RL closer to real-world applications. Deep reinforcement learning algorithms have been used for optimizing DSA [99, 100, 101] and random access in IoT [102]. By combining classical RL algorithms with function approximation, these algorithms enable the application of RL in complex domains, where a coarse discretization of the state-action space may lead to sub-optimal performance of the learned solutions. However, convergence of deep RL to optimal strategies is an open issue and training has to be performed offline and centrally due to the computational complexity of training deep neural networks. This, in combination with the fact that, if the network conditions change significantly retraining of the system is required, suggests that deep RL may not be computationally feasible in resource-constrained networks of realistic size, where the topology and channel conditions may vary with time.

How to select the reward function? Arguably the most important and difficult step in formulating and successfully solving an RL task is that of defining the reward function. Rewards, a vital component of RL algorithms, are directly influenced by the hardware and software characteristics of devices. In general, the more sophisticated the rewards an algorithm has been designed for is, the more restrictive is its use in resource-constrained networks. Thus, algorithms designed with the simplest type of reward, an acknowledgement signal transmitted by the receiver in the case of a successful transmission, can be seen as universal. In contrast, solutions following a CSMA approach, i.e., assuming that devices have the capability of sensing the medium prior to transmission, are limited to specific types of devices and networks. LoRaWAN [14] is an example of an emerging standard developed for IoT networks, which does not employ sensing information. This simplification of the reward signal, however, can cause problems in the application of RL. For example, as we explained in Chapter 5, rewards exhibit partial observability in multi-player settings due to collisions between players. We will address this challenge in Chapter 9.

6.3 A review of multi-player bandit algorithms for dynamic spectrum access

Early on in the study of DSA, researchers realized that the problem faced by secondary users is identical to the exploration/exploitation dilemma formulated in the MAB framework [103]. Solutions are traditionally called algorithms and are primarily decentralized and contention-based, due to the ad hoc nature of networks and the lack of a centralized coordinator. While some algorithms follow the CSMA approach [42, 41, 104], recent solutions for IoT networks do not assume that sensing is possible and follow an RA approach [40, 105, 106]. The main objective when designing such algorithms under the MAB framework is to provide theoretical guarantees for the optimality of a solution while avoiding communication between devices and minimizing learning time.

The primary objective of our short review is to describe the different options in designing an algorithm for multi-player bandits and explain how these affect properties of the solution, such as its optimality, complexity and applicability. In addition to a high-level description of the different algorithms, we compare their upper regret bounds in Figure 6.2. A summary of the properties of these algorithms is presented in Table 6.1.

Theoretical guarantees versus simplicity An important consideration when designing a bandit algorithm is whether we aim to derive an upper regret bound for it. The benefits of accompanying an algorithm with a bound should not be understated: (i) the bound is a quantifiable measure that can be used to position the algorithm in the related literature in terms of performance; and, (ii) studying a bound helps understand the difficulty of the learning problem. It should, therefore, be a surprise that most works in this area, including our own, provide regret guarantees [42, 41, 105]. Nevertheless, the machine learning community has always been receptive to heuristic solutions, which work well in practice despite their lack of a theoretical understanding. Practices such as assuming independence among agents or using neural networks to approximate the Q-value function [94], are extensively used today even though it is known that they do not have convergence guarantees. In stochastic multi-player bandits, the family of *Selfish* algorithms [40] encompasses algorithms that ignore the violation of stochasticity in the non-sensing setting and follow a classical UCB approach. As we witnessed in Figure 5.1, Selfish algorithms can fail in some simple problem settings. Their advantage lies in their simplicity and quick convergence rate in most bandit instances.

To sense or not to sense? Another important distinction when it comes to algorithms for multi-player bandits is whether they employ sensing information or learn using partially observable rewards. Arguably due to the difficulty of theoretically analyzing the no-sensing setting, the majority of existing algorithms assume that devices can sense [42, 41, 40]. Notable exceptions that have a regret bound are the algorithm proposed in [105], DYN-MMAB [42] and our proposed algorithms, CR-UCB and DYN-CR-UCB. As we discussed in Chapter 1 and previously in this chapter, contemporary resource-constrained devices are not always capable of sensing. In addition to being applicable in a large variety of application, our proposed algorithms perform closely to state-of-the-art algorithms that employ sensing, as we can see in Figure 6.2. This suggests that the lack of sensing information does not harm the

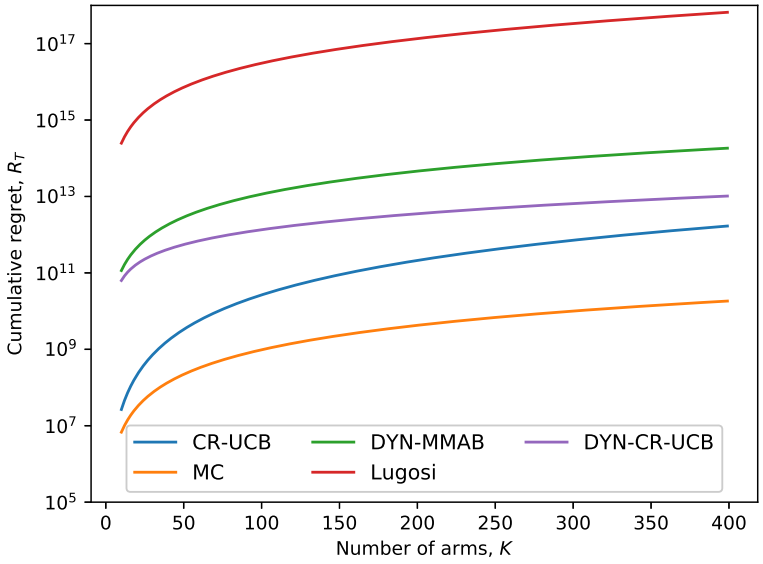


Figure 6.2: Comparison of the upper bounds on the duration of algorithms that do not require the observation of collisions (CR-UCB, SIC-MMAB [42], DYN-MMAB [42], the algorithm in [105]) and MC [41], which observes both collisions and availabilities. The load in all problem settings is 0.5 and mean availabilities are randomly sampled in the range $[0.1, 1]$.

operation of devices significantly as long as a mechanism for resolving collisions is employed. We should note for the algorithm in [105], that we have also taken into account the time required to compute M, in order to ensure a fair comparison with the rest of the algorithms, which do not require knowledge of M.

Synchronous versus semi-synchronous learning An implicit assumption in the early study of multi-player bandits was that all players start learning simultaneously. As was proven in [42], this setting, which is termed *synchronous*, is particularly convenient, as players can leverage pulls to indirectly communicate information to other players and, thus, reach a consensus on the optimal assignment. In contrast, the *semi-synchronous* setting considers that each player arrives in the game at an arbitrary time step, unknown to all players. The reason why this setting is not termed asynchronous is that it still makes the minimum assumption that players experience time steps in a synchronous manner. Traditionally, algorithms designed for the synchronous setting are termed *static*, while algorithms for the semi-synchronous setting are termed *dynamic*.

In this thesis, we study both static and dynamic settings. Although a definitive proof does not exist, dynamic algorithms face a tougher problem than static ones, so they are expected to exhibit larger regret bounds. On the other hand, dynamic algorithms can provide solutions for a larger variety of applications, such as wireless networks where devices exhibit mobility or sensor networks where sensors hibernate to reduce energy consumption.

We conclude this chapter with a summary of the discussed bandit algorithms in Table 6.1. In particular, we indicate whether the algorithms assume that sensing information is

Table 6.1: Review of MMAB algorithms

Algorithm	Sensing	Dynamic	Regret bound
Selfish [40]	✗	✓	✗
SIC-MMAB [42]	✓	✗	✓
DYN-MMAB [42]	✗	✓	✓
Musical Chairs [41]	✓	✗	✓
MCTopM [40]	✓	✗	✓
CR-UCB	✗	✗	✓
DYN-CR-UCB	✗	✓	✓

available, whether they can be employed in dynamic settings and whether an upper regret bound exists. CR-UCB and DYN-CR-UCB are the algorithms that we propose in Chapter 9 of this thesis.

Chapter 7

Optimizing IRSA using independent Q-learning

As we discussed in Chapter 1, designing efficient and low-complexity MAC protocols is an important frontier in resource-constrained networks. Among the various RA protocols discussed in Section 3.1, we distinguished IRSA [29] as a RA protocol that holds great promise due to its low complexity and clear theoretical understanding that guarantees close to optimal performance. SIC is an integral part of the IRSA protocol, where devices transmit multiple replicas in different time slots and the mechanism of interference cancellation ensures that transmission is successful with high probability. IRSA, however is traditionally optimized for infinite frame sizes and networks with low traffic, which makes its performance sub-optimal in networks with moderate numbers of devices and high traffic. The research question that led us to was: *"is it possible to design a variant of IRSA that can adapt its operation to variability in the network size and environmental conditions without introducing significant complexity?"*

We tackle this question by introducing a reinforcement learning algorithm for optimizing the degree distribution of IRSA that requires no communication between devices. This algorithm is independent Q-learning, where each device is an agent that learns based on a finite history of observations of its own state. We model the wireless network as a Dec-POMDP in order to capture the properties of decentralization and partial observability. Simulations show that the degree distributions learned using our algorithm surpass the throughput of designs classically employed in IRSA [29].

As our learning-based solution improves upon the IRSA protocol, it borrows some of its limitations that may render it in appropriate for some applications. In particular: (i) IRSA belongs to the family of RA protocols whose aim is to trade optimality for low complexity. Thus, our solution may not be preferred in data-critical applications that impose strict requirements on the loss of packets due to overflows; (ii) the SIC mechanism in IRSA requires that the base station can perform SIC. This functionality is available in many IoT networks, but may not be satisfied in ad hoc networks such as VANETS and sensor networks with epidemic routing. Furthermore, the existing analysis relies on the assumption that no impairments and capture effect take place at the physical layer. This assumption can be however raised by employing the analysis of the effect of these phenomena on the ability of SIC to resolve collisions, presented in [29]; (iii) devices employing the IRSA protocol need

to be able to make multiple transmissions per frame. This slightly increases the energy consumption, but is not prohibitive for most applications as the number of replicas is bounded and relatively small (e.g. at most 8 in a network of up to 200 devices [29]);

7.1 Motivation for independent Q-learning

To design adaptive MAC protocols appropriate for resource-constrained networks, we first view MAC design as a multi-agent system, where agents learn how to transmit their packets by interacting with their environment. Specifically, we assume that each device is an agent aiming at maximizing the throughput of the network. Agents in our formulation perform actions that correspond to the coefficients of the degree distribution employed by IRSA to determine the number of copies of packets. For more details on degree distributions we refer the reader to Section 2.1, where we described them. The objective of the network is to maximize the common channel throughput, while rewards are associated with the success of transmission, which is guaranteed if devices choose collision-free slots. We also consider partial observability in our definition of states, as devices can observe only information that is local to them and is possibly inaccurate due to the limited capabilities of devices. For example, a device may be able to observe its own battery level, but not the battery level of other devices forming a network.

The reinforcement learning algorithm employed in our solution is Q-learning [107], which has been extensively used for rendering wireless networks adaptive [108], as it achieves a fine balance between low complexity and satisfactory performance. Q-learning does not require a model of the environment, a particularly advantageous trait for real-world ad hoc networks. As we discussed in Chapter 6, it is possible to derive a model of the environment in order to perform model-based reinforcement learning [107, 92], but this would significantly affect the time and computational complexity of optimization.

We propose to tackle the complexity associated with multi-agent reinforcement learning in partially observable settings by employing two techniques: (i) adopting finite histories of observations to approximate the continuous beliefs of Belief MDPs, which significantly reduces the size of the state space and, as we prove in Section 7.3.1, can still lead to policies with near-optimal performance. For a reminder on Belief MDPs, we refer the reader to Section 4.3; (ii) assuming that each device learns independently from other devices, by updating its local Q-function based on its individual observations and actions. Although naïvely ignoring agents' interactions, this technique has been found to converge when coupled with exploitive exploration strategies in [109] and exhibits low complexity.

Furthermore, we investigate the effect that partial observability has on the quality of the solution and prove the existence of near-optimal solutions for a POMDP employing a finite history of past observations. This result is novel and can be employed for characterizing the optimality of learning solutions for resource allocation problems that exhibit the waterfall effect. Our proof is based on a similar analysis that corresponds to a centralized setting [110]. We, therefore, need to investigate how decentralization and the assumption of independent learning affect the quality of the learned policies. The conditions under which independent learners are guaranteed to converge to equilibrium points were derived in [109], but it was observed that these points can correspond to sub-optimal solutions and that escaping them

can prove hard, as it may require extensive exploration or some form of coordination among agents. In our analysis, we justify why the properties of the problem under investigation are such that only two equilibrium points arise and, based on this observation, derive a technique to avoid sub-optimal solutions during learning.

Apart from the quality of the solution, convergence rate is also an important criterion when evaluating a learning technique. In Q-learning, convergence to a stationary policy is guaranteed at the end of an episode, which refers to the period of interaction of an agent with its environment until a terminal state is reached, provided that the environment is stationary. However, in wireless networks, the time-varying nature of the network and channel conditions renders the learning environment non-stationary, which effectively means that the optimal policy can change during the course of an episode. Thus, Q-learning will exhibit sub-optimal performance, if the environment changes at a rate quicker than its convergence rate. To address this issue, our solution equips Q-learning with the concept of virtual experience (VE) [111], where an agent updates multiple state-actions pairs at each Q-learning iteration by "imagining" state visits. These visits correspond to state-action pairs termed virtual, whose defining property is that they are equivalent in front of the unknown environment dynamics. Our theoretical analysis formulates the effect that virtual experience has on the convergence rate of Q-learning, which we also empirically measure in our simulations.

7.2 RL-IRSA: an adaptive medium access control protocol

The discussion will proceed with a description of the independent Q-learning algorithm for optimizing the degree distribution of IRSA. In particular, we model the network as an Dec-POMDP and, then, describe how Q-learning can be employed to find the optimal transmission strategy.

7.2.1 The learning algorithm

In our setting, two parameters comprise the environment's state: the channel traffic G and a device's condition C_t. We first assume that the network is a single agent that interacts with its environment, which includes the channel and itself. This modelling was depicted in Figure 4.1a. We model the problem as an MDP with state:

$$S = \times_{1 \leq i \leq M} S^i \times S^u \tag{7.1}$$

where $S \in \mathcal{S}$ is the state of the agent, $\mathcal{S}$ is the set of all states, S^i represents the state of device i and S^u stands for the part of the environment that is uncontrolled by the devices and, in our formulation, corresponds to G. Finally, M is the number of devices.

The transition probabilities of this MDP can be cast as:

$$P(S^u_{t+1} \mid S^u_t, \times_{1 \leq i \leq M} S^i_t, K) = P(S^u_{t+1} \mid S^u_t) \tag{7.2}$$

$$P(S^i_{t+1} \mid S^u_t, \times_{1 \leq i \leq M} S^i_t, K_t, F^i_t) \propto -T^i_t(K, G) + F^i_t \tag{7.3}$$

where $T_t^i(\cdot)$ is the individual throughput of device i, i.e., its number of successfully transmitted packets, that depends on the current values K_t and G_t. Specifically, the throughput of a device can be computed as $T_t^i = -b_t^i + b_{t-1}^i + f_t^i$ and the throughput of the network as a whole as $T_t = \sum_i^{G*N} T_t^i$, where G is the channel load and N is the frame size. Note that we did not provide a strict definition for the transition probabilitity of the state, but defined it to be proportional to the number of messages that will be added to the buffer before the next frame. This is because the state transition can vary for different applications (e.g. devices dropping packets to avoid congestion or packets being stochastically corrupted by noise at the receiver). In our simulations, we assume that all packets are successfully added to the buffer, unless there is an overflow, in which case they are dropped.

From (7.2) we observe that the transitions of the uncontrollable state S^u are independent of the transmission strategy and the states of individual devices. In particular, we assume that the channel probabilistically switches states based on the arrival and departure of devices in the network. Further, from (7.3), we observe that individual transitions in the state of a device depend on the states and actions of other devices, channel traffic, noise conditions and packet throughput.

The action $A \in \mathcal{A}$ of the agent, with $\mathcal{A}$ being the action space, consists of the joint actions of all devices in the network. These actions represent the values of the coefficients Λ_l of the degree distribution, as it was described in Section 3.3. Thus, the action space is:

$$A = \mathbf{A}^i \times \cdots \times \mathbf{A}^M, \text{ with } \mathbf{A}^i = \{\Lambda_1^i, \cdots, \Lambda_{\Lambda_{\max}}^i\} \tag{7.4}$$

where Λ_l^i denotes the coefficient Λ_l of the degree distribution of device i. Recall that $\Lambda_{\max}$ is the maximum number of replicas a device is allowed to send.

The above MDP formulation, although genuinely modeling the IRSA optimization problem, leads to a continuous action space, that scales exponentially with the number of devices. This renders learning of the optimal action intractable for large-sized problems. To circumvent this drawback, we redefine the actions as the number of replicas to send:

$$A = A^i \times \cdots \times A^M, \text{ with } A^i = l \text{ and } l \in \{1, \cdots, \Lambda_{\max}\} \tag{7.5}$$

During the learning phase the agent finds a deterministic policy $\pi(a|s)$, with $s \in \mathcal{S}$ and $a \in \mathcal{A}$, by choosing the optimal A^i for each device, except for exploratory moves where a random action is performed. After learning has completed, the probability distribution $\Lambda(x)$ is computed using the information of visited state-action pairs. Therefore, upon execution of the protocol the policy is probabilistic with $\pi(a|s) = \Lambda_a$, where Λ_a is a coefficient in $\Lambda(x)$. This technique allows us to leverage the benefits of maintaining a small action space, while using a stochastic policy.

The choice of the reward function is guided by our aim to design self-interested agents, attempting to improve their transmission rate while lacking access to a global performance measure, i.e., the channel throughput. We, thus, define the immediate reward R_t as the negation of the number of packets in the buffer of the device in the current time index, ,i.e., $R_t = -b_t$. In addition to helping with the avoidance of overflows, this reward makes devices with more packets more eager to transmit, instead of making the decisions purely based on the outcome of the current transmission.

In our setting, the network and devices cannot observe $S^u = G$, as this requires a global view of the environment. We, therefore, constrain observability to information only locally available to devices. We assume that the only state-related information a device has access to is the number of messages stored in its buffer, that is:

$$\Omega = \Omega^1 \times \cdots \times \Omega^M, \quad \text{with} \quad \Omega^i = b^i \tag{7.6}$$

where b is the buffer state, originally defined in our description of the buffer model in Section 3.4.

A Dec-POMDP, which we originally presented in Definition 3 of Chapter 4, can be used to extend the single-agent POMDP model by considering joint actions and observations. In our case $A^i \in \{1, 2, \cdots, \Lambda_{\max}\}$, $\Omega^i \in \{0, 1, ..., B\}$ and R^i is the individual reward agent i observes, as described above. Thus, our algorithm does not need a common reward function, as agents individually measure their rewards based on their observations. Note that the state space of the Dec-POMDP coincides with the state space of the POMDP, defined in (7.1). In order to initialize the states, without loss of generality, we uniformly sample values in the range $[0, 1, \cdots, B]$. As we prove in Section 7.3.1, the existence of an ϵ-optimal solution is independent of this initialization.

The aim is to find the optimal policy, i.e. the policy that maximizes the expected reward for all states, defined as:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} \big(Q^*(s, a)\big), \quad \text{with } s \in \mathcal{S} \tag{7.7}$$

where $Q^*(s, a)$ denotes the optimal Q-function. Q-learning in a Dec-POMDP can be described by the following update mechanism:

$$Q(H_t, A_t) = (1 - \alpha)Q(H_t, A_t) + \alpha[R_t + \gamma \max_a Q(H_{t+1}, a)] \tag{7.8}$$

which we have explained in detail in Chapter 4.

7.2.2 Virtual experience

Q-learning proves to be inefficient for real-time applications, as extensive interaction with the environment is required in order to converge to the optimal solution. To improve the convergence rate of standard Q-learning, we employ VE [37, 112], where an agent updates multiple state-action pairs at each Q-learning iteration by "imagining" state visits.

The intuition behind VE is that an agent can update not only the state-action pairs that it has visited, but also pairs that are equivalent in terms of the unknown environment dynamics. In our case, the unknown environment dynamics include the arrival and collision model, take place after the selection of the number of packet replicas, and determine the reward the agent experiences, as well as the next observation $\omega \in \Omega$. As defined in (4.5), an agent's history of observations is a tuple of past buffer states. Based on this information, an agent chooses the preferred number of replicas to send. Although agent's i observation vector h_t^i is essential for determining the optimal action, we should point out that the unknown dynamics do not

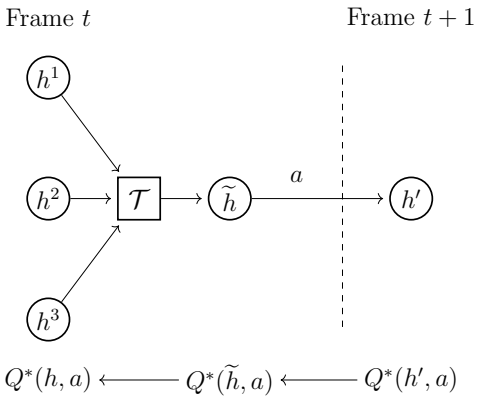


Figure 7.1: Illustration of virtual experience: h^1, h^2 and h^3 are states of the Dec-POMDP that are mapped to the same virtual state $\widetilde{h}$ through the transformation $\mathscr{T}$. After action a is performed and the next h' is observed, the Q-table is updated for all states.

directly depend on h_t^i. In particular, if the observation tuple is $h_t^i = \{b_{t-w+1}^i, \cdots, b_{t-1}^i, b_t^i\}$, then the unknown dynamics view states of the following form as equivalent:

$$\begin{aligned} H_t^{i'} &= \{b_{t-w+1}^{i'}, \cdots, b_{t-1}^{i'}, b_t^{i'}\} \\ &= \{b_{t-w+1}^{i'}, \cdots, b_{t-2}^{i'} - \delta b_{t-1}^i, b_{t-1}^{i'} - \delta b_t^i\}, \\ \text{with} \qquad \delta b_t^i &= b_{t-1}^i - b_t^i \end{aligned} \tag{7.9}$$

Note that, we have substituted observations with buffer states from the original definition of the observation tuple in (4.5), as these are equivalent in our current formulation.

The reason for the above formulation is that collisions should intuitively depend on the number of transmitted packets, as they determine the channel congestion. The value of the buffer state is useful in shaping the eagerness of agents to transmit packets. Formally and according to [111], a pair $(\tilde{s}, \tilde{a})$ is equivalent to a pair (s, a) if $p(s'|s,a) = p(s'|\tilde{s},\tilde{a}),\ \forall s' \in S$ and the reward $R(\tilde{s},\tilde{a})$ can be derived from $R(s,a)$. VE can be viewed as applying the following transformation on visited states, and then updating all states that have the same representation:

$$H_t = \{b_{t-w+1}, \cdots, b_{t-2} - \delta b_{t-1}, b_{t-1} - \delta b_t\} \tag{7.10}$$

$$\xrightarrow{\mathcal{T}} \widetilde{H}_t = \{\delta b_{t-w+2}, \cdots, \delta b_t\} \tag{7.11}$$

We term $\widetilde{h}$ a virtual state, as it is neither visited, nor directly used in the Q-learning update, but serves as an intermediate state in order to acknowledge states equivalent towards the unknown environment dynamics. We illustrate this concept in Figure 7.1.

Following the above observation for each move of an agent a batch update on all pairs (s^j, a^j) with $\mathcal{T}(s^j) = \widetilde{h}$ and $a^j = a$ will be performed. Note that we cannot extrapolate experience to states with different actions, as the collision dynamics depend on the performed action. Algorithm 1 contains the pseudocode of RL-IRSA. We present the algorithm from the perspective of a single device, as all devices learn and transmit in parallel. We note that the computation of the reward in Line 6 implies that the base station performs SIC to

remove collisions and returns the ACK signal to all players.

Algorithm 1: RL-IRSA

for $\tau \in \{1, \ldots, T\}$ **do**
  Observe buffer state b_t
  Update current history $\vec{h}_t$ with b_t
  Choose number of replicas $a_t = \arg\max_a Q(\vec{h}_t)$
  Transmit a_t replicas in randomly chosen slots
  Observe reward $r_t = -b_t$
  Update $Q(h_t, a_t)$ using (7.8)
end
Derive degree distribution $\Lambda(x)$ as $\Lambda_i = Q(s,i)/(\sum_{i\in\{1,\ldots,\Lambda_{\max}\}} Q(s,i))$

7.3 RL-IRSA: theoretical analysis

In this section, we theoretically analyze three important properties of the proposed learning algorithm: (i) the quality of the solution (ii) the rate of convergence and (iii) the computational complexity.

7.3.1 Optimality analysis

Our analysis begins by studying the waterfall effect and its relation to the optimal performance of IRSA. First, we associate this performance with the cost of a POMDP employed to optimize the degree distribution of IRSA. We then leverage our observations to justify that, a known result about the near-optimality of POMDPs with finite history approximations [110], is applicable in our setting. As this result is for a single agent, and thus corresponds to a centralized approach, we investigate the impact that replacing a centralized POMDP agent with independent learners has on the Q-learning solution. To this aim, we use observations derived in [109] and our own analysis of the equilibrium points of the problem under study.

In [29], the theoretical analysis of SIC in asymptotic settings reveals that the performance of IRSA is governed by the following waterfall effect: there is a sub-space in the space of all valid degree distributions, called a stability region, where SIC can successfully resolve all collisions with a probability close to 1. This leads to near-optimal channel throughput in the stability region. Outside of this region, this probability is bounded away from 0. The degree distribution can be used to calculate an upper bound on the channel traffic G^*, which signifies the limits of the stability region. The aforementioned observations can be consolidated in the following result:

Observation 1. ([29]) In asymptotic settings ($N \to \infty$), the probability of IRSA having optimal throughput is close to 1 if $G^* \leq \frac{1}{\lambda_2 \Lambda}$. Otherwise, this probability is bounded away from 0.

Furthermore, we know that the optimal solution of IRSA in our setting corresponds to the case where all devices successfully transmit one packet in each frame, i.e. $T^* = G$. If we, thus, define the cost as $J^\pi(\beta) = G - T^\pi(\beta)$, where π denotes the learned policy, which

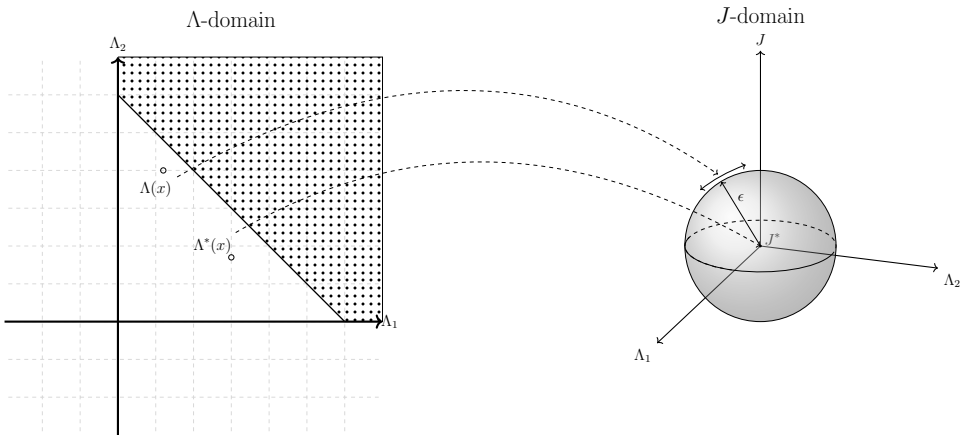


Figure 7.2: Visualization of the quality of the solution achieved by a POMDP for a wireless network where d, the maximum number of replicas, equals 2. On the left, the optimal solution and the one our algorithm converges, both of which lie in the stability region of IRSA, are depicted. On the right, we indicate the optimal cost of the POMDP as a point at 0, while the cost of our solution is at most ϵ, and therefore lies in a sphere with a radius equal to ϵ.

corresponds to the optimized degree distribution and, β is the underlying belief state of the finite-history POMDP, we can conclude that the optimal cost for IRSA is constant and does not depend on properties of the Dec-POMDP. We thus derive the following observation:

Observation 2. The optimal cost, $J^*(\beta)$, that can be achieved for IRSA is constant and equal to 0.

In [110], the optimality of POMDPs with finite histories of observations was studied and the following theorem was derived:

Theorem 3. (Proposition 2.2 in [110]) If the optimal cost, $J^*(\cdot)$, is a constant function, then, for any $\epsilon > 0$, there exists an integer w, which corresponds to the history window of the POMDP, and an ϵ-optimal policy $\pi^* \in \Pi$ such that π^* at each state depends only on the history of the most recent w stages.

We exploit Observations 1 and 2 and Theorem 3 to derive the main result of our optimality analysis:

Theorem 4. When the degree distribution of IRSA is optimized using a POMDP with a finite history window w, then, for any ϵ, there exists an ϵ-optimal solution. Furthermore, this solution lies in the stability region of IRSA and, therefore, throughput is optimal with a probability close to 1.

Proof. The first part of Theorem 4 is a direct result of Theorem 3, which in its turn is valid for the optimization of IRSA due to Observation 2. We will prove the second part of the theorem using a simple argument ad absurdum. Assume that there exists no solution with a probability of optimal performance close to 1. This would mean that all solutions found by the POMDP are bounded away from the optimal solution. But, according to the first part of the theorem, there exists an ϵ-optimal solution for any $\epsilon > 0$, where ϵ is a finite constant that

can be arbitrarily close to 0. Thus, we conclude that there exist solutions with a probability of optimal performance close to 1, with closeness to the optimal solution indicated by the parameter ϵ. □

Figure 7.2 offers a visual representation of how the solution of the IRSA optimization problem can be mapped into the space of POMDP policies for a simplified scenario where the degree distribution consists of two coefficients. As indicated, the sub-space of valid degree distributions that corresponds to the stability region of IRSA is mapped to a sphere of finite radius ϵ.

