2019

Learning Optimal Resource Allocations In Wireless Systems

Mark Randall Eisen
University of Pennsylvania, eisen.markr@gmail.com

Learning Optimal Resource Allocations In Wireless Systems

Abstract

The goal of this thesis is to develop a learning framework for solving resource allocation problems in wireless systems. Resource allocation problems are as widespread as they are challenging to solve, in part due to the limitations in finding accurate models for these complex systems. While both exact and heuristic approaches have been developed for select problems of interest, as these systems grow in complexity to support applications in Internet of Things and autonomous behavior, it becomes necessary to have a more generic solution framework. The use of statistical machine learning is a natural choice not only in its ability to develop solutions without reliance on models, but also due to the fact that a resource allocation problem takes the form of a statistical regression problem.

The second and third chapters of this thesis begin by presenting initial applications of machine learning ideas to solve problems in wireless control systems. Wireless control systems are a particular class of resource allocation problems that are a fundamental element of IoT applications. In Chapter 2, we consider the setting of controlling plants over non-stationary wireless channels. We draw a connection between the resource allocation problem and empirical risk minimization to develop convex optimization algorithms that can adapt to non-stationarities in the wireless channel. In Chapter 3, we consider the setting of controlling plants over a latency-constrained wireless channel. For this application, we utilize ideas of control-awareness in wireless scheduling to derive an assignment problem to determine optimal, latency-aware schedules.

The core framework of the thesis is then presented in the fourth and fifth chapters. In Chapter 4, we formally draw a connection between a generic class of wireless resource allocation problems and constrained statistical learning, or regression. From here, this inspires the use of machine learning models to parameterize the resource allocation problem. To train the parameters of the learning model, we first establish a bounded duality gap result of the constrained optimization problem, and subsequently present a primal-dual learning algorithm. While any learning parameterization can be used, in this thesis we focus our attention on deep neural networks (DNNs). While fully connected networks can be represent many functions, they are impractical to train for large scale systems. In Chapter 5, we tackle the parallel problem in our wireless framework of developing particular learning parameterizations, or deep learning architectures, that are well suited for representing wireless resource allocation policies. Due to the graph structure inherent in wireless networks, we propose the use of graph convolutional neural networks to parameterize the resource allocation policies.

Before concluding remarks and future work, in Chapter 6 we present initial results on applying the learning framework of the previous two chapters in the setting of scheduling transmissions for low-latency wireless control systems. We formulate a control-aware scheduling problem that takes the form of the constrained learning problem and apply the primal-dual learning algorithm to train the graph neural network.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Electrical & Systems Engineering

LEARNING OPTIMAL RESOURCE ALLOCATIONS IN WIRELESS SYSTEMS

Mark Eisen

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania
in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy
2019

Supervisor of Dissertation

Alejandro Ribeiro, Professor of Electrical and Systems Engineering

Graduate Group Chairperson

Victor M. Preciado, Associate Professor and Graduate Group Chair

Dissertation Committee
George J. Pappas, Joseph Moore Professor and Chair of Electrical and Systems Engineering
Daniel D. Lee, Tisch University Professor of Electrical and Computer Engineering, Cornell Tech
Dave Cavalcanti, Senior Staff Research Scientist, Intel Corporation

LEARNING OPTIMAL RESOURCE ALLOCATIONS IN WIRELESS SYSTEMS

Acknowledgments

The years that I spent as a Ph.D. student were some of the most valuable in my life for both professional and personal reasons. I am extremely grateful to be able to appreciate and acknowledge the influence of my advisor, collaborators and friends on writing this thesis.

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Alejandro Ribeiro for inviting me into his research group. I began working with Dr. Ribeiro as an undergraduate at Penn, and the past 7 years we have worked together have been instrumental in my personal, professional, and intellectual growth. Without his his intellectual insights and mentorship throughout the course of my PhD, this thesis would not have been possible.

I would like to further thank Dr. George G. Pappas, Dr. Daniel D. Lee, and Dr. Dave Cavalcanti for agreeing to serve in my doctoral committee and for giving me the opportunity to collaborate with them in different projects. All of these collaborations played key roles in developing the content for this thesis.

Writing this thesis will not have been possible without the joint effort of my collaborators. I would like to thank Dr. Konstantinos Gatsis, Dr. Mohammad M. Rashid, Clark Zhang, and Luiz F. O. Chamon for collaborating with me on the work used in this thesis. Over the course of my PhD, I've been fortunate to be a part of other meaningful collaborations, the work of which was not included in this thesis. Specifically, I benefited greatly from working closely with Dr. Aryan Mokthari and Dr. Santiago Segarra.

I would also like to extend a deep sense of gratitude to my friends in the laboratory to which I belonged over the past five years. Their support and friendship made me feel at home and has made the whole experience vastly more enjoyable and meaningful. I would like to specifically mention the wonderful friends I made in my research group, including but not limited to: Ceyhun Eksin, Santiago Segarra, Alec Koppel, Aryan Mokhtari, Weiyu Huang, Santiago Paternain, Fernando Gama, Shi-Ling Phuong, Luiz Chamon, Luana Ruiz, Maria Peifer, Kate Tolstaya, Harshat Kumar, Clark Zhang, Arbaaz Khan, Vinicius Lima, Zhan Gao, Mahyar Fazylab, and Mohammad Fereydounian. Among the many memorable experiences we had, I would like to highlight the Thirsty Thursdays and weekend Asados as things I will never forget.

I would like to further thank my many friends from Philadelphia and Langhorne who have provided me support over the years of my Ph.D. Special thanks to my parents, Steve and Debbi, who first suggested I pursue a Ph.D, and my brothers Jon and David. Their love and support will always mean more to be than anything else I may accomplish in my professional career, and none of this would have been possible without my family.

Mark Eisen, Philadelphia, May 2019

ABSTRACT

LEARNING OPTIMAL RESOURCE ALLOCATIONS IN WIRELESS SYSTEMS

Mark Eisen
Alejandro Ribeiro

The goal of this thesis is to develop a learning framework for solving resource allocation problems in wireless systems. Resource allocation problems are as widespread as they are challenging to solve, in part due to the limitations in finding accurate models for these complex systems. While both exact and heuristic approaches have been developed for select problems of interest, as these systems grow in complexity to support applications in Internet of Things and autonomous behavior, it becomes necessary to have a more generic solution framework. The use of statistical machine learning is a natural choice not only in its ability to develop solutions without reliance on models, but also due to the fact that a resource allocation problem takes the form of a statistical regression problem.

The second and third chapters of this thesis begin by presenting initial applications of machine learning ideas to solve problems in wireless control systems. Wireless control systems are a particular class of resource allocation problems that are a fundamental element of IoT applications. In Chapter 2, we consider the setting of controlling plants over non-stationary wireless channels. We draw a connection between the resource allocation problem and empirical risk minimization to develop convex optimization algorithms that can adapt to non-stationarities in the wireless channel. In Chapter 3, we consider the setting of controlling plants over a latency-constrained wireless channel. For this application, we utilize ideas of control-awareness in wireless scheduling to derive an assignment problem to determine optimal, latency-aware schedules.

The core framework of the thesis is then presented in the fourth and fifth chapters. In Chapter 4, we formally draw a connection between a generic class of wireless resource allocation problems and constrained statistical learning, or regression. From here, this inspires the use of machine learning models to parameterize the resource allocation problem. To train the parameters of the learning model, we first establish a bounded duality gap result of the constrained optimization problem, and subsequently present a primal-dual learning algorithm. While any learning parameterization can be used, in this thesis we focus our attention on deep neural networks (DNNs). While fully connected networks can be represent many functions, they are impractical to train for large scale systems. In Chapter 5, we tackle the parallel problem in our wireless framework of developing particular learning parameterizations, or deep learning *architectures*, that are well suited for representing wireless resource allocation policies. Due to the graph structure inherent in wireless networks,

we propose the use of graph convolutional neural networks to parameterize the resource allocation policies.

Before concluding remarks and future work, in Chapter 6 we present initial results on applying the learning framework of the previous two chapters in the setting of scheduling transmissions for low-latency wireless control systems. We formulate a control-aware scheduling problem that takes the form of the constrained learning problem and apply the primal-dual learning algorithm to train the graph neural network.[1]

[1]Work presented in this thesis has been published and submitted for review to IEEE Transactions on Signal Processing, IEEE Internet of Things Journal, and the Proceedings of the American Control Conference, International Conference on Acoustics, Speech, and Signal Processing, Asilomar Conference on Signals, Systems, and Computers, and International Workshop on Signal Processing Advances in Wireless Communications (SPAWC). Submissions available at [30–33, 35–40]. Work in this thesis is supported by ARL DCIST CRA W911NF-1, 7-2-0181 and Intel Science and Technology Center for Wireless Autonomous Systems (ISTC-WAS).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The advent of the Internet of Things (IoT) brings way towards the rise of integrating fully autonomous systems into our infrastructure and daily lives. With applications ranging from industrial robotics to smart grid to autonomous vehicles, the future of these technologies invariably depend up our ability to increase the capacity of the underlying technology of autonomous IoT systems to support their increasing scale and complexity. One of the most fundamental of such underlying technologies is our wireless communication systems [7,79,128]. Indeed, a primary feature of future autonomous systems is their ability to communicate sensing and actuation information over the wireless medium—whether it be through 5G, LTE, Bluetooth, etc.—to make autonomous decisions. It is thus increasingly necessary to optimally design wireless systems that can support the various demands—e.g. capacity, latency, throughput, etc.—placed by these autonomous systems.

The defining feature of wireless communication is fading, or the various random disturbances experienced by the signal as it propagates through the air. The role of optimal wireless system design is to allocate resources across fading states to optimize long term system properties. Mathematically, we have a random variable $\mathbf{h}$ that represents the instantaneous fading environment, a corresponding instantaneous allocation of resources $\mathbf{p}(\mathbf{h})$, and an instantaneous performance outcome $\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)$ resulting from the allocation of resources $\mathbf{p}(\mathbf{h})$ when the channel realization is $\mathbf{h}$. The instantaneous system performance tends to vary too rapidly from the perspective of end users for whom the long term average $\mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)\big]$ is a more meaningful metric. This interplay between instantaneous allocation of resources and long term performance results in distinctive formulations where we seek to maximize a utility of the long term average $\mathbf{x}$ subject to the constraint $\mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)\big]$. Problems of this form range from the simple power allocation in wireless fading channels – the solution of which is given by water filling – to the optimization of frequency division multiplexing [126], beamforming [8, 112], and random access [61, 62].

Optimal resource allocation problems are as widespread as they are challenging. This

is because of the high dimensionality that stems from the variable $\mathbf{p}(\mathbf{h})$ being a function over a dense set of fading channel realizations and the lack of convexity of the constraint $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$. For resource allocation problems, such as interference management, heuristic methods have been developed [18, 111, 130]. Generic solution methods are often undertaken in the Lagrangian dual domain. This is motivated by the fact that the dual problem is not functional, as it has as many variables as constraints, and is always convex whether the original problem is convex or not. A key property that enables this solution is the lack of duality gap, which allows dual operation without loss of optimality. The duality gap has long being known to be null for convex problems – e.g., the water level in water filling solutions is a dual variable – and has more recently being shown to be null under mild technical conditions despite the presence of the nonconvex constraint $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ [103, 136]. This permits dual domain operation in a wide class of problems and has lead to formulations that yield problems that are *more* tractable, although not necessarily tractable without resorting to heuristics [31, 42, 47, 76, 92, 124, 138]. All such approaches invariably require accurate system models and may require prohibitively large computational complexity for each allocation decision.

In contrast to such model-based heuristics, more recent work has applied machine learning and regression techniques to solve resource allocation problems. Machine learning methods train a generic learning model, most commonly a deep neural network (DNN), to make resource allocation decisions for a wide variety of problems. One such approach follows the tenants of supervised learning, or in other words fitting a neural network to a training set of solutions obtained using an existing algorithm [70, 115, 121, 131]. These techniques are useful in their simplicity and relative effectiveness—neural networks are well suited for finding good local minima in loss functions typically used in supervised learning, e.g. Euclidean loss. However, supervised learning techniques are limited by both the availability of solutions needed to build a training set as well as the accuracy of such solutions. The former limitation implies supervised learning can only be used in resource allocation problems with existing heuristic solutions, while the latter limitation implies that the learning model will only meet the performance of such heuristics but never exceed them. For many of the open problems in wireless autonomous system design, a training set required to perform supervised training methods is unavailable.

A crucial observation in the understanding of wireless resource allocation problems and their connection to machine learning is the fact that the expectation $\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ has a form that is typical of learning problems. Indeed, in the context of learning, $\mathbf{h}$ represents a feature vector, $\mathbf{p}(\mathbf{h})$ the regression function to be learned, $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ a loss function to be minimized, and the expectation $\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ the statistical loss over the distribution of the dataset. We may then learn without labeled training data by directly minimizing

the statistical loss with stochastic optimization methods which merely observe the loss $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ at sampled pairs $(\mathbf{h}, \mathbf{p}(\mathbf{h}))$. In this way, we follow the interpretation of wireless autonomous system design as a *constrained learning problem*, in which we seek a resource allocation function, or policy, that minimizes a statistical loss that represents the physical performance of the system, subject to the necessary constraints.

In Chapter 2, we present our first method to exploit this interpretation in tackling the problem of the *non-stationarity* of wireless channels in practical systems. The optimal resource allocation policy to close a series of wireless control systems is inherently linked to the statistics of the wireless channel. In most practical applications, the statistics will invariably change over time. To address this problem, we utilize the statistical learning interpretation of resource allocation to leverage ideas of empirical risk minimization (ERM). ERM substitutes the statistical loss with a deterministic, empirical loss. With the application of second order convex optimization methods, we demonstrate how we can quickly adapt the resource allocation policy as the wireless channel distribution changes over time.

We proceed in Chapter 3 to apply techniques from machine learning in studying another wireless resource allocation problem of growing interest in autonomous system design—namely, the challenge of designing ultra-reliable, low-latency communications (URLLC). As in Chapter 2, we may formulate a constrained resource allocation problem that models the performance of a resource allocation decision relative to the expected performance of the control system. In such a manner, the resource allocation is a scheduling policy that minimizes total latency while meeting a control performance constraint. For added practical value, we look specifically at a formulation employing the IEEE 802.11ax WiFi architecture. The combinatorial size of the scheduling decision space inspires the use of so-called *assignment methods* to find solutions to the resulting optimization problem.

While these initial approaches for solving complex wireless resource allocation problems prove effective, they are limited in that they are custom designed to tackle specific problems in wireless autonomous systems, and moreover utilize a large degree of model information. In this thesis, we are ultimately interested in developing a comprehensive learning framework for addressing open wireless resource allocation problems. This requires both the generality in the algorithmic approach, as well as an ability to operate when knowledge of the model—e.g. system dynamics, capacity model, etc.—-is unavailable, as is most often the case in complex systems of practical interest.

A more promising approach in learning for resource allocation uses the learning model, e.g. deep neural network, to directly parameterize the resource allocation policy in the optimization problem [25, 38, 69, 73, 83, 132]. This can be considered unsupervised in that such techniques can train neural networks with respect to an arbitrary system performance measure and thus does not require the acquisition of a training set—or, in other words,

reinforcement learning. These techniques are further beneficial in that they can be applied to any arbitrary resource allocation problem and have the potential to exceed performance of existing heuristics. This setting is typical of, e.g., reinforcement learning problems [117], and is a learning approach that has been taken in several *unconstrained* problems in wireless optimization [25, 93, 94, 134]. In general, wireless optimization problems *do* have constraints as we are invariably trying to balance capacity, power consumption, channel access, and interference. We develop such a constrained learning framework in Chapters 4 and 5 of this dissertation.