Note that, in contrast to finite approximations for POMDPs that require an explicit transition probability model to generate beliefs [113], Theorem 3 can be used in model-free approaches. Therefore, Q-learning can be the method of finding the near-optimal policies, the existence of which is guaranteed by Theorem 4.

Our analysis has so far assumed that the optimization is performed by a single agent, which corresponds to a centralized solution. However, to avoid the complexity associated with centralized learning approaches, we allow agents to act independently from each other and learn the Q-values only for their own individual actions. As was observed in [109], the existence of multiple agents renders the environment non-stationary. This suggests that Q-learning is not guaranteed to converge to the optimal solution, as stationarity of the learning environment is one of the traditional requirements for convergence [65]. In order, therefore, for our solution to be optimal, it is first essential to guarantee that all devices have converged in their attempt to find an optimal policy, as this ensures that the environment becomes stationary.

In [109], sufficient conditions for independent agents to converge were derived and it was observed that a decreasing learning rate and exploitive, such as Boltzmann, exploration, play an important part. However, guaranteed convergence is not adequate to prove the optimality of independent learning, as the reached equilibrium may correspond to sub-optimal solutions [109].

If we remember our definition of rewards, we can draw some useful conclusions related to the equilibrium points of our problem. While devices are learning, and before the optimal transmission strategy has been reached, the rewards are constantly decreasing. This is because, as we assumed in Section 3.4, packets arrive at a steady rate, equal to the optimal rate of transmission. After having learned the optimal policy and acting on it for a few iterations, a device reaches an equilibrium point where its rewards are constantly 0, as any packet is successfully transmitted upon arrival. If learning fails to find the optimal policy, then the devices quickly converge to a "bad" equilibrium point, where their buffers over-flow and the rewards converge to the negation of the buffer capacity. Due to this structure of the problem, we know that, if devices converge to the first equilibrium point, then the found solution is optimal. Otherwise, we discard the sub-optimal solution and restart the learning episode, as we explain in our discussion regarding Fig 7.6 in Section 7.4.

7.3.2 Rate of convergence analysis

Although the convergence of Q-learning is well-understood [65], employing virtual experience, as described in Section 7.2.2, alters the number of visited states per learning iteration. Under

the VE framework the conditions under which Q-learning is guaranteed to converge [65], are still applicable, but the convergence rate of Q-learning changes. Inspired by the work in [114], where the convergence rate of classical Q-learning is analyzed, we study how VE affects convergence time and derive a lower bound for it. We limit our analysis to asynchronous learning using an exponentially decreasing learning rate $\alpha = 1/t^{\phi}$, where ϕ is a parameter that determines how fast the learning rate converges to zero, as this is the learning scheme implemented in our simulations, and extend the results in [114] by considering multiple updates in each learning iteration.

We first investigate how VE affects coverage time L, i.e., the learning iterations necessary to visit all state action pairs at least once, and then proceed with bounding convergence time. Our remarks are based on Lemma 33 in [114].

Lemma 5. Assume that P is the probability of visiting all state-action pairs in a time interval k, which corresponds to a time period of L iterations. Then, using virtual experience, the probability of visiting all state-action pairs P in an interval k is $|\widetilde{\mathcal{H}}|P$, where $|\widetilde{\mathcal{H}}|$ denotes the size of the virtual state space.

Proof. The probability P is calculated as the percentage of unique state-action pairs visited, i.e., $P = L_u/(|\mathcal{S}||\mathcal{A}|)$, where L_u is the number of iterations where a pair was visited for the first time and the denominator represents the size of the state-action space. We assume that states are sampled with replacement from an i.i.d. probability distribution. As we note in Section 7.3.3, virtual experience increases the number of states updated in a learning iteration by $|\widetilde{\mathcal{H}}|$. It follows then that $L_u^v = |\widetilde{\mathcal{H}}|L_u$, where L_u^v is the number of iterations where the visited pair was unique using virtual experience. Thus, $P^v = |\widetilde{\mathcal{H}}|P$. □

Lemma 6. Assume that from any initial state we visit all state-action pairs with probability $|\widetilde{\mathcal{H}}|P$ in L steps. Then, from any initial state, we visit all state-action pairs in $L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$ steps for a learning period of length $\left\lceil \frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)} \right\rceil$ with probability $1-\delta$.

Proof. The probability of not visiting all state-action pairs in k consecutive intervals is $(1-|\widetilde{H}|P)^k$. If we define k as $\log_{1-|\widetilde{\mathcal{H}}|P}(\delta)$, then this probability equals δ and $L\log_{1-|\widetilde{\mathcal{H}}|P}(\delta) = L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$ steps will be necessary to visit all state-action pairs. □

Based on Lemma 6 we can derive the following property of VE:

Corollary 6.1. Virtual experience alters coverage time L by a factor of $\frac{\log_2(1-P)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}$.

Similar to the work in [114], we thus express how the convergence time of virtual experience depends on the covering time in the following theorem:

Theorem 7. Let Q_t be the Q-value computed by the asynchronous Q-learning algorithm using exponentially decreasing learning rate at time t. Then, with probability at least $1-\delta$, we have $\|Q_t - Q^*\| \leq \epsilon$, given that

$$t = \Omega\Big(\big(L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}\big)^{3+1/\phi} + \big(L\frac{\log_2(\delta)}{\log_2(1-|\widetilde{\mathcal{H}}|P)}\big)^{1/(1-\phi)}\Big)$$

Proof. This theorem is a direct result of Corollary 6.1 of our work and Theorem 4 in [114], where the corresponding bound for classical Q-learning is found to be $\Omega(L^{3+1/\phi} + L^{1/(1-\phi)})$. □

7.3.3 Computational complexity

The proposed protocol is a computationally attractive alternative to transmission strategies that are based on finite length analysis [64]. In our framework, at each learning iteration an agent has to choose its transmission strategy and then update its local Q-table. In contrast to the work in [93], in the proposed scheme the action space is discrete and increases linearly with the maximum number of allowed replicas. The size of the observation space, which coincides with the size of the Q-table, is $(B+1)^w$. Recall that B is the capacity of devices' buffer and w is the history window. The observation space scales exponentially with w and linearly with B. Finally, the complexity associated with the number of devices is $O(1)$, as devices learn independently of each other.

Equipping Q-learning with virtual experience increases computational complexity, as instead of updating one entry of the Q-table in each learning iteration, all (h, a) pairs with the same virtual state $\widetilde{h}$ are updated. This complexity increase is equal to the number of those pairs, which we denote by $|\widetilde{H}|$ and can be bounded as:

$$0 \leq |\widetilde{\mathcal{H}}| \leq \min\{B + 1 - B_{\min}, B_{\max}\} \qquad \text{where} \tag{7.12}$$

$$B_{\min} = \arg\min_{b}\{b - \sum_{t=w-1}^{\tau} \delta b_t \geq 0\} \qquad \text{and} \tag{7.13}$$

$$B_{\max} = \arg\max_{b}\{b - \sum_{t=w-1}^{\tau} \delta b_t \leq B\} \qquad \forall \tau \in [0, w-2) \tag{7.14}$$

where $B_{\min}$ and $B_{\max}$ are used to avoid considering virtual states with numbers of packets in their buffers that are either negative or exceed the maximum capacity B.

7.4 Simulations

This section begins with a performance comparison of the proposed RL-IRSA protocol and classical IRSA, which, as we explained in Section 3.3 does not employ learning and is optimized in [29] using differential evolution. It subsequently studies the effect of different learning schemes on the performance of independent learning with the two-fold goal of drawing conclusions about the behavior of agents and providing a guideline for configuring system parameters to determine the optimal transmission strategy. Finally, we evaluate the proposed scheme advanced with virtual experience to show the reduced convergence time. Unless stated otherwise, the simulation parameters are as indicated in Table 7.1. Note that a simulation includes learning of a degree distribution for L_E number of learning iterations and its evaluation for L_T transmission trials.

Table 7.1: Simulation setup

Simulation parameters	value
N, frame size	10
G, channel traffic	$[0.1, 0.2, \cdots, 1.0]$
$B_{\max}$, buffer size	3
w, history window	4
L_T number of transmission trials	1000
L_E, number of learning iterations	1500
N_E, number of independent simulations	20
c_{level}, confidence level	97.5%
ϵ, exploration	0.05
α, learning rate	$\frac{1}{(1+i)^0.9}$
γ, discount factor	0.98

Table 7.2: Degree distributions of classical IRSA and our proposed solution for different frame sizes

Method	$\Lambda(x)$
IRSA	$0.25x^2 + 0.60x^3 + 0.15x^8$
RL-IRSA$_{10}$	$0.4354x^2 + 0.2445x^3 + 0.173x^4 + 0.0855x^5 + 0.0437x^6 + 0.001x^7 + 0.008x^8$
RL-IRSA$_{50}$	$0.0556x^1 + 0.0278x^2 + 0.2732x^3 + 0.1654x^4 + 0.1027x^5 + 0.1178x^6 + 0.0878x^7 + 0.0902x^8$
RL-IRSA$_{200}$	$0.0402x^1 + 0.0754x^2 + 0.3036x^3 + 0.1607x^4 + 0.1131x^5 + 0.1548x^6 + 0.0952x^7 + 0.0476x^8$

7.4.1 Protocol comparison

In Figure 7.3, a statistical analysis on the performance of the two protocols under consideration is performed. From this figure, it is obvious that RL-IRSA is superior to classical IRSA for the whole channel traffic range, with the difference gap becoming wider for channel traffic above 0.6. We also observe that performance has higher variations for high channel traffic. Figure 7.4 illustrates convergence time for independent learning in different channel traffic. From this figure, we can see that convergence is guaranteed and is fast for low channel traffic. For $G = 0.2$ only four learning iterations are necessary, while for $G = 0.4$ seven iterations are needed. In the case of high channel traffic RL-IRSA fails to transmit messages faster than their arrival rate, the devices' buffers thus quickly overflow and the values of the rewards saturate to $-B$. For maximum channel traffic ($G = 1$), the saturation occurs very soon at the course of the episode (7 iterations), while for $G = 0.8$, 17 learning iterations take place before all buffers overflow and, even after this time point, some devices manage to empty their buffers. The highest channel traffic for which convergence is achieved is $G = 0.6$, which is expected if we consider the fact that packets are arriving with probability 1 at each iteration. Thus, a throughput above 0.5 is required on average to avoid saturation. Based on these observations, we design a mechanism for agents to avoid sub-optimal solutions: if the rewards deteriorate for three consecutive iterations, we classify the episode as "*bad*", discard the learned policy, and reset the POMDP to an arbitrary state.

To draw further insights into how our solution differs from an asymptotic optimization, we present in Table 7.2 the IRSA degree distributions, as optimized in [29] and our learned degree distributions for different frame sizes. In contrast to the work in [29], where the majority of

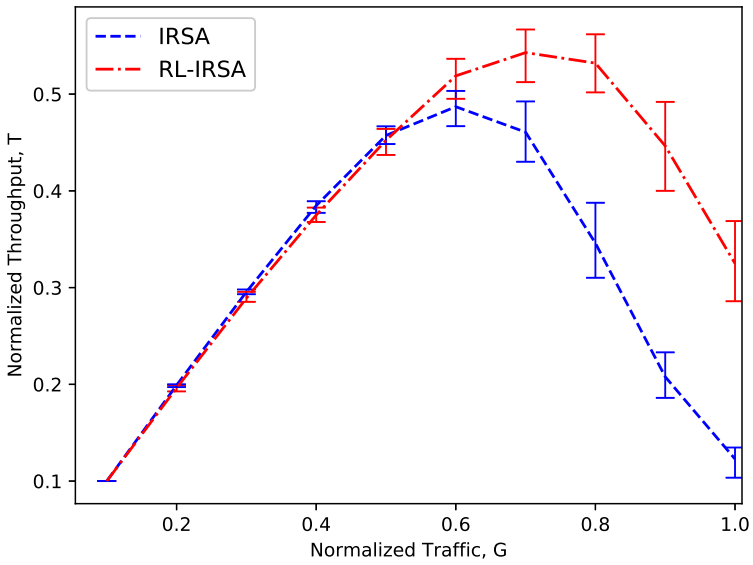


Figure 7.3: Achieved throughput comparison of IRSA and RL-IRSA on a toy network for various values of the channel traffic G.

the coefficients was a priori set to 0 to reduce the complexity of optimization, our learned $\Lambda(x)$ are more dense. We also observe that, as the frame size N increases, higher degrees become more prevalent, while degrees are left-concentrated for small frame sizes. To explain this tendency, we can borrow theoretical and empirical insights from the design of random error-correction channel codes [59]. This is due to the similarity between SIC and decoding on graphs, first observed in [29] and discussed in Chapter 2. In particular, the analysis in [59] proves that higher degrees lead to better probability of successful decoding, while observing that the simulated performance of the codes in finite settings deviates from the asymptotic analysis, with the gap becoming more evident as the maximum degree increases. Note that the maximum simulated frame size in IRSA does not usually exceed 200, while simulations in [59] were done for codewords of length orders of magnitude larger. It is thus natural to restrict the maximum degree in our simulations to lower values than the ones employed in [59].

Figure 7.5 illustrates how the throughput achieved by RL-IRSA and classical IRSA changes with frame size and $G \in \{0.6, 0.8, 1\}$. Note that these results can be easily translated into throughput variation with respect to the number of devices, as, for a constant channel traffic, the number of devices increases with the frame size according to the formula $M = G \cdot N$. Thus, the results in Figure 7.5 correspond to $M \in \{6, 12, 30, 60, 120\}$ devices ($G = 0.6$), $M \in \{8, 16, 40, 80, 160\}$ devices ($G = 0.8$) and $M \in \{10, 20, 50, 100, 200\}$ devices ($G = 1$). As regards scalability of RL-IRSA, its performance increases with larger frame sizes. This can be attributed to the fact that learning is more meaningful in more complex networks, where collisions occur more often and learning to avoid other agents has a more profound impact on the overall throughput. Classical IRSA also improves its performance for increased frame sizes, as it provingly works better in asymptotic settings. This is attributed

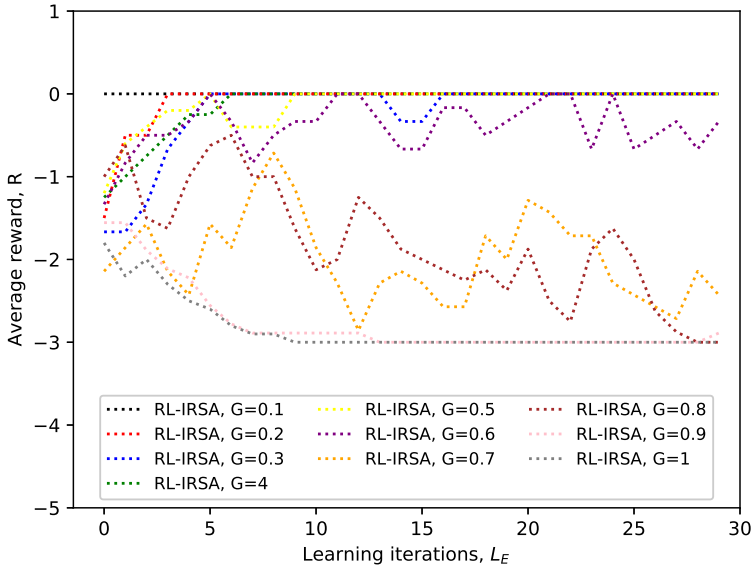


Figure 7.4: Average rewards of RL-IRSA and IRSA for different values of the channel traffic G.

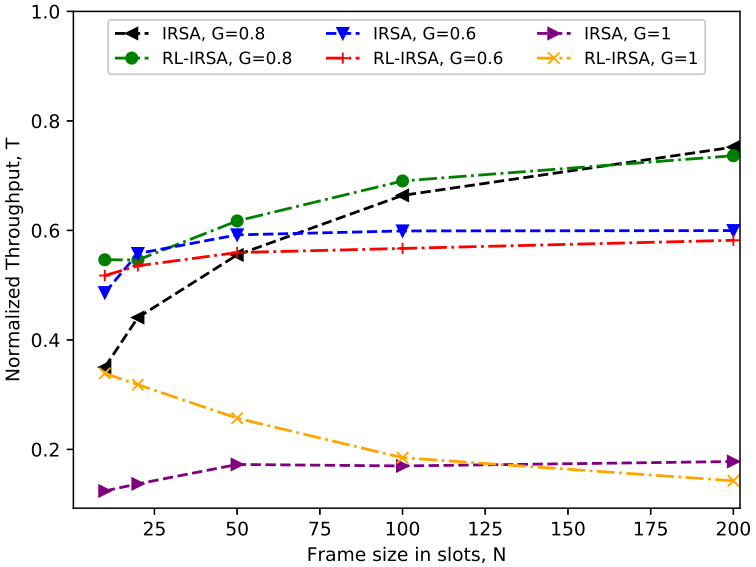


Figure 7.5: Achieved throughput comparison of IRSA and RL-IRSA for varying frame sizes N and number of devices M with channel traffic $G \in \{0.6, 0.8, 1\}$.

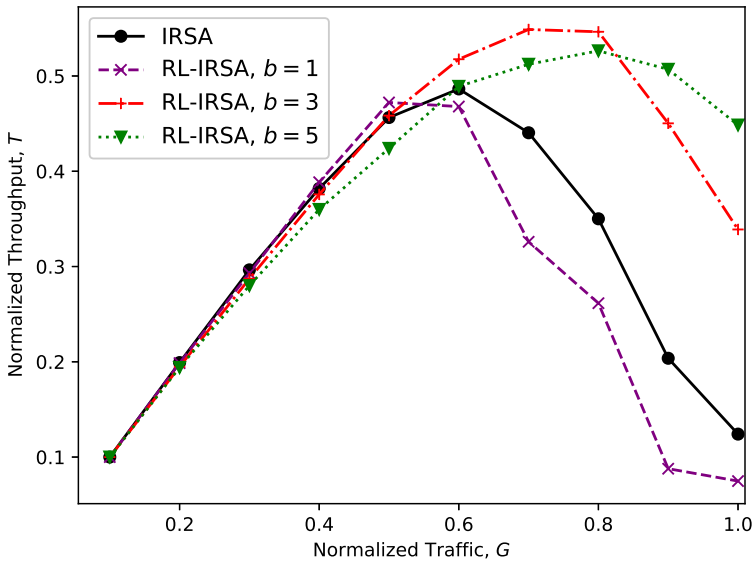
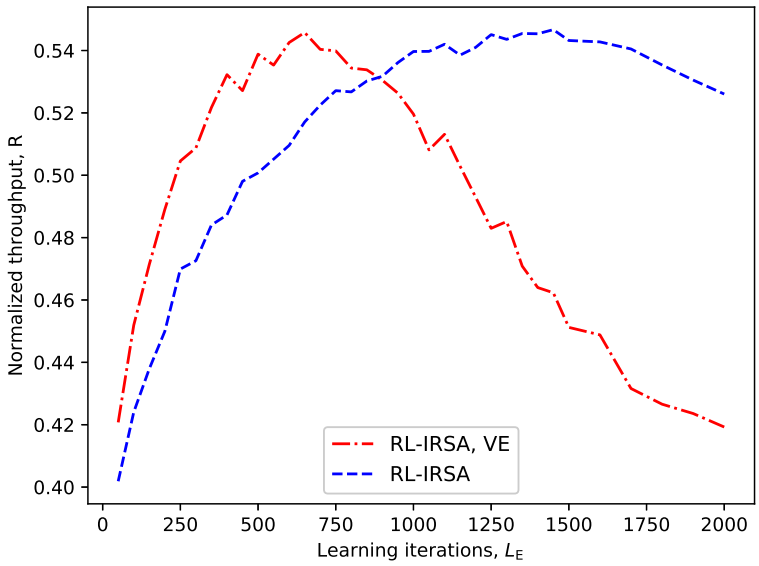


Figure 7.6: Comparison of achieved throughput for different buffer sizes of devices.

to the fact that the probability distribution $\Lambda(x)$ is computed using asymptotic analysis and is therefore closer to optimal for frames whose size exceeds 200 slots. Nevertheless, the performance gap between these two methods remains high in heavy channel traffic ($G = 1$), due to the waterfall effect of classical IRSA. To conclude scalability analysis, the slight superiority of standard IRSA manifested for low G in asymptotic settings is irrelevant to practical scenarios, as the assumption of very large frame size N leads to inefficient implementations, as was discussed in Section 9.1.

7.4.2 Effect of state space size

The size of the state space depends on the history window, as well as the maximum value of observations, which equals $B + 1$. Increasing B has a two-fold effect. Firstly, it increases the size of the state space, thus urging for longer exploration. Secondly, it dilates the range of rewards, which makes agents more eager to transmit. Assuming buffer sizes of constant size, constrained by devices' characteristics, one anticipates to improve performance of learning by increasing the history window, as that will lead to better approximation of the underlying beliefs. Nevertheless, letting memory constraints aside, this will result in an exponential increase of the state-action space, leading either to intractable problems or high time complexity. Thus, it is crucial to determine the minimum amount of information necessary for agents to derive efficient policies. Note that, for the sake of a fair comparison, learning iterations were also increased to 3000 for increased history window and buffer size. Figure 7.6 demonstrates that using a value of $B = 1$, i.e., only one packet is kept in the buffer, leads to lower throughput for channel traffic above 0.6, as agents are not made eager enough to transmit. On the other hand, increased buffer size improves the perceived throughput for traffic above 0.8, but it slightly degrades it for the rest.



]

Figure 7.7: Comparison of throughput of RL-IRSA with and without VE for different number of learning iterations and channel traffic $G = 0.7$.

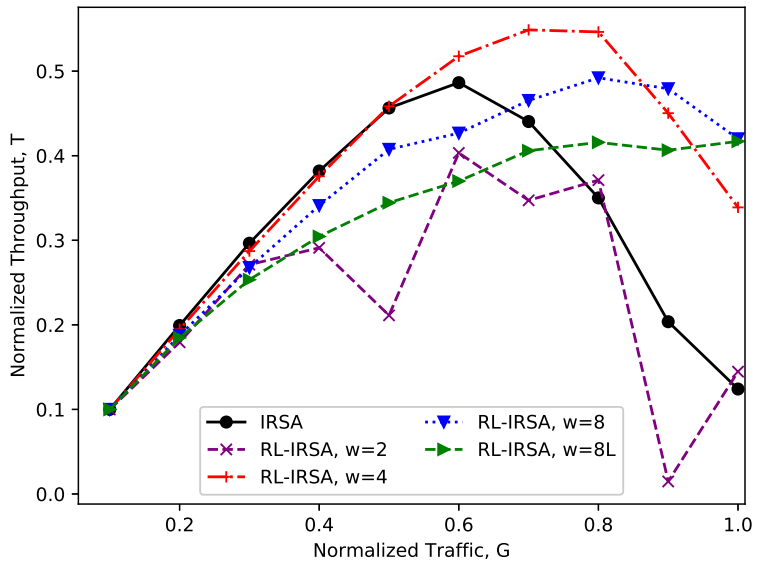


Figure 7.8: Comparison of achieved throughput for different values of the history window w.

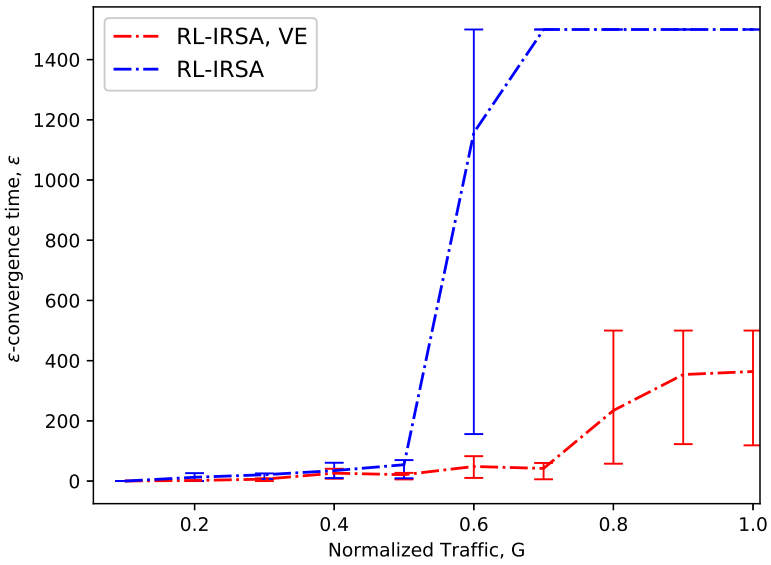


Figure 7.9: Statistical comparison of ϵ-convergence times for simple RL-IRSA and RL-IRSA using virtual experience, with $\epsilon = 0.5$.

Regarding the history window w, Figure 7.8 reveals that the effect of increased world size is more profound. This is due to size scaling exponentially with w, in contrast to its linear scaling with B. We observe that by decreasing the window to $w = 2$, a severe degradation in performance is observed, suggesting that the information provided to the agents through the observation tuples is not substantial. Increasing the learning iterations for $w = 8$ has a counterintuitive effect, as performance is degraded, whereas we would expect that an increased world size would benefit from longer training times. This expectation is based on the fact that increasing the dimensionality of the state space will require more state-action pairs to be visited. Thus, a higher number of learning iterations to find a well performing solution will be required, a problem often termed as curse of dimensionality. In this case however, 800 learning iterations perform optimally for $w = 4$, so we can assume that equipping agents with larger memory leads to learning better policies, thus less iterations are required. If we continue to train after this optimal point, performance degrades due to over-training, a phenomenon that generally refers to learning a behaviour that performs well during training, but not during the evaluation of the policy. In particular, after a good policy has been learned, the distribution of visited states is no longer representative of the original problem. We conclude that, considering the current parameterization, $w = 4$ is the best performing choice. Note that the optimal value for w is dictated by the learning dynamics (α, γ and ϵ) and is the one that defines a state-space of size appropriate for a satisfactory exploration/exploitation balance.

7.4.3 Virtual experience

Virtual experience was introduced in our solution to reduce convergence time, which we, similarly to the work in [37], measure here using the ϵ-convergence time, i.e., the number of

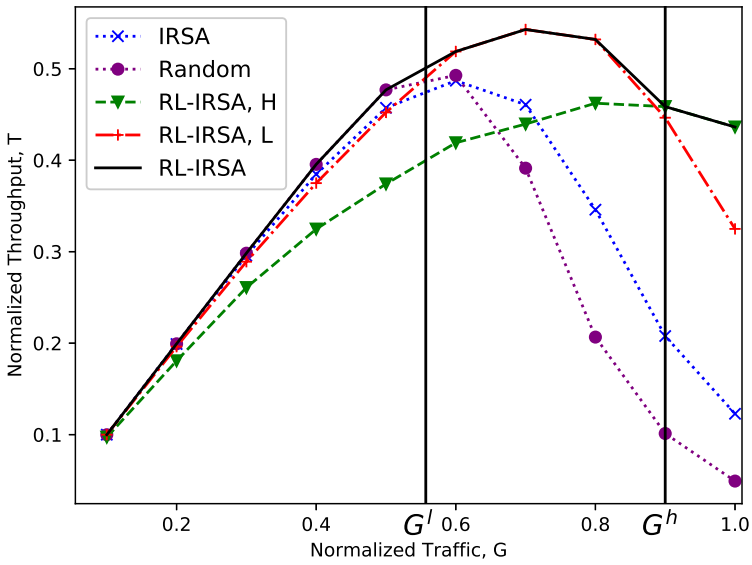


Figure 7.10: Performance comparison of classical IRSA, a random strategy, RL-IRSA optimized for low G and RL-IRSA optimized for high values of G.

iterations required to learn an ϵ-optimal policy. Figure 7.7 shows how throughput changes with learning iterations and reveals that, using VE, the optimal number of iterations was reduced from 1500 to 500. Note that the degradation in performance with increasing learning iterations, manifested at around 1500 iterations for RL-IRSA and 500 using VE, is due to over-training. Figure 10 is also useful for understanding the complexity of our learning solution. Although finding the optimal policy requires around 700 iterations (using VE), satisfactory performance is achieved very early during the course of the episode. From Figure 10, it is clear that even at 100 iterations, the algorithm is not very far from the optimum, while performance improves rapidly. Therefore, we can conclude that this solution will give good results even if the problem changes often, i.e. every 100-200 iterations. To gain some intuition on how simulation time maps to real time, we note that an iteration in our simulations required 3 milliseconds using an Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz processor. As the code was not optimized in terms of time complexity, we expect even smaller time requirements from software solutions specialized for a specific hardware architecture.

Figure 7.9 performs a statistical analysis on ϵ-convergence time for different values of the channel traffic using a 95% confidence level on 40 independent simulations and $\epsilon = 0.5$. We observe that convergence is fast for low traffic regardless of the use of VE. For $G \geq 0.6$, however, we observe that VE exhibits an improvement of around 80%, which can be attributed to increasing the number of batch updates by a factor of $|\widetilde{\mathcal{H}}|$. Also, RL-IRSA without VE usually fails to converge for high channel traffic (i.e. convergence has not been achieved within the budget of 15000 iterations), although throughput remains close to optimal. This observation suggests that, in this case, there are different policies that lead to optimal behavior, so RL-IRSA without VE is less biased to the optimal one. From Figure 7.9 we can also observe that the confidence interval observed for RL-IRSA without

VE for channel traffic $G = 0.6$ is large. This traffic value is a transitional case, as for some simulations Q-learning converged, similarly to $G < 0.6$, while for others convergence was not achieved by the end of the episode, similarly to $G > 0.6$. This is because, as we can see in Figure 5, the learned transmission strategies up to channel traffic $G = 0.5$ are optimal, i.e. $T = G$, while $T < G$ for $G >= 0.6$, which suggests that a unique optimal policy has not been found.

7.4.4 Waterfall effect

As explained in Section 7.3.1, the performance of IRSA has been proven to be governed by a stability condition [29], which leads to a waterfall effect in the performance of SIC. From a learning perspective, this profoundly changes the nature of the problem and, thus, the learning objective. In the realm of low channel traffic ($G < G^*$), where resources are abundant, agents must learn to coordinate their actions, as there is a number of replicas to transmit that optimizes packet throughput. Observe that, for low channel traffic ($G \leq 0.5$), even a random strategy, implemented by sampling the number of replicas l uniformly from $\{1, \cdots, d\}$, is appropriate, as illustrated in Figure 7.10, so learning is of no practical interest. In the realm of high channel traffic ($G \geq G^*$), however, we can acknowledge the task as a Dispersion game [115], where agents need to cooperate in order to avoid congesting the channel by exploiting it in different time frames. Different problem nature urges for different learning behavior, thus we expect that parameterization of learning should vary with G. Figure 7.10 illustrates the performance of three different parameterizations, each one optimal for a different range of values for G. Note that G^* and G^{low} stand for the threshold traffic value below which the probability of unsuccessful transmission is negligible and a random strategy is optimal, respectively. We observe that by optimizing the parameters for a particular range of G values, we obtain significant gains in the region of interest ($G > 0.6$).

7.5 Conclusions - Towards optimality guarantees

We have examined the problem of IRSA design for finite frame sizes from a reinforcement learning perspective and proved that learning degree distributions can be beneficial even under the assumption of devices' independence in learning. The theoretical analysis proves that our proposed algorithm, RL-IRSA, finds a near-optimal solution. Simulations suggest that the waterfall effect of the problem, common in optimization problems where agents compete for common resources, leads to different learning dynamics, and thus, demands adaptive solutions. Our method's superiority is particularly manifested in high channel traffic and small frame sizes, where the degree distributions of classical IRSA exhibit low throughput. Our simulations indicate that making learning tractable for online application scenarios requires achieving fast convergence. We observed that even when maintaining a small observation space, by restricting the history window to 2, the performance remains satisfactory. Finally, the results show that we significantly reduced convergence time by introducing virtual experience into Q-learning.

A question that arises from our simulations is: *"is it possible to improve throughput for higher traffic even more?"*. As we observed, employing learning helped across all values of

channel traffic, but the high competition for resources in congested networks still leads to failing to resolving collisions with some probability. Our intuition is that avoiding collisions in congested networks in order to achieve optimal throughput can become easier if devices employ some form of communication with each other. As all devices aim at maximizing their probability of successfully transmitting, then acting independently when resources are not enough may make them inadvertently harm the transmission of other devices. In the next chapter, we raise the assumption of independence and allow devices to coordinate their actions prior to transmission, taking into consideration that this additional communication will not incur high complexity.

Chapter 8

Optimizing IRSA using coordinated Q-learning

In Chapter 7, we observed that there is a limit to what independent Q-learners can achieve on the task of optimizing the IRSA degree distribution. This observation fits in our broader discussion regarding the necessity of communication among devices competing for a pool of common resources, presented in Chapter 1. We now venture into the design of a learning-based solution that addresses the question: *"is it possible to achieve a finer balance between complexity and optimality when designing RL solutions for resource-constrained networks?"*

To tackle this question, we employ coordinated Q-learning to optimize the IRSA degree distribution. Our solution maps the MAC resource allocation problem first to a bipartite graph, and then, based on the dependencies between devices, transforms it into a coordination graph, on which the max-sum algorithm is employed to find the optimal transmission actions for devices. Thus, coordination takes place only when local dependencies arise between agents and the complexity of a centralized approach is avoided. We have theoretically analyzed our algorithm and determined the convergence guarantees for decentralized coordinated learning in the considered networks. As part of this analysis, we derive a novel sufficient condition for the convergence of max-sum on graphs with cycles and employ it to render the learning process robust. In addition, we reduce the complexity of applying max-sum to our optimization problem by expressing coordination as a Multiple Knapsack Problem (MKP). Our simulations reveal the benefits coming from adaptivity and coordination, both inherent in the proposed learning-based MAC.

8.1 Motivation for coordinated Q-learning

Similarly to Chapter 7, the objective of this chapter is to design the degree distribution of IRSA. Actions in our POMDP formulation correspond to the number of replicas sent by each device. We follow the observation in [29] that, in IRSA, transmissions can be represented by a bipartite graph and employ this graph to derive the CG, which is updated at the beginning of each frame. This approach differs significantly from works where the connectivity of the network is not provided but designed to improve inference [116].

As we explained in Chapter 3, SIC often successfully collisions occurring under trans-

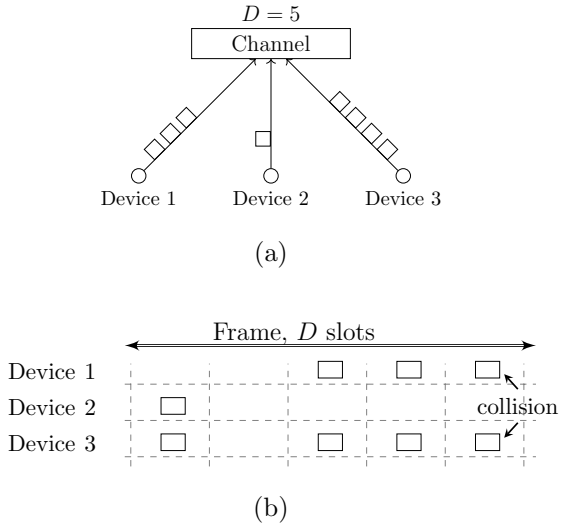


Figure 8.1: Transmission under IRSA: (a) a sensor network consisting of three sensors that wirelessly transmit replicas of their packets to a common channel, and (b) transmissions of replicas in a frame.

mission using IRSA. Nevertheless, it can fail for small frame sizes, particularly if collisions are too many. This happens because the iterative algorithm employed gets stuck in cycles [29]. In IRSA, this can be avoided if devices indirectly avoid each other, by transmitting a number of replicas that will result in the smallest probability of collision, as slots are assigned to packets uniformly at random. Therefore, our work equips devices with the ability to coordinate their transmission policies in order to avoid simultaneously sending too many (unresolved collisions) or too few (underutilized channel) replicas. For example, in Figure 8.1a, devices 1 and 3 choose to transmit 3 and 4 replicas respectively, which are both large numbers considering a frame of 5 slots, presented in Figure 8.1b. In this frame, the transmission will fail for all devices, even though sensor 2 sent only one replica. This could lead all sensors to send a small number of replicas in the next frame and, thus, successfully transmit their packets.

In this work, we adopt a coordinated learning approach, where CGs help us leverage the particular structure of the coordination problem. Devices learn by using a Q-learning based algorithm, where the global Q-function is decomposed into a summation of Q-functions, each one corresponding to a different group in the CG. [117, 118, 119]. In order to derive the optimal actions, max-sum, an algorithm for approximate distributed probabilistic inference, is applied to the CG. We provide more details regarding the max-sum algorithm and our implementation of coordinated Q-learning in Sections 8.2 and 8.3 respectively. To the best of our knowledge, this is the first attempt to optimize resource management in communication networks using coordinated Q-learning.

As we are aiming at designing a solution for resource-constrained networks, we need to reduce further and bound the complexity of coordination. To achieve this, in Section 8.4, we formulate the optimization objective of max-sum as a multiple knapsack problem (MKP) [120], where devices are removed from groups to satisfy the complexity constraints. This is done by eliminating edges on the CG, while constraints are controlled by the capacities of the knapsacks. We should note that this formulation is generic and can be used to achieve

the desired balance between the sparsity of a bipartite graph and the quality of approximate inference. By appropriately redefining weights, the MKP can account for different types of constraints, such as the battery lifetime of devices.

Furthermore, in Section 8.5 we prove that our Q-learning based algorithm converges to the optimal solution under the GDD-POMDP framework. Thereafter, we derive a sufficient condition for the convergence of max-sum, inspired by the work in [121], which contains an analysis of the convergence of sum-product, an algorithm that, along with max-sum, belongs in the family of belief propagation algorithms. This condition permits us to a priori evaluate whether max-sum will converge on a specific CG. Due to this mechanism, coordinated learning is rendered robust, as devices can choose to coordinate only when convergence is guaranteed, while independent learning can be performed otherwise. In this way, devices can reduce their energy consumption without degrading the overall throughput. We believe that our analysis is an important step towards addressing the convergence uncertainty inherent in non-stationary learning environments.

8.2 Coordination graphs and the max-sum algorithm

Our problem can be seen, on a higher-level, as a group of agents that attempt to maximize the overall throughput by coordinating their packet replica transmissions. Specifically, each agent (device) i chooses an individual action a_i from a set $\mathcal{A}_i$, and the resulting joint action $a = \langle a_1, \cdots, a_C \rangle$ generates a payoff $f(a)$ for the network. In our problem, action a_i corresponds to the number of packet replicas device i sends and the payoff $f(a)$ to the overall throughput T. The aim of the coordination problem is to find the optimal joint action a^* that maximizes $f(a)$. The optimized degree distribution $\Lambda(x)$ can then be estimated from the history of actions. An obvious approach to determine the optimal action is to consider all possible joint actions and select the one that maximizes $f(a)$. However, this approach quickly becomes impractical, since the joint action space grows exponentially with the number of devices.

Fortunately, coordination problems in wireless networks exhibit the property that the payoff matrix $f(a)$ is sparse. This suggests that each agent is affected only by the decisions made by a small subset of the agents, as only devices that collide have to coordinate their actions. In this paper, we consider the use of a CG to account for such dependencies. This allows us to decompose the global payoff function $f(a)$ into a linear combination of group payoff functions, each involving only a smaller number of agents. For example, a payoff function involving the three devices of Figure 8.1 can be decomposed as follows:

$$f(a) = f_{13}(a_1, a_3) + f_{23}(a_2, a_3) \tag{8.1}$$

We can map function $f(a)$ to a CG $(\mathcal{N}, \mathcal{L})$, as the one depicted in Figure 8.2a. Each node in $\mathcal{N}$ represents an agent, while an edge in $\mathcal{L}$ indicates a coordination dependency (a collision that occurred in the previous time frame). Only connected agents have to coordinate their actions at any particular time instance. The global optimization problem is, thus, recast as a number of local coordination problems, each involving a subset of the total number of agents, that can be solved distributively. Thus, agents can find their optimal values independently of agents that do not participate in their local coordination problem. Sparsity of the CG is

directly related to the complexity of coordination, as the computational complexity of max-sum scales exponentially with the number of variables on which the group payoff functions depend. Edges in a CG represent dependencies and, thus, increase the arity of group payoff functions. Note, however, that the number of exchanged messages varies linearly with the number of agents. As such, the increase in complexity is not due to communication overhead, but should be attributed to the exponential growth of the search space [122].

Variable elimination is an approach to finding the optimal joint action a^* [123]. This algorithm eliminates the agents from the graph one by one, and always finds the optimal joint action. However, its execution time is non-deterministic, due to its dependence on the order of elimination [124], as well as impractical for large wireless networks, as it increases exponentially with the induced width of the CG [124]. To avoid the complexity required to determine the optimal solution, in this work, we adopt the use of the max-sum algorithm, which performs approximate inference on the CG.

The coordination problem in Figure 8.2a can be alternatively represented using a bipartite graph, like the one presented in Figure 8.2b, where interactions (collisions) between agents (devices) are now explicitly drawn. In the bipartite representation, nodes in the lower row are termed as variable nodes (VNs), and correspond to devices, while the upper row, consisting of the check nodes (CNs), represents the shared network resources, which in our case are time slots. Bipartite graphs offer much richer problem representations than simple coordination graphs, as they can represent k-ary ($k \geq 1$) relationships. Hereafter, we denote VNs with lower case letters, CNs with uppercase, the set of CNs as $\mathcal{F}$ and the set of VNs as $\mathcal{V}$. Also, the neighborhood of a variable node i is denoted as $\mathcal{N}_i = \{I \in \mathcal{F},\ i \in \mathcal{N}_I\}$ and the neighborhood of a check node I as $\mathcal{N}_I = \{i \in \mathcal{V},\ I \in \mathcal{N}_i\}$. If we express the dependencies in terms of utilities, i.e. define a quantity $u_I(a_i)$, where $I \in \mathcal{F}$, that expresses the utility of agent i when interacting with other agents, then we get the equivalent, interaction-based bipartite graph, shown in Figure 8.2c.

The max-sum algorithm can be applied on the bipartite graph of Figure 8.2c to solve the inference problem of finding the value assignment of a set of variables that maximizes a factored probability distribution. Consider $|\mathcal{V}|$ discrete random variables x_i for $i \in \mathcal{V} := \{1, 2, \cdots, |\mathcal{V}|\}$, with x_i taking values in $\mathcal{X}_i$ and $|\mathcal{X}_i|$ the size of the space of variable's i values. Note that, in our problem formulation, the random variable x_i corresponds to the possible number of actions, i.e. $x_i \triangleq a_i$ and $\mathcal{X}_i = \{1, \cdots, d\}$. We are interested in estimating the optimal action:

$$a^* = \arg\max_a f(a) = \arg\max_a \prod_{I \in \mathcal{F}} u_{\mathcal{N}_I}(a_i) \tag{8.2}$$

$$\equiv \arg\max_a \sum_{I \in \mathcal{F}} \ln u_{\mathcal{N}_I}(a_i) \tag{8.3}$$

where $u_{\mathcal{N}_I}$ is the utility function of CN I. Due to instabilities arising from the multiplication of potentially small quantities, the problem is formulated as a summation of logarithms. During the application of the max-sum algorithm, variable and check nodes exchange messages

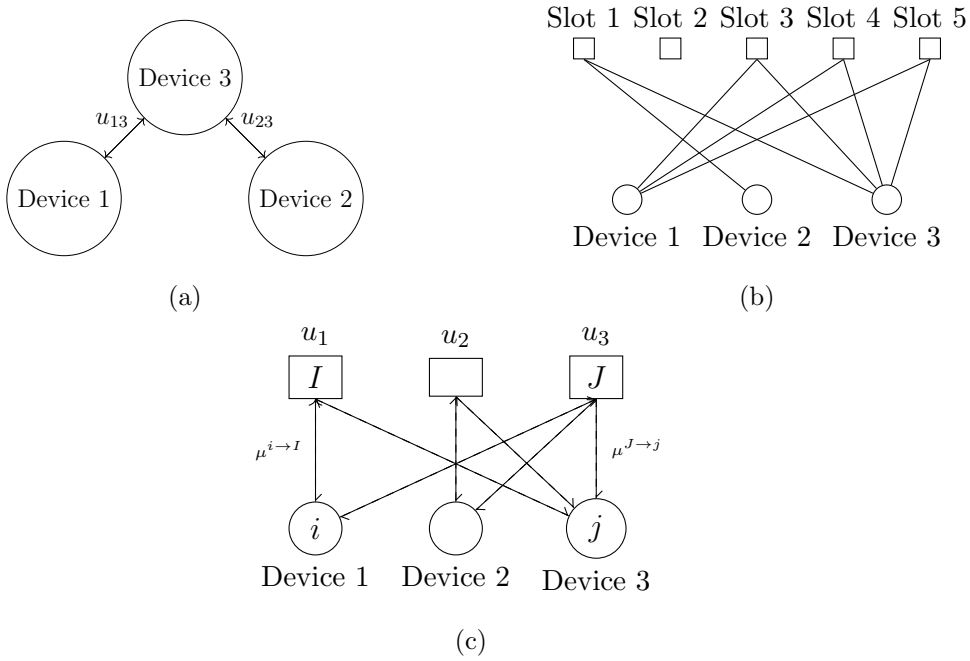


Figure 8.2: Representing coordination in the device network: (a) the simple coordination graph, (b) the interaction-based bipartite graph (each check-node represents a slot), and (c) the utility-based bipartite graph (each check node computes the utility (u_I) of a variable node v_i that belongs to group I).

of the following form:

$$\begin{aligned} \text{From VN } j \text{ to CN } I\text{:} \quad & \tilde{\mu}_{j\to I}(a_j) = \sum_{J\in\mathcal{N}_j\setminus I} \mu_{J\to j}(a_j) \\ \text{From CN } I \text{ to VN } i\text{:} \quad & \tilde{\mu}_{I\to i}(a_i) = \max_{a_I\setminus i} \big[\ln(u_{\mathcal{N}_I}(a_I)) \\ & \qquad + \sum_{j\in\mathcal{I}\setminus i} \mu_{j\to\mathcal{I}}(a_j)\big] \end{aligned} \tag{8.4}$$

where $\mu(\cdot)$ corresponds to the current message and $\tilde{\mu}(\cdot)$ to the updated message, which will be used in the next iteration.

It is known that, for tree-structured graphs, max-sum converges to the optimal solution within a finite number of iterations [117]. Although it also empirically exhibits good performance for graphs with cycles [117, 125], there are no guarantees for convergence in this case. In Section 8.5, we derive a sufficient condition for max-sum to converge for arbitrary graphs and employ it to devise an optimal learning algorithm, as failure to converge is generally associated with solutions of bad quality [121, 126].

8.3 Coordinated Q-learning based optimization of IRSA

We now present our algorithm for optimizing the degree distribution of IRSA using coordinated Q-learning. The algorithm employs various steps, which we summarize in Figure

8.3. In our formulation, each device is an agent that interacts with its environment (channel and device network) by performing actions (number of replicas to send), accepts rewards (similarly to Chapter 7, defined as the negation of number of packets in the transmission buffer) and makes observations (number of packets in its buffer). Partial observability in our framework arises due to the inability of devices to observe the underlying state of the whole network, denoted as s. Instead of employing the framework of Belief MDPs [66], in our setting, we use an approximation to beliefs based on a fixed history window of size w. Therefore, agents' state consists of a finite set of successive observations. The mathematical formulation of this finite-history POMDP for a device is:

$$\text{Observation:} \quad \omega^t = b^t \quad \text{Reward:} \quad r^t = -b^t$$
$$\text{Action:} \quad a^t = l^t \quad \text{State:} \quad \vec{h}^t = \langle \omega^{t-w+1}, \cdots, \omega^t \rangle$$

where b^t is the number of packets in the buffer at time t and l^t is the number of replicas to send. For the sake of simplicity, we have omitted the device index from the above variables.

This definition of rewards and observations only requires information that is local to the devices, in particular, the number of packets in their buffers. Note that rewards can implicitly provide information about the success of transmission, as packets are added to the devices' buffer due to a packet arrival or stay in the buffer because of a transmission failure. Simultaneously, rewards do not depend only on the current success of transmission, but incentivize devices to avoid an overflow of their buffers, as this would result in packet loss. Similarly, for the observations, a device can discriminate states based on how full its buffer is, which leads the device to adapt its strategy to low and high data traffic.

Our goal is to compute a joint policy $\pi^*(\vec{h}, a)$ that maximizes the total expected reward of all agents over a finite horizon τ, termed as the optimal policy. In Chapter 4, we described Q-learning [65] as an approach to determining $\pi^*(\vec{h}, a)$. We remind readers that Q-learning employs the following update mechanism to estimate the value of a history-action pair:

$$Q(\vec{h}^t, a^t) = (1-\alpha)Q(\vec{h}^t, a^t) + \alpha[r^t + \gamma \max_a Q(\vec{h}^{t+1}, a)] \tag{8.5}$$

where α is the learning rate, dictating how quickly new acquired information overrides past one, and γ is the discount factor, determining how much future information is discounted.

Next, we present the coordinated Q-learning employed by each device to learn the optimal policy by coordinating its actions using the max-sum algorithm. The process of finding the optimal actions, which takes place every time devices interact with their environment to update their Q-function, consists of the following steps:

Step 1. At the beginning of each frame a bipartite graph, of the form presented in Figure 8.2b, is built based on the collisions that occurred in the previous time frame.

Step 2. The graph is converted to a utility-based representation (Figure 2c). In this graph, each VN is a device and each CN involves its Q-function. This suggests that, in our setting, the utility functions, as described in Section 8.2, are mapped to the Q-functions, i.e. $u_{\mathcal{N}_I}(a_i) \triangleq Q_l(\vec{h}_l, a_l)$, where $l \in \mathcal{L}$ indicates the index of a group of devices, $\mathcal{L}$ refers to the set of groups, and I is the CN all devices in group l are connected to.

Step 3. Max-sum is applied on the graph to distributively calculate the optimal joint action of the global Q-function, defined as:

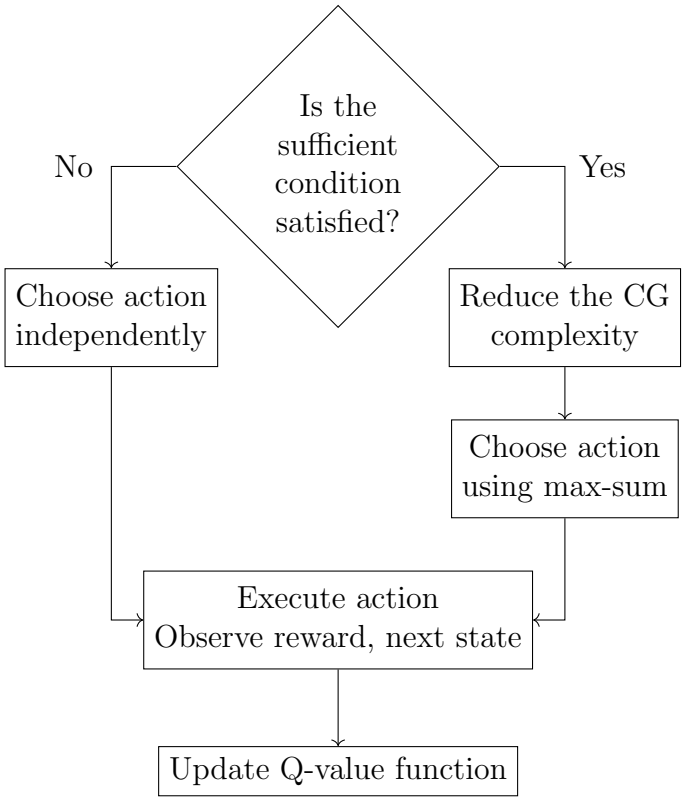


Figure 8.3: Our proposed algorithm for optimizing IRSA using coordinated Q-learning.

$$a^* = \arg\max_a Q(\vec{h}, a) \tag{8.6}$$

Note that, in the preceding equation, we dropped time index t, as max-sum is performed independently for each frame. Thus, the term $\vec{h}$ remains constant throughout the application of max-sum. The groupwise decomposition of the global Q-function is expressed as:

$$\hat{Q}(\vec{h}, a) = \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l, a_l) \tag{8.7}$$

Based on (8.7), the update rule presented in (8.5) can be expressed using $\hat{Q}(\vec{h}, a)$ as:

$$\sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t) = (1-\alpha) \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t) + \tag{8.8}$$

$$\alpha[\sum_{l \in \mathcal{L}} r_l^t + \gamma \max_a \hat{Q}(\vec{h}^{t+1}, a)] \tag{8.9}$$

Note that term $\max_a \hat{Q}(\vec{h}^{t+1}, a)$ cannot be further decomposed into a sum of local discounted future rewards, as this would require knowledge of the joint optimal action. Therefore, we define:

$$a^* = \arg\max_a \hat{Q}(\vec{h}^{t+1}, a) \tag{8.10}$$

$$\max_a \hat{Q}(\vec{h}^{t+1}, a) = \hat{Q}(\vec{h}^{t+1}, a^*) = \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^{t+1}, a_l^*) \tag{8.11}$$

where the last equality is due to (8.7).

The update mechanism for each group is thus:

$$Q_l(\vec{h}_l^t, a_l^t) = (1-\alpha)Q_l(\vec{h}_l^t, a_l^t) + \alpha[r_l^t + Q_l(\vec{h}_l^{t+1}, a_l^*)] \tag{8.12}$$

where we employ max-sum to determine a_l^*. We, then, perform the optimal actions a^* (or an exploratory action) and, based on the received rewards, update the group Q-functions Q_l.

The messages exchanged to choose the optimal actions when solving (8.6) are very small, indicating the probability that one of the D actions is optimal, as the max-sum algorithm dictates. Their exchange will require communication between devices which, in practice, can be implemented through the base station. As the base station is aware of the structure of the bipartite graph, a requirement imposed by the need to perform SIC, transmitting these messages does not introduce additional assumptions. More importantly, their transmission does not require that each device knows the structure of the bipartite graph.

8.4 Complexity reduction

8.4.1 Motivation

Despite exploiting locality of interaction, the application of max-sum can still be prohibitive for resource-constrained wireless networks, if the CGs are not sparse enough. In particular, as the channel load or frame size increases, collisions also increase in number, especially at the beginning of the learning process, when agents have not yet learned how to avoid transmitting in a way that will lead to unresolved collisions. We are, thus, still in need of techniques that will reduce the search space of max-sum without affecting the quality of its solution.

We start with the observation that, often, a small fraction of the original variables involved in a complex mathematical problem is required to determine the optimal solution [120]. In this paper, we use column generation [127], which exploits this observation by expressing the problem as an integer program and considering only a subset of its original variables. A common approach is to formulate the optimization objective as a multiple knapsack problem [120]. The intuition behind using column generation is that an agent can, under circumstances, ignore some of the agents it collided with, and still compute its optimal action correctly. We can, thus, reduce the complexity of the original problem by pruning some of the variables involved in Q_l.

Definition 2. The 0-1 MKP is: given a set of N items and M knapsacks ($M \leq N$), where each item has a value p_j and a weight w_j, and each knapsack has a capacity c_i, to select M disjoint subsets of items, that can be assigned to a knapsack whose capacity is no less than the total weights of items in it, so that the total profit of the selected items is maximized.

Formally:

$$\max \sum_{i=1}^{M} \sum_{j=1}^{N} p_j x_{ij} \tag{8.13}$$

$$\text{subject to} \quad \sum_{j=1}^{N} w_j x_{ij} \leq c_i, \quad i \in \mathcal{M} = \{1, \cdots, M\} \tag{8.14}$$

$$\sum_{i=1}^{M} x_{ij} \leq 1, \quad j \in \mathcal{N} = \{1, \cdots, N\} \tag{8.15}$$

$$x_{ij} \in \{0, 1\}, \quad i \in \mathcal{M},\ j \in \mathcal{N} \tag{8.16}$$

$$\text{where} \quad x_{ij} = \begin{cases} 1, \text{ if item } j \text{ is assigned to knapsack } i \\ 0, \text{ otherwise} \end{cases} \tag{8.17}$$

8.4.2 Multiple knapsack formulation of max-sum

In our setting, each Q_l function corresponds to a knapsack and each agent to an item. Our objective is to determine a_l for each group $l \in \mathcal{L}$, so that the sum of the knapsacks, which denotes the overall utility of the device network, is maximized. The MKP will determine which agents to include in each Q-table. Note that condition (8.15) must be modified in our case, as agents can be included simultaneously in different knapsacks. We, thus, replace (8.15) with $\sum_{i=1}^{M} x_{ij} \leq M,\ j \in \mathcal{N}$, which forces an agent to be in a maximum of M knapsacks.

We define the value of an agent j in (8.13) as:

$$p_j = \max_{a_j} Q_j(\vec{h}_j, a_j), \quad a_j \in \{1, \cdots, \Lambda_{\max}\} \tag{8.18}$$

where $\Lambda_{\max}$ is the maximum allowed number of replicas. This definition suggests that an agent is evaluated based on the maximum value of its local Q-table, which can be interpreted as the maximum contribution this agent expects to have to the maximization of the group Q-table. Note that this is not equal to $Q_j(\vec{h}_j, a_l^*[j])$, i.e., the component of the globally optimal solution that corresponds to agent's j action. This is because $a_l^*[j]$ is not the same with the action of j that maximizes its local Q-table, as the effect that this action will have on the different Q_l agent j participates in, is not considered. Thus, the solution provided by solving the MKP will not be globally optimal and max-sum should still be applied.

In order to define the weights w_j, we should measure how the participation of an agent in the Q-function of a group increases the computational complexity of applying max-sum. In particular, in this paper, we measure complexity as the time required until the convergence of max-sum. To determine this time we make use of the fact that max-sum exhibits a linear convergence rate, as we prove in Section 8.5. The convergence rate is governed by $|x^* - x^0|$, where x^* is the optimal value assignment and x^0 is the value max-sum is initialized with. We, therefore, know that, the further from the optimal solution max-sum starts, the more time it will need to converge. We can, thus, define the weight as $|x^* - x^0|$. Note that variables x^* and x^0 correspond to probability distributions over the agents' decision variables, as they are the

messages sent from check to variable nodes during the application of max-sum. Formally:

$$\begin{aligned} x^0 &= \mu_i^n(a_i) \\ &= \sum_{I \in N_i} \mu_{I \to i}(a_i), \quad a_i \in \{1, \cdots, d\}, \quad n = 0 \end{aligned} \tag{8.19}$$

$$\begin{aligned} x^* &= \mu_i^n(a_i) \\ &= \sum_{I \in N_i}^{n} \mu_{I \to i}(a_i), \quad a_i \in \{1, \cdots, d\}, \; n = N_{\max} \end{aligned} \tag{8.20}$$

where n is the index of the max-sum iteration, and $N_{\max}$ is the maximum allowed number of iterations. With a slight abuse of notation, we refer to the messages received by VN i as $\mu_i(a_i)$. Note that (8.20) is valid only when max-sum converges to the optimal solution. Although restricting the number of iterations in graphs with cycles can lead to sub-optimal solutions, if $N_{\max}$ is chosen to be appropriately high, it will not significantly affect the quality of the solution. This is due to the empirical observation that, if max-sum converges, this happens within the first few iterations [125].

We calculate the distance between x^* and x^0 as their Kullback - Leibler (KL) divergence, i.e:

$$w_j = D_{KL}(x^* || x^0) = \sum_{i=1}^{d} x^*(i) \log \frac{x^0(i)}{x^*(i)} \tag{8.21}$$

Our choice of the KL divergence was motivated by the observation in [128] that this measure naturally describes the lack of fit of an approximation of a distribution when preferences (beliefs) are expressed by a logarithmic function. The employed max-sum algorithm is an example of such a case, as beliefs are the logarithms of utilities. Furthermore, the additive property of the KL divergence is useful, both for representing the collective belief of agents, as well as for summing the weights of the items in a knapsack to determine the total weight assigned to it.