In Chapter 4, we formally draw an equivalence between resource allocation problems and constrained statistical learning—or constrained regression—to develop a theoretical and algorithmic framework for learning resource allocation policies for a generic class of resource allocation problems. In particular, this involves using the universal approximation properties of fully connected neural network (FCNNs) is to recover the duality results of [103, 136]. From there, we present a so-called primal-dual learning method that can be used to learn the optimal weights of the FCNN for a generic class of resource allocation problems. We moreover demonstrate how the proposed primal-dual method can be performed model-free, or agnostic to particular system model knowledge.

The primal-dual framework provides an algorithmic approach for training learning parameterizations of a resource allocation policy. The next question of interest concerns precisely which learning parameterizations are best suited for representing resource allocation policies. FCNNs are an immediate candidate due to their general universality property—that is, they can theoretically approximate any continuous function arbitrarily well. However, just as FCNNs are made a naturally viable choice from their universality property [38, 115], so too are they limited. The practical challenge of training FCNNs to parameterize strong performing policies is well documented in empirical study , as their full expressive power inherently requires performing optimization over a very high dimensional space. Moreover, the completely generic structure of FCNN contains no intrinsic invariance to input scaling or variation; any change in wireless network size or shuffling of the network labels renders the current FCNN-based policy ineffective. Convolutional neural networks architectures (CNNs), on the other hand, have proved a solution to this problem for many learning domains such as image classification and recommender systems by preserving invariances in the architecture itself. While some existing work has used CNNs for wireless resource allocation [69, 121, 131], they utilize standard temporal or spatial CNNs and thus do not leverage the true invariances present in wireless networks. In particular, the structure—and subsequent invariances—of wireless networks comes from the links between transmitters and receivers that result from fading. The work in [23] uses spatial CNNs that utilize the geometric structure of the network, but in doing so does not incorporate the fading link structure. In Chapter 5,

we incorporate a recent development in CNNs that perform convolutions on arbitrarily structured data—called graph neural networks [45, 55]—to fully utilize the network of fading links in parameterizing a resource allocation policy. When such learning architectures are used in conjunction with the model-free algorithmic learning approach developed in Chapter 4, we obtain a more complete framework for learning effective resource allocation policies in large wireless systems for application in autonomous system settings.

We conclude in Chapter 6 to apply the developed learning framework in the low-latency wireless control system problem previously tackled in Chapter 3. In this chapter, we demonstrate how a scheduling problem to address low-latency control systems can be formulated as the constrained statistical learning problem, and utilize both the primal-dual learning algorithm and graph neural network parameterization discussed in Chapters 4 and 5, respectively, to solve.

The outline for our developed framework for learning in wireless systems is presented in Figure 1.1. Chapters 2 and 3 discuss initial applications that utilize machine learning techniques, i.e. empirical risk minimization and assignment methods, to address specific problems. These applications are shown in green diamonds in the first row of Figure 1.1. Chapters 4 and 5, shown in the following row in red rectangles, develop the two pillars of a specific framework for addressing generic resource allocations problems. One pillar involves the formulation of the resource allocation problem and the resulting algorithm—we discuss the primal-dual learning framework in Chapter 4—and the parallel pillar is in specific learning architectures and parameterization to utilize in the learning process—we discuss the use of Graph Neural Networks in Chapter 5. Finally, we utilize these two pillars to again address an application of interest, namely the low-latency wireless control system, in Chapter 6.

The work presented in this thesis develops a learning framework to be applied to a wide variety of complex and large-scale wireless resource allocation problems. Such problems are becoming increasingly prevalent in the design of the wireless autonomous systems that will compose the future of IoT technology and infrastructure. As these problems grow in both scale and complexity, machine learning has an enormous potential to become a fundamental feature of next generation wireless technology, from 6G [19, 46] to Industry 4.0 [79]. The methodology proposed here is intended to provide a pathway towards successful application of machine learning technology to communication systems. Indeed, there still exist many important extensions to be made to fully address the challenges posed by wireless IoT systems. We conclude the thesis in Chapter 7 by discussing the numerous opportunities for future work.

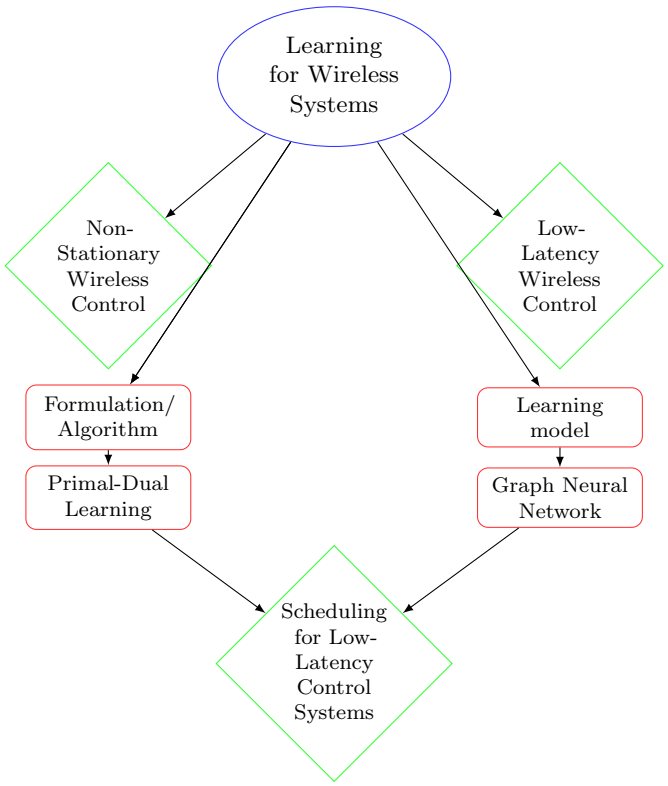


Figure 1.1: Design of learning framework for wireless resource allocation problems. Applications are notated with green diamonds, while foundation pillars of the learning framework are notated with red rectangles.

Chapter 2

Empirical Risk Minimization for Non-Stationary Wireless Control Systems

2.1 Introduction

The recent developments in autonomy in industrial control environments, teams of robotic vehicles, and the Internet-of-Things have motivated intelligent design of wireless systems. Even though wireless communication facilitates connectivity, it also introduces uncertainty that may affect stability and performance. To guarantee performance and safety of the control application it is common to employ model-based approaches. However wireless communication is naturally uncertain and time-varying due to effects that are not always amenable to modeling, such as mobility in the environment. In this chapter we propose an alternative learning-based approach, where autonomy relies on collected channel samples to optimize performance in a non-stationary environment. The connection between the two approaches is based on the observation that a sampled version of the model-based design approach can be cast as an empirical risk minimization (ERM) problem, a typical machine learning problem. Even so, standard techniques developed for solving ERM problems in machine learning do not address the additional challenges present in wireless autonomous systems, namely the non-stationarity of sample distributions.

The traditional model-based approach is motivated by the desire to build wireless control systems with stability and optimal performance. To counteract channel uncertainties it is natural to include a model of the wireless communication, for example an i.i.d. or Markov link quality, alongside the model of the physical system to be controlled. These models have been valuable to help analysis and control/communication design. For example, one can

characterize that it is impossible to estimate and/or stabilize an unstable plant if its growth rate is larger than the rate at which the link drops packets [51,56,106,113], or below a certain channel capacity [105,120]. Additionally models facilitate the design of controllers [22,41,63], as well as the allocation of communication resources to optimize control performance, for example power allocation over fading channels with known distributions [48, 100], or event-triggered control [5, 54, 80, 81, 101].

In practice wireless autonomous systems operate under unpredictable channel conditions following unknown time-varying distributions. While one approach would be to estimate the distributions using channel samples and then follow the above model-based design approach, in this chapter we propose an alternative learning-based approach which bypasses the channel-modeling phase. We exploit channel samples taken from the time-varying channel distributions with the goal to learn directly the solution to communication design problems. To apply this approach we exploit a connection between the model-based and the learning-based design problems. Existing works [20, 47, 49] study related problems in multiple-access wireless control systems and resource allocation problems in wireless systems but under a stationary channel distribution. These works generally employ first-order stochastic methods, which have slow convergence rates and hence not suitable for the present framework. A significant challenge remains in how to continuously learn optimal policies over a wireless channel that is time-varying. This shortcoming of existing sample-based approaches used in [20,47,49] and more general machine learning scenarios motivates the higher-order learning approach proposed in this chapter. Some existing machine learning methods account for nonstationarity by optimizing an averaged objective over all time [11, 65, 86]. Our approach differs in that we seek and track optimality locally with respect to the current channel distribution at every time epoch.

In this chapter we consider a wireless autonomous system where the design goal is to maximize a level of control performance for multiple systems while meeting a desired transmit power budget over the wireless channel (Section 2.2). The wireless channel is modeled as a fading channel with a time-varying and unknown distribution, and only available through samples taken over time. We derive in Section 2.2.1 a wireless control problem that finds optimal power allocation policies for an individual time epoch where the wireless channel distribution does not change, and then proceed to derive the Lagrange dual (Section 2.2.2). We show in Section 2.3 that the dual of the power allocation problem can be rewritten using channel samples as an empirical risk minimization problem, a common machine learning problem in which an expected loss function over an unknown distribution is approximated by optimized over a set of samples. Here the risk is loosely related to how far the current solution is from the desired optimal power allocation.

Because the wireless channel is varying over time, we develop a new approach to solving

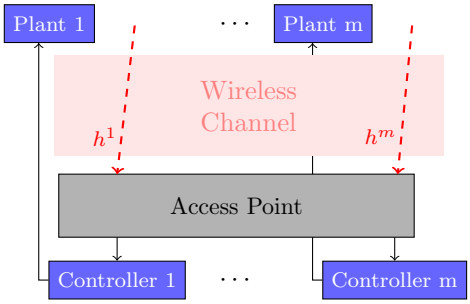


Figure 2.1: Wireless control system. Plants communicate state information to access point/controllers over wireless medium.

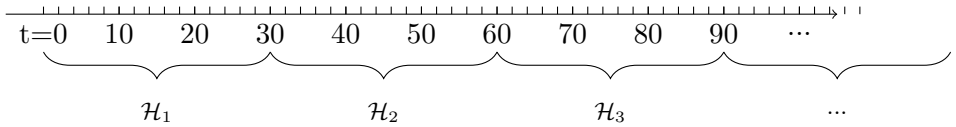


Figure 2.2: Time axis showing evolution of time t and epochs k. Each channel distribution $\mathcal{H}_k$ is stationary for a set of time instances.

a sequence of ERM problems. We collect and store a window of channel samples taken from consecutive distributions to reduce sampling complexity and employ Newton's method to learn new policies quickly (Section 2.4). More specifically, the quadratic convergence rate of Newton's method is shown to be sufficient to find approximate solutions to slowly varying objectives with a single update. Using Newton's method, we propose an algorithm that uses channel samples to approximate the solution of a power allocation wireless control problem over a non-stationary channel. We prove that, under specific conditions, the algorithm reaches an approximately optimal point in a single iteration of Newton's method (Section 2.5). This result establishes both a suboptimality bound with respect to the sampled problem (Section 2.5.1) as well as with respect to control performance metric in the wireless control problem (Section 2.5.2). We additionally show a stability result for a particular problem description common in wireless control systems (Section 2.5.3) and provide considerations for practical implementation of the method (Section 2.6). These results are further demonstrated in a numerical demonstration of learning power allocation policies across multiple control systems over a time-varying channel (Section 2.7).

2.2 Wireless Control Problem

We consider a wireless control problem (WCP) with m independent control systems labeled $i = 1, \ldots, m$, as shown in Figure 2.1. Each control system/agent i communicates at time t its state x_t^i over a wireless channel in order to close a loop and maximize a level of control performance. In particular, system i tries to close the control loop over the wireless channel

by transmitting with power level $p^i \in [0, p_0]$. Due to propagation effects the channel fading conditions that each system i experiences, denoted by $h^i \in \mathbb{R}_+$, change unpredictably over time [50, Ch. 3]. Together, the channel fading h^i and transmit power p^i determine the signal-to-noise ratio (SNR) at the receiver for system i, which in turn affects the probability of successful decoding of the transmitted packet at the receiver. We consider a function $q(h^i, p^i)$ that, given a current channel state and transmit power, determines the probability of successful transmission and decoding of the transmitted packet – see, e.g., [48, 100] for more details on this model. Transmission are assumed on different frequencies/bands and are not subject to contention – see [47, 49] for alternative formulations.

Because these fading conditions vary quickly and unpredictably, they can be modeled as independent random variables drawn from distribution $\mathcal{H}$ that itself is non-stationary, or time-varying. Channel fading is assumed constant during each transmission slot and it is independently distributed over time slots (block fading). Furthermore, the channel distribution $\mathcal{H}$ may vary across *time epochs*, but will in general be stationary within a single time epoch. In particular, consider an epoch index $k = 0, 1, \ldots$ that specifies a particular channel distribution $\mathcal{H}_k$ with realization h_k^i for system i. In Figure 2.2, we display a time axis rendering of this model. The state variables change at each transmission slot t, while the channel changes at scale k, which will in general contain multiple time steps. This is to say that we assume that the channel distribution $\mathcal{H}_k$ changes at a rate slower than the system evolution, and that within a single time epoch the channel is effectively stationary.

We proceed to derive a formal description of the wireless control problem of interest within a single time epoch, where the channel is assumed stationary. In Section 2.4 we extend this formulation to the non-stationary setting.

2.2.1 WCP in single epoch

Within a particular time epoch k with channel distribution $\mathcal{H}_k$, we can derive a formulation that characterizes the optimal power allocations between the m control systems so as to maximize the aggregate control performance across all systems, where p_0 reflects a maximum transmission power of the system. Given a random channel state $h_k^i \in \mathbb{R}_+$ drawn from the distribution $\mathcal{H}_k$. We wish to determine the amount of transmit power $p_k^i(h_k^i) : \mathbb{R}_+ \to [0, p_0]$ to be used when attempting to close its loop—see [48] for details. We note that we are looking for transmit power as a function of current channel conditions, as the power necessary to close the loop will indeed change with channel conditions. We assume the current channel gain h_k^i is available at the transmitter at each slot, as this can generally be obtained via short pilot signals—see [48]. Then the probability of closing the loop is given by the value

$$y_k^i := \mathbb{E}_{h_k^i} \left\{ q(h_k^i, p_k^i(h_k^i)) \right\}. \tag{2.1}$$

The variable $y_k^i \in [0,1]$ is the expectation of successful transmission over the channel distribution $\mathcal{H}_k$.

Using the variable y_k^i we use a monotonically increasing concave function $J^i : [0,1] \to \mathbb{R}$ that returns a measure of control system performance as a function of the probability of successful transmission. Such a function can take on many forms and, in general, can be derived in relation to the particular control task of interest. In the following example, we derive such a measure for a typical wireless control problem setting, namely the quadratic control performance of a switched linear dynamical system – see, e.g., [51, 106].

Example 1. *Consider for example that a control system i is a scalar linear dynamical system of the form*

$$x_{t+1}^i = A_o^i x_t^i + B^i u_t^i + w_t^i \tag{2.2}$$

where $x_t^i \in \mathbb{R}$ is the state of the system at transmission time t, A_o^i is the open loop (potentially unstable) dynamics of the system, $u_t^i \in \mathbb{R}$ is the control input applied to the system at time t, and w_t^i is some zero-mean i.i.d. disturbance process with variance W^i. Consider a given linear state feedback is applied to the system as the control input when a transmission is successful, i.e.,

$$u_t^i = \begin{cases} K^i x_t^i & \text{if loop closes} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

As a result, the system switched between an open loop mode A_o^i and a closed loop stable mode $A_c^i = A_o^i + B^i K^i$, as in

$$x_{t+1}^i = \begin{cases} A_o^i x_t^i + w_t^i & \text{if loop closes} \\ A_o^i x_t^i + w_t^i & \text{otherwise} \end{cases} \tag{2.4}$$

The goal is to regulate the system state close to zero, i.e., the system attempts to close the loop at a high rate in order to minimize an expected quadratic control cost objective of the form

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=0}^{N-1} \mathbb{E}(x_t^i)^2 \tag{2.5}$$

Assuming the control loop in (2.4) is closed with the success probability y_k^i in (2.1) at all time steps, it is possible to express the above cost explicitly as a function of y_k^i. Using the system dynamics (2.4), the variance of the system state satisfies the recursive formula

$$\mathbb{E}(x_{t+1}^i)^2 = y_k^i \, (A_c^i)^2 \, \mathbb{E}(x_t^i)^2 + (1 - y_k^i) \, (A_o^i)^2 \, \mathbb{E}(x_t^i)^2 + W^i \tag{2.6}$$

that is, with probability y_k^i the variance grows according to the open loop dynamics, and with probability $1 - y_k^i$ the variance shrinks according to the closed loop stable dynamics.