The value of x^* in (8.20) could be found by employing the max-sum algorithm, but the calculation of weights should not require this. We, therefore, approximate x^* based on the system's Gibbs free energy [129], an alternative approach to finding the solution of (8.2). Instead of employing message-passing, this solution computes the optimal probability distribution x^* by minimizing the KL divergence between x^* and the joint probability distribution, given by:

$$p(a) = \frac{1}{Z} \exp \sum_{I \in \mathcal{F}} \ln u_{\mathcal{N}_I}(a_{\mathcal{N}_I}) \tag{8.22}$$

where Z is a normalizing constant to ensure that $p(\cdot)$ sums to 1. Thus, x^* can be found by

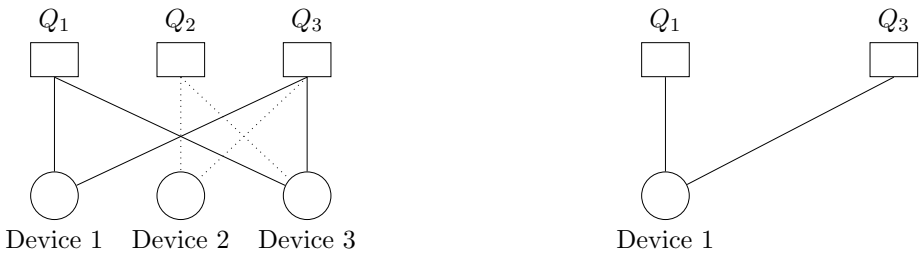


Figure 8.4: (a) Agents 1 and 3 have collided, so they participate in each other's Q-function. The MKP needs to decide whether device 1 will be included in Q_3. (b) We form the sub-graph for calculating the weight of device 1 by assuming that Q_1 is independent from the messages from device 3.

solving the following optimization problem:

$$\min \sum_{a_i} q(a_i) \log p(a_i) - \sum_{a_i} q(a_i) \log q(a_i) \tag{8.23}$$

$$\text{where} \quad q(a_i) = \sum_{J \in N_i} \mu_{J \to i}(a_i) \text{ and } p(a_i) \tag{8.24}$$

$$= -\frac{1}{Z} \exp \sum_{c \in C_i} \ln u_{\mathcal{N}_I}(a_{\mathcal{N}_I}) \tag{8.25}$$

Figure 8.4 illustrates how the weight of an agent is determined. In this example, when calculating agent's 1 effect on Q_3, we ignore agent's 3 effect on Q_1, otherwise self-referentiality would not allow us to solve the problem. As the weight of an agent j depends on the knapsack j it is evaluated for, we henceforth include both indexes when denoting a weight (w_{ij}), and define w_j as the average weight of agent j.

Finally, we define the capacity of a knapsack, i.e. the time available to a Q-table to converge, based on the problem's feasibility constraints. We know that, in general, $c_i \geq \min w_{ij},\ \forall j \in \mathcal{N}$ and $w_{ij} \leq \max c_i,\ \forall i \in \mathcal{M}$. If a knapsack violates the first constraint, it can be ignored because no items can be added to it. Similarly, if an agent violates the second constraint, then it does not fit anywhere, and can, therefore, be eliminated. By sampling c_i from the range $[w_{ji}, \sum_{j \in \mathcal{M}} w_{ij})$, we ensure that a Q-table can at least include the agent it is associated with and that, not all the agents can fit in it.

8.4.3 Solving the multiple knapsack problem

We solve (8.13) by obtaining a tight bound on the optimal solution using the Lagrangian relaxation method [130]. We do not employ Branch-and-Bound algorithms [120], although they offer an exact solution to the MKP problem. This is because their high computational complexity would defeat the original purpose of reducing the complexity of coordination. Besides, even if these algorithms were used, the final solution would still have to be found by the max-sum algorithm, as explained in Section 8.4.2.

The Lagrangian relaxation of (8.13) can be formulated as:

$$L(MKP, \lambda) = \max \sum_{i=1}^{M} \sum_{j=1}^{N} \tilde{p}_j x_{ij} + M \sum_{j=1}^{N} \lambda_j \tag{8.26}$$

$$\text{where} \quad \tilde{p}_j = p_j - \lambda_j, \quad j \in \mathcal{N},\ i \in \mathcal{M} \tag{8.27}$$

Similarly to [120], we find the optimal dual variables λ_j associated with the constraints in (8.15):

$$\lambda_j = p_j - w_j \frac{p_c}{w_c} \text{ , if } j < c, \text{ and 0, otherwise} \tag{8.28}$$

where c denotes the critical item. This corresponds to the first item that does not fit in the knapsack, if we consecutively insert items in decreasing order of value per weight unit, and is formally defined as:

$$c = \min \left\{ j : \sum_{i=1}^{j} w_{ij} |N_i| > \sum_{k=1}^{M} c_k \right\} \tag{8.29}$$

where $|N_i|$ denotes the number of neighbors of VN i. Note that our definition of the critical item differs from the classical one [120] due to the way that capacities are defined in our setting.

The relaxed problem can be subsequently decomposed into a series of independent single knapsack problems of the form:

$$\max \sum_{i=1}^{M} \sum_{j=1}^{N} \tilde{p}_j x_{ij} \tag{8.30}$$

$$\text{subject to} \quad \sum_{j=1}^{N} w_{ij} x_{ij} \leq c_i, \quad \text{and} \quad x_{ij} \in \{0, 1\},\ j \in \mathcal{N} \tag{8.31}$$

As explained in [120], the optimization problem described by (8.30) is the langrangian relaxation of a single knapsack and can be solved by setting

$$\tilde{x}_j = \begin{cases} 1, \text{ if } \tilde{p}_j > 0 \\ 0, \text{ if } \tilde{p}_j < 0 \end{cases} \tag{8.32}$$

where the value of $\tilde{x}_j$ is immaterial when $\tilde{p}_j = 0$. Thus, by defining $J(\lambda) = \{j : p_j / w_j > \lambda\}$, the optimal solution of the multiple knapsack is:

$$z(L(MKP, \lambda)) = \sum_{j \in J(\lambda)} \tilde{p}_j + M\lambda, \tag{8.33}$$

$$\text{where} \lambda = \sum_{j=1}^{N} \lambda_j. \tag{8.34}$$

8.5 Optimality analysis

This section begins by introducing GDD-POMDPs, which map the properties of learning performed by wireless networks into a mathematical framework for decision making. Subsequently, we derive guarantees for the convergence of coordinated Q-learning in this framework. The proof consists of two parts. First, we prove that the joint Q-function can be decomposed into a sum of group Q-value functions. Since max-sum is employed to choose the actions, we also have to ensure that the solution it converges to is optimal. Therefore, in the second part of the analysis, we derive a sufficient condition for the convergence of max-sum.

8.5.1 The GDD-POMDP framework

Based on our solution, as presented in Section 8.3, the actions of devices depend on each other only when their packets collide upon transmission. Dependence among agents regards both actions and states: two collided agents will have to coordinate their actions due to sharing the same utility function, and, they will also affect the state transition of each other. In our setting, transition and observation independence is not guaranteed for each agent (device). According to our formulation, it only concerns agents belonging to different groups, i.e., devices whose packets have not collided. We refer to this framework as GDD-POMDPs and ascribe to it the property of groupwise observability [117].

Definition 3. A GDD-POMDP is defined as a tuple $\langle \mathcal{M}, \mathcal{S}, \mathcal{A}, T, R, \Omega, O, w, \xi^0 \rangle$, where

$\mathcal{M} = \{1, \cdots, |\mathcal{M}|\}$ is the set of agent indices.

$\mathcal{S} = \times_{i \in \mathcal{M}} \mathcal{S}_i \times \mathcal{S}_u$. $\mathcal{S}_i$ refers to the local state of agent i. $\mathcal{S}_u$ refers to a set of uncontrollable states that are independent of the actions of the agents.

$\mathcal{A} = \times_{i \in \mathcal{M}} \mathcal{A}_i$, where $\mathcal{A}_i$ is the set of actions for each agent.

$\Omega = \times_{i \in \mathcal{M}} \Omega_i$ is the joint observation set.

$T(s'|s,a) = T_u(s'_u|s_u) \cdot \prod_{l \in \mathcal{L}} T_l(s'_l|s_l, s_u, a_l)$, is the transition probability function, where $a = \langle a_i, \cdots, a_M \rangle$ is the joint action performed in joint state $s = \langle s_i, \cdots, s_M \rangle$ and s_u is the current value of the uncontrollable state, the transitions of which are not affected by the actions of devices, but are controlled by external factors (e.g. arrival/departure of devices to/from a network). The transition probability distribution of the network is decomposable among groups of agents, indexed by l, with $l \in \mathcal{L}$, where $\mathcal{L}$ denotes the set of all groups. Note that we employ model-free learning and, thus, do not require a particular form for T_l. If $k = |l|$ agents with indices $\{i_1, \cdots, i_k\}$ are involved in a particular group l, then s_l denotes the state of group l, i.e. $s_l = \langle s_{l1}, \cdots, s_{lk} \rangle$ and, similarly, $a_l = \langle a_{l1}, \cdots, a_{lk} \rangle$. This decomposition models the transition independence between agents belonging to different groups.

$R = \sum_{i \in \mathcal{M}} R_i(s_i, s_u, a_i)$ is the immediate reward function. Thus, rewards are inherently local in this framework.

$O(\omega|s,a) = \prod_{l \in \mathcal{L}} O_l(\omega_l|s_l, s_u, a_l)$ is the observation probability function. This decomposition models the observation independence among groups.

w is the history window.

ξ^0 is the initial state distribution at time $t = 0$.

Definition 4. GDD-POMDPs are said to have groupwise observability if, $\forall l \in \mathcal{L}$, the set of observations $\omega_l = \langle \omega_{l1}, \cdots, \omega_{lk} \rangle$, made by agents belonging in group l, fully determine the current uncontrolled state s_u, i.e., if $\forall l,\ \forall \omega_l,\ \exists s_u$: $Pr(s_u|\omega_l) = 1$.

In our framework, this property implies that, given the joint observation of a group $l \in \mathcal{L}$, ω_l, the observation and the transition probability function of the group l do not depend on actions and observations of agents in other groups.

8.5.2 Q-function decomposition

We base our analysis on [117], where the Q-function was also proven to be decomposable into a sum of group Q-functions. One notable difference between our setting and the one in [117] is that our framework is not Networked Distributed POMDPs, as in our case independence holds only for agents that do not belong in the same group. Another difference is that, in [117], the reward has the same decomposition as the Q-function and agents get their rewards by evenly distributing the group reward, whereas in our case rewards are individual to each device.

Theorem 1. For GDD-POMDPs with groupwise observability, under basic assumption of Q-learning and by means of update rule (8.12), $Q_l(\vec{h}_l, a_l)$ converges to the optimal $Q_l^*(\vec{h}_l, a_l)$ for all $l \in \mathcal{L}$, and thus, policy $\pi^*(\vec{h}) = \arg\max_a \sum_{l \in \mathcal{L}} Q_l^*(\vec{h}_l, a_l)$ is globally optimal.

In order to prove the above theorem, we first establish that a Q-function defined over states is decomposable. Then, we prove that a Q-function based only on histories of observations is also decomposable.

The Bellman equation for the global Q-function is:

$$Q(s^t, a^t) = R(s^t, a^t) + \gamma \sum_{s^{t+1}, \omega^{t+1}} T_u^t T^t Q^{t*} \tag{8.35}$$

Equivalently, for the group Q-functions:

$$Q_l(s_l^t, a_l^t) = R(s_l^t, s_u^t, a_l^t) + \gamma \sum_{s_l^{t+1}, \omega_l^{t+1}} T_u^t T_l^t Q_l^{t*}, \quad \forall l \in \mathcal{L} \tag{8.36}$$

Recall that T_l^t cannot be further decomposed into a product of individual probability functions due to the absence of independence within a group.

In the case of Belief MDPs, we know that for the global Q-function:

$$Q(b^t, a^t) = \sum_{s \in S} b^t(s) Q(s^t, a^t) \tag{8.37}$$

If we replace continuous beliefs with finite histories of observations, then the above equations take the following form:

$$Q(\vec{h}^t, a^t) = \sum_{s \in S} b^t(s) Q(s^t, \vec{h}^t, a^t) \tag{8.38}$$

$$= \sum_{s_l \in \mathcal{S}_l, s_u \in \mathcal{S}_u} b^t(s_u, s_l) Q(s_l^t, \vec{h}_l^t, a_l^t) \tag{8.39}$$

We can thus treat histories as a substitute for states in the Q-learning framework.

Lemma 2. In GDD-POMDPs, the global Q-function $Q(s^t, a^t)$ for any finite horizon τ is decomposable, that is:

$$Q(s^t, a^t) = \sum_{l \in \mathcal{L}} Q_l(s_l^t, a_l^t) \tag{8.40}$$

Proof. We prove the lemma by mathematical induction. For $t = \tau - 1$ we have by definition $Q(s^t, a^t) = R(s^t, a^t) = \sum_{l \in \mathcal{L}} r_l(s_l^t, a_l^t)$ and there is no future reward, as τ corresponds to the last iteration. Assume that for $1 \leq t \leq \tau - 1$ the global Q-function is decomposable, i.e. $Q(s^t, a^t) = \sum_{l \in \mathcal{L}} Q_l(s_l^t,\ a_l^t)$. Then, we have

$$\begin{aligned}
Q(s^{t-1}, a^{t-1}) &= R(s^{t-1}, a^{t-1}) + \gamma \sum_{s^t,\ \omega^t} T_u^{t-1} T^{t-1} Q^* \\
&= \sum_{l \in \mathcal{L}} r_l^{t-1} + \gamma \sum_{s^t,\ \omega^t} T_u^{t-1} T^{t-1} \sum_{l \in \mathcal{L}} Q_l^* \\
&= \sum_{l \in \mathcal{L}} \left[r_l^{t-1} + \gamma \sum_{s^t, \omega^t} p_u^{t-1} P_l^{t-1} Q_l^* \right] \\
&= \sum_{l \in \mathcal{L}} Q_l^{t-1}
\end{aligned}$$

where the last equality is valid by the assumption of mathematical induction. □

Lemma 3. In GDD-POMDPs with groupwise observability, the global Q-function $Q(\vec{h}^t, a^t)$ for any finite horizon τ is decomposable, that is:

$$Q(\vec{h}^t, a^t) = \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t) \tag{8.41}$$

Proof.

$$\begin{aligned}
Q(\vec{h}^t, a^t) &= \sum_{s_u, s} b_u^t(s_u^t) b_s^t(s^t) \sum_{l \in \mathcal{L}} Q_l(s_l^t, a_l^t) \\
&= \sum_{l \in \mathcal{L}} \big[\sum_{s_u, s_l} b_l^t(s_u^t, s_l^t) Q_l(s_l^t, s_u^t, a_l^t) \big] \\
&= \sum_{l \in \mathcal{L}} Q_l(\vec{h}_l^t, a_l^t)
\end{aligned}$$

where the last equality arises from (8.40). Note that the decomposition of beliefs is valid due to Lemma 4.

Theorem 1 is a direct result of Lemmas 2 and 3. □

8.5.3 Convergence analysis of the max-sum algorithm

In this section, we derive a sufficient condition for the convergence of max-sum, based on the analysis in [121], where sufficient conditions for the sum-product algorithm were formulated. The intuition behind the proof is that the update mechanism for the messages can be expressed as a mapping in the vector space of messages. Thereafter, the conditions under

which this mapping is a contraction can be derived, so that convergence to a fixed point is guaranteed. Our analysis can be applied to arbitrary graphs and depends on both the structure of the graphs and the involved utility functions. The derived sufficient condition can be used during learning in the following way: after the CG is formed, and before max-sum is employed, we evaluate the condition. If it is false, we can decide to avoid coordination for the current iteration and employ independent learning, otherwise, we can proceed with coordination.

We start by formulating the max-sum update equation, initially presented in (8.4) as:

$$\tilde{\mu}_{I\to i}(a_i) = \max_{a_{\mathcal{N}_I\setminus i}} \left[\ln(u_{\mathcal{N}_I}(a)) + h_{\mathcal{N}_I\setminus i}(a) \right] \tag{8.42}$$

where we expressed everything in terms of messages from check to variable nodes and defined:

$$h_{\mathcal{N}_I\setminus i}(a) \triangleq \sum_{j\in\mathcal{N}_I\setminus i} \sum_{J\in\mathcal{N}_i\setminus I} \mu_{J\to j}(a_j) \tag{8.43}$$

To simplify the notation, we denote the utility function $u_{\mathcal{N}_I}(a_{\mathcal{N}_I})$ as $u_{\mathcal{N}_I}(a)$ and messages $h_{\mathcal{N}_I\setminus i}(a_{\mathcal{N}_I\setminus i})$ as $h_{\mathcal{N}_I\setminus i}(a)$. The following theorem is the main tool employed by our analysis:

Theorem 4. (Banach's fixed-point theorem) Let $f : \mathcal{X} \to \mathcal{X}$ be a contraction of a complete metric space $(\mathcal{X}, d)$, where d represents the distance metric. Then, f has a unique fixed point $x^\infty \in \mathcal{X}$ and $\forall x \in \mathcal{X}$, the sequence $x, f(x), f^2(x), \cdots$ obtained by iterating f converges to x^∞. The rate of convergence is at least linear to $d(f(x), x^\infty)$, since $d(f(x), x^\infty) \leq Kd(x, x^\infty)$ for all $x \in \mathcal{X}$, where K satisfies $0 \leq K < 1$ and $d(f(x), f(y)) \leq Kd(x, y),\ \forall x, y \in \mathcal{X}$.

As suggested by Lemmas 1 and 2 in [121], in order to prove that the message update equation is a contraction, we bound its derivative. Directly taking the derivative of (8.42) would result in a trivial bound, we thus re-parameterize messages in terms of a monotonically increasing function:

$$\nu_{I\to i}(a_i) = e^{\mu_{I\to i}(a_i)} \tag{8.44}$$

The derivative of (8.44) can be calculated as:

$$\frac{\partial\tilde{\nu}_{I\to i}(a_i)}{\partial\nu_{J\to j}(y_j)} = e^{\max_{a_{\mathcal{N}_I\setminus i}} \left(Q_{\mathcal{N}_I}(a)+h_{\mathcal{N}_I\setminus i}(a)\right)} \mathbf{1}_{\mathcal{N}_j\setminus I}(J)\mathbf{1}_{\mathcal{N}_I\setminus i}(j) \tag{8.45}$$

$$\leq e^{\max_{a_{\mathcal{N}_I\setminus i}} Q_{\mathcal{N}_I}(a)+\max_{a_{\mathcal{N}_I\setminus i}} h_{\mathcal{N}_I\setminus i}(a)} \mathbf{1}_{\mathcal{N}_j\setminus I}(J)\mathbf{1}_{\mathcal{N}_I\setminus i}(j) \tag{8.46}$$

$$= e^{\max_{a_{\mathcal{N}_I\setminus i}} Q_{\mathcal{N}_I}(a)} e^{\max_{a_{\mathcal{N}_I\setminus i}} h_{\mathcal{N}_I\setminus i}(a)} \mathbf{1}_{\mathcal{N}_j\setminus I}(J)\mathbf{1}_{\mathcal{N}_I\setminus i}(j) \tag{8.47}$$

We define:

$$A_{I\to i,J\to j} = e^{\max_{x_{\mathcal{N}_I\setminus i}} Q_{\mathcal{N}_I}(a)} \mathbf{1}_{N_j\setminus I}(J)\mathbf{1}_{\mathcal{N}_I\setminus i}(j) \tag{8.48}$$

$$B_{I\to i}(\nu) = e^{\max_{a_{\mathcal{N}_I\setminus i}} h_{\mathcal{N}_I\setminus i}(a)} \tag{8.49}$$

Note that we have absorbed all ν-dependence in the term $B_{I\to i}(\nu)$, while term $A_{I\to i,J\to j}$ captures the structure of the bipartite graph, as well as the effect of the utility functions. In order to bound (8.47), we employ the following theorem:

Theorem 5. (Theorem 2 in [121]) Let $f : \mathbb{R}^m \to \mathbb{R}^m$ be differentiable and suppose that $f'(x) = B(x)A$, where A has nonnegative entries and B is diagonal with bounded entries $|B_{ii}(x)| \leq 1$. If the spectral radius of matrix A is strictly less than 1, then for any $x \in R^m$, the sequence $x, f(x), f^2(x), \cdots$ obtained by iterating f converges to a fixed point x_∞, which does not depend on x.

If we assume that $h_{\mathcal{N}_I \setminus i}(x)$ is normalized in the range (0,1), we can bound $B_{I \to i}(\nu)$ as:

$$\sup |B_{I \to i}(\nu)| \leq e \tag{8.50}$$

This bound corresponds to a worst-case analysis, where messages from all nodes involved in (8.48) are vectors with all their elements, except for one, set to 0. In addition, the index of the non-zero element must be the same for all nodes. Although this situation may arise, we expect that the messages exchanged in reality are more uniform. We also anticipate that, the more nodes are involved in (8.48), the less probable it is that agents agree on the maximum index. In Section 8.6.3, we present a heuristic that significantly refines this bound.

If we multiply all elements of matrix $A_{I \to i, J \to j}$ by the bound on the right-hand side of (8.50) and form matrix $\bar{A}_{I \to i, J \to j}$ with the new elements, then, based on Theorems 4 and 5, we derive the main result of our convergence analysis:

Theorem 6. If the spectral radius of matrix $\bar{A}_{I \to i, J \to j}$ is strictly smaller than 1, then the max-sum algorithm converges to a unique fixed point irrespective of the initial messages. Furthermore, the rate of convergence is at least linear to $d(f(x), x^\infty)$.

8.6 Simulations

8.6.1 Simulation Setup

To evaluate the proposed solution, we first examine its performance on a toy network with frames of size $C = 10$ and channel load $G \in [0.1, \cdots, 1]$. We set the buffer size to $B = 3$, the maximum number of replicas $\Lambda_{\max}$ to 8 and the maximum number of max-sum iterations $N_{\max}$ to 10. In each frame, a device accepts a new packet with a probability of 0.5. Regarding the learning parameters, we define a constant exploration rate ϵ of 0.05, a learning rate α of the form $0.9\alpha_b^i$, where α_b is a constant dictating the rate of decay and i denotes the number of visits of the current history-action pair. Furthermore, we employ a constant value for γ. Parameters α_b and γ were tuned for different ranges of G, and we observed that a value of $\gamma = 0.4$ and $\alpha_b = 0.4$ is optimal for $G \leq 0.6$, while $\gamma = 0.98$ and $a_b = 0.98$ works best for high loads. We observed that the tuning of hyperparameters was sensitive to the selection of the load G and the history window h. Finally, we employ a fixed window w of 4 observations. Unless stated otherwise, performance is averaged over 1000 Monte Carlo trials. Confidence intervals are calculated based on 20 independent experiments with 97.5% confidence level. To prove the superiority of our solution to traditional MAC, we compare its performance with an IRSA protocol where $\Lambda(x) = 0.25x^2 + 0.60x^3 + 0.15x^8$, which proved superior to other commonly used distributions [29] as well as with our solution employing independent Q-learning, presented in Chapter 7.

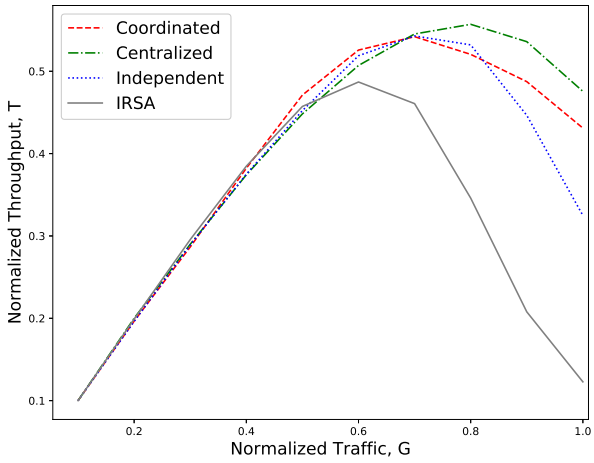


Figure 8.5: Comparison of independent, coordinated and centralized Q-learning on the task of optimizing the degree distribution of IRSA in terms of the throughput of the network for various levels of traffic, defined as the ratio of devices to time slots in a frame. We also present the performance of IRSA optimized using the technique proposed in [29], which is the state-of-the-art in asymptotic settings.

8.6.2 Throughput evaluation

The purpose of our evaluation is twofold. First, we aim to prove that learning-based protocols surpass in performance the current state-of-the-art RA protocol, IRSA. Second, we want to draw insights into how coordinated learning compares with independent and centralized approaches. In Figure 8.5, we observe that all learning-based methods improve upon IRSA in terms of the normalized throughput, and, thus, confirm the necessity of adaptive solutions. In addition, we observe that the throughput achieved by the proposed, coordinated approach, is higher than the independent learning case, and lower than the centralized solution for $G < 0.8$. This result was anticipated, as agents that coordinate their actions using the max-sum algorithm converge to neighbourhood optima, in contrast to independent agents that get stuck in local optima. In contrast, a centralized approach solves the problem in the original, joint space, Q-learning can thus converge to the global optimum. Note that, due to memory restrictions, we were not able to apply the centralized solution on networks with frame size larger than $D = 5$.

Figure 8.6 exhibits the benefits of coordination in terms of convergence rate and achieved throughput. We observe that coordinating agents converge early to higher throughput than independent agents, which converge slowly and experience oscillations. Although convergence of multi-agent reinforcement learning algorithms is a largely unexplored area, Figure 8.6 is in accordance with our expectations: when devices act independently, they learn in an uncertain environment and, thus, require more learning iterations. In contrast, agents that coordinate learn more steadily and converge to a well performing solution within a few iterations.

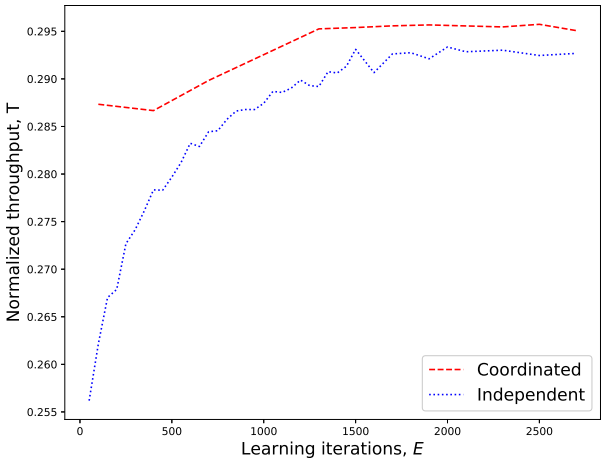


Figure 8.6: Convergence rate of coordinated and independent agents for channel traffic $G = 0.3$.

8.6.3 Robustness evaluation

We now evaluate the robustness of learning based on the sufficient condition presented in Section 8.5. In Figure 8.7, we present the probability of convergence of the max-sum algorithm, measured as the percentage of times that the condition indicated a possible failure to converge ($\|\bar{A}_{I\to i,J\to j}\| > 1$). We observe that convergence is less likely to be guaranteed for higher G. Additionally, this figure presents how convergence varies with the Q-table initialization: the values of all entries of the local Q-tables of devices are initialized by randomly sampling in the interval $[-B - c, -B + c]$, where B is the capacity of the buffer of agents and c is a constant that was chosen to have very low ($c = 0.01$) or very high ($c = 4$) value. Note that this initialization of the Q-tables corresponds to a uniform distribution with mean equal to B and variance given by $4c^2/12$. We observe that higher randomization weakens the convergence guarantees.

In order to gain further insights into how convergence depends on the CG realizations encountered during learning, we separately evaluate the spectral radius of matrix $A_{I\to i,J\to j}$ and the bound of matrix $B_{I\to i}$. In Figure 8.8, we present how the spectral radius of $A_{I\to i,J\to j}$ evolves with learning iterations. In particular, we calculate its moving average for a window of 70. We observe that the spectral radius in general increases with learning time for low ($G = 0.2$), intermediate ($G = 0.5$) and high ($G = 0.7$) channel loads. Also, it is significantly higher for intermediate channel loads. Both these observations can be justified by closely examining (8.48): matrix $A_{I\to i,J\to j}$ has mostly zero entries, denoting the absence of collisions. Also, the number of collisions tends to decrease as the learning process proceeds, due to agents learning how to avoid each other, and increase with the channel load. However, the non-zero entries acquire higher values as the learning process proceeds, due to agents becoming more certain of which actions are optimal. In addition, collisions are more common for high G, thus Q-tables, as well as the matrix $A_{I\to i,J\to j}$, have lower entry values, as collisions likely lead to failure to transmit and thus, lower rewards. Note that this is only evident for high loads ($G = 0.7$). As indicated in Figure 8.5, throughput is optimal for both $G = 0.2$

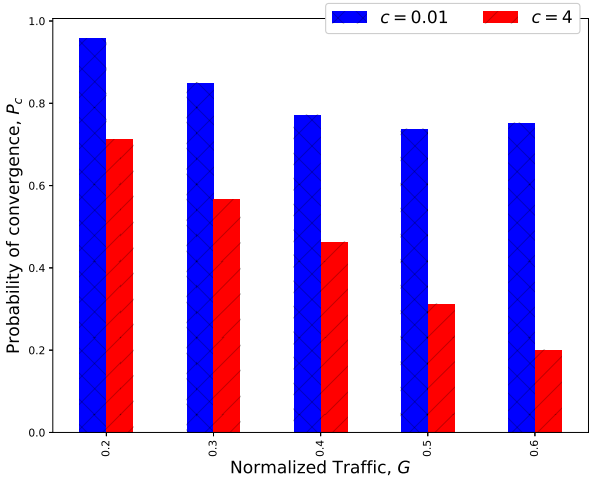


Figure 8.7: Evaluation of the sufficient condition for robustness for different channel loads and different initialization for the local Q-tables.

and $G = 0.5$, the lower values for $G = 0.2$ can be, therefore, attributed to the higher sparsity of the coordination graph.

In Figure 8.9, we evaluate how the bound of $B_{I \to i}$ differs from the worst-case scenario, based on a heuristic evaluation. In particular, our simulations indicate that devices tend to send different number of replicas for different frames for medium channel load ($G \in [0.4, 0.5, 0.6]$), in order to efficiently exploit the available slots, while they all agree to send a few replicas (1 or 2) when the channel load is high ($G = 0.7$). Based on these observations, we evaluate the bound of (8.50) using the current messages of check nodes, and observe that it takes significantly lower values than the worst-case analysis. It is worth noting that this heuristic evaluation gives lower values for $G = 0.7$ than for $G = 0.5$, as, with increasing sizes of the Q-tables, it is less likely that all agents associated with a Q-table will agree on a common action.

8.6.4 Evaluation of complexity

Figure 8.10 exhibits the benefits of our complexity reduction technique, which we measure as the average number of agents' collisions. We observe that throughput is low when complexity reduction is not employed, as the learning algorithm exhausts the time budget before a good policy is found. We can, thus, conclude that the reduction in complexity achieved is particularly significant for high channel loads ($G \geq 0.6$), where collisions are frequent and lead to CGs of low sparsity. Finally, in Figure 8.11, we present the time complexity of the different techniques employed in our solution. Note that, in these simulations, learning was rendered robust by employing the sufficient condition derived in Section 8.5, we therefore avoid cases that would require exhausting the number of message-passing iterations. Furthermore, by employing the MKP complexity reduction technique, CGs become more sparse and, therefore, significantly reduced time is required for coordination. It is, thus, anticipated, that the computation time of max-sum would have been much higher, had these two

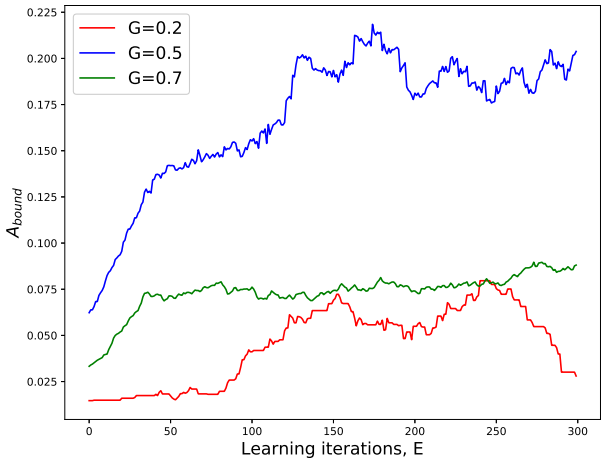


Figure 8.8: The evolution of spectral radius of matrix A with the learning iterations for varying values of load G.

techniques not been employed. We observe that calculating the condition for robustness is the main bottleneck of the operation and time complexity increases significantly with the channel load G.

8.6.5 Evaluation on different topologies

Finally, we perform an experiment that examines how our solution performs for different wireless network topologies. Figure 8.12 presents the throughput and time required for learning for two types of networks: a fully-connected (simple) wireless network with $C = 16$ and $D = 20$, and a wireless network with devices clustered into 4 clusters of the same size. We assume that devices can collide only with devices within their cluster. From Figure 8.12, we observe that, in a fully-connected network, learning is significantly slower, as the time budget is exhausted at 100 learning iterations. This suggests that the complexity of coordination is high due to CGs not being sparse enough. Regarding throughput, the clustered network achieves significantly higher performance, which can be attributed to: (i) experiencing more learning iterations in the same time period, and, (ii) coordination being more important, as clustering dependencies are persisting, whereas dependencies that arise due to collisions may change at each learning iteration. We base this conclusion on the observation that learning in the fully-connected network achieves lower throughput, even when the time budget is not exhausted ($L_E < 100$). Note that the complexity of our solution does not depend on the number of clusters, but increases with the size of the clusters. Due to the locality of interaction in wireless networks, it is natural to decompose the network into clusters that operate independently from each other. The sizes of these clusters should remain small compared to the size of the network and their value is dictated by practical limitations of devices, such as their transmission power. We also simulated networks consisting of clusters of 4 devices, channel load $G = 0.8$ and $C \in \{160, 320, 480\}$, and confirmed that the achieved throughput remained approximately 0.56 in all cases.

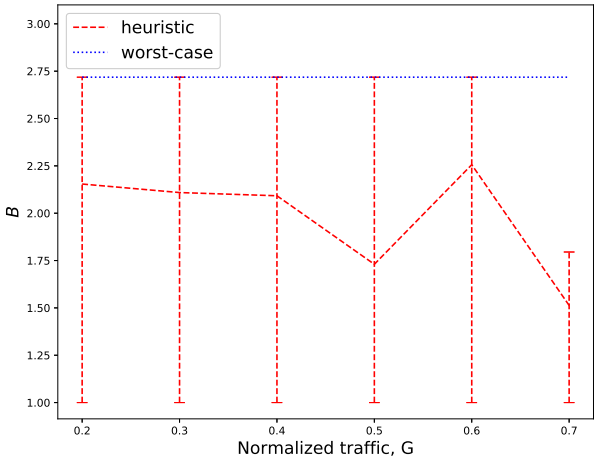


Figure 8.9: Evaluation of the bound of matrix B for different channel loads based on the worst-case theoretical analysis and heuristically calculated based on the values of the messages during simulation.

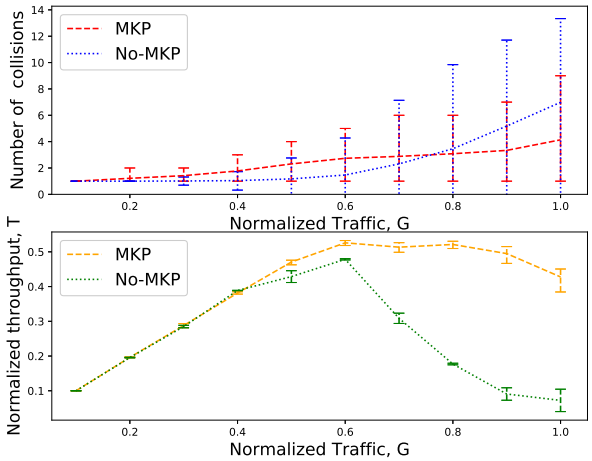


Figure 8.10: Number of collisions devices experience upon transmission and achieved throughput for coordinated agents before and after reducing complexity using the MKP technique.

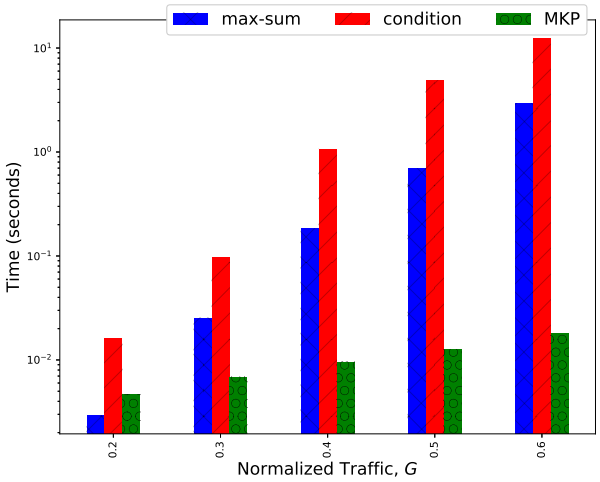


Figure 8.11: Time (measured in seconds) for the different stages of learning: application of max-sum, calculation of robustness condition and solution of the MKP.

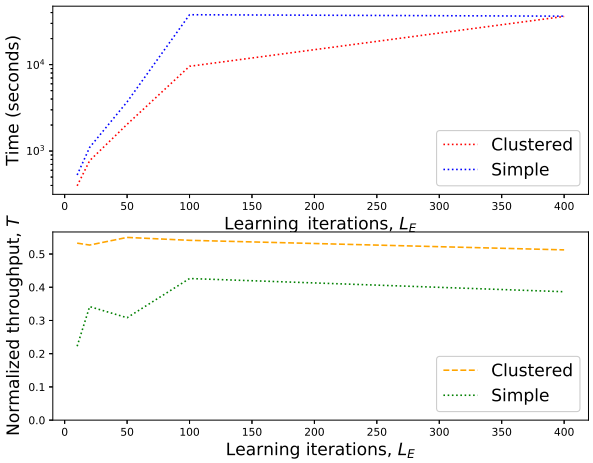


Figure 8.12: Comparison of time complexity and achieved average throughput for a fully-connected network (simple) with 16 devices, and a clustered one with 4 group of devices with 4 devices each for $G = 0.8$

8.7 Conclusions - Towards communication-free coordination

In this chapter, we presented an adaptive optimization technique for IRSA that uses coordinated Q-learning, where the max-sum algorithm is applied on CGs, used to describe the dependencies among devices, to find the optimal actions. We presented a technique for bounding complexity based on a multiple knapsack formulation, as well as a sufficient condition for the convergence of max-sum that can be used to evaluate whether coordination will lead to finding the optimal solution. In addition, we derived convergence guarantees for our Q-learning based algorithm in our learning framework, which we termed as Groupwise-Dependent Decentralized POMDP. Our simulations confirm that coordination is beneficial in terms of throughput and convergence rate, when compared with classical IRSA, as well as solutions employing independent learning. Furthermore, coordination must exploit structural properties, as well as complexity reduction techniques, so that computational complexity does not prohibit learning.

Despite our efforts to reduce the complexity of coordination, it remains possible that applications where energy efficiency is a pressing concern will not be able to afford the cost of communication required by employing the max-sum algorithm. In our next chapter, we present our work on answering the following question in the affirmative: *"is it possible to guarantee an optimal solution for Multiple Access without communication in a fully decentralized network?"*

Chapter 9

Multi-player bandit algorithms for dynamic spectrum access

Our contributions in Chapters 7 and Chapter 8 have provided solutions for time slot access. In this chapter, we remain concerned with resource-constrained networks, but shift our attention to optimizing channel access. As we explained in Chapter 1, Dynamic Spectrum Access is an important application created for addressing the problem of spectrum scarcity, that has acquired a lot of significance in emerging wireless applications. The communication scenario considered in this chapter differs significantly from the one considered in Chapters 7 and 8: devices are accessing channels, each corresponding to a different transmission frequency, to discover the one with the highest availability and have the ability of accessing multiple channels simultaneously. In contrast, in Chapters 7 and 8 devices where transmitting in a single frequency and accessing multiple time slot in a given frame. From the perspective of our learning-based solution for MA, the defining difference between time slot and channel access is that the former are always available to communication devices, while a channel becomes available with an unknown probability.

Adhering to the recent trend of simplifying the hardware of wireless devices, we focus on networks where devices do not have the ability of sensing the medium prior to transmission. By additionally prohibiting communication between devices, we formulate a problem setting where rewards are partially observable, as they contain information about both the arm availability and the occurrence of collisions. We, thus, venture into the design of a learning algorithm that answers the following question in the affirmative: *"is it possible to guarantee that devices will find a collision-free channel without sensing and without communicating?"*

The main novelty in our algorithms is the introduction of the collision resolution (CR) mechanism, which players employ to compute unbiased estimates of the arms' mean availability. We propose two algorithms that make use of this mechanism: CR-UCB, which has a sub-logarithmic upper regret bound for static problem settings, and DYN-CR-UCB, which is designed for dynamic problem settings, where players arrive at different time steps. Both algorithms improve upon the regret bound of state-of-the-art algorithms for the MMAB setting without sensing information. We perform empirical simulations for a variety of bandit instances and observe that players employing our algorithms converge to the optimal assignment of devices to channels more quickly than players employing existing state-of-the-art algorithms. We also analyze the CR mechanism theoretically using the AND-OR tree

analysis in order to derive an upper bound on the probability of its failure.

9.1 Introduction

The problem of dynamic spectrum access in cognitive radio [30] has recently reignited the interest in MMAB algorithms, where channels can be viewed as arms and communication devices as players searching for an optimal assignment in a decentralized and communication-free manner. In multi-armed bandits (MABs) a player sequentially interacts with a finite set $\mathcal{K} = \{1, ..., K\}$ of arms that incur rewards following unknown probability distributions, with the objective of maximizing the reward it accrues by the end of the problem horizon T. Multi-player multi-armed bandits (MMABs) are a generalization of this framework to the case where a set $\mathcal{M} = \{1, \cdots, M\}, M >= 2$ players compete for the same set of arms.

In this work, we consider problem settings where the availability of an arm k follows a Bernoulli distribution $Y_{k,t}$ with mean μ_k, the problem horizon T is known and decision-making is fully decentralized, i.e., there is no communication among players, and their number, M, is unknown. We study both *static* settings, where players start learning simultaneously and *dynamic* settings, where players may arrive at different time steps. Such a dynamic setting has recently been studied in [42] and does not consider departures of players, as these can be easily addressed by allowing players to leave at specific time intervals [42, 41]. Instead, dealing with arrivals is more difficult, as players cannot leverage collisions to indirectly communicate with each other [42]. When more than one players simultaneously pull an arm, a collision occurs and all players observe a reward of zero. As we explained in Chapter 5, it is customary to separate the possible feedback provided by an arm into three types of information: (i) the availability of the arm, $y_k(t)$; (ii) an indication of collision, $n_k(t)$; and (iii) the reward of the arm $r_k(t)$, which is equal to $y_k(t)$ if $n_k(t) = 0$, and zero otherwise. In our setting, often referred to as the no-sensing setting [40], we assume that only rewards are available. This suggests that, when a player receives a reward of zero, it is not clear whether this occurs due to a collision or an unavailable arm.

We first propose the Collision Resolution-UCB (CR-UCB) algorithm which unfolds in four phases, with the first one, the μ-estimation phase making use of the main novelty in our work, the Collision Resolution (CR) mechanism. During this phase, players compute an unbiased estimate $\hat{\mu}_k$ for each arm despite collisions. This was also achieved in [105] (see Algorithm 1), by assuming that M is known and multiplying each estimate of the rewards with the probability of collision. In our work, the CR mechanism offers an alternative technique for dealing with partial observability with significantly reduced *sample complexity*, which refers to the number of time steps required for a player to find an optimal arm. The M-estimation phase that follows makes use of the previously computed $\hat{\mu}_k$ to compute an estimate $\hat{M}$ of the number of players M. This phase requires significantly less time than Algorithm 3 in [105] to estimate M, while, to the best of our knowledge, no other solution exists in the no-sensing setting. CR-UCB completes with the Musical Chairs (MC) routine, which consists of an assignment and an exploitation phase. The original MC routine requires the observation collisions [41], so in our work we make use of the variant proposed and theoretically analyzed in [105]. We, also, present Dynamic CR-UCB (DYN-CR-UCB) an algorithm designed for dynamic settings, where players arrive at different times and begin learning immediately. We

do not follow the approach proposed in [41], which converts a static algorithm to a dynamic one by running it in epochs, as it forces players to wait for a long time before initiating learning.

The proposed CR mechanism is inspired by the realization that the no-sensing setting creates a learning process that can be studied using the AND-OR tree analysis [131]. The benefits coming from such an analysis is that we can theoretically bound the number of time steps required to compute the unbiased estimates $\hat{\mu}_k$. The μ-estimation phase divides its T_μ time steps into rounds of equal duration $I_{\max}$ and employs the CR mechanism, described in detail in Algorithm 2, independently in each round. The theoretical analysis of the CR mechanism is based on the evolution of random processes on bipartite graphs, which represent a resource allocation task using a set of nodes for representing players and another set of nodes for resources. The random process approximates the transmission mechanism of players, where initial collisions caused due to random sampling are *resolved* when a channel returns an acknowledgement message indicating successful transmission. To our knowledge, our work is the first attempt to transfer the AND-OR tree analysis to bandit settings, which differ from already studied resource allocation tasks [29, 46] in that resources are not always available.

Differently from the family of Selfish algorithms [40, 106], which, as we exhibited in Section 5.4, incur linear regret in some settings, the algorithms proposed in this work are accompanied by upper regret bounds, which improve upon known regret bounds of existing bandit algorithms in the no-sensing setting [42, 105].

9.2 Bandit model

We consider a K-armed bandit, where each arm is a Bernoulli distribution with mean availability μ_k. We denote with $Y_{k,t}$ the i.i.d. random variable associated with each arm, that satisfies $P(Y_{k,t} = 1) = \mu_k$ and refer to it as the availability of the arm. At each round t, M players simultaneously choose which of the K arms to pull, an action we represent as $a^m(t)$. Upon being pulled, an arm returns a reward of one if it is available, i.e., $y_k(t) = 1$, and only one player pulled it. Otherwise, it returns a reward of 0.

The reward model is formally:

$$r_m(t) = y_{a_m}(t)(1 - n_{a_m}(t)) \tag{9.1}$$

where $n_{a_m}(t)$ indicates whether a collision occurred on arm a_m. In this work, we consider that players only observe the reward $r_m(t)$.

This process is repeated for T time steps, where T is termed the problem horizon, and is fixed and known in advance. The objective of the team of players is to minimize the system expected cumulative regret at the end of the horizon, defined as:

$$R(T) = T \sum_{k=1}^{M} \mu_k^*(t) - \sum_{t=1}^{T} \sum_{m=1}^{M} r_m(t) \tag{9.2}$$

where μ_k^* is the mean availability of an arm belonging to the set of arms with the M highest means, which we denote as $\mathcal{M}^*$. We refer to such an arm as an M-best arm.

We should note that we choose the regret as our evaluation metric in accordance with the bandit literature. In contrast, in Chapters 7 and 8 we employed throughput, which is the most common evaluation metric in MAC protocol design. We should note that one can use throughput and system regret interchangeably, as there is an one-to-one correspondence between them. In particular, throughput in our bandit setting can be defined as:

$$D_t = \frac{\sum_{m=1}^{M} r_m(t)}{N} \tag{9.3}$$

The optimal value of throughput depends on the load of the system, i.e., $L = M/K$, and can be calculated as:

$$D^* \begin{cases} = \sum_{m=1}^{M} \mu_m^*/K, & \text{if } L \leq 1, \\ = \sum_{m=M-K}^{K} \mu_m^*/K, & \text{otherwise} \end{cases} \tag{9.4}$$

9.3 Algorithm for static settings

In the following, we introduce the collision resolution (CR) mechanism and then describe CR-UCB, the bandit algorithm that employs the mechanism to achieve state-of-the-art performance in static settings.

9.3.1 Collision-based multi-player bandits revisited

In collision-based multi-armed bandits, players adopt strategies that can lead to collisions and attempt to learn the optimal allocation of arms despite them. Recently, the classical UCB-based strategies UCB1 [72] and klUCB [73] were evaluated in this setting and were found to be well-performing. From the perspective of reinforcement learning, ignoring collisions is a multi-armed bandit is equivalent to employing independent Q-learning in an MDP, while from the perspective of RA protocols, it is equivalent to the ALOHA protocol. It is, therefore, not surprising that this approach is not optimal, as collisions are frequent in networks where the load, i.e., the ratio of devices to channels, is high.

To clarify the relation between the classical problem of RA and our setting, we abstractly present in Figure 9.2 a time step in a bandit setting with $K = 4$ and $M = 3$. We adopt the common practice of mapping the resource allocation task to a bipartite graph with two sets of nodes: lower nodes represent players and upper nodes represent arms. In the time step presented in Figure 9.1, players 2 and 3 have both chosen arm 2, so regardless of its availability, μ_2, they will receive a reward of zero.

The main motivation behind our approach is the realization that the reward model presented in (9.1) can be studied using the AND-OR tree analysis, a tool developed to analyze the evolution of random processes on graphs [131]. To see how this is possible, we slightly modify the bandit model by considering the case where a player has the capability of pulling multiple arms per sampling step. To decouple the discussion of the CR mechanism from the

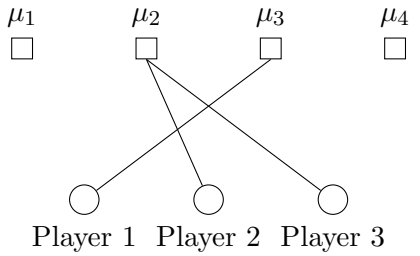


Figure 9.1: Single-pull multi-player multi-armed bandit.

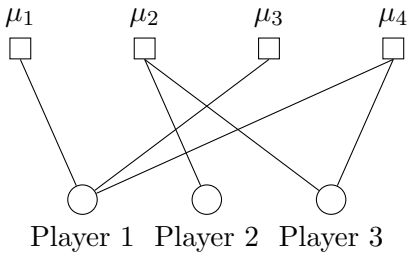


Figure 9.2: Multiple-pull multi-player multi-armed bandit.

sampling problem, we assume that $\mu_k = 1,\ \forall k \in \mathcal{K}$ in this example, so that a reward of zero is observed only in the case of a collision.

In Figure 9.2, player 1 pulls 3 arms, player 2 pulls 1, and player 3 pulls 3. At first sight, this strategy looks counter-intuitive, as players experience more collisions than previously, while the end result remains the same: only player 1 observes a reward of one. This, however, changes if we employ the following CR mechanism: all players repeat their actions (i.e. pulling the same subset of arms) until they observe a reward of one from one of them. When this happens, the player stops pulling the other arms and keeps pulling the arm that gave the reward of one. This process lasts for a fixed number of $I_{\max}$ iterations, which form a single CR round. In our example, this mechanism will lead to a collision-free assignment within 3 time steps. First, player 1 receives a reward of one from arm 1, so they stop pulling arms 3 and 4. Then, player 3 receives a reward of one from arm 4 and stops pulling arm 2, and finally, player 2 receives a reward of one from arm 2. For the remaining iterations, up to $I_{\max}$, the three players can continue pulling the arms that gave a reward of one and update two variables for each arm: (i) S_k, the cumulative rewards for arm k; (ii) T_k, the number of pulls of arm k. The empirical mean rewards of arm k can be, then, computed as $\mu_k = \frac{S_i}{T_i}$.

To make the discussion more formal we introduce some additional notation. Let d_m be the number of arms player m pulls, a random variable following a probability distribution $\Lambda_m(x) = \sum_{d=1}^{\Lambda_{\max}} \Lambda_d x^d$, where $\Lambda_{\max}$ is the maximum number of simultaneous pulls allowed to a player. We consider that all players are employing the same degree distribution, i.e. $\Lambda_m(x) = \Lambda(x),\ \forall m \in \mathcal{M}$. We denote as $\lambda_d(x)$ the probability that an edge, corresponding to a pull, is connected to a player of degree d, where the degree of a node is defined as the number of edges connected to it. We, also, denote as $\rho_d(x)$ the probability that an edge is connected to an arm of degree d. These events are also random variables, sampled from a distribution $\lambda(x) = \sum_{d=1}^{\Lambda_{\max}} \lambda_d x^{d-1}$ and $\rho(x) = \sum_{d=1}^{\Lambda_{\max}} \rho_d x^{d-1}$. The probability of failure of the CR mechanism can be described using the AND-OR tree analysis, derived in Section 9.4.2, where players consider that the resolution is successful if at least one pull has been resolved (OR function) and an arm considers resolution successful if all pulls have been resolved (AND function). A CR round completes successfully when all players have found a collision-free arm. If a player finds a collision-free arm i within a round, it updates its estimate of the mean availability of that arm. Algorithm 2 presents the pseudocode of a

collision resolution (CR) round.

Algorithm 2: CR Round

Data: $\mathcal{K}, \Lambda, I_{\max}$

Decide d, the number of arms to pull simultaneously, by sampling $\Lambda(x)$
Form a random sub-set $\vec{k}$ by randomly choosing d out of the K arms
free = False
for $\tau \in \{1, \ldots, I_{max}\}$ **do**
    Pull all arms in $\vec{k}$
    Observe rewards $\vec{r}_k$
    if $\exists i : r_i == 1$ *and not free* **then**
        Remove all elements except i from $\vec{k}$ ▷ Detect collision-free arm
        $free = True$
    end
    if *free* **then**
        $S_{\vec{k}} = S_{\vec{k}} + r_{\vec{k}}$ ▷ Update sum of rewards for collision-free arm
        $T_{\vec{k}} = T_{\vec{k}} + 1$ ▷ Update number of pulls for collision-free arm
    end
end
Return free, $\vec{k}$

An important step in the theoretical analysis of our proposed algorithms is to determine the conditions under which the CR mechanism succeeds with high probability. We consider that a round completes successfully for a player when it has found a collision-free arm by the end of it. The following theorem states that the probability of failure of the CR mechanism is upper bounded by a value that depends on the number of players and the probabilistic structure of the bipartite graph.

Theorem 1. Assume that a CR round of $I_{\max}$ iterations takes place among M players accessing K arms and the degree distribution is $\Lambda(x)$. Then, with probability at least equal to $1 - \delta_4$, a CR round fails for a player with probability at most:

$$p_f = q_{I_{\max}} + \sqrt{\frac{-\ln(\delta_4)}{\eta M}},$$

where $q_{I_{\max}}$ is the probability of failure in an asymptotic setting ($M, K \to \infty$) at the end of the CR round and η is a constant whose value depends on the structure of the graph and is given in Lemma 4 in Appendix 9.4.2.

Proof (Sketch). Our proof requires results found in different works from the field of information theory [88, 132, 133, 45, 131, 134]. We, therefore, deemed it necessary for the completeness of our work to gather these results and adjust them to our problem setting. In Appendix 9.4.1, we provide a general description of bipartite graphs and present Lemma 1, which bounds the probability that a bipartite graph of finite size does not have a tree structure, and in Appendix 9.4.2, we present the analysis of the CR mechanism. The first step of the proof, in Lemma 2, is to derive the condition under which the probability of failure of the

CR mechanism is monotonically decreasing with each iteration in a round. This condition, referred to as the stability condition, is derived assuming that M is infinite, which simplifies the analysis as it guarantees that the graph is cycle-free, meaning that the probability of failure evolves independently for each player. Thus, we can compute the duration of a CR round, $I_{\max}$, based on a target probability of error for the CR mechanism, $q_{I_{\max}}$. Note that we have slightly modified the existing analysis to take into account the effect of arm availabilities, i.e., μ_k, have on the calculation of $I_{\max}$, as the original proof considered a setting with $\mu_k = 1, \forall k \in \mathscr{K}$. In order to transfer the analysis to settings with finite M and K, where cycles may appear on the bipartite graph, in Lemma 4 we formulate the process of resolving collisions as a martingale and derive a concentration inequality that describes how the probability of failure diverges from its asymptotic expectation. We make use of Lemma 4 by setting the right-hand side of (9.13) to be equal to δ_4, which leads to the value of $\alpha = \sqrt{\frac{-\ln(\delta_4)}{\eta M}}$. □

We should note that the bound appearing in Theorem 1 is not valid unconditionally. In particular, Lemma 4 introduces a condition on the minimum number of players M, i.e., $M > 2\gamma/\alpha$, where γ is a constant that depends on the probabilistic structure of the bipartite graph and is defined in Lemma 1, and α was defined above, for the result to be valid. In order to derive this condition, the analysis of Richardson and Urbanke [43] makes a very conservative estimation which relies on the assumption that cycles of any length in the bipartite graph can lead to failure of the CR mechanism, by requiring that $l = I_{\max}$ in the estimation of γ. While it has been empirically observed that only cycles of very small length affect the performance of random processes on graphs [43], this conjecture remains to be theoretically proven.

9.3.2 The CR-UCB algorithm

In a similar spirit to the MC algorithm, we break down the task of finding the optimal assignment of arms to players into smaller tasks. The major difference with the setting considered by the MC algorithm is that players cannot observe collisions. An inspection of the MC algorithm reveals two ways in which this can be problematic. First, players do not know whether a reward of zero is due to an unavailable arm or a collision, so they cannot compute an unbiased estimate $\hat{\mu}$. Second, a player cannot have an unbiased estimate of the probability of collision, which is required to determine the number of players, due to not being able to observe the occurrence of collisions.

In the following, we divide CR-UCB into four phases and explain how each phase helps address these problems. We assume that K arms become simultaneously available for a horizon T to M players, whose number remains static. All phases are described from the perspective of a single player, as players act independently and in parallel. The pseudocode for CR-UCB is presented in Algorithm 3.

$\mu-$estimation phase

During this phase, a player aims to compute estimates of the availability $\hat{\mu}_k$ for all arms, accurately enough to have an ϵ-correct ranking of the arms. We denote a ranking as ϵ-correct,

when arms whose mean availabilities differ by at least ϵ are ranked correctly. This phase has a duration of T_μ steps and is divided into $T_\mu/I_{\max}$ CR rounds of equal duration $I_{\max}$. [1] The CR mechanism, which was described in Section 9.3.1, guarantees that, at the end of each round, a player has found a collision-free arm. Therefore, during a round a player collects at least one and at most $I_{\max}$ unbiased samples of the mean availability of an arm. We should note that players need to employ the same value of $I_{\max}$, which remains fixed throughout learning and needs to be high enough to guarantee resolution with high probability.

Algorithm 3: CR-UCB

Data: $\mathcal{K}, \Lambda, I_{\max}, T, T_\mu, T_M$
Result: M strategies

Initialize $S_k = 0, T_k = 0, \ \forall k \in \mathcal{K}$
for $t \in \{1, \ldots, T_\mu/I_{max}\}$ **do**
  CR Round$(\mathcal{K}, \Lambda, I_{\max})$
end
Compute $\hat{\mu} = S_k/T_k, \ \forall k \in \mathcal{K}$
Initialize $F_k = 0, \ \forall k \in \mathcal{K}$
for $t \in \{1, \ldots, T_M\}$ **do**
  Select arm k randomly from K
  Observe reward r_k
  if $r_k == 1$ **then**
    $F_k = F_k + 1$
  end
end

Compute $\hat{M}$ using (9.5)
$r_k = 0$
while *not* r_k **do**
  Select arm k randomly from M^*
  Observe reward r_k
end
while $t \leq T$ **do**
  Pull arm k
end

μ-estimation phase.
M-estimation phase.
Assignment phase.
Exploitation phase.