Operating recursively and using the geometric series sum, we can rewrite the variance at time t as

$$\mathbb{E}(x_t^i)^2 = [y_k^i\,(A_c^i)^2 + \,(1-y_k^i)\,(A_o^i)^2]^t \mathbb{E}(x_0^i)^2 \tag{2.7}$$

$$+ W^i \frac{1-[y_k^i\,(A_c^i)^2 + \,(1-y_k^i)\,(A_o^i)^2]^t}{1-[y_k^i\,(A_c^i)^2 + \,(1-y_k^i)\,(A_o^i)^2]}. \tag{2.8}$$

As follows from the above expression, the system is stable, i.e., the variance is bounded, if the packet success rate satisfies $[y_k^i\,(A_c^i)^2 + \,(1-y_k^i)\,(A_o^i)^2] < 1$ so that the sum above is bounded – see also [51, 106]. In that case, the state variance as well as the average (2.5) *converge to the same limit value, which we can define as our control performance function*

$$J^i(y_k^i) = -\frac{W^i}{1-\left[y_k^i(A_c^i)^2 + (1-y_k^i)(A_o^i)^2\right]} \tag{2.9}$$

This control performance function satisfies the assumption of concavity, and it is also monotonically increasing because we have added the negative sign in front of the expression. It is also possible to extend this analysis to include a cost on the control input, as is common in the Linear Quadratic Control problem, i.e., replace the cost in (2.5) *with $\mathbb{E}(x_t^i)^2 + (u_t^i)^2$.*

Remark 1. In Example 1, observe that the control system performance in (2.5) is a long term objective asymptotically for $t \to \infty$. As the channel fading distribution $\mathcal{H}_k$ will change unpredictably in the future it is hard to define an accurate value of this control performance. As a surrogate, in the above example we write a control system performance in (2.9) with respect to the current channel distribution $\mathcal{H}_k$, i.e., as if this channel distribution is stationary and will not change in the future. Later, in Section 2.5.3 we argue that this approximation and the power allocation algorithm we develop can indeed guarantee system stability.

To derive the full formulation of the wireless control problem for current channel distribution $\mathcal{H}_k$, we first define using boldface vectors the set of m channel states $\mathbf{h}_k := [h_k^1; h_k^2; \ldots; h_k^m] \sim \mathcal{H}_k^m$ observed by the control systems and the set of power allocation policies $\mathbf{p}_k(\mathbf{h}_k) := [p_k^1(h_k^1); p_k^2(h_k^2); \ldots; p_k^m(h_k^m)] \in \mathcal{P} := [0, p_0]^m$. We further define the vector of transmission probabilities at specific channel states $\mathbf{q}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k)) := [q(h_k^1, p_k^1(h_k^1)); \ldots; q(h_k^m, p_k^m(h_k^m))]$ and expected transmission probabilities $\mathbf{y}_k := [y_k^1; y_k^2; \ldots; y_k^m]$ from (2.1). The goal is to select $\mathbf{p}_k(\mathbf{h}_k)$ whose expected aggregate value is within a maximum power budget $p_{\max}$ while maximizing the total system performance $\sum J^i$ over m agents. Because J^i is monotonically increasing, we can relax the equality in (2.1) to an inequality constraint and write the

following optimization problem.

$$\{\mathbf{p}_k^*(h), \mathbf{y}_k^*\} := \underset{\mathbf{p}_k \in \mathcal{P}, \mathbf{y}_k \in \mathbb{R}^m}{\operatorname{argmax}} J(\mathbf{y}_k) := \sum_{i=1}^{m} J^i(y_k^i) \qquad (\text{WCP}_k)$$
$$\text{s. t.} \qquad \mathbf{y}_k \leq \mathbb{E}_{\mathbf{h}_k}\left\{\mathbf{q}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k))\right\},$$
$$\sum_{i=1}^{m} \mathbb{E}_{h_k^i}(p_k^i(h_k^i)) \leq p_{\max}$$

The problem in (WCP_k) states the optimal power allocation policy $\mathbf{p}_k^*(\mathbf{h}_k)$ is the one that maximizes the expected aggregate control performance over channel states while guaranteeing that the expected total transmitting power is below an available budget $p_{\max}$. We stress that this only provides the optimal policy with respect to a particular channel distribution $\mathcal{H}_k$. In the non-stationary wireless setting we are interested in solving (WCP_k) for all k.

2.2.2 Dual formulation of (WCP_k)

Solving this optimization problem directly has a number of significant challenges. The first is that the problem is non-convex, in particular due to the first constraint in (WCP_k). The second challenge is that the problem is optimized over an infinite-dimensional variable $\mathbf{p}_k(\mathbf{h}_k)$. It is very difficult to solve such a problem if there is no assumed parameterization of $\mathbf{p}_k^*(\mathbf{h}_k)$. We can show, however, from a result in [103] that a naturally occurring parameterization of $\mathbf{p}_k^*(\mathbf{h}_k)$ indeed can be derived from Lagrangian duality theory.

We proceed then to derive the dual problem from the constrained problem in (WCP_k). To simplify the presentation, we first introduce a set of augmented variables, denoted with tildes. Define the augmented vectors $\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k)) \in \mathbb{R}^{m+1}$ and $\check{\mathbf{y}}_k \in \mathbb{R}^{m+1}$ as

$$\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k)) := \begin{bmatrix} q(h_k^1, p_k^1(h_k^1)) \\ \vdots \\ q(h_k^m, p_k^m(h_k^m)) \\ -\sum_{i=1}^m p_k^i(h_k^i)] \end{bmatrix} \quad \check{\mathbf{y}}_k := \begin{bmatrix} y_k^1 \\ \vdots \\ y_k^m \\ -p_{\max} \end{bmatrix}. \qquad (2.10)$$

The augmented $\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k))$ includes transmission probabilities augmented with the total power allocation while $\check{\mathbf{y}}_k$ includes auxiliary variables augmented with total power budget. Using this new notation, the Lagrangian function is formed as

$$\mathcal{L}_k(\mathbf{p}_k(\mathbf{h}_k), \mathbf{y}_k, \boldsymbol{\mu}_k) := \sum_{i=1}^{m} J^i(y_k^i) \qquad (2.11)$$
$$+\boldsymbol{\mu}_k^T\left(\mathbb{E}_{\mathbf{h}_k}\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k)) - \check{\mathbf{y}}_k\right),$$

where $\boldsymbol{\mu}_k := [\mu_k^1; \ldots; \mu_k^m; \tilde{\mu}] \in \mathbb{R}_+^{m+1}$ contains the dual variables associated with each of the $m+1$ constraints in (WCP_k). From the Lagrangian function in (2.11), the Lagrangian dual loss function is defined as $L_k(\boldsymbol{\mu}_k) := \max_{\mathbf{p}_k, \mathbf{y}_k} \mathcal{L}_k(\mathbf{p}_k(\mathbf{h}_k), \mathbf{y}_k, \boldsymbol{\mu}_k)$—see, e.g., [14]—and the corresponding dual problem as

$$\tilde{\boldsymbol{\mu}}_k^* := \operatorname*{argmin}_{\boldsymbol{\mu}_k \geq 0} L_k(\boldsymbol{\mu}_k) \tag{2.12}$$
$$L_k(\boldsymbol{\mu}_k) := \mathbb{E}_{\mathbf{h}_k}\left\{ \max_{\mathbf{p}_k, \mathbf{y}_k} \sum_{i=1}^{m} J^i(y_k^i) + \boldsymbol{\mu}_k^T \left(\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k)) - \check{\mathbf{y}}_k\right) \right\}.$$

Note in (2.12) that the expectation operator and maximization were exchanged without loss of generality—see, e.g. [47, Proposition 2]. It is important to stress here the connection between the dual problem in (2.12) with the original problem in (WCP_k). While (WCP_k) is indeed not convex, problems of this form can be shown to exhibit zero duality gap under the technical assumption that the primal problem is strictly feasible and that the channel probability distribution is non-atomic [103]. This implies that the optimal primal variable $\mathbf{p}_k^*(\mathbf{h}_k)$ in (WCP_k) can be recovered from the optimal dual variable $\tilde{\boldsymbol{\mu}}_k^*$ in (2.12). Thus, the power allocation policy for each agent i is found indirectly by solving (2.12) and recovering as

$$p_k^i(h_k^i, \boldsymbol{\mu}_k) = \operatorname*{argmax}_{p_k^i \in [0, p_0]} \mu_k^i q(h_k^i, p_k^i(h_k^i)) - \tilde{\mu} p_k^i(h_k^i), \tag{2.13}$$
$$y_k^i(\boldsymbol{\mu}_k) = \operatorname*{argmax}_{y_k^i} J^i(y_k^i) - \mu_k^i y_k^i. \tag{2.14}$$

The optimal policy is subsequently recovered using the optimal dual variable as $\mathbf{p}_k^*(\mathbf{h}_k) := [p_k^1(h_k^1, \tilde{\boldsymbol{\mu}}_k^*); \ldots; p_k^m(h_k^m, \tilde{\boldsymbol{\mu}}_k^*)]$. Observe that the problem in (2.12) is a simply constrained stochastic problem that is known to always be convex from duality theory, and can be solved efficiently with a variety of projected stochastic descent methods [12,26,29,47,49]. Thus, the non-convex, infinite-dimensional optimization problem in (WCP_k) can be solved indirectly but exactly with the convex, finite-dimensional problem in (2.12).

Remark 2. The problem formulation given in (WCP_k) that we use in this chapter assumes there is a fixed power budget and the metric to be optimized is a measure of control performance. An alternative formulation of resource allocation that may be more relevant in some settings would instead fix a bound on the required control performance, typically derived from a stability margin for the control system. Here the objective would instead be to minimize total power usage, subject to the constraint on control performance. Indeed, these two problems are very similar when reformulated in the dual domain, and can thus be studied almost identically as such. We specifically focus on the problem in (WCP_k) in this

chapter but stress that all the results will apply to this alternative problem as well.

2.3 ERM Formulation of (WCP_k)

The stochastic program in (2.12) features an objective that is the expectation taken over a random variable, and can thus be considered as a particular case of the empirical risk minimization (ERM) problem. Empirical risk minimization is a common problem studied in machine learning due to its ubiquity in training classifiers, and the same structure appears naturally in the dual formulation of the WCP. A generic ERM problem considers a convex loss function $f(\boldsymbol{\mu}_k, \mathbf{h}_k)$ of a decision variable $\boldsymbol{\mu}_k \in \mathbb{R}^{m+1}$ and random variable $\mathbf{h}_k$ drawn from distribution $\mathcal{H}_k$ and seeks to minimize the expected loss $L_k(\boldsymbol{\mu}_k) := \mathbb{E}_{\mathbf{h}_k}[f(\boldsymbol{\mu}_k, \mathbf{h}_k)]$. For the WCP in (WCP_k), we rewrite the loss function L and associated ERM problem in terms of a function $f(\boldsymbol{\mu}_k, \mathbf{h}_k)$ using its dual as

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_k^* &:= \operatorname*{argmin}_{\boldsymbol{\mu}_k \geq \mathbf{0}} L_k(\boldsymbol{\mu}_k) := \operatorname*{argmin}_{\boldsymbol{\mu}_k \geq \mathbf{0}} \mathbb{E}_{\mathbf{h}_k} f(\boldsymbol{\mu}_k, \mathbf{h}_k), \\ f(\boldsymbol{\mu}_k, \mathbf{h}_k) &:= J(\mathbf{y}_k(\boldsymbol{\mu}_k)) + \boldsymbol{\mu}_k^T \left(\check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}_k(\mathbf{h}_k, \boldsymbol{\mu}_k)) - \check{\mathbf{y}}_k(\boldsymbol{\mu}_k)\right). \end{aligned} \tag{2.15}$$

Typically the distribution $\mathcal{H}_k$ is not known by the user, so the expected loss cannot be evaluated directly, but is instead replaced by an *empirical risk* by taking n samples labeled $\mathbf{h}_k^1, \mathbf{h}_k^2, \ldots, \mathbf{h}_k^n \in \mathcal{H}_k^m$, (where $\mathbf{h}_k^l := [h_k^{1,l}; \ldots; h_k^{m,l}]$). In practice, such samples can be obtained through the use of short pilot signals sent from the users to measure channel conditions—see [48]. We then consider the empirical average loss function

$$\hat{L}_k(\boldsymbol{\mu}_k) := \frac{1}{n} \sum_{l=1}^{n} f(\boldsymbol{\mu}_k, \mathbf{h}_k^l) := \frac{1}{n} \sum_{l=1}^{n} f_k^l(\boldsymbol{\mu}_k). \tag{2.16}$$

To characterize the closeness of the empirical risk $\hat{L}_k(\boldsymbol{\mu}_k)$ with n samples with respect to the expected loss $L_k(\boldsymbol{\mu}_k)$, we define a constant V_n called the *statistical accuracy* of $\hat{L}_k$. The statistical accuracy V_n provides a bound of the difference in the empirical and expected loss for all $\boldsymbol{\mu}_k$ with high probability (i.e. at least $1 - \gamma$ for some small γ). In other words, we define V_n to be the constant that satisfies

$$\sup_{\boldsymbol{\mu}_k} |\hat{L}_k(\boldsymbol{\mu}_k) - L_k(\boldsymbol{\mu}_k)| \leq V_n \quad \text{w.h.p.} \tag{2.17}$$

The upper bounds on V_n are well studied in the learning literature and in general may involve a number of parameters of the loss function f as well as, perhaps most importantly, the number of samples n. For $\hat{L}_k(\boldsymbol{\mu}_k)$ defined in (2.16), a bound for the statistical accuracy V_n can be obtained in the order of $\mathcal{O}(1/\sqrt{n})$ or, in some cases, $\mathcal{O}(1/n)$ [13,122]. This further

implies a suboptimality of $\hat{L}_k^* := \min \hat{L}_k(\boldsymbol{\mu}_k)$ of the same accuracy, i.e. $|L_k^* - \hat{L}_k^*| \leq 2V_n$ [13].