$M-$ estimation phase

During this phase, players compute an estimate of M, $\hat{M}$, in a decentralized manner. The phase evolves as follows: for T_M time steps, each player pulls a randomly selected arm out of the K arms and observes the reward. If the reward is one, the player increments an arm-specific counter F_k by 1. The probability of observing a reward of one during T_M time

[1] To ensure that the last round runs to completion we set $T\mu = \lfloor (T_\mu/I_{\max}) \rfloor I_{\max}$.

steps for an arm with mean availability is $\hat{\mu}_k$ is

$$1/K(1-1/K)^{M-1}\mu_k$$

where we have taken into account that each player pulls a single arm uniformly at random. Thus, a player can use the value of the counter F_k at this end of this phase to compute the following K estimates of M:

$$\hat{M}_k = \left\lfloor \frac{\log(\frac{F_k K}{T_M \mu_k})}{\log(1-\frac{1}{K})} + 1 \right\rfloor, \quad \forall k \in \mathcal{K} \tag{9.5}$$

To get a single estimate $\hat{M}$ the player adds the K equations in (9.5). We observe that the estimates $\hat{\mu_k}$ computed in the previous phase are vital in the success of this phase. Also, this phase does not introduce any additional need for synchronization. The number of steps T_M does not need to be pre-determined nor communicated: a player can compute T_M using a concentration inequality and $\mu_{\min}$ so that $\hat{M}$ is an accurate estimate of M with a high probability.

Assignment and exploitation phase

At the beginning of this phase, each player knows $\hat{M}$ and which are the M-best arms. Their objective now is to select one of them and continue pulling it until the end of the horizon T, without experiencing collisions. The MC routine, described and analyzed in a partially observable setting in [105], can be used to achieve this. The routine consists of two phases, the assignment and the exploitation phase. During the first phase players randomly sample an arm and observe its reward. As soon as a player finds a arm with a reward of one, they enter the exploitation phase for all time steps that follow.

Thus, players that are in the assignment phase coexist with players in the exploitation phase, the former causing collisions due to their random selections of arms. As soon as all players enter the exploitation phase, the system stops accumulating regret (provided that estimates $\hat{M}$ and $\hat{M}^*$ are correct).

The observant reader may wonder why we did not employ the CR mechanism to find a collision-free arm for each player within a single CR round during this phase. Although this is possible, we can prove that the number of iterations required for all players to enter the exploitation phase would be larger using this approach. As M players are competing for the M-best arms, the load of the network is effectively equal to one. From the analysis of the IRSA protocol, it is known that, when the load is larger than approximately 0.97, the strategy of randomly accessing a single resource has a higher probability of succeeding than accessing multiple resources simultaneously and resolving collisions [29]. Although our setting differs in that resources correspond to channels and not time slots, it is still true that the large number of collisions due to having a load of one would defeat our objective of finding a collision-free arm as quickly as possible.

9.3.3 Regret Analysis of CR-UCB

The main result of our theoretical analysis, a high-probability upper bound for the cumulative system regret, is presented in Theorem 2. In order to derive it, we need to estimate the duration of the three phases, which will guarantee that CR-UCB succeeds with high probability. We derive bounds for the duration of the μ-estimation and M-estimation phases in Appendices 9.4.3 and 9.4.4, respectively. As the MC phase is identical to the one used by Lugosi and Mehrabian [105], we directly use Lemma 2.3 of that work to bound its duration.

Theorem 2. Let $\Delta > 0$ be the gap between the expected availability of the M-th and the $(M+1)$-th best arm. Then, for all $\epsilon < \Delta$, with probability $\geq 1-\delta$, the expected regret of players using the CR-UCB algorithm with K arms for T rounds, when the length of the μ-estimation phase is $T_\mu = \left\lceil \max\left\{ \frac{1}{8\mathbb{E}[A_{i,j}(t)]} \ln\left(\frac{K^2}{\delta_2}\right), 2\frac{1}{\mathbb{E}[A_{i,j}(t)]} \ln\left(\frac{2K^2}{\delta_1}\right)\frac{2}{\epsilon^2} \right\} \right\rceil$, the length of the M-estimation phase is $T_M = \frac{\log(2K^2/\delta_3)}{0.4802}$ and the length of the MC phase is $T_{MC} = \frac{4M\log(M^2T)}{\mu_{\min}}$, is at most:

$$\Big(T_\mu + T_M + T_{MC}\Big)M, \tag{9.6}$$

where $\mathbb{E}[A_{i,j}(t)] = \frac{\left(1-p_f\right)\sum_{d=1}^{D}\Lambda_d\frac{d}{K}}{I_{\max}}$, and the duration of a CR round is chosen so that the expected probability of failure is lower than a target δ_5, i.e. $I_{\max} = \min_t(t \in \{1,\ldots,T\} : q_t < \delta_5)$.

Our theoretical analysis follows closely the analysis of the MC algorithm [41, in particular Appendices A.1.1 and A.1.2]. There are however a couple of essential differences: during the μ-estimation phase players do not directly observe the availability of arms at every time step t, but get at least one observation every $I_{\max}$ steps provided that the CR mechanism succeeds in a given round. During the M-estimation phase, players do not estimate the number of players $\hat{M}$ by calculating a single probability of collision on all arms, but maintain K estimates of the probability of observing a reward of one, taking into account their $\hat{\mu}_k$ estimates. We elaborate on how these differences affect the regret of CR-UCB in Sections 9.4.3 and 9.4.4.

To get a clearer picture of the sample complexity of CR-UCB, we present the relationship between the number of players and the bounds on the number of time steps that its different phases require, which can be computed based on Theorem 2. In particular, in Figure 9.3, we present the bounds on the duration of the μ-estimation phase (T_μ), the M-estimation phase (T_M) and the MC routine that consists of the assignment and exploitation phase (T_{MC}). We observe that the M-estimation phase requires the most time and that the duration of all phases does not increase significantly as the size of the problem increases, which indicates that CR-UCB is scalable.

9.4 Theoretical analysis

9.4.1 Useful Properties of Bipartite Graphs

This appendix includes results related to bipartite graphs that are useful for analyzing the CR mechanism.

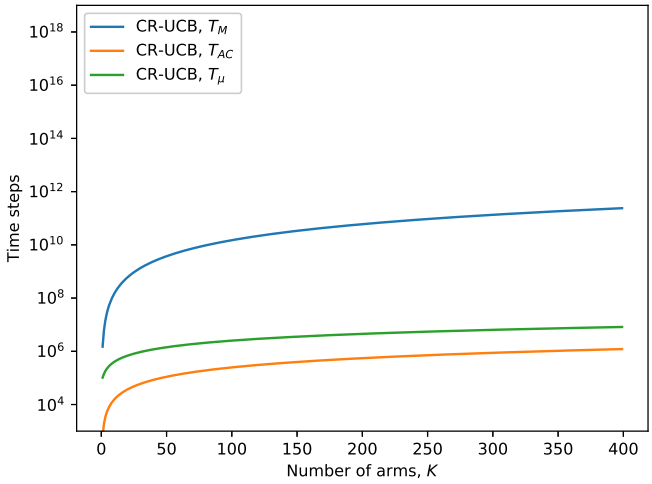


Figure 9.3: The upper bounds on the duration of the different phases of CR-UCB for a problem setting with load 0.5 and mean availabilities μ_k randomly sampled in the range $[0.1, 1]$.

We denote a bipartite graph describing a problem setting of M players and K channels as $G(\mathscr{M}, \mathscr{K}, \mathscr{E})$, where $\mathscr{E}$ is the set of edges representing arms pulled by players at a given time step. In our analysis, we refer to nodes representing players as player nodes (PNs), nodes representing arms as arm nodes (ANs) and denote an edge between player m and arm k as $\vec{e} = (m, k)$. An example of a bipartite graph is presented in Figure 9.4.

Players pull random subsets of arms, with the size of the subset being determined by sampling the degree distribution $\Lambda(x) = \sum_{l=1}^{D} \Lambda_l x^l$, where D is the maximum number of pulls allowed to a player. We denote an ensemble of bipartite graphs as $\mathscr{G}(M, K, \Lambda(x))$, i.e. the family of bipartite graphs that can be generated using this random process. From the perspective of ANs, $\Psi(x) = \sum_{l=1}^{D} \Psi_l x^l$ is the distribution describing the number of pulls on each arm. Thus, the average number of pulls for a player can be denoted $\bar{\Lambda} = \sum_{l=1}^{D} l\Lambda_l$ and the average number of pulls for an arm can be equivalently denoted as $\bar{P} = \sum_{l=1}^{D} lP_l$. This leads to the following relationship for the load of the network: $L = M/K = \Psi'(1)/\Lambda'(1)$. In addition to the $\Lambda(x)$ and $P(x)$ degree distributions, which we term as node-perspective, we also often refer to the edge-perspective degree distributions $\lambda(x) = \sum_{l=2}^{D} \lambda_l x^{l-1}$ ($\rho(x) = \sum_{l=2}^{D} \rho_l x^{l-1}$), where $\lambda_l(\rho_l)$ denotes the percentage of edges that are connected to a PN (AN) of degree l.

An important trait of our theoretical analysis is that it concerns randomly built graphs. As a result, the exact connections between PNs and ANs cannot be known in advance and vary for different CR rounds. In order to analyze the performance of the CR mechanism, and thus the regret of CR-UCB and DYN-CR-UCB, we need to ensure that the performance of a given graph is close to that of its ensemble. This is termed the *concentration* property of the ensemble, and will be proven in Appendix 9.4.2 for our setting.

An important concept in our analysis is that of a sub-graph, $G_{\vec{e}}^{2l}$, which is obtained by the following process: choose an edge $\vec{e} = (m, k)$ uniformly at random from among all edges of a bipartite graph $G(\mathscr{M}, \mathscr{K}, \mathscr{E})$, and then consider the sub-graph induced by the left node

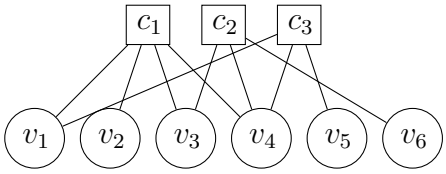


Figure 9.4: Illustration of a bipartite graph where player nodes' (PNs) degrees are either 2 or 3.

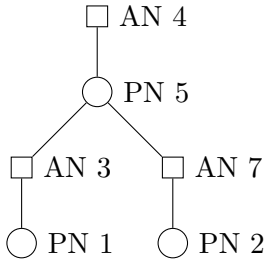


Figure 9.5: The induced sub-graph for edge (5, 4) and $l = 1$. This sub-graph is tree-like, because no node appears twice.

m and all its neighbors within distance $2l$ after deleting the edge (m, k). Sub-graphs are useful because they help us describe how each step of the CR mechanism affects the structure of the original bipartite graph. An alternative way to describe a sub-graph is through the neighborhood $\vec{\mathcal{N}_e^{2l}}$, which is the set of all nodes and edges included in the corresponding sub-graph $G_{\vec{e}}^{2l}$. Figure 9.5 presents a $G_{\vec{e}}^{2}$ sub-graph induced for the edge $\vec{e} = (5, 4)$. As the nodes appearing are distinct, the sub-graph is characterized as tree-like.

As we will see in the analysis of the CR mechanism, a tree-like structure is essential for ensuring that all players manage to find a collision-free arm. Lemma 1 proves that the probability that a sub-graph is not tree-like is negligible for a large enough number of players M. We have derived it by extending existing analysis [43, Appendix A], which concerned regular bipartite graphs.

Lemma 1. Consider a randomly constructed graph G. Let $G_{\vec{e}}^{2l^*}$ be the sub-graph of fixed length $2l^*$ for a given edge $\vec{e}$. Then, for some constant γ:

$$\mathbb{P}(\mathcal{N}_e^{2l^*} \text{is not tree-like}) \leq \frac{\gamma}{M}$$

Proof. Denote with $\Lambda_{\max}$ the maximum degree of a PN and $P_{\max}$ the average degree of an AN. (Note that in other parts of the paper we also refer to $\Lambda_{\max}$ as D.) Under the assumption that the sub-graph is tree-like, the number of PNs in the sub-graph is:

$$M_{l^*} = \sum_{i=0}^{l^*} (\Lambda_{\max} - 1)^i (P_{\max} - 1)^i \tag{9.7}$$

and the number of ANs is:

$$K_{l^*} = 1 + (\Lambda_{\max} - 1) \sum_{i=0}^{l^*-1} (\Lambda_{\max} - 1)^i (P_{\max} - 1)^i. \tag{9.8}$$

The proof proceeds constructively. First, we prove that removing an edge connected to a PN does not change the tree-structure form of a sub-graph with high probability. We, then, prove an equivalent result for an AN. Then, considering that a sub-graph with $l = 0$ trivially has a tree-structure, we prove a lower bound for the sub-graph of length $2l^*$.

Consider $l < l^*$. Assume that $\mathscr{N}_{\vec{e}}^{2l}$ is tree-like and that k edges have been removed so far. The probability that removing another edge connected to a PN will not create a loop can be computed by considering whether expanding the sub-graph from that edge will not, at any level of the sub-tree, randomly hit an AN that is already in the neighborhood. This probability is equal to $\frac{(K-K_l-k)P_{\max}}{KP_{\max}-K_l-k}$. Assuming that K is sufficiently large, we get: $\frac{(K-K_l-k)P_{\max}}{KP_{\max}-K_l-k} = 1 - \frac{(K_l+k)(P_{\max}-1)}{KP_{\max}-K_l-k} \geq 1 - \frac{K_l^*}{K}$

Since $(K_{l+1} - K_l)$ ANs are added to the sub-graph at this step, the probability that the edge removal will lead to a tree-like sub-graph is $(1 - \frac{K_l^*}{K})^{(K_{l+1}-K_l)}$.

Equivalently for an AN, the probability that removing an edge connected to it does not create a loop is $\frac{(M-M_l-k)\Lambda_{\max}}{M\Lambda_{\max}-M_l-k}$. Assuming that M is sufficiently large, we find that:

$$\frac{(M - M_l - k)\Lambda_{\max}}{M\Lambda_{\max} - M_l - k} = 1 - \frac{(M_l + k)(\Lambda_{\max} - 1)}{M\Lambda_{\max} - M_l - k} \geq 1 - \frac{M_l^*}{M}.$$

Since $(M_{l+1} - M_l)$ PNs are added to the sub-graph at this step, the probability that this edge removal will lead to a tree-like sub-graph is at least: $(1 - \frac{M_l^*}{M})^{(M_{l+1}-M_l)}$.

We now transfer these results to the original sub-graph $\mathscr{N}_{\vec{e}}^{2l^*}$, where the probability that the sub-graph is tree-like is lower-bounded by

$$\mathbb{P}(\mathscr{N}_e^{2l}\text{is tree-like}) \geq \left(1 - \frac{K_{l^*}}{K}\right)^{K_{l^*}} \left(1 - \frac{M_{l^*}}{M}\right)^{M_{l^*}}.$$

We then use the Taylor series of $(1 - x/n)^x$ and approximate the preceding bound with the first term of the series:

$$\mathbb{P}(\mathscr{N}_e^{2l} \text{ is tree-like}) \geq \left(1 - \frac{M_{l^*}^2}{M}\right)\left(1 - \frac{K_{l^*}^2}{K}\right).$$

This leads to

$$\mathbb{P}(\mathscr{N}_e^{2l} \text{ is not tree-like}) \leq \frac{M_{l^*}^2 + \frac{P_{\max}}{\Lambda_{\max}} K_{l^*}^2}{M}. \tag{9.9}$$

□

In order for the bound proposed by the above Lemma to correspond to a probability that converges to 0 as M grows, we need to make sure that $\gamma < M$. Based on the proof, and in particular Eq. (9.9), this suggests that the number of players needs to satisfy the following constraint:

$$M \geq 4 \cdot M_l^2 \tag{9.10}$$

The intuition behind this constraint is that the maximum number of pulls of a single player, D needs to be adjusted based on the total number of players, in order to ensure that the bipartite graph is sparse enough to ensure the resolution of collisions.

9.4.2 Analysis of the CR mechanism

The analysis in this section establishes the conditions under which the CR mechanism succeeds for all players. A CR round is successful for a player when they have found a collision-free arm within this round. Exiting a round without having found a collision-free arm is considered a failure for a player.

We begin by assuming that all sub-graphs are tree-like and then proceed with relaxing this assumption. The analysis consists of the following steps: first, we derive a condition under which the expected value of the probability of failure is monotonically decreasing with each iteration t of the CR round (Lemma 2). Under this condition, which we refer to as the *stability condition* of the CR mechanism, a collision-free arm is found by all players with a probability that approaches 1 at a rate exponential in t (Lemma 3). Then, we prove that the probability of failure concentrates around its expected value at a rate exponential in M, where the expectation is taken over all possible realizations of bipartite graphs (Lemma 4). Finally, we show in Lemma 4 that with high probability, an exponentially small number of players have not found a collision-free arm by a certain iteration t.

We consider a setting with K arms where the mean availability μ_k of arm k is randomly sampled in $[\mu_{\min}, 1]$. We assume that M players are using the CR mechanism described in Section 9.3.2. The following Lemma presents the stability condition of the CR mechanism. It is based on Lemma 1 of Luby, Mitzenmacher, and Shokrollahi [131], which derived a similar stability condition for a problem setting where arms corresponded to time slots instead of channels and were always available ($\mu_k = 1,\ \forall k \in \mathscr{K}$). Note that, in the following Lemma, t refers to an iteration within a single CR round.

Lemma 2. Consider a cycle-free bipartite graph derived by the edge-perspective degree distribution $\lambda(x)$. Denote with q_t the probability that a player has not found a collision-free arm at iteration t. Then, the probability that a player has not found a collision-free arm approaches 0 as t grows to infinity if, for all $q_t \in (0, 1]$:

$$\lambda(1 - \rho(1 - q_t)) < q_t \tag{9.11}$$

Proof. We will first prove that the quantity on the left-hand side represents the probability of failure at the next iteration of the CR round, q_{t+1}.

Consider a PN of degree l. Denote by q the probability that an edge has not been removed, given that each of the other $l-1$ edges has been removed with probability $1-p$. The edge of a player is removed when at least one of the other edges is removed. Thus, $q = p^{l-1}$. As edge-perspective degrees follow the degree distribution $\lambda(x)$, we can infer that

$$q_t = \sum_{l=1}^{D-1} \lambda_l p_{t-1} = \lambda(p_{t-1})$$

Similarly, consider an AN of degree l, where p denotes the probability that an edge has not been removed given that each of the other $l-1$ edges have been removed with probability $1-q$. As we know, the edge of an arm is removed when all other edges have been removed (no collisions) and the arm is available, which happens with probability μ_k, a random variable taking values in $[\mu_{\min}, 1]$. As a player keeps pulling the same set of arms until a reward of one

is observed, we can ignore the effect of μ_k at this step and set $1-p=(1-q)^{l-1}$. Considering that the edge-perspective degrees of ANs follow the degree distribution $P(x)$, we can infer that

$$p_t = \sum_{l=1}^{D-1} p_l(1-(1-q_t)) = 1-\rho(1-q_t).$$

By inserting the expression of p_t into the expression of q_t, we get:

$$q_{t+1} = \lambda(1-\rho(1-q_t)).$$

In order for the CR mechanism to succeed we need to ensure that q goes to 0 as t grows. A necessary condition for this to happen is that $q_{t+1} < q_t,\ \forall q_t \in [0,1]$. The following expression is the stability condition of the CR mechanism:

$$\lambda\big(1-\rho(1-x)\big) < x, \forall x \in [0,1].$$

An alternative formulation of the stability condition that will prove useful in the analysis that follows is:

$$\lambda\big(1-\rho(1-x)\big) < x(1-\epsilon), \quad \forall x \in (0,1], \tag{9.12}$$

where ϵ is a positive constant.

□

Thus, in order to make sure that the CR mechanism succeeds with a target probability δ_5, we need to set $I_{\max}$ to the minimum number of time steps that satisfy $q_t < \delta_5$, multiplied by $1/\mu_{\min}$. This multiplication is due to the fact that any arm needs to be sampled at least $1/\mu_{\min}$ to return a reward of 1, which is necessary for the CR mechanism to continue.

The following trivial lemma, originally proposed by Luby, Mitzenmacher, and Shokrollahi [131], states that the probability of failure for a single player decreases exponentially with the iteration index t and that, for any upper bound on the probability of failure, there exists an iteration that satisfies it.

Lemma 3. If the stability condition in (9.12) is satisfied, then, for any $\gamma > 0$ we can set t to a constant such that $y_t < \gamma$.

Proof. From the stability condition in Eq. (9.12) it holds that $x_t < x_{t-1}(1-\epsilon) < x_{t-2}(1-\epsilon)^2 < \cdots < (1-\epsilon)^t$. If we set $t = c/\epsilon$, for some $c > 1$, then $x_t < (1-\epsilon)^{c/\epsilon} \leq e^{-c}$, where the last inequality can be confirmed by studying the monotonicity of $\ln(1-x)+x$. We set $\gamma = e^{-c}$ and the proof is complete. □

Our analysis has so far assumed that all sub-graphs have a tree structure and does not take into account how performance on arbitrary graphs concentrates around its expected value. Using Lemma 1, we can prove that all sub-graphs are tree-like with high probability. We, therefore, need to just study the concentration of the performance of tree-like sub-graphs around their expected value, denoted as q_t in Lemma 2. We should note that the following Lemma, originally formulated by Luby et al. [88, Theorem 1], is valid independently of whether the stability condition is satisfied.

Lemma 4. Let t denote the iteration in a CR round and Z_t be the random variable describing the fraction of players that have not found a collision-free arm after t iterations. Let $\mathbb{E}[Z_t]$ denote the expected value of Z_t, where the expectation is over all bipartite graphs and notice that it is equal to q_t, appearing in Lemma 2. Then, there is a sufficiently large constant M, such that for any $\alpha > 0$ and some constant η:

$$\mathbb{P}(|MZ_t - Mq_t| > M\alpha] < e^{-\eta\alpha^2 M} \tag{9.13}$$

Proof. The proof requires two intermediate steps. First, we need to bound the probability that the CR mechanism will create sub-graphs that do not have a tree structure. Then, we need to prove that the probability of failure for all graphs with a tree-structure concentrates around its expected value. We prove both results by formulating the edge removal process under the CR mechanism as a martingale and employing Azuma's inequality to prove a concentration bound.

Let M^* be the number of players for which the sub-graph of up to $2l$ levels is a tree. From Lemma 1, we know that the probability that this sub-graph fails to be a tree is upper bounded by γ/M. For large M, this bound can be upper bounded as follows: $\gamma/M < \alpha/4$. Thus, the expected number of players with tree-structured sub-graphs is lower bounded as: $\mathbb{E}[M^*] \geq M(1 - \alpha/4)$.

We now obtain a concentration result for M^* by describing edge removal as a martingale. We define Z_t to be the expected value for M^*, given the effect of the first t removals. In particular, $Y_0 = \mathbb{E}[M^*], Y_M = M^*$ and we define a filtration $\{\mathscr{F}_0, \cdots, \mathscr{F}_t\}$, where $\mathscr{F}_t$ is a σ-algebra containing the sub-graph at step t. Then, the sequence $Z_t = \mathbb{E}[Y|\mathscr{F}_t]$ forms a standard Doob's martingale with $\mathbb{E}[Y_{t+1|Y_t}] = \mathbb{E}[Y_t]$. Using the additional observation that consecutive values of Y_t differ only by a constant [88, Lemma 1] and Azuma's inequality, we can derive the following concentration inequality for the number of players with sub-graphs without a tree-structure:

$$\mathbb{P}(|M^* - M| > M\alpha/2) < \frac{1}{e^{\eta_1 \alpha^2 M}}, \tag{9.14}$$

where η_1 is an appropriate constant.

Now, let M' denote the number of players, out of M^* total players, which have not found a collision-free arm after t steps. By definition, $\mathbb{E}[M'] = M^* q_t$. Again, we define Y_t as the expected value of M', given the results of the first t rounds. Since resolving collisions for a PN can only affect players in its sub-graph, the expression $|Y_{t+1} - Y_t|$ is a constant. Thus, using Azuma's inequality for the martingale $Z_t = \mathbb{E}[Y|\mathscr{F}_t]$ we get the following concentration result:

$$\mathbb{P}(|M' - M^* q_k| > M\epsilon/2) < \frac{1}{e^{\eta_2 \epsilon^2 M}}, \tag{9.15}$$

where η_2 is another constant. It is easy to verify that the random variables M, M^*, M' satisfy the following inequalities:

$$M' \leq MZ_l \leq M' + |M^* - M|, \tag{9.16}$$

where the final inequality is due to the observation that the inability of a player to resolve its collisions may be either due to that player not having a tree-structured sub-graph or having a tree-structured sub-graph but not having resolved a collision yet.

By combining the concentration inequalities in (9.14) and (9.15), we get a new concentration inequality:

$$\begin{aligned}\mathbb{P}(|M^* - M + M - M^* q_l| > M\epsilon) &< \frac{1}{e^{\eta\epsilon^2 M}} \\ \rightarrow \mathbb{P}(|MZ_l - M^* q_l| > M\epsilon) &< \frac{1}{e^{\eta\epsilon^2 M}},\end{aligned}$$

where the last inequality is due to (9.16) and $\eta = \eta_1 + \eta_2$. We use the value $\eta = 1/(544\bar{\Lambda}^{2l-1}\bar{P}^{2l})$ for this constant, as proposed by Richardson and Urbanke [43, Theorem 2], who advised however that it does not lead to a tight bound. This completes the proof. □

A direct conclusion from Lemma 4 is that the probability that more than $\gamma' M$ players have not found a collision-free arm at iteration t is exponentially small in M. As M_t corresponds to the size of the sub-graph at time-step t, its value increases quickly with t and depends on $\bar{\Lambda}$. Thus, the right-hand side of (9.13) can be very large for small values of M. We should note that Lemma 4 is only valid when the condition $M > 2\gamma/\alpha$ is satisfied.

9.4.3 Analysis of the μ-estimation phase

To theoretically analyze the μ-estimation phase, we need to bound the probability that players have not found an ϵ-correct ranking of the arms within T_{mu} time steps. Our proof leverages results from the analysis of the MC algorithm [41, Appendix A.1.1.] and takes into account the probability of failure of the CR mechanism, derived in the preceding appendix.

The analysis by Rosenski, Shamir, and Szlak [41] showed that the number of observations of each arm by each player, denoted as C, needs to satisfy the following condition in order to ensure that an ϵ-correct ranking is computed with probability at least $1 - \delta_1$:

$$C > \ln\left(\frac{2KM}{\delta_1}\right)\frac{2}{\epsilon^2}. \tag{9.17}$$

In addition, with probability at least $1 - \delta_2$, players will collect C observations within a number of time steps:

$$\begin{aligned}T_\mu = \max\Bigg\{ & 2\frac{1}{\mathbb{E}[A_{i,j}(t)]}\ln\left(\frac{2K^2}{\delta_1}\right)\frac{2}{\epsilon^2}, \\ & \frac{1}{8\mathbb{E}[A_{i,j}(t)]}\ln\left(\frac{K^2}{\delta_2}\right)\Bigg\},\end{aligned} \tag{9.18}$$

where $A_{i,j}(t)$ denotes the expected number of observations of arm j by player i. Under CR-UCB, an arm is observed at least once in a CR round if it belongs to the sub-set of selected arms and the CR mechanism succeeds for that player. Having bounded the probability of failure of the CR mechanism in Theorem 1, we conclude that, with probability at least $1 - p_f$, a player succeeds to observe a collision-free arm. Thus, in expectation a player will fail with probability p_f. Considering that a CR round lasts for $I_{\max}$ time steps and that a player

selects a random sub-set of arms of length d by sampling the degree distribution $\Lambda(x)$, we get:

$$\mathbb{E}[A_{i,j}(t)] = \frac{(1-p_f)}{I_{\max}} \sum_{d=1}^{D} \Lambda_d \frac{d}{K} \tag{9.19}$$

9.4.4 Analysis of the M-estimation phase

We now bound the number of iterations T_M required to compute a correct estimate of the number of arms, $\hat{M}$, with high probability. We consider the estimate correct when $|\hat{M} - M| < \gamma$, with $\gamma < 0.5$. Our analysis follows the spirit of the previous MC analysis [41, Appendix A.1.2], but differs in that players compute K estimates of the probability of an arm giving a reward of 1, whereas in the MC algorithm, a single estimate of the probability of observing a collision in a fully observable setting was computed. The CR mechanism is not employed during this phase, which significantly simplifies its analysis.

As was explained in Section **??**, the probability that an arm k returns a positive reward is $p_k = \frac{1}{K}(1-\frac{1}{K})^{M-1}\mu_k$ and players estimate M using Eq. (9.5). Thus, in order for this phase to complete successfully, we require that each arm k satisfies:

$$\left| \frac{\log\left(\frac{\hat{p_k}K}{\mu_k}\right)}{\log\left(1-\frac{1}{K}\right)} - \frac{\log\left(\frac{p_kK}{\mu_k}\right)}{\log\left(1-\frac{1}{K}\right)} \right| \leq \gamma.$$

Remember that $\hat{p}_k = F_k/T_M$ is a player's estimate of p_k, where F_k is the number of times a reward of one is observed within the T_M steps of this phase.

The above is equivalent to requiring that:

$$\left| \frac{\log\left(\frac{\hat{p_k}}{p_k}\right)}{\log\left(1-\frac{1}{K}\right)} \right| \leq \gamma.$$

Let β denote the actual difference between $\hat{p}_k$ and p_k, so that $\hat{p}_k = p_k + \beta$. We can then rewrite the above as:

$$-\gamma \leq \frac{\log\left(\frac{p_k+\beta}{p_k}\right)}{\log\left(1-\frac{1}{K}\right)} \leq \gamma,$$

which after some manipulation results in

$$(p_k(1-(1-\frac{1}{K})^{-\gamma}) \leq \beta \leq (p_k(1-(1-\frac{1}{K})^{\gamma}). \tag{9.20}$$

We can, therefore, combine the K inequalities into a single one by summing each corresponding part:

$$\sum_{k\in\mathscr{K}} \left(p_k(1-(1-\frac{1}{K})^{-\gamma} \right) \leq \beta K \leq \sum_{k\in\mathscr{K}} \left(p_k(1-(1-\frac{1}{K})^{\gamma} \right).$$

If we set $p_k = \frac{1}{K}(1-\frac{1}{K})^{M-1}\mu_k$ and request that $\gamma = 0.49$ and $|\hat{p}_k - p_k| \leq \xi$, we get the following value for the error ϵ:

$$\xi = \min\left\{ \left| \left(\sum_{k\in\mathcal{K}} \mu_k \frac{1}{K}(1-\frac{1}{K})^{M-1}(1-(1-\frac{1}{K})^{-\gamma}) \right) \right|, \right.$$
$$\left. \left| \sum_{k\in\mathcal{K}} \mu_k \frac{1}{K}\left(1-\frac{1}{K}\right)^{M-1}\left(1-\left(1-\frac{1}{K}\right)^{\gamma}\right) \right| \right\}. \tag{9.21}$$

By studying the monotonicity of $\ln(1-x)+x$, we can show that $(1-\frac{1}{K})^{K-1} \geq \frac{1}{e}$. Also, as was previously shown [41], $(1-(1-\frac{1}{K})^{-\gamma}) \geq \frac{0.49}{K}$.

Thus, the first term in (9.21) gives

$$\xi_1 \geq \sum_{k\in\mathcal{K}} \mu_k \frac{0.49}{eK^2} \geq \frac{0.1K\mu_{\min}}{K^2} \geq \frac{0.1\mu_{\min}}{K},$$

where $\mu_{\min}$ is the smallest μ_k among all arms. The second term in (9.21) leads to the same bound.

Finally, we can bound T_M using Hoeffding's inequality: with probability at least $1-\delta_3$, $|\hat{p}_k(T_M)-p| \leq \xi$ provided that $T_M \geq \frac{\log(2/\delta_3)}{2\epsilon^2} \geq \frac{\log(2/\delta_3)}{2\left(\frac{0.49}{K}\right)^2} = \frac{\log(2K^2/\delta_3)}{0.4802}$.

9.4.5 Optimizing the degree distribution

This section describes how the degree distribution $\Lambda(x)$ can be optimized in asymptotic settings for a given target load M/K so as to satisfy the stability condition of the CR mechanism, presented in Lemma 2.

The main idea is to use Eq. (9.11) in order to compute an upper bound on the maximum load, L^*, i.e. ratio of players to arms, below which a given $\Lambda(x)$ achieves collision resolution with high probability, and then search for the degree distribution that has the highest L^*. Deriving the bound L^* requires that we analyze the degree distribution from the perspective of ANs ($P(x)$) and edges connected to ANs ($P(x)$). The average number of collisions per arm is, thus, $\bar{P}$ and the probability that a player pulls a certain arm is $\bar{P}/m$. Thus, the probability that an AN has degree l is given by:

$$P_l = \binom{M}{l}\left(\frac{\bar{P}}{M}\right)^l\left(1-\frac{\bar{P}}{M}\right)^{M-l}.$$

The node-perspective AN degree distribution has the form:

$$P(x) = \sum_l P_l x^l = \left(1-\frac{\bar{P}}{M}(1-x)\right)^M.$$

If we let $M \to \infty$, then we observe that the number of collisions follows a Poisson distribution ($P_l = \frac{1}{l!}(L\bar{\Lambda})^l e^{-L\bar{\Lambda}}$). Thus, the preceding equation becomes $P(x) = e^{-\bar{P}(1-x)} = e^{-L\bar{\Lambda}(1-x)}$.

The edge-perspective distribution for ANs is:

$$P(x) = \frac{\Psi'(x)}{\bar{P}} = e^{-L\bar{\Lambda}(1-x)}.$$

By replacing the above equation into (9.11), the threshold L^* can be defined as the maximum value of L that satisfies the following condition:

$$q > \lambda(1 - e^{-qL\bar{\Lambda}}),\ \forall q \in (0, 1].$$

Define now $f(q) \triangleq \lambda(1 - e^{-qL\bar{\Lambda}})$. If the derivative of this function is a contraction for $q \to 0$, then for $L \leq L^*$ the above condition becomes true. Since $f'(0) = \lambda'(0)\bar{\Lambda}L = \lambda_2\Lambda'L$, this leads to the following bound:

$$L^* \leq \frac{1}{\lambda_2\bar{\Lambda}}.$$

We can, thus, choose $\Lambda(x)$ by solving the following linear program:

$$\begin{aligned} &\text{Maximize } \frac{1}{\lambda_2\Lambda'(1)} \\ &\text{subject to } \sum_l \lambda_l = 1, \sum_l \Lambda_l = 1, \lambda_2 = \frac{\Lambda_2 2}{\sum_l \Lambda_l l} \end{aligned}$$

where $\lambda_2, \Lambda_l \in [0, 1]$.

Solving this optimization problem will ensure that the CR mechanism succeeds almost surely for any load $L \in [0, 1)$ as $M \to \infty$.

Optimization is traditionally performed using differential evolution. Various solutions were presented by **IRSA**, including the degree distribution $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ with $L^* = 0.938$. To the best of our knowledge, the best-performing solution in the literature has $L^* = 0.965$ [**IRSA**] It is known that, if we allow a sufficiently large value for D, the maximum number of pulls, we can get solutions arbitrarily close to 1. However, Lemma 4 introduces a trade-off, as increasing D increases the value of γ, leading to a larger deviation from asymptotic performance for finite M. Thus, for small problem settings D should be kept sufficiently low.

9.5 Algorithm for dynamic settings

In the following, we describe a dynamic version of CR-UCB, DYN-CR-UCB, for problem settings with players arriving at different time steps $\tau_m \in \{0, \ldots, T-1\}$, where τ_m is unknown to all. We denote the learning horizon of player m by T_m. A player knows the time elapsed since joining the network and observes a common clock with period $I_{\max}$ and can be in one of the two phases: (i) in the exploration phase, the player is employing the CR mechanism, as it was described in Section 9.3.2, and experiences CR rounds of equal duration $I_{\max}$; (ii) in the exploitation phase, the player is pulling a single arm until the end of the horizon.

During the exploration phase, a player computes unbiased estimates $\hat{\mu}_k$ for all arms and a confidence bound $B_t = 2\sqrt{\frac{\log(T_m)}{t}}$. Thus, it knows that the true mean of an arm lies

with high certainty in the range $[\hat{\mu}_k - B_k, \hat{\mu}_k + B_k]$. The player keeps an initially empty *preferences* list, $\vec{p}$ and inserts an arm in it once it detects that its lower bound is higher than the upper bound of all other arms. The player will exploit an arm in $\vec{p}$ as soon as it gives a positive reward. As each player employs the UCB algorithm with confidence bound $B_k(t) = \sqrt{\frac{\log(T_m)}{T_k}}$, using Hoeffding's inequality, we can prove that:

$$\mathbb{P}[|\hat{\mu}_k - \mu_k| > B_k] \leq 4/T_m^2, \tag{9.22}$$

which suggests that all players will acquire a correct estimate of all free arms at the end of their individual horizon T_m with probability $1 - \mathcal{O}(1/(T_m)^2)$. In addition, we know that a sub-optimal arm is detected within $K \log T_m / \Delta_k^2$ time steps, where $\Delta_k = \min_{i=1,\cdots,k} |\mu_i - \mu_{i+1}|$ indicates the difficulty of ranking an arm.

An important element of DYN-CR-UCB is how the detection of exploited arms happens. In contrast to DYN-MMAB, where players sample arms randomly and cannot discern between an occupied and an unavailable arm, players in our setting are employing the CR mechanism. Due to this, they know that, at the end of a CR round, at least one arm will be observed as available, provided that the CR round completes successfully. Thus, the probability of an arm being unavailable even if it is not exploited is equal to p_f and is, thus, independent of its mean availability, which significantly reduces the sample complexity.

The number of consecutive rounds of observing no rewards from an arm required to declare it as occupied, L, needs to be high enough to guarantee that it is not falsely detected as occupied and low enough to ensure that detection does not incur unnecessary regret. By setting $L \geq 2 \log T_m / (1 - p_f)$, we ensure that the probability of observing L successive rounds with all-zero rewards is smaller than $\frac{1}{(T_m)^2}$, due to the inequality $(1 - p_\mathrm{f})^L \leq e^{-L(1-p_f)}$. In order to prove that a player will pull an arm L times with probability $1/T_m^2$, we make use of the Hoeffding bound of the binomial distribution which takes the value one with probability equal to the probability of sampling arm k, denoted as $p_k = \sum_{l=0}^{d} \Lambda_d l / K$. This leads to $L_2 = \frac{L p_k \pm L^2((p_k - 1)p_f + \log T_m)}{p_k - \log T_m}$. Thus, if a player occupies an arm at time step $t_0 + \tau_j$, then it is correctly detected as occupied within $\mathcal{O}(I_{\max} L_2) + \tau_j$ steps, where we have taken into

account that a round lasts for $I_{\max}$ iterations.

Algorithm 4: DYN-CR-UCB

Data: $\mathcal{K}, \Lambda, I_{\max}, T_m$

1 Initialize $p = 0$, *occupied* = [], *preferences*= [], *unresolved* = [], phase="explore", L2 as in Eq. (9.23)
2 **while** *phase ==“explore”* **do**
3 free, k = CR Round($\mathcal{K}, \Lambda, I_{\max}$)
4 $\mu_k = S_k/T_k$
5 $B = 2\sqrt{\frac{\log T_m}{t}}$
6 **if** $k ==$ *preferences*$[p]$ *and free* **then**
7 phase = “exploit”
8 **end**
9 **if** *preferences*$[p] \in$ *occupied* **then**
10 $p = p + 1$
11 **end**
12 **if** *not free* **then**
13 **if** k *not in unresolved* **then**
14 Insert k to *unresolved* ▷ arm is potentially occupied
15 $C_k = 1$
16 **else**
17 $C_k = C_k + 1$
18 **if** $C_k > L_2$ **then**
19 Insert k to *occupied* ▷ arm is certainly occupied
20 **end**
21 **else**
22 **if** k *in unresolved* **then**
23 $C_k = 0$ ▷ arm is certainly not occupied
24 **end**
25 **if** $\exists i, \mu_{min}[i] > \mu_{max}[k]$ $\forall k$ *not in preferences and occupied* **then**
26 Insert k to *Preferences*
27 **end**
28 **if** $\exists i$ *not in preferences*$[1:p]$ *such that* $\mu_{min}[i] > \mu_{max}[$*preferences*$[p]]$ **then**
29 Insert *preferences*$[p]$ to *occupied*
30 **end**
31 **end**
32 Pull k until T^m ▷ Exploitation phase

The pseudocode of DYN-CR-UCB is presented in Algorithm 4. In addition to the *preferences* list that a player updates when high quality arms are detected (Lines 27-29) and checks to find an arm to exploit (Lines 6-8), a player also updates an *unresolved* list, with arms that have given only zero consecutive rewards and are potentially being exploited by other players. If an arm remains in this list for more than L_2 time steps, it is transferred to the *occupied* list. By making use of Lemma 10 presented by **sic˙mmab** and the preceding discussion we derive the following regret bound for DYN-CR-UCB:

Theorem 1. In the dynamic setting, the regret of DYN-CR-UCB is upper bounded as follows:

$$\mathbb{E}[R_T] \leq \frac{MK \log T}{\Delta^2(M)} + MI_{\max}L_2$$

, where $\bar{\Delta}^2(M) = \min_{i=1,\ldots,M} |\mu_i - \mu_{i+1}|$, $p_k = \sum_{l=0}^{d} \Lambda_d l/K$ and

$$L_2 = \min\Big\{\Big(\frac{Lp_k + L^2((p_k-1)p_f + \log T_m)}{p_k - \log T_m}, \\ \frac{Lp_k - L^2((p_k-1)p_f + \log T_m)}{p_k - \log T_m}\Big)\Big\} \tag{9.23}$$

9.6 Simulations

Simulations in this section aim at empirically evaluating the performance of CR-UCB. We first examine the ability of the CR mechanism to successfully resolve collisions and, then, evaluate the regret achieved by our proposed algorithms, CR-UCB and DYN-CR-UCB. We also compare the performance of our solution with other state-of-the-art algorithms in the related literature.

We initially study problem settings where the number of players is $M = 200$. As was shown in [29], this is the minimum value for M for which the performance predicted by the asymptotic analysis of the CR mechanism is approximated closely by simulations for finite values of M. We employ the degree distribution $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$, which was found to lead to successful collision resolution with high probability in [29].

9.6.1 Evaluation of the collision resolution mechanism

As was described in Section 9.3.2, CR rounds take place during the μ-estimation phase. Each round consists of a fixed number of $I_{\max}$ iterations, where each player initially chooses a random subset of the K arms of size d, where d is determined by sampling the degree distribution $\Lambda(x)$, and repeatedly pulls the arms in the subset until they observe a reward of one from one of them. From the perspective of a single player, a round has failed if no reward of one is observed during its duration, as this means that the player has not found a collision-free arm. During the μ-estimation phase, this has the negative effect that the player cannot update their estimate of the mean.

Our theoretical analysis has proven that the probability of failure is negligible when $M \to \infty$. We now examine practical scenarios, where M is finite to confirm that the probability of failure remains low. In the following simulations, we measure the number of iterations that each player required to find a collision-free arm within a round during the μ-estimation phase. This evaluation also provides an answer to the following design question: *what should the value of I_{max} be in order to guarantee collision resolution with high probability?*

Figure 9.6 presents the histogram of failures of the CR mechanisms in different problem settings. During simulations with $T = T_\mu = 10000$, we compute the number of iterations

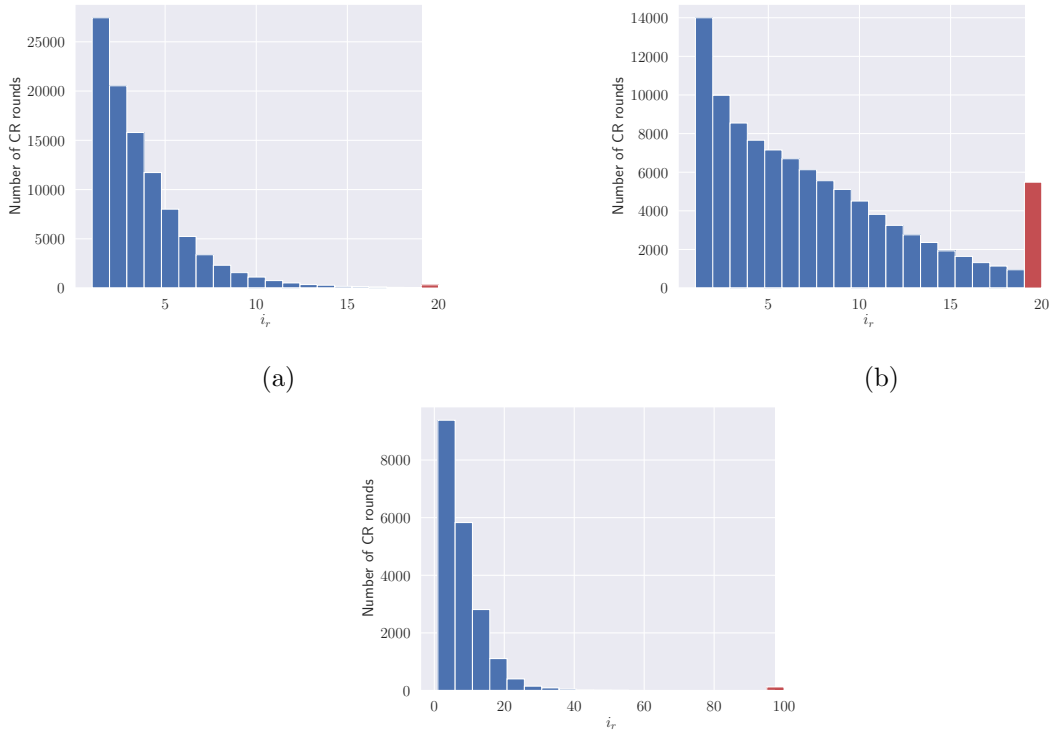


Figure 9.6: Evaluation of the CR mechanism using the resolution time i_r for three problem settings: (a) $M = 200, K = 400, I_{\max} = 20$, (b) $M = 200, K = 280, I_{\max} = 20$, (c) $M = 200, K = 280, I_{\max} = 100$

each player needs to find a collision-free arm in a round. We denote this random variable as i_r, with $\bar{i_r}$ being its mean value, and refer to it as the resolution time. In Figure 9.6(a), we consider a setting with $M = 200, K = 400, I_{\max} = 20$ and present the histogram and mean value of resolution times. The bar that corresponds to $i_r = I_{\max}$, highlighted in red, represents the number of failed trials. We observe that players need on average 4 iterations before finding a collision-free arm and that the number of failed trials is negligible. In Figure 9.6(b), we increase the load L from 0.5 to 0.7 and keep $I_{\max}$ the same. We observe that the value of $\bar{i_r}$ increases to 7.1 and that a considerable number of rounds for each player have failed. In Figure 9.6(b), we keep the problem setting the same and increase $I_{\max}$ to 100, in order to allow more time to the CR mechanism to resole collisions. We observe that by doing so, the number of failed trials is again negligible and that i_r has a mean value of 8. This behaviour is anticipated: the higher the load, the higher the probability of experiencing more collisions at the beginning of a round and, therefore, the more iterations player will need to discover collision-free arms. While $\bar{i_r}$ is low across problem settings, it is important to keep $I_{\max}$ high enough to ensure that the CR mechanism succeeds with high probability.

9.6.2 Evaluation of regret

In this section, we evaluate the regret, as defined in Eq. (9.2), achieved by different MMAB algorithms. We study both static and dynamic networks. As our work concerns the no-

sensing setting, we compare our algorithms with DYN-MMAB [42], a state-of-the-art algorithm that can be employed in both static and dynamic settings. As we explained in Chapter 5, the algorithm proposed in [105] is also applicable, but requires prohibitively long learning times. We also evaluate the MC algorithm [41], which requires sensing information, with a two-fold objective: (i) visualizing how much harder is the no-sensing from the sensing setting; (ii) evaluating how much the CR mechanism in our algorithms helps deal with the absence of sensing information.

Static networks

In Figure 9.7 we evaluate CR-UCB, DYN-CR-UCB, DYN-MMAB and MC in a network with $M = 5$ players and $K = 10$ arms. The true means μ_k are randomly sampled in $[0.1, 1]$ and the minimum distance between them is 0.001. For CR-UCB, we determine the values of $T_\mu, T_M, T_{\text{MC}}$ based on Theorem 2 and employ a horizon of length $T_\mu + T_M + T_{\text{MC}}$ For DYN-CR-UCB, we determine L_2, the number of rounds required to detect an arm as occupied, based on our discussion in Section 9.5. For DYN-MMAB and MC, we equivalently define the hyper-parameters as these were described in [42] and [41]. We observe that DYN-MMAB requires the longest time to learn the optimal assignment and, thus, accrues the largest regret. The MC algorithm is the quickest to converge. This is not surprising, as, in contrast to all other three algorithms, MC can observe collisions. We observe that our proposed algorithms, CR-UCB and DYN-CR-UCB, accrue comparable regret, with CR-UCB converging slightly sooner. This is anticipated, as CR-UCB has been specifically designed for static settings. Although it converges quicker, CR-UCB accrues larger regret than DYN-CR-UCB, as all players enter the exploitation phase simultaneously. In contrast, players employing DYN-CR-UCB stop exploring at different time steps, facilitating the exploration of the remaining active players.

Dynamic networks

We now examine dynamic networks, where players arrive at different time steps. In this setting, the regret bound of algorithms designed for dynamic settings, such as CR-UCB and MC, are no longer valid. In Figure 9.8, we compare the performance of our proposed algorithm, DYN-CR-UCB, and DYN-MMAB [42] in a network with $K = 10$ arms, where 5 players arrive at the distinct time steps $[1, 5 \cdot 10^5, 2.437.500, 4.375.000, 6.312.500]$. All players have a horizon of $T_m = 10^6$ and remain in the network until the end of the experiment. Thus, the last player to arrive has the shortest time to budget to find an arm to exploit (5312.500 time steps), which is however long enough according on the regret bound of each algorithm.

We observe that players using our proposed algorithm DYN-CR-UCB, find an optimal arm significantly quicker than players employing DYN-MMAB. In particular, by the time the final player arrives ($t = 6.312.500$), the existing players have already entered their exploitation phase. Players employing DYN-MMAB accrue about ten times the regret accrued by our algorithm and converge to the optimal assignment close to 410^5 time steps later.

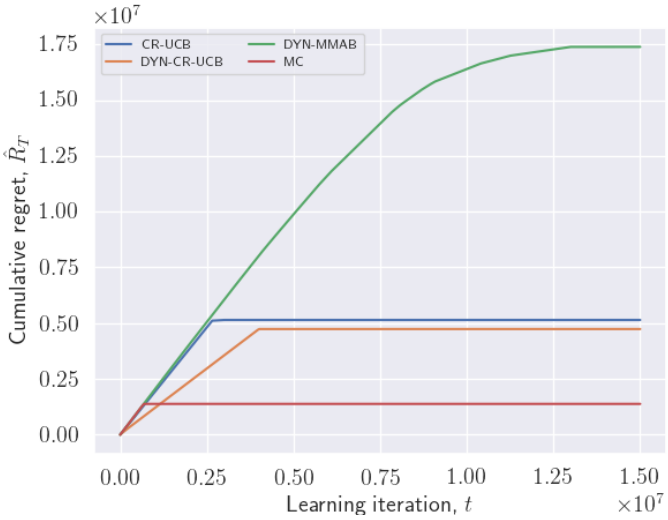


Figure 9.7: Cumulative regret achieved by different bandit algorithms in a network with $K = 10$ arms and $M = 5$ devices. Our proposed algorithms, CR-UCB and DYN-CR-UCB are compared against the DYN-MMAB algorithm [42] and the MC algorithm [41]. In contrast to all other algorithms, the latter assumes that players can observe collisions.

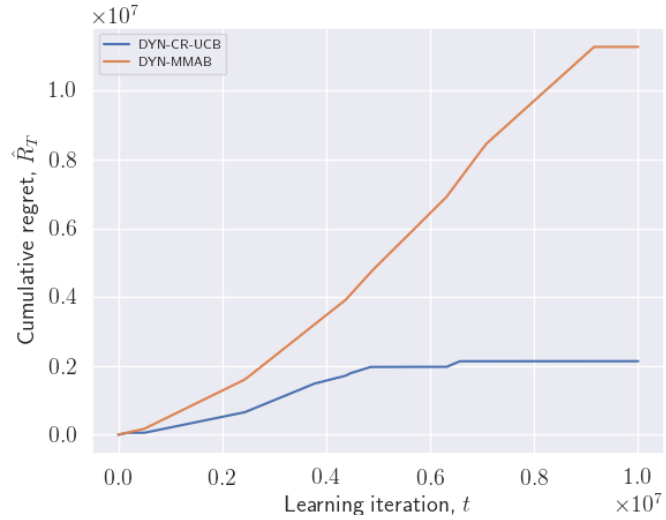


Figure 9.8: Cumulative regret achieved by DYN-CR-UCB and DYN-MMAB in a dynamic network with $K = 10$ arms and $M = 5$ devices arriving at time steps $[2 \cdot 10^6, 9.75 \cdot 10^6, 1.75 \cdot 10^7, 2.525 \cdot 10^7]$.

9.7 Conclusions - Towards scalable and optimal DSA

In this chapter, we presented solutions for Dynamic Spectrum Access in the framework of Multi-player Multi-armed Bandits. We first proposed the CR-UCB algorithm, which achieves state-of-the-art performance in static networks, where devices enter the network simultaneously. We, then, examined dynamic networks and proposed DYN-CR-UCB, which also improves upon the state of the art. We evaluated our algorithms both theoretically, by deriving an upper regret bound and, empirically, by simulating networks of various sizes and loads.

Our work is the first one to introduce a novel collision resolution mechanism in DSA. An important property of this mechanism is that, as indicated by the regret bound, its performance improves with the size of the network. Thus, our algorithms are *scalable*, in contrast to previous algorithms that were designed and evaluated having small networks in mind.

An important existing question in MMABs is the following: *"how much more difficult is the decentralized setting than the centralized one?"* Answering this question requires computing a lower regret bound for MMABs and will enable the design of an optimal solution for DSA. As is also explained in [42], analyzing regret in MMABs is more complex than in single-player settings, as regret can be dominated by factors such as M and K, even if it is sub-linear in the problem horizon. While multi-player bandits are a relatively young area, the study of optimality of decentralized MA protocols has a long-standing history. We, therefore, believe that tools developed in the area of MA and, more generally, in the analysis of bipartite graphs, can shed light into this question.

Chapter 10

Conclusion and future work

10.1 Conclusion on our contributions

We started this thesis with a broad historical and technical discussion on the needs and opportunities in the development of wireless networks. We analyzed the challenges that reinforcement learning algorithms face when employed in the optimization of wireless resource-constrained networks and motivated our main research question: *"can we design reinforcement learning algorithms that achieve a fine balance between optimality and complexity for contemporary resource-constrained networks?"* We, then, provided background on bipartite graphs and their use in analyzing and optimizing Multiple Access. Prior to presenting our reinforcement-learning based solutions, in Chapters 7, 8 and 9, we also provided a description of the reinforcement learning frameworks employed in our work, namely Markov Decision Processes and Multi-armed Bandits. Our exploration of this field was targeted, as we began with general tools considered fundamental in these frameworks and, then, delved deeper into theoretical challenges and solutions particularly associated with decentralized resource-constrained systems. We have accompanied Chapters 7, 8 and 9 with a dedicated discussion on their contributions. We now venture into a deeper analysis of our contributions and highlight the common objectives underlying them.

Our main contribution is the design of reinforcement learning algorithms for MA for rendering resource allocation in wireless networks adaptive. In particular, our algorithm based on independent Q-learning for optimizing time access using the IRSA protocol can be used when low complexity has the highest priority, while its counterpart, which uses coordinated Q-learning, can be used to achieve a desired balance between complexity and optimality. CR-UCB and DYN-CR-UCB, our proposed bandit algorithms, are low-complexity solutions with optimality guarantees for optimizing channel access. Moreover, the design process itself, and the accompanying discussion on the trade-offs that guided our design choices, can find a more general applicability in the optimization of resource-constrained systems using machine learning.

The approach that we advocate is, thus, grounding reinforcement learning algorithms in the needs of resource-constrained networks in order to achieve a fine balance between optimality and complexity of practical solutions. We propose the use of algorithms that make minimum requirements on the software and hardware capabilities of agents, in order to ensure

that our learning-based solutions have a wide applicability. A large part of the theoretical analysis of our algorithms is devoted to deriving optimality and complexity, primarily in terms of learning time, guarantees. We believe that this is a constructive attitude towards successfully embedding learning-based solutions in communication networks. Quality of service, enforced through minimum requirements on performance measures such as reliability, speed and lifetime, is becoming increasingly important in future communication standards. It would, therefore, be unrealistic to expect that learning-based solutions will become a norm, unless they are accompanied by a rigorous understanding of their guarantees.

In addition to ensuring optimality, theoretically analyzing an algorithmic solution can lead to a better understanding of the difficulties and opportunities arising in a problem setting. An example of a difficulty in our work is the mobility of devices in spectrum access. As we saw in Chapter 9, algorithms designed assuming the absence of mobility cannot solve the problem optimally, while our dynamic algorithm requires longer learning time to find the optimal solution. Thus, an algorithm faces a trade-off between being efficient and solving tougher settings. Locality of interaction on the other hand, was an example of an opportunity for reducing the complexity of a learning task while maintaining optimality guarantees. As we saw in Chapter 8, resource allocation problems often exhibit dependencies that cannot be ignored when searching for an optimal solution. However, leveraging the structure of dependencies to restrict communication at a minimum can bring large benefits compared to a centralized solution.

From an application point of view, we have provided realistic simulations of our proposed solutions for time slot and channel access in resource-constrained networks. We have evaluated our algorithms on a large range of network sizes and loads and examined how the requirements in convergence time scale with the size of the problem. In particular, our study of multi-player bandit algorithms is the first to introduce scalability in terms of the size of the network as an important concern and opportunity in DSA. The CR mechanism, introduced in our work, which can be seen as a generalization of the SIC mechanism to settings where resources are not always available, helped us design algorithms that exhibit performance close to the one achieved by algorithms that assume sensing information.

10.2 Future work

Through our work in the various Chapters of this thesis we identified the following possible directions for future research.

On multiple access in wireless networks MA is a long-standing problem in wireless networks and, yet, not adequately solved. This is arguably due to the constant shift of the needs of these networks towards better quality of service and the development of new types of devices and applications, which make the design of MA solutions a moving target problem. Recent forecasts estimate that about 28 billion smart devices will be connected across the global world by 2021 [135], making ad hoc networks a considerable part of the communication ecosystem. Based on the extensive review on past and current MA solutions performed in this thesis and the conclusions reached through the design of adaptive protocols, we advocate random access, as an efficient, flexible and low-complexity mechanism for resource

allocation. We believe that the focus of future protocols will be on performance guarantees and self-configuration. The reinforcement learning-based protocols proposed in this work for optimizing slot and channel access are an important step towards this direction. Based on our analysis of multi-player bandits in the no sensing setting, we believe that an important research question for future solutions in DSA will be the following: *"what is the minimum time required to find the optimal assignment of users to channels in a decentralized manner?"*. Answering this question will lead us to finding an algorithm that is optimal for DSA and can be tackled by computing a lower regret bound for our setting, in addition to the upper bounds computed in this thesis.