As is often the case in machine learning problems, the statistical accuracy informs the proper use of regularization terms in the empirical loss function. We can add regularizations to prescribe desirable properties on the empirical risk $\hat{L}_k(\boldsymbol{\mu}_k)$, such as strong convexity, without adding additional bias beyond that already accrued by the empirical approximation. In other words, as $\hat{L}_k^*$ will be of order V_n from the optimal expected value L^*, any additional bias of order V_n or less is permissible. With that in mind, we add the regularization term $\alpha V_n/2\|\boldsymbol{\mu}_k\|^2$ where $\alpha > 0$ to the empirical risk in (2.16) to impose strong convexity. We can further remove the non-negativity constraint on the dual variables in (2.15) through the use of a logarithmic barrier. To preserve smoothness for small $\boldsymbol{\mu}_k$, we use an ϵ-thresholded log function, defined as

$$\log_\epsilon(\boldsymbol{\mu}_k) := \begin{cases} \log(\boldsymbol{\mu}_k) & \boldsymbol{\mu}_k \geq \epsilon \\ \ell_{2,\epsilon}(\boldsymbol{\mu}_k - \epsilon) & \boldsymbol{\mu}_k < \epsilon, \end{cases} \tag{2.18}$$

where $\ell_{2,\epsilon}(\boldsymbol{\mu}_k)$ is a second order Taylor series expansion of $\log(\boldsymbol{\mu}_k)$ centered at ϵ for some small $0 < \epsilon < 1$. We then use $-\beta V_n \mathbf{1}^T \log_\epsilon \boldsymbol{\mu}_k$ where $\beta > 0$ as a second regularization term, and obtain a regularized empirical risk function

$$R_k(\boldsymbol{\mu}_k) := \frac{1}{n}\sum_{l=1}^{n} f_k^l(\boldsymbol{\mu}_k) + \frac{\alpha V_n}{2}\|\boldsymbol{\mu}_k\|^2 - \beta V_n \mathbf{1}^T \log_\epsilon \boldsymbol{\mu}_k. \tag{2.19}$$

From here, we can seek a minimizer of the strongly convex regularized risk $R_k(\boldsymbol{\mu}_k)$ without explicitly enforcing a non-negativity constraint on $\boldsymbol{\mu}_k$ and find a solution with suboptimality of order $\mathcal{O}(V_n)$ with respect to (2.15). Such a deterministic and strongly convex loss function as in (2.19) can be minimized using a wide array of optimization methods [26, 34, 64, 85]. However, all such methods only solve the problem for a particular epoch k, or otherwise assume a stationary channel distribution $\mathcal{H}_k$ as is typical in machine learning settings.

2.4 ERM over non-stationary channel

The ERM problem we are interested in solving in wireless autonomous systems is further complicated by the non-stationarity of $\mathcal{H}$, making existing solution methods insufficient. This is due to the fact that finding the minimizer to $R_k(\boldsymbol{\mu})$ will only provide an optimal power allocation for the respective channel distribution $\mathcal{H}_k$. In wireless systems, we instead must *continuously learn* optimal policies as the channel varies, or in other words, find optimal points for $R_k(\boldsymbol{\mu})$ for $k = 0, 1, \ldots$. To formulate the non-stationarity, however, we first define an epoch-indexed empirical risk function. While we may use a simple empirical risk as we

did in (2.16), we instead define a more general statistical loss function for a non-stationary channel using samples from the previous M epochs. We define a windowed empirical loss function $\tilde{L}_k(\boldsymbol{\mu})$ at epoch k as

$$\tilde{L}_k(\boldsymbol{\mu}) := \frac{1}{M} \sum_{j=k-M+1}^{k} \hat{L}_j(\boldsymbol{\mu}) \tag{2.20}$$

By keeping a window of samples, we may retain $N = Mn$ total samples while drawing only n new samples at each epoch. If the successive channel distributions $\mathcal{H}_{k-M+1}, \dots, \mathcal{H}_k$ are not very different, we may expect the old channel samples to still be of interest. We define the associated statistical accuracy $\tilde{V}_N$ as the constant that satisfies

$$\sup_{\boldsymbol{\mu}} |\tilde{L}_k(\boldsymbol{\mu}) - L_k(\boldsymbol{\mu})| \leq \tilde{V}_N \quad \text{w.h.p.} \tag{2.21}$$

Here we stress that the bounds on this constant $\tilde{V}_N$ are not as easily obtainable or well-studied as in the stationary setting. Such a bound over non-i.i.d. samples may be dependent upon many parameters such as the sample batch size n, window size M, and correlation between successive distributions $\mathcal{H}_j$ and $\mathcal{H}_{j+1}$. Therefore, finding precise bounds on $\tilde{V}_N$ would require a sophisticated statistical analysis and is outside the scope of this work. We instead define a user-selected accuracy $\hat{V}$ that may estimate the statistical accuracy $\tilde{V}_N$. We assume that $\hat{V} \geq \tilde{V}_N$, with equality holding in cases where $\tilde{V}_N$ is known. Using the same regularizations introduced previously, we obtain the regularized windowed empirical loss function

$$\tilde{R}_k(\boldsymbol{\mu}) := \frac{1}{M} \sum_{j=k-M+1}^{k} \hat{L}_j(\boldsymbol{\mu}) + \frac{\alpha \hat{V}}{2} \|\boldsymbol{\mu}\|^2 - \beta \hat{V} \mathbf{1}^T \log_\epsilon \boldsymbol{\mu}. \tag{2.22}$$

We subsequently define $\tilde{\boldsymbol{\mu}}_k^* := \operatorname{argmin}_{\boldsymbol{\mu}} \tilde{R}_k(\boldsymbol{\mu})$. The definition of the loss function in (2.22) includes the batches of n samples taken from the previous M channel distributions $\mathcal{H}_{k-M+1}, \dots, \mathcal{H}_k$. This definition is, in a sense, a generalization of the simpler empirical risk $R_k(\boldsymbol{\mu})$ in (2.19). Observe that, by using a window size of $M = 1$, we use only samples from the current channel and recover $R_k(\boldsymbol{\mu})$. In the following proposition we establish the accuracy of an optimal point of our regularized empirical risk function $\tilde{R}_k(\boldsymbol{\mu})$ relative to the optimal point of the original dual loss function $L_k(\boldsymbol{\mu})$.

Proposition 1. *Consider $L_k^* = L_k(\boldsymbol{\mu}_k^*)$ and $\tilde{L}_k^* = \min_{\boldsymbol{\mu} \geq \mathbf{0}} \tilde{L}_k(\boldsymbol{\mu})$, and define $\tilde{R}_k^* := \min_{\boldsymbol{\mu}} \tilde{L}_k(\boldsymbol{\mu}) + \alpha \hat{V}/2 \|\boldsymbol{\mu}\|^2 - \beta \hat{V} \mathbf{1}^T \log \boldsymbol{\mu}$ as the optimal value of the regularized empirical risk. Define $\tilde{V}_N$ by (2.21). Assuming $\tilde{V}_N \leq \hat{V}$, the difference $|L_k^* - \tilde{R}_k^*|$ is upper bounded on*

the order of statistical accuracy $\hat{V}$, i.e. for some $\rho > 0$

$$|L_k^* - \tilde{R}_k^*| \leq 2\tilde{V}_N + \rho\hat{V} \leq (2+\rho)\hat{V}, \quad w.h.p. \tag{2.23}$$

Proof: To obtain the result in (2.23), consider expanding and upper bounding $|L_k^* - \tilde{R}^*| = |L_k^* - \tilde{L}_k^* + \tilde{L}_k^* + \tilde{R}_k^*| \leq |L_k^* - \tilde{L}_k^*| + |\tilde{L}_k^* + \tilde{R}_k^*|$. The first term is bounded by $2\tilde{V}_N$ as previously discussed. The second term, can be decomposed into the bias introduced by the logarithmic barrier $-\beta\hat{V}\mathbf{1}^T \log \boldsymbol{\mu}$ and the bias introduced by the quadratic regularizer $c\hat{V}/2\|\boldsymbol{\mu}\|^2$. The former of these is known to produce an optimality bias of $(m+1)\beta\hat{V}$ [14, Section 11.2.2], while the latter is known to introduce a bias on the order of $\mathcal{O}(\hat{V})$ [108]. Combining these, we get a total suboptimality between the regularized risk function optimal and the true optimal of $2\tilde{V}_N + \rho\hat{V}$ for some constant $\rho > 0$. As we assume that $\tilde{V}_N \leq \hat{V}$, the rightmost bound in (2.23) follows. ■

A key observation to be made here is that any exact solution to (2.22) only minimizes the expected loss L_k to within accuracy $\hat{V}$ (assuming $\hat{V} \geq \tilde{V}_N$). There is therefore no need to minimize (2.22) exactly but is in fact sufficient to find a $\hat{V}$-accurate solution, as this incurs no additional error relative to the statistical approximation itself. While many optimization methods can be used to find a minimizer to (2.22), we demonstrate in the next section that fast second order methods can be used to learn *approximate* minimizers—and by Proposition 1 approximately solve (2.15)—at each epoch k with just single updates as the channel distribution $\mathcal{H}_k$ changes, thus tracking near-optimal points at every epoch. This is done by exploiting an important property of second order optimization methods, namely *local quadratic convergence.*

Remark 3. Observe in the text of Proposition 1 that we define $\tilde{R}_k^*$ to be the optimal point of the loss function $\tilde{L}_k(\boldsymbol{\mu}_k)$ regularized with a standard log barrier $-\log(\boldsymbol{\mu}_k)$, rather than the thresholded barrier $-\log_\epsilon(\boldsymbol{\mu}_k)$ used in the definition in (2.22). Indeed, using the thresholded barrier does not explicitly enforce nonnegativity for values smaller than ϵ. However, this thresholding is necessary to preserve smoothness of the barrier, which will be necessary for the proof of Lemma 1 in Section 2.5. The threshold ϵ can be made as small as necessary to enforce nonnegativity, although this comes at the cost of a worse smoothness constant. In practice, however, we observe this thresholding to not be explicitly needed and is just included here for ease of analysis. We also stress that the smoothness constant itself does not play a pivotal role in the proceeding analysis.

2.4.1 Learning via Newton's Method

In this chapter, we use Newton's method to approximately minimize (2.22) efficiently as the channel $\mathcal{H}_k$ changes over epochs. Motivated by the recent use of Newton's method in

solving large scale ERM problems through adaptive sampling policies [34, 85], we use the N samples drawn from recent distributions to find an iterate $\boldsymbol{\mu}_k$ that approximately solves for $\tilde{\boldsymbol{\mu}}_k^*$. At the next epoch, the iterate $\boldsymbol{\mu}_k$ provides a "soft" start towards finding a point $\boldsymbol{\mu}_{k+1}$ that approximately minimizes $\tilde{R}_{k+1}(\boldsymbol{\mu})$. In this way, with single iterations we may find near-optimal solutions for each regularized empirical loss function, and thereby efficiently learn the optimal power allocation of the wireless channel as the channel distribution evolves over time epochs.

We proceed by presenting the details of Newton's method. At epoch k, we compute a new iterate $\boldsymbol{\mu}_{k+1}$ by subtracting from the current iterate $\boldsymbol{\mu}_k$ the product of the Hessian inverse and the gradient of the function $\tilde{R}_{k+1}(\boldsymbol{\mu}_k)$. For the empirical dual loss function $\tilde{R}_k$ defined in (2.22), we define the gradient $\nabla\tilde{R}_k(\boldsymbol{\mu})$ and Hessian $\nabla^2\tilde{R}_k(\boldsymbol{\mu})$. The new approximate solution $\boldsymbol{\mu}_{k+1}$ is then found from current approximate solution $\boldsymbol{\mu}_k$ using the Newton update

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \mathbf{H}_{k+1}^{-1}\nabla\tilde{R}_{k+1}(\boldsymbol{\mu}_k), \tag{2.24}$$

where we use $\mathbf{H}_{k+1} := \nabla^2\tilde{R}_{k+1}(\boldsymbol{\mu}_k)$ as simplified notation.

To understand the full algorithm, consider that $\boldsymbol{\mu}_k$ is a $\hat{V}$-accurate solution of current loss function $\tilde{R}_k$, i.e. $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}$. Recall that the new loss function $\tilde{R}_{k+1}$ differs from $\tilde{R}_k$ only in the discarding of old samples $\hat{L}_{k-M+1}$ and inclusion of samples $\hat{L}_{k+1}$ drawn from $\mathcal{H}_{k+1}$. If we consider that the distributions are varying slowly across successive time epochs, i.e. $\mathcal{H}_{k+1}$ is close to $\mathcal{H}_k$, then the respective loss functions $\tilde{R}_{k+1}$ and $\tilde{R}_k$ and their optimal values $\tilde{R}_{k+1}^*$ and $\tilde{R}_k^*$ will also not differ greatly under some smoothness assumptions. Therefore, under such conditions a single step of Newton's method as performed in (2.24) can in fact be sufficient to reach a $\hat{V}$-accurate solution of the new loss function $\tilde{R}_{k+1}$. This is possible precisely because of the Newton method's property of *local quadratic convergence*, meaning that Newton's method will find a near-optimal solution very quickly when it is already in a local neighborhood of the optimal point. Given then a $\hat{V}$-accurate solution $\boldsymbol{\mu}_0$ of initial loss $\tilde{R}_0$, the proceeding and all subsequent iterates $\boldsymbol{\mu}_k$ will remain within the statistical accuracy of their respective losses $\tilde{R}_k$ as the channel distribution varies over time. The formal presentation of the exploitation of this property and other technical details of this result are discussed in Section 2.5 of this chapter.

The learning algorithm is presented in Algorithm 1. After preliminaries and initializations in Steps 1-4, the backtracking loop starts in Step 5. Each iteration begins in Step 6 with the the drawing of n samples from the new channel distribution $\mathcal{H}_k$ and discarding of old samples from $\mathcal{H}_{k-M}$ to form $\tilde{R}_k$. Note that samples will be only be discarded for $k > M$. The gradient $\nabla\tilde{R}_k$ and Hessian $\mathbf{H}_k$ of the regularized dual loss function are computed in Step 7. The Newton step is taken with respect to $\tilde{R}_{k+1}$ in Step 8. In Step 9, the optimal primal variables are computed with respect to the updated dual variables. This includes

Algorithm 1 Learning via Newton's Method

1: **Parameters:** Sample size increase constants $n_0 > 0$, $M_0 \geq 1$ backtracking params $0 < \gamma < 1 < \Gamma$, and accuracy $\hat{V}$.
2: **Input:** Initial sample size $n = m_0$ and argument $\boldsymbol{\mu}_n = \boldsymbol{\mu}_{m_0}$ with $\|\nabla \tilde{R}_n(\boldsymbol{\mu}_{k+1})\| < (\sqrt{2\alpha})\hat{V}$
3: **for** [**do**main loop]$k = 0, 1, 2, \ldots$
4: Reset factor $n = n_0$, $M = M_0$.
5: **repeat**[sample size backtracking loop]
6: Draw n samples from $\mathcal{H}_k$, discard from $\mathcal{H}_{k-M}$.
7: Compute Gradient $\nabla \tilde{R}_k(\boldsymbol{\mu}_{k-1})$, Hessian $\mathbf{H}_k$.
8: Newton Update [cf. (2.24)]:

$$\boldsymbol{\mu}_k = \boldsymbol{\mu}_{k-1} - \mathbf{H}_k^{-1} \nabla \tilde{R}_k(\boldsymbol{\mu}_{k-1})$$

9: Determine power allocation, aux. variables [cf. (2.13), (2.14)]:

$$p_k^i(h_k^i, \boldsymbol{\mu}_k) = \underset{p_k^i \in [0, p_0]}{\operatorname{argmax}} \, \mu_k^i q(h_k^i, p_k^i(h_k^i)) - \tilde{\mu} p_k^i(h_k^i),$$
$$y_k^i(\boldsymbol{\mu}_k) = \underset{y_k^i}{\operatorname{argmax}} \, J^i(y_k^i) - \mu_k^i y_k^i.$$

10: Backtrack sample draw $n = \Gamma n$, window size $M = \gamma M$.
11: **until** $\|\nabla \tilde{R}_k(\boldsymbol{\mu}_k)\| < (\sqrt{2\alpha})\hat{V}$
12: **end for**

both the auxiliary variables $\mathbf{y}(\boldsymbol{\mu}_k)$ and the power allocation policy $\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)$ itself. Because there are function and channel system parameters that are not known in practice, we include a backtracking step for the parameters n and M in Step 10 to ensure the new iterate $\boldsymbol{\mu}_k$ is within the intended accuracy $\hat{V}$ of $\boldsymbol{\mu}_k^*$. Further details on the specifics of the backtracking procedure are discussed in Section 2.6 after the presentation of the theoretical results.

2.5 Convergence Analysis

In this section we provide a theoretical analysis of the Newton learning update in (2.24). We do so by first analyzing the convergence properties of the ERM problem in (2.22). We subsequently return to the WCP in (WCP_k) and establish a control performance result.

2.5.1 Convergence of ERM problem

We begin by analyzing the ERM formulation of the power allocation problem in (2.22) and establish a theoretical result that, under certain conditions, guarantees each iterate $\boldsymbol{\mu}_k$ is within the statistically accuracy of the risk function at epoch k. Our primary theoretical result characterizes such conditions dependent on statistical accuracy and rate of non-stationarity. We begin by presenting a series of assumptions made in our analysis regarding the dual loss functions f.

AS1. *The expected loss function L_k and empirical loss functions $f(\boldsymbol{\mu}, \mathbf{h}_k)$ are convex with respect to $\boldsymbol{\mu}$ for all values of $\mathbf{h}_k$. Moreover, their gradients $\nabla L_k(\boldsymbol{\mu})$ and $\nabla f(\boldsymbol{\mu}, \mathbf{z})$ are Lipschitz continuous with constant Δ.*

AS2. *The loss functions $f(\boldsymbol{\mu}, \mathbf{h})$ are self-concordant with respect to $\boldsymbol{\mu}$ for all $\mathbf{h}$, i.e. for all i,*

$$|\partial^3/\partial\mu_i^3 f(\boldsymbol{\mu}, \mathbf{h})| \leq 2\partial^2/\partial\mu_i^2 f(\boldsymbol{\mu}, \mathbf{h})^{3/2}.$$

Assumption 1 implies that the regularized empirical risk gradients $\nabla\tilde{R}_k$ are Lipschitz continuous with constant $\Delta + c\hat{V}$ where $c := \alpha + \beta/\epsilon^2$ and α, β, ϵ are the regularization constants in (2.22). The function $\tilde{R}_k$ is also strongly convex with constant $\alpha\hat{V}$. This implies an upper and lower bound of the eigenvalues of the Hessian of $\tilde{R}_k$, namely

$$\alpha\hat{V}\mathbf{I} \preceq \mathbf{H}_k \preceq (\Delta + c\hat{V})\mathbf{I}. \tag{2.25}$$

Assumption 2 states the loss functions are additionally self concordant, which is a common assumption made in the analysis of second-order methods—see, e.g. [14, Ch. 9], for such an analysis. It also follows that the functions $\tilde{R}_{k+1}$ are therefore self concordant because both the quadratic and thresholded log regularizers are self-concordant. We present a brief remark regarding the implications of these assumptions on the dual risk function on the wireless control problem.

Remark 4. We state the preceding assumptions in terms of the sampled dual functions f due to their direct use in the proceeding analysis. However, they indeed have implications on the primal domain problem in (WCP_k). While the dual function is always convex, the smoothness condition in Assumption 1 can be obtained from the strong concavity of the control performance $\sum_i J^i$ with strong concavity $1/\Delta$. The self-concordance property on the dual function in Assumption 2, however, is not easily derived from properties of $J^i(\cdot)$ or $q(\cdot)$. We point to work that establishes self concordance of the dual for various machine learning problems [88, 95].

The two preceding assumptions deal specifically with the properties of the empirical dual loss functions used in the ERM problem. To connect the solving of the sampled functions f^l with the expected loss function L, we additionally include two assumptions regarding the statistics of the expected and empirical losses.

AS3. *The difference between the gradients of the empirical risk $\hat{L}_k$ and the statistical average loss L_k is bounded by $V_N^{1/2}$ for all $\boldsymbol{\mu}$ and k with high probability,*

$$\sup_{\boldsymbol{\mu}} \|\nabla L_k(\boldsymbol{\mu}) - \nabla\hat{L}_k(\boldsymbol{\mu})\| \leq V_N^{1/2}, \qquad \textit{w.h.p.} \tag{2.26}$$

AS4. *The difference between two successive expected loss functions* $L_k(\boldsymbol{\mu}) = \mathbb{E}_{h_k} f(\boldsymbol{\mu}, h_k)$ *and* $L_{k+1}(\boldsymbol{\mu}) = \mathbb{E}_{h_{k+1}} f(\boldsymbol{\mu}, h_{k+1})$ *and the difference between their gradients are bounded respectively by a bounded sequence of constants* $\{D_k\}, \{\bar{D}_k\} \geq 0$ *for all* $\boldsymbol{\mu}$*,*

$$\sup_{\boldsymbol{\mu}} |L_k(\boldsymbol{\mu}) - L_{k+1}(\boldsymbol{\mu})| \leq D_k, \tag{2.27}$$

$$\sup_{\boldsymbol{\mu}} \|\nabla L_k(\boldsymbol{\mu}) - \nabla L_{k+1}(\boldsymbol{\mu})\| \leq \bar{D}_k. \tag{2.28}$$

Assumption 3 bounds the difference between gradients of the expected loss and the empirical risk with N samples by $V_N^{1/2}$, which can be readily obtained using the law of large numbers. Assumption 4 bounds the point-wise difference in the expected loss functions and their gradients at epochs k and $k+1$. This can be interpreted as the rate at which the channel evolves between epochs, and is used to establish that optimal dual variables for two consecutive empirical risk functions $\tilde{R}_k$ and $\tilde{R}_{k+1}$ are not very different. We discuss practical implications of this assumption in Section 2.6.

Remark 5. Observe that the bounds provided in Assumption 4 are with respect to the dual function rather than explicitly on the non-stationary statistics of the channel. They are provided as such because this is the manner in which the non-stationarity appears in the proceeding analysis. To see how the channel characteristics play a role in the provided bound, consider that, e.g., (2.27) can be expanded using the definition of the dual function $L_k(\boldsymbol{\mu})$ as

$$\begin{aligned} \sup_{\boldsymbol{\mu}} |\mathbb{E}_{\mathbf{h}_k} \left\{ \max_{\mathbf{p} \in \mathcal{P}} \boldsymbol{\mu}^T \check{\mathbf{q}}(\mathbf{h}_k, \mathbf{p}(\mathbf{h}_k)) \right\} \\ - \mathbb{E}_{\mathbf{h}_{k+1}} \left\{ \max_{\mathbf{p} \in \mathcal{P}} \boldsymbol{\mu}^T \check{\mathbf{q}}(\mathbf{h}_{k+1}, \mathbf{p}(\mathbf{h}_{k+1})) \right\} | \leq D_k. \end{aligned} \tag{2.29}$$

The exact condition this imposes upon the channel distribution variation thus depends both on the form of the distributions $\mathcal{H}_k$, $\mathcal{H}_{k+1}$, and the function $q(\mathbf{h}, \mathbf{p})$. Thus, the exact manner in which the varying channel conditions effect this bound are indeed problem-specific, and a generic condition on non-stationarity of the channel is only present in the proceeding analysis indirectly through the condition in (2.29).

The proceeding analysis is organized in the following manner. Our goal is to establish conditions on the parameters of the statistical accuracy—$\hat{V}$—and the non-stationarity—D_k and $\bar{D}_k$—that guarantee that, starting from an approximate solution to $\tilde{R}_k$, a single step of Newton's method generates an approximately accurate solution to $\tilde{R}_{k+1}$. From there, we can recursively say that, assuming an initial point $\boldsymbol{\mu}_0$ that is within the intended accuracy

of $\tilde{R}_0$, the method will continue to find a $\hat{V}$-accurate solution at each epoch as the channel distribution changes with k. We achieve this result in two steps. We first find a condition that guarantees that a $\hat{V}$-accurate solution of $\tilde{R}_k$ is also in the quadratic convergence region of $\tilde{R}_{k+1}$. Second, we find a condition that guarantees that such a point within the quadratic convergence region of $\tilde{R}_{k+1}$ will reach its intended accuracy with a single update as in (2.24).

We begin by establishing the condition in the first step, namely that a $\hat{V}$-accurate solution to $\tilde{R}_k$, labeled $\boldsymbol{\mu}_k$ is in in the quadratic convergence region of $\tilde{R}_{k+1}$ if certain conditions hold. The quadratic convergence region is a region local to the optimum in which Newton's method is known to converge at a fast quadratic rate. The analysis of Newton's method commonly characterizes quadratic convergence in terms of a quantity called the Newton decrement, explicitly defined as $\lambda_{k+1}(\boldsymbol{\mu}) := \|\nabla^2 \tilde{R}_{k+1}(\boldsymbol{\mu})^{-1/2} \nabla \tilde{R}_{k+1}(\boldsymbol{\mu})\|$. We say the dual iterate $\boldsymbol{\mu}$ is in the quadratic convergence region of $\tilde{R}_{k+1}$ when $\lambda_{k+1}(\boldsymbol{\mu}) < 1/4$—see [14, Chapter 9.6.4]. In the following proposition, we give conditions under which any iterate $\boldsymbol{\mu}_k$ that is within the accuracy $\hat{V}$ of the optimal point $\tilde{R}_k^* = \min_\mu \tilde{R}_k(\mu)$ is also within the quadratic convergence region of the subsequent loss function $\tilde{R}_{k+1}$.

Lemma 1. *Consider $\boldsymbol{\mu}_k$ as a $\hat{V}$-accurate optimal solution of the loss $\tilde{R}_k$, i.e., $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}$. In addition, define $\lambda_{k+1}(\boldsymbol{\mu}) := \left(\nabla \tilde{R}_{k+1}(\boldsymbol{\mu})^T \nabla^2 \tilde{R}_{k+1}(\boldsymbol{\mu})^{-1} \nabla \tilde{R}_{k+1}(\boldsymbol{\mu})\right)^{1/2}$ as the Newton decrement of variable $\boldsymbol{\mu}$ associated with the loss $\tilde{R}_{k+1}$. If Assumptions 1-4 hold, then Newton's method at point $\boldsymbol{\mu}_k$ is in the quadratic convergence phase for the objective function $\tilde{R}_{k+1}$, i.e., $\lambda_{k+1}(\boldsymbol{\mu}_k) < 1/4$, if we have*

$$\left(\frac{2(\Delta + c\hat{V})\hat{V}}{\alpha \hat{V}}\right)^{1/2} + \frac{2\tilde{V}_N^{1/2} + \bar{D}_k}{(\alpha \hat{V})^{1/2}} < \frac{1}{4}. \quad \textit{w.h.p.} \tag{2.30}$$

Proof: See Appendix. ■

Lemma 1 provides the first necessary condition in our analysis by identifying the statistical parameters under which every iterate $\boldsymbol{\mu}_k$ is in the quadratic region of $\tilde{R}_{k+1}$. From here we can show the second step, in which such a point in the quadratic convergence region of $\tilde{R}_{k+1}$ can reach its statistical accuracy with a single Newton step as given in (2.24). To achieve this, we first present the following lemma that upper bounds the sub-optimality of the point $\boldsymbol{\mu}_k$ with respect to the optimal solution of R_{k+1}^*.

Lemma 2. *Consider a point $\boldsymbol{\mu}_k$ that minimizes the loss function $\tilde{R}_k$ to within accuracy $\hat{V}$, i.e. $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}$. Provided that Assumptions 1-4 hold, the sub-optimality $\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_{k+1}^*$ is upper bounded w.h.p. as*

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_{k+1}^* \leq 4\tilde{V}_N + \hat{V} + 2D_k \tag{2.31}$$

Proof: See Appendix. ■

In Lemma 2 we establish a bound on the suboptimality of $\boldsymbol{\mu}_k$ with respect to $\tilde{R}_{k+1}$. The following lemma now bounds the suboptimality of $\boldsymbol{\mu}_{k+1}$ in terms of the suboptimality of $\boldsymbol{\mu}_k$ with a quadratic rate.

Lemma 3. *Consider $\boldsymbol{\mu}_k$ to be in the quadratic neighborhood of the loss $\tilde{R}_{k+1}$, i.e., $\lambda_{k+1}(\boldsymbol{\mu}_k) \leq 1/4$. Recall the definition of the variable $\boldsymbol{\mu}_{k+1}$ in (2.24) as the updated variable using Newton's method. If Assumptions 1-3 hold, then the difference $\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1}$ is upper bounded by*

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1} \leq 144(\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}^*_{k+1})^2. \tag{2.32}$$

Proof: See Appendix. ■

With Lemma 3 we establish the known quadratic rate of convergence of the suboptimality of the Newton update in (2.24). Now by substituting the upper bound on $\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}^*_{k+1}$ from Lemma 2, a condition can easily be derived under which the suboptimality of the new iterate is within the accuracy $\hat{V}$ of $\tilde{R}_{k+1}$. Using the results of Lemmata 1-3, we present our main result in the following theorem.

Theorem 1. *Consider Newton's method defined in (2.24) and the full learning method detailed in Algorithm 1. Define $\tilde{V}_N$ to be the statistical accuracy of $N = Mn$ samples by (2.21), with n samples taken from each of the M most recent channel distributions $\mathcal{H}_k$. Further consider the variable $\boldsymbol{\mu}_k$ as a $\hat{V}$-optimal solution of the loss $\tilde{R}_k$, and suppose Assumptions 1-4 hold. If the sample size n and window size M are chosen such that the following conditions*

$$\left(\frac{2(\Delta + c\hat{V})\hat{V}}{\alpha\hat{V}}\right)^{1/2} + \frac{2\hat{V}^{1/2} + \bar{D}_k}{(\alpha\hat{V})^{1/2}} < \frac{1}{4} \tag{2.33}$$

$$144(4\tilde{V}_N + \hat{V} + 2D_k)^2 \leq \hat{V} \tag{2.34}$$

are satisfied, then the variable $\boldsymbol{\mu}_{k+1}$ computed from (2.24) has the suboptimality of $\hat{V}$ with high probability, i.e.,

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1} \leq \hat{V}, \qquad w.h.p. \tag{2.35}$$

The inequalities 2.33-2.34 in Theorem 1 specify conditions under which $\boldsymbol{\mu}_{k+1}$ as generated by (2.24) is a $\hat{V}$-optimal solution of $\tilde{R}_{k+1}$. Note that these conditions come directly from the preceding lemmata. Thus, when these conditions are satisfied, single iterations of Newton's method at each epoch k—as detailed in Algorithm 1—successively generate approximately optimal dual parameters. A further discussion of the satisfaction of such conditions in regards to practical implementation is provided later in Section 2.6. We first extend the theoretical result of Theorem 1 to establish properties of the resulting WCP solution.

Remark 6. Observe in Theorem 1 that the provided conditions cannot be satisfied if the true statistical accuracy $\tilde{V}_N$ is greater than the selected $\hat{V}$. While we assume in our analysis this is not the case, (i.e. $\hat{V}$ is a conservative estimate of $\tilde{V}_N$), this may not be guaranteed if very little information is known about V_N. In the case $\hat{V} < V_N$, we point out that the results in Theorem 1 can simply be modified by replacing achieved accuracy $\hat{V}$ by V_N. In other words, the accuracy we can achieve is limited by the greater of these terms. We do not go through the details of this analysis for clarity of presentation, but such result can be obtained through the same steps of the preceding analysis.

2.5.2 Sub-optimality in wireless control system

Because the proposed Newton method indeed solves (2.15) to within a statistical approximation $\hat{V}$, it is important to consider the effect of such an approximation on the original WCP in (WCP_k). In this section we provide a sequence of results that characterize the accuracy of the solutions generated by the Newton update in (2.24) in the original primal control problem in (WCP_k). Firstly, recall the constraints in (WCP_k) reflect both a power budget limited by $p_{\max}$ and that the auxiliary variable y^i should not exceed the expected packet success function $q(\cdot)$. In solving the dual problem approximately, we may then also violate these constraints by a small margin. We can specifically characterize such a constraint violation, as well as address the suboptimality in terms of the primal objective. Both these results together can then be combined to demonstrate the stability of the switched system WCP introduced in Example 1. To do so, we first introduce an assumption regarding the feasibility and boundedness of the dual loss solutions L_k^* and the optimal dual point $\boldsymbol{\mu}_k^*$.