On multi-player bandits As this field has only recently started being intensively investigated by the bandit community, it is still characterized by major open questions, which are mostly related to the difficulty of the learning task: *"how should an optimal algorithm scale with the number of players?", "What is the minimum bound on regret and how does it depend on whether players attempt to communicate with each other?", "Is the multi-player setting inherently more difficult than the single-player setting?"* As such, we believe that the near future will bring new algorithms and theoretical results in this area, at a benefit of applications similar to DSA. In particular, we believe that implicit coordination mechanisms for multi-player bandits can reveal opportunities in providing more efficient solutions. Recent examples are the work in [42], which proved that implicit communication is possible if players are synchronized and intentionally pull arms to share information, and our proposed CR mechanism in Chapter 9, which revealed that guaranteeing collision-free learning is possible even in the absence of sensing information.

On online model-free learning Model-free RL algorithms have found wider applicability than their model-based counterparts and, combined with function approximation, have become the norm in solving complex tasks, such as Atari and RPG games. Recent years have seen a departure from the DQN algorithm [94], to algorithms employing policy gradients, such the Proximal Policy Optimization algorithm [136], which are more stable and intuitive to analyze theoretically. The need for training neural networks as part of solving an RL task has led to a new research frontier: *"how can we make RL algorithms less data hungry?"*. We believe that the efforts of the RL community will move along the following research axis: (i) hybrid model-based/model-free solutions. Although learning a model of the environment may entail prohibitive complexity, it is often the case that a part of the environmental dynamics is known. Particularly in real-world systems, incorporating prior knowledge about the physical properties of components can speed up learning significantly. Virtual experience, the technique that we employed in Chapter 7 and analyzed theoretically, is an example of a way to leverage knowledge about the state-action transition dynamics to augment the training data without increasing the learning time; (ii) improving the efficiency of exploration. The vast majority of deep RL algorithms address the problem of exploration using heuristics, such as the ϵ-greedy technique. This tradition arose at times when the tasks solved by RL were simple enough for learning complexity to not be a concern. Considering the complexity of contemporary RL tasks, sample efficiency has become an important concern. For this reason, solutions developed in the multi-armed bandit framework, a field

whose primary interest is exploration, are becoming increasingly relevant to the work of the MDP community. As such, online learning and the framework of bandits holds great promise for the future of reinforcement learning.

On multi-agent reinforcement learning for decentralized networks Real-world systems that are physically decentralized are a promising application for RL algorithms. Examples include communication networks, studied in this thesis, energy markets, smart factories and teams of autonomous robots. Multi-agent reinforcement learning has, therefore, attracted significant interest from the AI community, with efforts focusing on designing algorithms with low complexity and stable training dynamics. Designers of such systems need to address the following question: *"how will the agents coordinate with each other in order to pursue the team objective?"*. We believe that exploiting the structure of the system and problem is a promising way to designing multi-agent RL algorithms. In addition to the Coordination Graphs, employed in our work, recent works in MARL are turning towards emergent cooperation, where the structure is not known in advance but discovered in the process of learning to solve the RL task [137]. Regardless of the coordination scheme, we believe that techniques for reducing the complexity introduced by coordination, as the one based on a multiple knapsack proposed in Chapter 8, will play an important role in the adoption of learning-based solutions in real-world applications.

Bibliography

[1] W. S. H. M. W. Ahmad et al. "5G Technology: Towards Dynamic Spectrum Sharing Using Cognitive Radio Networks". In: *IEEE Access* 8 (2020), pp. 14460–14488.

[2] P. Yang et al. "6G Wireless Communications: Vision and Potential Techniques". In: *IEEE Network* 33.4 (2019), pp. 70–75.

[3] Marconi G. "Wireless Telegraphic Communication: Nobel Lecture, 11 December 1909". In: *Nobel Lectures. Physics 1901-1921* 1967:196-222 (Dec. 1909), p. 198.

[4] M. Schwartz and N. Abramson. "The Alohanet - surfing for wireless data [History of Communications]". In: *IEEE Communications Magazine* 47.12 (2009), pp. 21–25.

[5] P. V. Dudhe et al. "Internet of Things (IOT): An overview and its applications". In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. 2017, pp. 2650–2653.

[6] B. Ji et al. "Survey on the Internet of Vehicles: Network Architectures and Applications". In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41.

[7] G. Aceto, V. Persico, and A. Pescapé. "A Survey on Information and Communication Technologies for Industry 4.0: State-of-the-Art, Taxonomies, Perspectives, and Challenges". In: *IEEE Communications Surveys Tutorials* 21.4 (2019), pp. 3467–3501.

[8] R. Baldemair et al. "Future Wireless Communications". In: *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*. 2013, pp. 1–5.

[9] Chee-Yee Chong and S. P. Kumar. "Sensor networks: evolution, opportunities, and challenges". In: *Proceedings of the IEEE* 91.8 (2003), pp. 1247–1256.

[10] C.R. Srinivasan et al. "A review on the different types of internet of things (IoT)". In: *Journal of Advanced Research in Dynamical and Control Systems* 11 (Jan. 2019), pp. 154–158.

[11] C.K. -K Toh. *Ad Hoc Wireless Networks: Protocols and Systems*. 1st. USA: Prentice Hall PTR, 2001. ISBN: 0130078174.

[12] Shangguang Wang et al. "An Overview of Internet of Vehicles". In: *Communications, China* 11 (Oct. 2014), pp. 1–15.

[13] G. A. Akpakwu et al. "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges". In: *IEEE Access* 6 (2018), pp. 3619–3647.

[14] LoRa Alliance Technical Committee. *LoRaWAN specification v1.0*. Tech. rep. LoRa Alliance, Jan. 2015.

[15] "IEEE Standard for Low-Rate Wireless Networks". In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709. DOI: 10.1109/IEEESTD.2016.7460875.

[16] A. M. Abdullah. "Wireless lan medium access control (mac) and physical layer (phy) specifications". In: 1997.

[17] P. McDermott-Wells. "What is Bluetooth?" In: *IEEE Potentials* 23.5 (2005), pp. 33–35. DOI: 10.1109/MP.2005.1368913.

[18] S. Safaric and K. Malaric. "ZigBee wireless standard". In: *Proceedings ELMAR 2006.* 2006, pp. 259–262. DOI: 10.1109/ELMAR.2006.329562.

[19] Priyanka Rawat et al. "Wireless sensor networks: a survey on recent developments and potential synergies". en. In: *The Journal of Supercomputing* 68.1 (Apr. 2014), pp. 1–48. ISSN: 0920-8542, 1573-0484. DOI: 10.1007/s11227-013-1021-9. URL: http://link.springer.com/10.1007/s11227-013-1021-9 (visited on 01/31/2021).

[20] S. Haykin. "Cognitive radio: brain-empowered wireless communications". In: *IEEE Journal on Selected Areas in Communications* 23.2 (2005), pp. 201–220.

[21] K. A. Yau et al. "Cognition-Inspired 5G Cellular Networks: A Review and the Road Ahead". In: *IEEE Access* 6 (2018), pp. 35072–35090.

[22] J. Mitola. "Software Radios. Survey, Critical Evaluation and Future Directions". In: *IEEE National Telesystems Conference* (1992), pp. 13–15.

[23] J. Mitola and G. Q. Maguire. "Cognitive radio: making software radios more personal". In: *IEEE Personal Communications* 6.4 (1999), pp. 13–18.

[24] M. Hadded et al. "TDMA-Based MAC Protocols for Vehicular Ad Hoc Networks: A Survey, Qualitative Analysis, and Open Research Issues". In: *IEEE Communications Surveys & Tutorials* 17.4 (June 2015), pp. 2461–2492.

[25] I. Demirkol, C. Ersoy, and F. Alagoz. "MAC protocols for wireless sensor networks: a survey". In: *IEEE Communications Magazine* 44.4 (2006), pp. 115–121.

[26] V. Cionca, T. Newe, and V. Dadârlat. "TDMA Protocol Requirements for Wireless Sensor Networks". In: *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008).* 2008, pp. 30–35.

[27] Vijay K. Garg and Yih-Chen Wang. "7 - Wireless Network Access Technologies". In: *The Electrical Engineering Handbook.* Ed. by WAI-KAI CHEN. Burlington: Academic Press, 2005, pp. 1005–1009. ISBN: 978-0-12-170960-0.

[28] Wei Ye, John Heidemann, and Deborah Estrin. "An energy-efficient MAC protocol for wireless sensor networks". In: *Proceedings - IEEE INFOCOM* 3 (Jan. 2002), 1567–1576 vol.3.

[29] G. Liva. "Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA". In: *IEEE Trans. on Communications* 59.2 (Feb. 2011), pp. 477–487.

[30] Q. Zhao and B. M. Sadler. "A Survey of Dynamic Spectrum Access". In: *IEEE Signal Processing Magazine* 24.3 (2007), pp. 79–89.

[31] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.

[32] Aleksandrs Slivkins. "Introduction to Multi-Armed Bandits". In: *Foundations and Trends® in Machine Learning* 12.1-2 (2019), pp. 1–286. ISSN: 1935-8237.

[33] Eleni Nisioti and Nikolaos Thomos. "Decentralized Reinforcement Learning Based MAC Optimization". In: *IEEE Proc. of PIMRC 2018, Bologna, Italy*. Sept. 2018.

[34] E. Nisioti and N. Thomos. "Fast Q-Learning for Improved Finite Length Performance of Irregular Repetition Slotted ALOHA". In: *IEEE Transactions on Cognitive Communications and Networking* 6.2 (2020), pp. 844–857.

[35] E. Nisioti and N. Thomos. "Robust Coordinated Reinforcement Learning for MAC Design in Sensor Networks". In: *IEEE Journal on Selected Areas in Communications* 37.10 (2019), pp. 2211–2224.

[36] Caroline Claus and Craig Boutilier. "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems". In: *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. AAAI '98/IAAI '98. Madison, Wisconsin, USA: American Association for Artificial Intelligence, 1998, pp. 746–752. ISBN: 0262510987.

[37] N. Mastronarde and M. van der Schaar. "Fast Reinforcement Learning for Energy-Efficient Wireless Communication". In: *IEEE Trans. on Signal Processing* 59.12 (Dec. 2011), pp. 6262–6266.

[38] Carlos Guestrin, Michail G. Lagoudakis, and Ronald Parr. "Coordinated Reinforcement Learning". In: *Proceedings of the Nineteenth International Conference on Machine Learning*. ICML '02. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, pp. 227–234. ISBN: 1558608737.

[39] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero. " Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks". In: *IEEE Trans. on Wireless Communications* 6.4 (Apr. 2007), pp. 1408–1419.

[40] Lilian Besson and Emilie Kaufmann. "Multi-Player Bandits Revisited". In: (2017). arXiv: `1711.02317 [stat.ML]`.

[41] Jonathan Rosenski, Ohad Shamir, and Liran Szlak. "Multi-Player Bandits: A Musical Chairs Approach". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 155–163.

[42] Etienne Boursier and Vianney Perchet. "SIC-MMAB: Synchronisation Involves Communication in Multiplayer Multi-Armed Bandits". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 12071–12080.

[43] T. J. Richardson and R. L. Urbanke. "The capacity of low-density parity-check codes under message-passing decoding". In: *IEEE Transactions on Information Theory* 47.2 (Feb. 2001), pp. 599–618. ISSN: 1557-9654.

[44] A. Ephremides and B. Hajek. "Information Theory and Communication Networks: An Unconsummated Union". In: *IEEE Trans. Inf. Theor.* 44.6 (Sept. 2006), pp. 2416–2434. ISSN: 0018-9448. DOI: 10.1109/18.720543. URL: https://doi.org/10.1109/18.720543.

[45] E. Paolini, G. Liva, and M. Chiani. "Coded Slotted ALOHA: A Graph-Based Method for Uncoordinated Multiple Access". In: *IEEE Transactions on Information Theory* 61.12 (2015), pp. 6815–6832.

[46] L. Toni and P. Frossard. "Prioritized Random MAC Optimization Via Graph-Based Analysis". In: *IEEE Trans. on Communications* 63.12 (Dec. 2015), pp. 5002–5013.

[47] Michael G. Luby, Michael Mitzenmacher, and M. Amin Shokrollahi. "Analysis of Random Processes via And-Or Tree Evaluation". In: *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '98. San Francisco, California, USA: Society for Industrial and Applied Mathematics, 1998, pp. 364–373. ISBN: 0-89871-410-9.

[48] J. Yang and Z. Fei. "Bipartite Graph Based Dynamic Spectrum Allocation for Wireless Mesh Networks". In: *2008 The 28th International Conference on Distributed Computing Systems Workshops*. 2008, pp. 96–101.

[49] L. Wang et al. "Socially enabled wireless networks: resource allocation via bipartite graph matching". In: *IEEE Communications Magazine* 53.10 (2015), pp. 128–135.

[50] A. Dutta and P. Dasgupta. "Bipartite graph matching-based coordination mechanism for multi-robot path planning under communication constraints". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 857–862.

[51] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. *Deep Coordination Graphs*. 2019. arXiv: 1910.00091 [cs.LG].

[52] Eugenio Bargiacchi et al. "Learning to Coordinate with Coordination Graphs in Repeated Single-Stage Multi-Agent Decision Problems". In: ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 482–490.

[53] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521852293, 9780521852296.

[54] M. Luby et al. "Analysis of Low Density Codes and Improved Designs Using Irregular Graphs". In: *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*. STOC '98. Dallas, Texas, USA: ACM, 1998, pp. 249–258. ISBN: 0-89791-962-9.

[55] A. Shokrollahi. "Raptor codes". In: *IEEE Transactions on Information Theory* 52.6 (2006), pp. 2551–2567.

[56] Michael Luby. "LT Codes". In: *Proceedings of the 43rd Symposium on Foundations of Computer Science*. FOCS '02. USA: IEEE Computer Society, 2002, p. 271. ISBN: 0769518222.

[57] E. Moore and C. Shannon. "Reliable circuits using less reliable relays". In: *J. Franklin Inst.* 262 (1956), 191-208 and 281-297.

[58] Michael G. Luby et al. “Practical loss-resilient codes”. In: ACM Press, 1997, pp. 150–159.

[59] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. “Design of capacity-approaching irregular low-density parity-check codes”. In: *IEEE Transactions on Information Theory* 47.2 (Feb. 2001), pp. 619–637. ISSN: 0018-9448.

[60] F. Tobagi and L. Kleinrock. “Packet Switching in Radio Channels: Part II - The Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution”. In: *IEEE Transactions on Communications* 23.12 (1975), pp. 1417–1433.

[61] Norman M. Abramson. “THE ALOHA SYSTEM: Another Alternative for Computer Communications”. In: *Proc. of joint Computing Conf. AFIPS'70*. Honolulu, HI, USA, Nov. 1970.

[62] G. L. Choudhury and S. S. Rappaport. “ Diversity ALOHA A Random Access Scheme for Satellite Communications”. In: *IEEE Trans. on Communications* 31.3 (Mar. 1983), pp. 450–457.

[63] R. Storn and K. Price. “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces”. In: *Journal of Global Optimization* 11.4 (Dec. 1997), pp. 341–359.

[64] E. Paolini. “Finite Length Analysis of Irregular Repetition Slotted Aloha (IRSA) Access Protocols”. In: *Proc. of IEEE Int. Conf. on Communication Workshop, ICCW'15*. London, UK, June 2015.

[65] C. J. .C. H. Watkins and P. Dayan. “Q-learning”. In: *Machine Learning* 8.3 (May 1992), pp. 279–292.

[66] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. “Planning and Acting in Partially Observable Stochastic Domains”. In: *Artif. Intell.* 101.1-2 (May 1998), pp. 99–134.

[67] D. S. Bernstein, S. Zilberstein, and N. Immerman. “The Complexity of Decentralized Control of Markov Decision Processes”. In: *Mathematics of Operations Research* 27.4 (Nov. 2002), pp. 819–840.

[68] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. “The Complexity of Decentralized Control of Markov Decision Processes”. In: *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. UAI'00. Stanford, California: Morgan Kaufmann Publishers Inc., 2000, pp. 32–37. ISBN: 1558607099.

[69] William R Thompson. “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”. In: *Biometrika* 25.3-4 (Dec. 1933), pp. 285–294. ISSN: 0006-3444.

[70] Afef Ben Hadj Alaya-Feki, Eric Moulines, and Alain LeCornec. “Dynamic spectrum access with non-stationary Multi-Armed Bandit”. In: (July 2008), pp. 416–420.

[71] Tali Sharot. “The optimism bias”. In: *Current Biology* 21.23 (2011), R941–R945. ISSN: 0960-9822.

[72] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. "Finite-Time Analysis of the Multiarmed Bandit Problem". In: *Mach. Learn.* 47.2–3 (May 2002), pp. 235–256. ISSN: 0885-6125. DOI: 10.1023/A:1013689704352.

[73] Aurélien Garivier and Olivier Cappé. "The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond". In: ed. by Sham M. Kakade and Ulrike von Luxburg. Vol. 19. Proceedings of Machine Learning Research. Budapest, Hungary: JMLR Workshop and Conference Proceedings, Sept. 2011, pp. 359–376.

[74] T.L Lai and Herbert Robbins. "Asymptotically efficient adaptive allocation rules". In: *Advances in Applied Mathematics* 6.1 (1985), pp. 4–22. ISSN: 0196-8858.

[75] C. Morlet and A. Ginesi. "Introduction of Mobility Aspects for DVB-S2/RCS Broadband Systems". In: *2006 International Workshop on Satellite and Space Communications*. 2006, pp. 93–97.

[76] P. Popovski. "Random Access". In: *Wireless Connectivity: An Intuitive and Fundamental Guide*. 2020, pp. 27–49.

[77] M. Berioli et al. "Modern Random Access Protocols". In: *Found. Trends Netw.* 10 (2016), pp. 317–446.

[78] E. Paolini, G. Liva, and M. Chiani. "High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA". In: *2011 IEEE International Conference on Communications (ICC)*. 2011, pp. 1–6.

[79] G. Liva et al. "Spatially-coupled random access on graphs". In: *2012 IEEE International Symposium on Information Theory Proceedings*. 2012, pp. 478–482.

[80] S. Kudekar and H. D. Pfister. "The effect of spatial coupling on compressive sensing". In: *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2010, pp. 347–353.

[81] V. Aref et al. "Lossy source coding via spatially coupled LDGM ensembles". In: *2012 IEEE International Symposium on Information Theory Proceedings*. 2012, pp. 373–377.

[82] Cedomir Stefanovic, Petar Popovski, and Dejan Vukobratovic. "Frameless ALOHA Protocol for Wireless Networks". English. In: *I E E E Communications Letters* 16.12 (2012), pp. 2087–2090. ISSN: 1089-7798.

[83] C. Stefanovic and P. Popovski. "ALOHA Random Access that Operates as a Rateless Code". In: *IEEE Transactions on Communications* 61 (2013), pp. 4653–4662.

[84] C. Stefanović et al. "Joint estimation and contention-resolution protocol for wireless random access". In: *2013 IEEE International Conference on Communications (ICC)*. 2013, pp. 3382–3387.

[85] Murali Kodialam and Thyaga Nandagopal. "Fast and Reliable Estimation Schemes in RFID Systems". In: *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*. MobiCom '06. Los Angeles, CA, USA: Association for Computing Machinery, 2006, pp. 322–333. ISBN: 1595932860.

[86] M. Molle and G. Polyzos. "Conflict Resolution Algorithms and their Performance Analysis". In: 1993.

[87] Y. Yu and G. B. Giannakis. "High-Throughput Random Access Using Successive Interference Cancellation in a Tree Algorithm". In: *IEEE Transactions on Information Theory* 53.12 (2007), pp. 4628–4639.

[88] M. G. Luby et al. "Improved low-density parity-check codes using irregular graphs". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 585–598.

[89] S. Kudekar, T. J. Richardson, and R. L. Urbanke. "Threshold Saturation via Spatial Coupling: Why Convolutional LDPC Ensembles Perform So Well over the BEC". In: *IEEE Transactions on Information Theory* 57.2 (2011), pp. 803–834.

[90] A. Jimenez Felstrom and K. S. Zigangirov. "Time-varying periodic convolutional codes with low-density parity-check matrix". In: *IEEE Transactions on Information Theory* 45.6 (1999), pp. 2181–2191.

[91] Zhenzhen Liu and I. Elhanany. "RL-MAC: A QoS-Aware Reinforcement Learning based MAC Protocol for Wireless Sensor Networks". In: *2006 IEEE International Conference on Networking, Sensing and Control*. 2006, pp. 768–773.

[92] Y. Chu, P. D. Mitchell, and D. Grace. "ALOHA and Q-Learning based medium access control for Wireless Sensor Networks". In: *2012 International Symposium on Wireless Communication Systems (ISWCS)*. 2012, pp. 511–515.

[93] Laura Toni and Pascal Frossard. "IRSA Transmission Optimization via Online Learning". In: *CoRR* abs/1801.09060 (2018). arXiv: 1801.09060. URL: http://arxiv.org/abs/1801.09060.

[94] V. Mnih et al. "Playing Atari with Deep Reinforcement Learning". In: *Proc. of NIPS Deep Learning Workshop, NIPS'13*. Lake Tahoe, CA, USA, Dec. 2013.

[95] M. A. L. Thathachar and P. S. Sastry. "Varieties of learning automata: an overview". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 32.6 (2002), pp. 711–722.

[96] N. A. Shinkafi et al. "Learning Automata Based Q-Learning RACH Access Scheme for Cellular M2M Communications". In: *2019 IEEE Global Conference on Internet of Things (GCIoT)*. 2019, pp. 1–6.

[97] T. Wang, S. Wang, and Z. Zhou. "Machine learning for 5G and beyond: From model-based to data-driven mobile wireless networks". In: *China Communications* 16.1 (2019), pp. 165–175.

[98] Alessio Zappone, Marco Di Renzo, and Mérouane Debbah. *Wireless Networks Design in the Era of Deep Learning: Model-Based, AI-Based, or Both?* 2019. arXiv: 1902.02647 [eess.SP].

[99] Oshri Naparstek and Kobi Cohen. "Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access". In: *arXiv:1704.02613 [cs]* (Apr. 9, 2017). arXiv: 1704.02613. URL: http://arxiv.org/abs/1704.02613.

[100] Ursula Challita, Li Dong, and Walid Saad. "Proactive Resource Management in LTE-U Systems: A Deep Learning Perspective". In: *arXiv:1702.07031 [cs, math]* (Feb. 22, 2017). arXiv: 1702.07031. URL: http://arxiv.org/abs/1702.07031.

[101] Shangxing Wang et al. "Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks". In: *arXiv:1802.06958 [cs]* (Feb. 19, 2018). arXiv: 1802.06958. URL: http://arxiv.org/abs/1802.06958.

[102] Y. Zhao et al. "Deep Reinforcement Learning Aided Intelligent Access Control in Energy Harvesting based WLAN". In: *IEEE Transactions on Vehicular Technology* (2020), pp. 1–1.

[103] A. Anandkumar et al. "Distributed Algorithms for Learning and Cognitive Medium Access with Logarithmic Regret". In: *IEEE Journal on Selected Areas in Communications* 29.4 (2011), pp. 731–745.

[104] Ilai Bistritz and Amir Leshem. "Distributed Multi-Player Bandits - a Game of Thrones Approach". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio et al. Curran Associates, Inc., 2018, pp. 7222–7232.

[105] Gábor Lugosi and Abbas Mehrabian. "Multiplayer bandits without observing collision information". In: *CoRR* abs/1808.08416 (2018). arXiv: 1808.08416. URL: http://arxiv.org/abs/1808.08416.

[106] Rémi Bonnefoi et al. "Multi-Armed Bandit Learning in IoT Networks: Learning helps even in non-stationary settings". In: (2018). arXiv: 1807.00491 [cs.NI].

[107] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.

[108] Yaohua Sun et al. "Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues". In: *arXiv:1809.08707 [cs]* (Sept. 2018). arXiv: 1809.08707. URL: http://arxiv.org/abs/1809.08707.

[109] C. Claus and G. Boutilier. "The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems". In: *Proc. of the AAAI '98/IAAI '98*. Madison, WI, USA, July 1998.

[110] Huizhen Yu and Dimitri P. Bertsekas. "On Near Optimality of the Set of Finite-State Controllers for Average Cost POMDP". In: *Mathematics of Operations Research* 33.1 (Feb. 2008), pp. 1–11. ISSN: 0364-765X, 1526-5471.

[111] Nicholas H. Mastronarde. "Online Learning for Energy-Efficient Multimedia Systems". PhD thesis. University of California, 2011.

[112] N. Thomos et al. "Adaptive Prioritized Random Linear Coding and Scheduling for Layered Data Delivery From Multiple Servers". In: *IEEE Trans. on Multimedia* 17.6 (June 2015), pp. 893–906. ISSN: 1520-9210.

[113] Huizhen Yu and Dimitri P. Bertsekas. "Discretized Approximations for POMDP with Average Cost". In: *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. UAI '04. Banff, Canada: AUAI Press, 2004, pp. 619–627. ISBN: 0-9749039-0-6.

[114] E. Even-Dar and Y. Mansour. "Learning Rates for Q-learning". In: *J. Mach. Learn. Res.* 5 (Dec. 2004), pp. 1–25.

[115] T. Grenager, R. Powers, and Y. Shoham. "Dispersion Games: General Definitions and Some Specific Learning Results". In: *Proc. of 18th National/14th Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence, AAAI '02*. Edmonton, AL, Canada, July 2002.

[116] Mark Paskin, Carlos Guestrin, and Jim McFadden. " A Robust Architecture for Distributed Inference in Sensor Networks". In: *Proc. IPSN '05*. Boise, ID, USA, Apr. 2005. ISBN: 0-7803-9202-7.

[117] C. Zhang and V. Lesser. "Coordinated Multi-Agent Reinforcement Learning in Networked Distributed POMDPs". In: *Proc. of the 25th Conf. on Artificial Intelligence, AAAI'11*. San Francisco, CA, USA, Aug. 2011.

[118] Carlos Guestrin, Michail G. Lagoudakis, and Ronald Parr. "Coordinated Reinforcement Learning". In: *Proc. ICML '02*. San Francisco, CA, USA, July 2002, pp. 227–234. ISBN: 1-55860-873-7.

[119] Yann-Michaël De Hauwere, Peter Vrancx, and Ann Nowé. "Generalized learning automata for Multi-agent Reinforcement Learning". In: *IOS Press Journal of AI Communications* 23.4 (Apr. 2010), pp. 311–324.

[120] Silvano Martello and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: John Wiley & Sons, Inc., 1990. ISBN: 0-471-92420-2.

[121] Joris M. Mooij and Hilbert J. Kappen. "Sufficient Conditions for Convergence of the Sum-Product Algorithm". In: *IEEE Trans. on Information Theory* 53.12 (Dec. 2007), pp. 4422–4437. ISSN: 0018-9448.

[122] A. Rogers et al. "Bounded approximate decentralised coordination via the max-sum algorithm". In: *Artificial Intelligence* 175.2 (Feb. 2011), pp. 730–759. ISSN: 00043702.

[123] Carlos Guestrin, Daphne Koller, and Ronald Parr. "Multiagent Planning with Factored MDPs". In: *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, 2002, pp. 1523–1530.

[124] Jelle R. Kok and Nikos Vlassis. "Using the Max-Plus Algorithm for Multiagent Decision Making in Coordination Graphs". In: *RoboCup 2005: Robot Soccer World Cup IX*. Vol. 4020. Berlin, Heidelberg, 2006, pp. 1–12. ISBN: 978-3-540-35437-6 978-3-540-35438-3.

[125] Jelle R. Kok and Nikos Vlassis. "Collaborative Multiagent Reinforcement Learning by Payoff Propagation". In: *J. Mach. Learn. Res.* 7 (Dec. 2006), pp. 1789–1828. ISSN: 1532-4435.

[126] Yair Weiss. "Correctness of Local Probability Propagation in Graphical Models with Loops". In: *Neural Computation* 12.1 (Jan. 2000), pp. 1–41. ISSN: 0899-7667, 1530-888X.

[127] Jr. L. R. Ford and D. R. Fulkerson. "A Suggested Computation for Maximal Multi-Commodity Network Flows". In: *Management Science* 5.1 (1958), pp. 97–101.

[128] Jose M. Bernardo and Adrian F.M. Smith. *Bayesian Theory*. John Willen & Sons, Inc., 1994, p. 75.

[129] Martin J. Wainwright and Michael I. Jordan. "Graphical Models, Exponential Families, and Variational Inference". In: *Foundations and Trends in Machine Learning* 1.1 (2007), pp. 1–305. ISSN: 1935-8237, 1935-8245.

[130] G. Terry Ross and Richard M. Soland. "A branch and bound algorithm for the generalized assignment problem". In: *Mathematical Programming* 8.1 (Dec. 1975), pp. 91–103. ISSN: 1436-4646.

[131] Michael G. Luby, Michael Mitzenmacher, and M. Amin Shokrollahi. "Analysis of Random Processes via And-Or Tree Evaluation". In: *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '98. San Francisco, California, USA: Society for Industrial and Applied Mathematics, 1998, pp. 364–373. ISBN: 0898714109.

[132] M. Sipser and D. A. Spielman. "Expander codes". In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1710–1722.

[133] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. "Design of capacity-approaching irregular low-density parity-check codes". In: *IEEE Transactions on Information Theory* 47.2 (2001), pp. 619–637.

[134] Michael G. Luby et al. "Practical Loss-Resilient Codes". In: *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*. STOC '97. El Paso, Texas, USA: Association for Computing Machinery, 1997, pp. 150–159. ISBN: 0897918886. DOI: 10.1145/258533.258573.

[135] Ericsson. *Cellular Networks for Massive IoT: Enabling Low Power Wide Area Applications*. Tech. rep. Stockholm, Sweden, 2016, pp. 1–13.

[136] John Schulman et al. "Proximal Policy Optimization Algorithms." In: *CoRR* abs/1707.06347 (2017). URL: http://dblp.uni-trier.de/db/journals/corr/corr1707.html#SchulmanWDRK17.

[137] Kris Cao et al. "Emergent Communication through Negotiation". In: *International Conference on Learning Representations*. 2018.