AS5. *For all epochs k, the problem in* (WCP_k) *under distribution $\mathcal{H}_k$ is strictly feasible. There also exists constants $\mathcal{K}$ and $\hat{\mathcal{K}}$ such that the optimal dual objective value L_k^* is bounded as $L_k^* \leq \mathcal{K}$ and optimal dual variable bounded as $\|\boldsymbol{\mu}_k^*\| \leq \hat{\mathcal{K}}$.*

From strict feasibility of the primal problem in (WCP_k), we also obtain a finite upper bound on the value of the dual function. This can be used with the suboptimality result in Theorem 1 to bound the norm of the dual variables $\boldsymbol{\mu}_k$ generated from the Newton update in (2.24). This is presented in the following corollary.

Corollary 1. *The norm of the dual variables $\boldsymbol{\mu}_k$ generated by the update in* (2.24) *is bounded as $\|\boldsymbol{\mu}_k\| \leq \sqrt{(2/\alpha)} + \hat{\mathcal{K}}$.*

Proof: From strong convexity we have that $\|\boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}}_k^*\|^2 \leq (2/\alpha\hat{V})(\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^*)$. Using the reverse triangle inequality with 2.35 and Assumption 5, we obtain the intended result. ■

Observe that the boundedness of the solutions to the regularized dual function in Assumption 5 in effect states that, for all distributions $\mathcal{H}_k$, the empirical, or sampled,

versions of the constrained problem in (WCP_k) will be strictly feasible. From here, we can establish through duality a bound on each constraint violation that may occur from solving the dual problem to its statistical accuracy. This result is stated in the following proposition.

Proposition 2. *Consider $\boldsymbol{\mu}_k$ to be a $\hat{V}$-optimal minimizer of $\tilde{R}_k$, i.e. $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}$. Further consider $\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)$ and $\mathbf{y}(\boldsymbol{\mu}_k)$ to be the Lagrangian maximizers over dual parameter $\boldsymbol{\mu}_k$. If Assumptions 1 and 5 hold, then the norm of the constraint violations in (WCP_k) can each be upper bounded as*

$$\left| \sum_{i=1}^{m} \mathbb{E}_{h_k^i}(p^i(h_k^i, \boldsymbol{\mu})) - p_{\max} \right| \leq \sqrt{2\Delta(\tilde{V}_N + C\hat{V})}, \tag{2.36}$$

$$\|\mathbf{y}(\boldsymbol{\mu}_k) - \mathbb{E}_{\mathbf{h}_k}\{\mathbf{q}(\mathbf{h}_k, \mathbf{p}(\mathbf{h}_k, \boldsymbol{\mu}_k))\}\| \leq \sqrt{2\Delta(\tilde{V}_N + C\hat{V})}, \tag{2.37}$$

where $C := 1 + \rho + \beta\kappa$ and κ such that $\mathbf{1}^T \log_\epsilon(\boldsymbol{\mu}_k) \leq \kappa$.

Proof: See Appendix. ■

In Proposition 2, we establish a bound that is proportional to $\hat{V}$ on the violation of the constraints in (WCP_k). There are two points to be stressed here. First, is that this constraint violation can indeed be made small by controlling the target accuracy $\hat{V}$. Additionally, we point out that the violation of the budget constraint can be controlled by adding a slack term to the maximum power as $\hat{p}_{\max} = p_{\max} - 2\Delta C\hat{V}$. In this way, any such violation will still be within the true intended budget $p_{\max}$.

We proceed by establishing suboptimality of the generated variables $\mathbf{y}(\boldsymbol{\mu}_k)$ in terms of control performance. Recall the final result in Theorem 1 that establishes at each epoch k, the current dual function value $\tilde{R}_k(\boldsymbol{\mu}_k)$ will be within accuracy $\hat{V}$ of the optimal value $\tilde{R}_k(\tilde{\boldsymbol{\mu}}_k^*)$ (after satisfying the necessary conditions). To establish that the control systems induced by such dual parameters $\boldsymbol{\mu}_k$ remain stable, we first connect the accuracy of the dual function value to the accuracy of associated primal variables $\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)$ and $\mathbf{y}(\boldsymbol{\mu}_k)$ with respect to their optimal values $\mathbf{p}_k^*(\mathbf{h}) := \mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k^*)$ and $\mathbf{y}_k^* := \mathbf{y}(\tilde{\boldsymbol{\mu}}_k^*)$. This bound is established in the following theorem.

Theorem 2. *Consider $\boldsymbol{\mu}_k$ to be a $\hat{V}$-optimal minimizer of $\tilde{R}_k$, i.e. $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}$. Further consider $\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)$ and $\mathbf{y}(\boldsymbol{\mu}_k)$ to be the Lagrangian maximizers over dual parameter $\boldsymbol{\mu}_k$. Under Assumptions 1-5 the primal objective function sub-optimality $J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*)$ can be upper bounded as*

$$J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*) \leq (1 + C)\Delta\left(\frac{1}{\alpha} + 2\hat{V}(\sqrt{2/\alpha} + \hat{\mathcal{K}})\right). \tag{2.38}$$

Proof: See Appendix. ■

In Theorem 2, we derive a bound on the suboptimality of the primal objective function $J(\mathbf{y})$ that is proportional also to the statistical accuracy $\hat{V}$ plus a constant. Recall that this function is, in general, a measure of the control performance of the system. Thus, solving the dual problem approximately indeed can be translated into approximate accuracy in terms of our original utility metric with respect to the control system. In many problems, the performance $J(\mathbf{y})$ will also effectively establish a stability margin for control systems that have unstable regions of operation. To demonstrate the effect of using the proposed Newton's method over a non-stationary wireless channel, we return to the switched dynamical system in Example 1.

2.5.3 Stability of switched dynamical system (Example 1)

Consider the switched dynamical system given in (2.4) and the derived performance metric $J(\mathbf{y})$ in (2.9) that tracks the asymptotic behavior of the state x_t. In this system, if the open loop gain is unstable $|A_o| > 1$ it can indeed cause the system to grow in an unstable manner if the system is not closed sufficiently often. As mentioned in Example 1 the system reaches instability if $yA_c^2 + (1-y)A_o^2$ becomes close to 1. A question of interest in this example is, using the power allocation policy found using Newton's method over a time-varying channel, whether or not the system remains stable over time. We can indeed demonstrate this to be true with the following argument.

From Theorem 2, we obtained that the primal suboptimality with respect to the control performance function $J(\mathbf{y})$ is bounded by a term proportional to $\hat{V}$. Assuming that $J(\mathbf{y}_k^*)$ is finite for all epochs k, it follows then that the generated performance $J(\mathbf{y}(\boldsymbol{\mu}_k))$ is also finite. Considering the expression for $J^i(y^i)$ given in (2.9), this is finite if and only if the denominator is positive, i.e., there exists a ω such that

$$1 - y^i(\boldsymbol{\mu}_k)((A_o^i)^2 - (A_c^i)^2) \leq \omega < 1 \tag{2.39}$$

at all epochs k.

Moreover from Proposition 2 we also have that the actual packet success rate during epoch k satisfies

$$\mathbb{E}_{h_k}\left\{q(h_k^i, p(h_k^i, \mu_k))\right\} \geq y^i(\boldsymbol{\mu}_k) - \sqrt{2\Delta(\tilde{V}_N + C\hat{V})}, \tag{2.40}$$

If the statistical accuracy at the right hand side of this expression is sufficiently small, then using (2.39) we also get that

$$1 - \mathbb{E}_{h_k}\left\{q(h_k^i, p(h_k^i, \mu_k))\right\}((A_o^i)^2 - (A_c^i)^2) \leq \tilde{\omega} < 1 \tag{2.41}$$

In particular this holds if $\sqrt{2\Delta(\tilde{V}_N + C\hat{V})}((A_o^i)^2 - (A_c^i)^2) < 1 - \omega$.

Substituting (2.41) back into the recursive expression in (2.6), we get that the variance of the state at each time step satisfies

$$\mathbb{E}(x_{t+1}^i)^2 \leq \tilde{\omega}\mathbb{E}(x_t^i)^2 + W^i. \tag{2.42}$$

Operating recursively and using the geometric series as in Example 1, we can bound (2.42) as

$$\mathbb{E}(x_{t+1}^i)^2 \leq \tilde{\omega}^{t+1}\mathbb{E}(x_0^i)^2 + W^i\frac{1-\tilde{\omega}^{t+1}}{1-\tilde{\omega}}. \tag{2.43}$$

As both terms on the right hand side of (2.43) are finite, we can conclude that the state variables remain bounded in variance for all t in the non-stationary channel.

2.6 Details of Implementation

In this section we provide a discussion of necessary considerations for practical implementation of the result in Theorem 1. Observe that the conditions in 2.33 and 2.34 are functions of four primary terms, $\hat{V}$, $\tilde{V}_N$, D_k, and $\bar{D}_k$. While $\hat{V}$ is user-selected, the latter three terms come directly from statistical properties of the control performance functions and the channel distribution. They can, however, be indirectly controlled for with some careful implementation techniques.

First, consider that the latter two terms D_k and $\bar{D}_k$ provide a bound on the difference of the neighboring expected loss functions L_k and L_{k+1} and their gradients, respectively. Thus, these terms collectively can be interpreted as a bound on the degree of non-stationarity of the channel distribution $\mathcal{H}$ between successive time iterations, or in other words the rate at which the channel changes over time epochs. In a practical sense, this rate is controllable by determining how much real time makes up a single discrete time epoch. That is, time epochs k and $k+1$ that are closer together in a real time-sense will naturally have a lower bound for D_k, and $\bar{D}_k$, assuming the rate of change of the channel distribution is indeed smooth. In this sense, D_k and $\bar{D}_k$ can be lowered to satisfy the conditions in 2.33 and 2.34 by considering shorter time between discrete epochs. This is to say that, because the channel conditions are not in our control, if necessary we may change the rate at which we apply our algorithm in a real time sense. By using shorter epochs, we collect channel samples and run the proposed Newton step more often to adapt to quickly changing channel conditions.

The second term present in the conditions of Theorem 1—namely $\tilde{V}_N$—represents the statistical accuracy of the non-i.i.d. samples taken from the window of M most recent channel distributions with respect to the current channel distribution. A condition on $\tilde{V}_N$ in

fact then indirectly provides conditions on the sample size n and window size M used to define $\tilde{R}_k$ necessary to learn a $\hat{V}$-optimal solution. We reiterate here that, in the simpler setting of $M = 1$, a well-studied bound on $\tilde{V}_N$ exists of the order $\mathcal{O}(1/\sqrt{n})$. For the case of windowed sampling the bound on $\tilde{V}_N$ can nonetheless still be varied through various choices of window size M and sample draw size n. However, because the exact nature of both $\tilde{V}_N$ and D_k come from statistical properties not known in practice, precise selection of such parameters n and M can be chosen via a standard backtracking procedure.

The details of the backtracking procedure can be seen in Steps 10 and 11 in Algorithm 1. At each epoch k, the parameters n and M are initialized to n_0 and M_0 in Step 4. In the inner loop, in Step 10 these parameters are respectively increased and decreased by factors of Γ and γ after performing the Newton step. In Step 11, the accuracy of the new dual iterate $\boldsymbol{\mu}_{k+1}$ is checked to be within the intended accuracy $\hat{V}$. Note that, while the sub-optimality cannot be checked directly without knowledge of $\tilde{R}^*_{k+1}$, it can be checked indirectly by checking the norm of the gradient $\|\nabla \tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1})\| < (\sqrt{2\alpha})\hat{V}$ from the strong convexity property in (2.25). If the condition in Step 11 is satisfied, the parameters n and M require no further modification. Otherwise, they are further modified until μ_{k+1} is within the target accuracy which in turn may imply that the conditions in 2.33 and 2.34 are satisfied. Note that the backtracking rates γ, Γ are standard parameters used in the definition of a backtracking algorithm and effectively tradeoff the speed of the backtracking search vs. its thoroughness or accuracy. Generally speaking, values closer to 1 will result in a slower, more careful backtracking search while values of γ and Γ that are, respectively, smaller and larger will result in a faster, more aggressive search. Tuning of these parameters should thus reflect the desired tradeoff. With this practical considerations in mind, we proceed by simulating a wireless control learning problem using the proposed use of Newton's method on the ERM relaxation.

2.7 Simulation Results

We simulate the performance of our second order learning method on a simple WCP. Consider the 1-dimensional switched dynamical system in 1 governed by the transition constants A_o and A_c for $m = 4$ systems/states. The control performance for the ith agent $J^i(y^i)$ measures the mean square error performance and is now given by the expression in (2.9). The open and closed loop control gains for each agents are chosen between $[1.1, 1.5]$ and $[0, 0.8]$, respectively. The probability of successful transmission for agent i is modeled as a negative exponential function of both the power and channel state, $q(h^i, p^i(h^i) := 1 - e^{-h^i p^i(h^i)}$, while channel states at epoch k are drawn from an exponential distribution with mean u_k. The channel varies over time by the mean u_k changing for different times. We draw $n = 200$ samples and store a window of the previous $M = 5$ distributions for a total of $N = 1000$

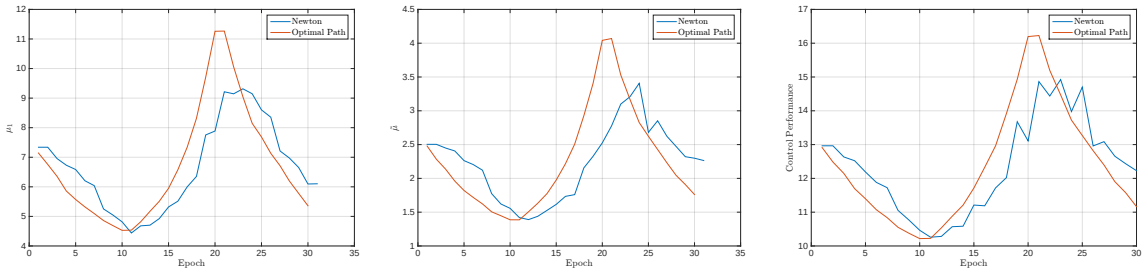


Figure 2.3: Convergence paths of optimal values vs. values generated by the Newton learning method for time-varying $\mathcal{H}_k$ for dual variables (left) μ^1, (center) $\tilde{\mu}$, and (right) control performance $\sum J^i(y^i)$. Newton's method is able to find an approximately optimal value for the dual variables and respective control performance at each iteration.

samples at each epoch. As we assume the that channel statistics vary only vary across time *epochs*, but stay constant within a single epoch, we may consider it reasonable to collect 200 channel samples within an epoch.

To demonstrate the ability of Newton's method to instantaneously learn an approximately optimal power allocation as the channel distribution varies over time, we perform Algorithm 1 over the ERM problem in (2.15) with the defined control performance $J(\cdot)$, transmission probabilities $q(\cdot)$ and channel distributions $\mathcal{H}_k$. In Figure 2.3 we show the path of Newton's method at each time k for the dual variables μ_k^1, $\tilde{\mu}_k$, and the control performance $\sum_{i=1}^m J^i(y_k^i)$. The red line of each figure plots the optimal values for the current distribution parameter u_k as it changes with k. These values are obtained by solving the optimization problem at each epoch offline a priori. The blue line, alternatively, plots the values generated by Newton's method for each epoch k in an online manner. The channel evolves at each iteration by a fixed rate $u_{k+1} = u_k \pm r$ for some rate r. Observe that within some small error Newton's method is indeed able to quickly and approximately find each new solution as the channel varies over time.

To compare the effect of selecting different choices of accuracy $\hat{V}$ numerically, we present in Figure 2.4 the simulation performance of two representative cases with respective accuracies of $\hat{V} = 0.01$ (left) and $\hat{V} = 0.03$ (right). In the top figures, we show the suboptimality relative to the optimal control performance and show on the bottom figures the resulting constraint violation (where a positive value reflect violation) over a set of time epochs where the channel varies. Here, we see an interesting case that highlights the need of proper selection or estimation of $\hat{V}$. Although the left hand figures strive for a better accuracy, the performance is better on the right hand figures. This is due to the fact that single iterations of Newton's method cannot reach accuracies of 0.01, resulting in a more suboptimal trajectory of resource allocation policies. On the other hand, the more moderate goal of 0.03 allows for the learning method to reach intended accurate goals with each step of Newton's method as the channel varies.

Figure 2.4: Comparison of suboptimality (top) and constraint violation (bottom) for the case of $\hat{V} = 0.01$ (left) and $\hat{V} = 0.03$ (right). Although the right-hand figures strive for less accuracy, they perform better because Newton's method can adapt to the intended accuracy more easily with single iterations.

Using the dual parameters found by Newton's method, we simulate the resulting dynamical system. The dual parameters are used to determine the power allocation policy, which is used to determine transmission probabilities given current channel conditions. In Figure 2.5 we show the resulting state evolution of x_t^i for each of the 4 state variables. The blue curve shows the process using the opportunistic transmission policy from Newton's method, while the red curve shows the process when the loop is always closed, i.e. no packet drops. Here, we observe that while there are some instances when the state variable grows large when the system is in open loop, overall the system remains stable over time.

2.8 Conclusion

In this chapter we considered the wireless control system over a non-stationary wireless channel. The problem of maximizing a control utility subject to resource constraints can be formulated as a stochastic optimization problem in the dual domain. Because the wireless channel is random and time-varying, channel samples must be taken, resulting in a relaxed empirical risk minimization (ERM) problem. Standard ERM techniques do not suffice in the wireless setting because the channel is constantly changing. We propose the use of Newton's method, whose local quadratic convergence property allows us to continuously learn and adapt our optimal power allocation policies to changes in the channel distribution. We derive specific conditions on achieving instantaneous convergence to an approximate solution and characterize the suboptimality and stability in the wireless control problem (WCP). We additionally provide numerical simulations that demonstrate the use of Newton's method to

Figure 2.5: Dynamic evolution of each of the 4 state variables over the time-varying channel. The blue curve shows the opportunistic power allocation policy found with Newton's method while the red curve shows the evolution assuming the loop can always be closed.

learn and track optimal power allocations over a time varying channel. While this chapter considers only resource allocation on contention-free links, consider the scheduling problem on a shared channel with non-stationary distributions remains an area of future work.

2.9 Appendix: Proof of Lemma 1

We start with the definition of the Newton decrement at time $k+1$. We can add and subtract $\nabla \tilde{R}_k(\boldsymbol{\mu}_k)$ and upper bound using the triangle inequality as

$$\begin{aligned}\lambda_{k+1}(\boldsymbol{\mu}_k) &= \|\mathbf{H}_{k+1}^{-1/2}\nabla \tilde{R}_{k+1}(\boldsymbol{\mu})\| = \|\nabla \tilde{R}_{k+1}(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}} \\ &\leq \|\nabla \tilde{R}_k(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}} + \|\nabla \tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \nabla \tilde{R}_k(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}}. \end{aligned} \tag{2.44}$$

First, we will upper bound the second term in (2.44). By adding and subtracting the expected losses $\nabla L_k(\boldsymbol{\mu}_k)$ and $\nabla L_{k+1}(\boldsymbol{\mu}_k)$ and using the triangle inequality to obtain

$$\begin{aligned}\|\nabla \tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \nabla \tilde{R}_k(\boldsymbol{\mu}_k)\| &\leq \|\nabla \hat{L}_{k+1}(\boldsymbol{\mu}_k) - \nabla L_{k+1}(\boldsymbol{\mu}_k)\| \\ &+ \|\nabla L_k(\boldsymbol{\mu}_k) - \nabla \hat{L}_k(\boldsymbol{\mu}_k)\| + \|\nabla L_{k+1}(\boldsymbol{\mu}_k) - \nabla L_k(\boldsymbol{\mu}_k)\|.\end{aligned}$$

The first two terms in the above sum are bounded by $\tilde{V}_N^{1/2}$ per (2.26), while the third term is the difference of two consecutive loss functions and is therefore bounded by $\bar{D}_k$ from (2.28). The norm weight $\mathbf{H}_{k+1}^{-1}$ additionally provides a bound of $\alpha\hat{V}$ as the strong convexity constant of $\tilde{R}_{k+1}$ providing an upper bound on the norm of Hessian inverse as in (2.25).

Combining these, we obtain

$$\|\nabla \tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \nabla \tilde{R}_k(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}} \leq \frac{2\tilde{V}_N^{1/2} + \bar{D}_k}{(\alpha \hat{V})^{1/2}}. \tag{2.45}$$

We now can bound the first term in (2.44) using the Lipschitz continuity of the gradient $\Delta + c\hat{V}$, i.e.

$$\|\nabla \tilde{R}_k(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}} \leq \left(\frac{2(\Delta + c\hat{V})\|\boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}}_k^*\|}{\alpha \hat{V}} \right)^{1/2} \tag{2.46}$$

Recall that $\boldsymbol{\mu}_k$ is given to be a $\hat{V}$-accurate minimizer of $\tilde{R}_k$. The difference $\|\boldsymbol{\mu}_k - \tilde{\boldsymbol{\mu}}_k^*\|$ can subsequently be bounded with $\hat{V}$, resulting in the final bound for the first term

$$\|\nabla \tilde{R}_k(\boldsymbol{\mu}_k)\|_{\mathbf{H}_{k+1}^{-1}} \leq \left(\frac{2(\Delta + c\hat{V})\hat{V}}{\alpha \hat{V}} \right)^{1/2} \tag{2.47}$$

To be in the quadratic convergence region, i.e. $\lambda_{k+1}(\boldsymbol{\mu}_k) < 1/4$, follows by summing (2.45) and 2.47 as in 2.30.

2.10 Appendix: Proof of Lemma 2

To prove this result, we start by expanding the term $\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_{k+1}^*$. By adding and subtracting $\tilde{R}_k(\boldsymbol{\mu}_k)$, $\tilde{R}_k^*$, and $\tilde{R}_k(\boldsymbol{\mu}_{k+1}^*)$, we obtain

$$\begin{aligned} \tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_{k+1}^* = \tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k) \\ + \tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \\ + \tilde{R}_k^* - \tilde{R}_k(\boldsymbol{\mu}_{k+1}^*) \\ + \tilde{R}_k(\boldsymbol{\mu}_{k+1}^*) - \tilde{R}_{k+1}^*. \end{aligned} \tag{2.48}$$

We now individually bound each of the four differences in 2.48. Firstly, the difference $\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k)$ becomes

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k) = \hat{L}_{k+1}(\boldsymbol{\mu}_k) - \hat{L}_k(\boldsymbol{\mu}_k), \tag{2.49}$$

Using the same reasoning as in (2.45) with the functional statistical accuracy bound in place of the bound for gradients in (2.26) and using (2.27) in place of (2.28), we obtain the equivalent bound

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k) \leq 2\tilde{V}_N + D_k. \tag{2.50}$$

For the second term in 2.48, we again use the fact that $\boldsymbol{\mu}_k$ as an $\hat{V}$-optimal solution for the sub-optimality $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^*$ to bound with the statistical accuracy as

$$\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}_k^* \leq \hat{V}. \tag{2.51}$$

We proceed with bounding the third term in 2.48. Based on the definition of $\boldsymbol{\mu}_k^*$ as the optimal solution of the loss $\tilde{R}_k$, the the difference $\tilde{R}_k^* - \tilde{R}_k(\boldsymbol{\mu}_{k+1}^*)$ is always negative, i.e.,

$$\tilde{R}_k^* - \tilde{R}_k(\boldsymbol{\mu}_{k+1}^*) \leq 0. \tag{2.52}$$

For the fourth term in 2.48, we use the triangle inequality to bound the difference $\tilde{R}_k(\boldsymbol{\mu}_{k+1}^*) - \tilde{R}_{k+1}^*$ in 2.48 as

$$\begin{aligned} \tilde{R}_k(\boldsymbol{\mu}_{k+1}^*) - \tilde{R}_{k+1}^* &= \hat{L}_k(\boldsymbol{\mu}_{k+1}^*) - \hat{L}_{k+1}(\boldsymbol{\mu}_{k+1}^*) \\ &\leq 2\tilde{V}_N + D_k. \end{aligned} \tag{2.53}$$

Observe that 2.53 uses the same reasoning as 2.50. Replacing the differences in 2.48 by the upper bounds in 2.50-2.53,

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}_{k+1}^* \leq 4\tilde{V}_N + \hat{V} + 2D_k \quad \text{w.h.p.} \tag{2.54}$$

2.11 Appendix: Proof of Lemma 3

The proof for this result follows from [85, Proposition 4], which we repeat here for completeness. We proceed by bounding the difference $\tilde{R}_{k+1}(\boldsymbol{\mu}) - \tilde{R}_{k+1}^*$ in terms of the Newton decrement parameter $\lambda_{k+1}(\boldsymbol{\mu})$. We first use the result in [89, Theorem 4.1.11], showing that

$$\begin{aligned} \lambda_{k+1}(\boldsymbol{\mu}) - \ln\left(1 + \lambda_{k+1}(\boldsymbol{\mu})\right) &\leq \tilde{R}_{k+1}(\boldsymbol{\mu}) - \tilde{R}_{k+1}^* \\ &\leq -\lambda_{k+1}(\boldsymbol{\mu}) - \ln\left(1 - \lambda_{k+1}(\boldsymbol{\mu})\right). \end{aligned} \tag{2.55}$$

We can use the Taylor's expansion of $\ln(1+a)$ for $a = \lambda_{k+1}(\boldsymbol{\mu})$ to show that $\lambda_{k+1}(\boldsymbol{\mu}) - \ln\left(1 + \lambda_{k+1}(\boldsymbol{\mu})\right)$ is bounded below by $(1/2)\lambda_{k+1}(\boldsymbol{\mu})^2 - (1/3)\lambda_{k+1}(\boldsymbol{\mu})^3$ for $0 < \lambda_{k+1}(\boldsymbol{\mu}) < 1/4$. Likewise, we have that $(1/6)\lambda_{k+1}(\boldsymbol{\mu})^2 \leq (1/2)\lambda_{k+1}(\boldsymbol{\mu})^2 - (1/3)\lambda_{k+1}(\boldsymbol{\mu})^3$ and subsequently $\lambda_{k+1}(\boldsymbol{\mu}) - \ln\left(1 + \lambda_{k+1}(\boldsymbol{\mu})\right)$ is bounded below by $(1/6)\lambda^2$. We again use Taylor's expansion of $\ln(1-a)$ for $a = \lambda_{k+1}(\boldsymbol{\mu})$ to show that $-\lambda_{k+1}(\boldsymbol{\mu}) - \ln\left(1 - \lambda_{k+1}(\boldsymbol{\mu})\right)$ is bounded above by $\lambda_{k+1}(\boldsymbol{\mu})^2$ for $\lambda_{k+1}(\boldsymbol{\mu}) < 1/4$; see e.g., [14, Ch. 9]. Considering these bounds and the inequalities in 2.55 we obtain that

$$\frac{1}{6}\lambda_{k+1}(\boldsymbol{\mu})^2 \leq \tilde{R}_{k+1}(\boldsymbol{\mu}) - \tilde{R}_{k+1}^* \leq \lambda_{k+1}(\boldsymbol{\mu})^2. \tag{2.56}$$

Because we assume that $\lambda_{k+1}(\boldsymbol{\mu}_k) \leq 1/4$, the quadratic convergence rate of Newton's method for self-concordant functions [14] implies that the Newton decrement has a quadratic convergence and we can write

$$\lambda_{k+1}(\boldsymbol{\mu}_{k+1}) \leq 2\lambda_{k+1}(\boldsymbol{\mu}_k)^2. \tag{2.57}$$

We combine the results in 2.56 and 2.57 to show that the optimality error $\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1}$ has an upper bound which is proportional to $(\tilde{R}_{k+1}(\mu_k) - \tilde{R}^*_{k+1})^2$. In particular, we can write $\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1} \leq \lambda_{k+1}(\boldsymbol{\mu}_{k+1})^2$ based on the second inequality in 2.56. This observation in conjunction with the result in 2.57 implies that

$$\tilde{R}_{k+1}(\boldsymbol{\mu}_{k+1}) - \tilde{R}^*_{k+1} \leq 4\lambda_{k+1}(\boldsymbol{\mu}_k)^4. \tag{2.58}$$

The first inequality in 2.56 implies that $\lambda_{k+1}(\boldsymbol{\mu}_k)^4 \leq 36(\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}^*_{k+1})^2$. Thus, we can substitute $\lambda_{k+1}(\boldsymbol{\mu}_k)^4$ in 2.58 by $36(\tilde{R}_{k+1}(\boldsymbol{\mu}_k) - \tilde{R}^*_{k+1})^2$ to obtain the result in 2.32.

Appendix: Proof of Proposition 2

We begin by bounding the gradient of the expected dual loss $L(\boldsymbol{\mu}_k)$ at the kth dual iterate $\boldsymbol{\mu}_k$ by using Lipschitz continuity, i.e.

$$\|\nabla L_k(\boldsymbol{\mu}_k)\|^2 \leq 2\Delta(L_k(\boldsymbol{\mu}_k) - L^*_k). \tag{2.59}$$

We expand the sub-optimality $L(\boldsymbol{\mu}_k) - L^*$ by adding and subtracting terms as follows

$$\begin{aligned}\frac{1}{2\Delta}\|\nabla L_k(\boldsymbol{\mu}_k)\|^2 \leq L_k(\boldsymbol{\mu}_k) - \tilde{L}_k(\boldsymbol{\mu}_k) + \tilde{L}_k(\boldsymbol{\mu}_k) \\ - \tilde{R}_k(\boldsymbol{\mu}_k) + \tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}^*_k + \tilde{R}^*_k - L^*_k,\end{aligned} \tag{2.60}$$

where we recall the notation $\tilde{R}^*_k := \tilde{R}_k(\tilde{\boldsymbol{\mu}}^*_k)$. We now proceed by bounding each successive pair of terms in (2.60). The first difference $L_k(\boldsymbol{\mu}_k) - \tilde{L}_k(\boldsymbol{\mu}_k)$ comes from the sampling and is thus bounded by the statistical accuracy $\tilde{V}_N$. The second difference $\tilde{L}_k(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k)$ can be bounded by the regularizers as

$$\tilde{L}_k(\boldsymbol{\mu}_k) - \tilde{R}_k(\boldsymbol{\mu}_k) \leq \beta\hat{V}\mathbf{1}^T \log_\epsilon(\boldsymbol{\mu}_k) - \frac{\alpha\hat{V}}{2}\|\boldsymbol{\mu}_k\|^2. \tag{2.61}$$

The second term on the right hand side of (2.61) is negative and can be ignored. Because the dual variable $\|\boldsymbol{\mu}_k\|$ was upper bounded in Corollary 1, we can place a finite bound on $\mathbf{1}^T \log_\epsilon(\boldsymbol{\mu}_k) \leq \kappa$ and then bound the term $\beta\hat{V}\mathbf{1}^T \log_\epsilon(\boldsymbol{\mu}_k) \leq \beta\hat{V}\kappa$. The third difference $\tilde{R}_k(\boldsymbol{\mu}_k) - \tilde{R}^*_k$ is bounded by the suboptimality $\hat{V}$ from the main result in 2.35 and the fourth

difference $\tilde{R}_k^* - L_k^*$ can be bounded by $\rho\hat{V}$ from (2.23). We can therefore bound the gradient of the dual loss as

$$\|\nabla L_k(\boldsymbol{\mu}_k)\|^2 \leq 2\Delta(\tilde{V}_N + C\hat{V}), \tag{2.62}$$

where $C := 1 + \rho + \beta m \log \kappa$. From here, consider that the norm of the dual gradient $\|\nabla L_k(\boldsymbol{\mu}_k)\|^2$ is the sum of squares of each constraint violation in (WCP_k), i.e.,

$$\begin{aligned}&\left(\sum_{i=1}^{m} \mathbb{E}_{h_k^i}(p^i(h)) - p_{\max}\right)^2 + \sum_{i=1}^{m}\left(y^i - \mathbb{E}_{h_k^i}\left\{q(h, p^i(h))\right\}\right)^2 \\ &\quad \leq 2\Delta(\tilde{V}_N + C\hat{V}).\end{aligned} \tag{2.63}$$

The results in (2.36) and (2.37) then follow from here.

Appendix: Proof of Theorem 2

Consider that, using the definitions of the primal maximizers $\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)$ and $\mathbf{y}(\boldsymbol{\mu}_k)$ at a dual point $\boldsymbol{\mu}_k$, we can write the dual function as

$$L(\boldsymbol{\mu}_k) = J(\mathbf{y}(\boldsymbol{\mu}_k)) + \boldsymbol{\mu}_k^T \left(\mathbb{E}_{\mathbf{h}}\check{\mathbf{q}}(\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)) - \check{\mathbf{y}}(\boldsymbol{\mu}_k)\right). \tag{2.64}$$

Likewise, we know from strong duality that the optimal dual values L_k^* is equivalent to the optimal primal objective value $J(\mathbf{y}_k^*)$. Therefore, we can write the suboptimality of dual functions as

$$\begin{aligned}L(\boldsymbol{\mu}_k) - L_k^* = J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*) \\ + \boldsymbol{\mu}_k^T \left(\mathbb{E}_{\mathbf{h}}\check{\mathbf{q}}(\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)) - \check{\mathbf{y}}(\boldsymbol{\mu}_k)\right).\end{aligned} \tag{2.65}$$

Using the bound on dual suboptimality that comes from combining strong convexity and the gradient bound in (2.62), we can upper bound (2.65) as

$$\begin{aligned}(1 + C)\Delta/\alpha \geq J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*) \\ + \boldsymbol{\mu}_k^T \left(\mathbb{E}_{\mathbf{h}}\check{\mathbf{q}}(\mathbf{p}(\mathbf{h}, \boldsymbol{\mu}_k)) - \check{\mathbf{y}}(\boldsymbol{\mu}_k)\right).\end{aligned} \tag{2.66}$$

We can lower bound the right hand side of (2.66) by taking the negative of the absolute value of the final term. Rearranging terms we obtain

$$\begin{aligned}(1+C)\Delta/\alpha + |\boldsymbol{\mu}_k^T \left(\mathbb{E}_{\mathbf{h}}\check{\mathbf{q}}(\mathbf{p}(\mathbf{h},\boldsymbol{\mu}_k)) - \check{\mathbf{y}}(\boldsymbol{\mu}_k)\right)| \\ \geq J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*).\end{aligned} \tag{2.67}$$

From here, we can upper bound the second term on the left hand side using the Cauchy-Schwartz inequality. The norm $\|\boldsymbol{\mu}_k\|$ is bounded by $\sqrt{2/\alpha} + \hat{\mathcal{K}}$ from Corollary 1 while the norm $\|\mathbb{E}_{\mathbf{h}}\check{\mathbf{q}}(\mathbf{p}(\mathbf{h},\boldsymbol{\mu}_k)) - \check{\mathbf{y}}(\boldsymbol{\mu}_k)\|$ is bounded by $2\Delta(1+C)\hat{V}$ from (2.37). This provides us the final result as

$$(1+C)\Delta\left(\frac{1}{\alpha} + 2\hat{V}(\sqrt{2/\alpha} + \hat{\mathcal{K}})\right) \geq J(\mathbf{y}(\boldsymbol{\mu}_k)) - J(\mathbf{y}_k^*). \tag{2.68}$$

Chapter 3

Control-Aware Scheduling for Low-Latency Wireless Systems

3.1 Introduction

The Internet-of-Things (IoT) promises enhanced modes of interaction with the physical world through the deployment of large numbers of sensing and control devices. Relative to conventional communication systems, the deployment of a control system over a communication network increases the sensitivity to packet loss and latency. This is not a major consideration if we rely on wired networks that can simultaneously achieve very low latency and ultra high reliability. However, the cost of installing and maintaining a wired network poses significant challenges [128, 137] that motivate the use of wireless communications. Consequently, there has been great effort in the design of wireless control systems that can achieve high performance in terms of reliability and latency [72, 123]. While this effort is widespread in its range of applications it is of note that wireless control systems hold promise in streamlining industrial control [10, 97, 123, 127, 137]—e.g. factory automation [15, 135].

The primary challenge in designing this ultra reliable low latency communications (URLLC) systems is the tradeoff between reliability and latency. To achieve ultra-high reliability we need significant protection against packet losses. This can be achieved by increasing packet length, thereby increasing latency, or by increasing bandwidth consumption. Such a tradeoff between latency and reliability is present in any communication medium but it is exacerbated in wireless communications because of fading and shadowing effects. The physical properties of a wireless channel impose inherent limitations in achieving both ultra high reliability and low latency. Such a mismatch has lead to the proposal of several radio resource allocation schemes have been proposed to improve the management of the latency reliability tradeoff in wireless systems [133]. This include seminal works on the design of delay-aware schedulers for communication systems in general [4, 78] and wireless control

systems in particular [129]. The exploitation of spatial and frequency diversity deserves particular mention as a technique that is increasingly recognized as a necessary component of URLLC [91, 97, 119].

In this chapter we propose an alternative approach in which we adapt the communication reliability to the dynamics and state of the plant. We expect this to provide significant advantages because high communication reliability is not a strict requirement of control reliability. In fact, control loops can, in general, drop packets with small impact if the plant is not close to unsafe states. The successful transmission of a packet becomes crucial only when in these unsafe regions. In adapting reliability to the state of the plant we expect that the resources that are saved with plants in safe states are available to achieve high reliability with plants close to unsafe states. Our specific contribution is to develop a control-aware scheduling protocol designed to enable larger scale low latency systems. This is done through the mathematical formulation of the control system design goal in the form of a Lyapunov function that ensures stability of the control system. This formulation naturally induces a bound on the packet delivery rate each control system needs to achieve to meet the control-based goal. Such packet delivery rates depend upon current control and channel states and thus dynamically change over the course of the system life time. In particular, we use IEEE 802.11ax WiFi [9] to allocate bandwidth and data rates to reduce total transmission times. As these control-based reliability targets may be significantly lower than traditional, high reliability communication demands, the proposed method is better suited to find scheduling configurations that can meet strict latency requirements imposed by the physical system.

We remark that in the context of wireless control systems, there have been a range of works that incorporates control system information in the networking and communication policies. For example, control system stability under fixed periodic protocols, e.g. round-robin, can be analyzed—see, e.g., [28,57,106,140]. Periodic sequences leading to stability [60], controllability and observability [139], or optimizing control objectives [68, 82, 102] have been proposed. More sophisticated schedulers do not rely on a predefined sequence but try to dynamically access the communication medium at each step. Initial approaches abstract control performance requirements in the time/frequency domain, e.g., how often a task needs resource access, employing algorithms from real-time scheduling theory [16, 74]. More recent scheduling approaches often depend on the current control system states, i.e., informally the subsystem with the largest state discrepancy is scheduled to communicate—see, e.g., [17,28,53,80,110]. Alternatively scheduling can take into account current wireless channel conditions opportunistically to meet target control system reliability requirements [47]. None of these approaches, however, are explicitly designed for low-latency communication systems, which is the subject of our work and a key contribution with respect to previous approaches.

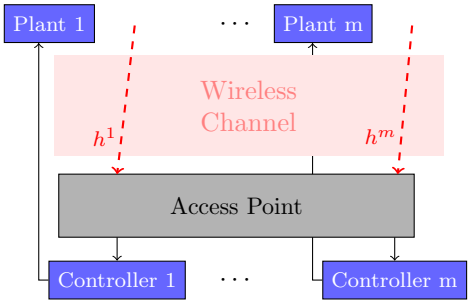


Figure 3.1: Wireless control system with m independent systems. Each system contains a sensor that measure state information, which is transmitted to the controller over a wireless channel. The state information is used by the controller to determine control policies for each of the systems.The communication is assumed to be wireless in the uplink and ideal in the downlink.

The chapter is organized as follows. We formulate the wireless control system in which state information is communicated to the control over a wireless channel. Due to the potential for random packet drops, this is modeled as a switched dynamical system (Section 3.2). A Lyapunov function is used to evaluate the stability of the control state, and the uncertainty in this measurement grows the more consecutive packets are lost for a particular system. We then discuss the scheduling parameters of the IEEE 802.11ax communication model (Section 3.2.1). From there, we derive a mathematical formulation of the optimal scheduling problem (Section 3.3). This can be formulation by minimizing a control cost with an explicitly latency constraint (Section 3.3.1) or minimizing transmission time with an explicit control performance constraint (Section 3.3.2).

Using this formulation, we develop the control-aware low latency scheduling (CALLS) method (Section 3.4). The CALLS method uses current control states and channel conditions to derive dynamic packet success rates for each user (Section 3.4.1). In this way, control systems that are closest to instability will be given priority in the scheduling so that they may close their control loops. The scheduling procedure consists of a random user selection procedure to reduce the number of required PPDUs that incur significant overhead (Section 3.4.2), followed by an assignment-method based scheduling of selected users to minimize total transmission time (Section 3.4.3). The performance of the CALLS method is analyzed in a series of simulation experiments in which its performance is compared against a control-agnostic procedure (Section 3.5). We demonstrate in numerous control systems that the control-aware, adaptive reliability approach may support more users than the alternative and achieve more robust overall performance.

3.2 Wireless Control Sysyem

Consider a system of m independent linear control systems, or devices, where each system $i = 1, \ldots, m$ maintains a state variable $\mathbf{x}_i \in \mathbb{R}^p$. The dynamics are discretized so that the state evolves over time index k. Applying an input $\mathbf{u}_{i,k} \in \mathbb{R}^q$ causes the state and output to evolve based on the discrete-time state space equations,

$$\mathbf{x}_{i,k+1} = \mathbf{A}_i \mathbf{x}_{i,k} + \mathbf{B}_i \mathbf{u}_{i,k} + \mathbf{w}_k \tag{3.1}$$

where $\mathbf{A}_i \in \mathbb{R}^{p \times p}$ and $\mathbf{B}_i \in \mathbb{R}^{p \times q}$ are matrices that define the system dynamics, and $\mathbf{w}_k \in \mathbb{R}^p$ is Gaussian noise with co-variance $\mathbf{W}_i$ that captures the errors in the linear model (due to, e.g., unknown dynamics or from linearizion of non-linear dynamics). We further assume the state transition matrix $\mathbf{A}_i$ is on its own unstable, i.e. has at least one eigenvalue greater than 1. This is to say that, without an input, the dynamics will drive the state $\mathbf{x}_{i,k} \to \infty$ as $k \to \infty$.

In the wireless control system model presented in Figure 3.1. Each system is closed over a wireless medium, over which the sensor located at the control system sends state information to the controller located at a wireless access point (AP) shared among all systems. Using the state information $\mathbf{x}_{i,k}$ received from device i at time k, the controller determines the input $\mathbf{u}_{i,k}$ to be applied. We stress in Figure 3.1 we restrict our attention to the wireless communications at the sensing, or "uplink", while the control actuation, or "downlink, is assumed to occur over an ideal channel. We point out that while a more complete model may include packet drops in the downlink, in practice the more significant latency overhead occurs in the uplink. We therefore keep this simpler model for mathematical coherence. In low-latency applications, a high state sampling rate is required be able to adapt to the fast-moving dynamics This subsequently places a tight restriction on the latency in the wireless transmission, so as to avoid losing sampled state information. This specific latency requirement between the sensor and AP we denote by $\tau_{\max}$, and is often considered to be in the order of milliseconds.

Because the control loop in Figure 3.1 is closed over a wireless channel, there exists a possibility at each cycle k that the transmission fails and state information is not received by the controller. We refer to this as the "open-loop' configuration; when state information is received, the system operates in "closed-loop." As such, it is necessary to define the system dynamics in both configurations. Consider a generic linear control, in which the input being determined as $\mathbf{u}_{i,k} = \mathbf{K}_i \mathbf{x}_{i,k}$ for some matrix $\mathbf{K}_i \in \mathbb{R}^{q \times p}$. Many common control policies indeed can be formulated in such a manner, such as LQR control. In general, this matrix $\mathbf{K}$ is chosen such as that the closed loop dynamic matrix $\mathbf{A} + \mathbf{BK}$ is stable, i.e. has all eigenvalues less that 1. Thus, application of this control over time will drive the state

$\mathbf{x}_{i,k} \to 0$ as $k \to \infty$. We assume that this choice of $\mathbf{K}$ is given—in other words, the controller is pre-designed with respect to ideal closed loop behavior. As the controller does not always have access to state information, we alternatively consider the estimate of state information of device i known to the controller at time k as

$$\hat{\mathbf{x}}_{i,k}^{(l_i)} := (\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i)^{l_i}\mathbf{x}_{i,k-l_i}, \tag{3.2}$$

where $k - l_i \geq k - 1$ is the last time instance in which control system i was closed. There are two important things to note in (3.2). First, this is the estimated state *before* a transmission has been attempted at time k; hence, $l_i = 1$ when state information was received at the previous time. Second, observe that in (3.2) we assume that the AP/controller has knowledge of the dynamics $\mathbf{A}_i$ and $\mathbf{B}_i$, as well as the linear control matrix $\mathbf{K}_i$. Any gap in this knowledge of dynamics is captured in the noise $\mathbf{w}_k$ in the actual dynamics in (3.35). Note that the estimated state (3.2) is used in place of the true state in both the determination of the control *and* the radio resource allocation decisions as discussed later in this chapter.

At time k, if the state information is received, the controller can apply the input $\mathbf{u}_{i,k} = \mathbf{K}_i\mathbf{x}_{i,k}$ exactly, otherwise it applies an input using the estimated state, i.e. $\mathbf{u}_{i,k} = \mathbf{K}_i\hat{\mathbf{x}}_{i,k}$. Thus, in place of (3.35), we obtain the following switched system dynamics for $\mathbf{x}_{i,k}$ as

$$\mathbf{x}_{i,k+1} = \begin{cases} (\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i)\mathbf{x}_{i,k} + \mathbf{w}_k, \text{ in closed-loop,} \\ \mathbf{A}_i\mathbf{x}_{i,k} + \mathbf{B}_i\mathbf{K}_i\hat{\mathbf{x}}_{i,k}^{(l_i)} + \mathbf{w}_k, \text{ in open-loop.} \end{cases} \tag{3.3}$$

The transmission counter l_i is updated at time k as

$$l_i \leftarrow \begin{cases} 1, \text{ in closed-loop,} \\ l_1 + 1, \text{ in open-loop.} \end{cases} \tag{3.4}$$

Observe in (3.3) that, when the system operates in open loop, the control is not applied relative to the current state $\mathbf{x}_{i,t}$ but on the estimated state $\hat{\mathbf{x}}_{i,k}^{(l_i)}$, which indeed may not be close to the true state. In this case, the state may not be driven to zero as in the closed-loop configuration. To see the effect of operating in open loop for many successive iterations, we can write the error between the true and estimated state as

$$\mathbf{e}_{i,k} := \mathbf{x}_{i,k} - \hat{\mathbf{x}}_{i,k}^{(l_i)} = \sum_{j=0}^{l_i-1} \mathbf{A}_i^j \mathbf{w}_{i,k-j-1}. \tag{3.5}$$

In (3.5), it can be seen that as l_i grows, the error $\mathbf{e}_{i,k}$ grows with the accumulation of the noise present in the actual state but not considered in the estimated state. Thus, if l_i is large and $\mathbf{w}_{i,k}$ is large (i.e., high variance), this error will become large as well.

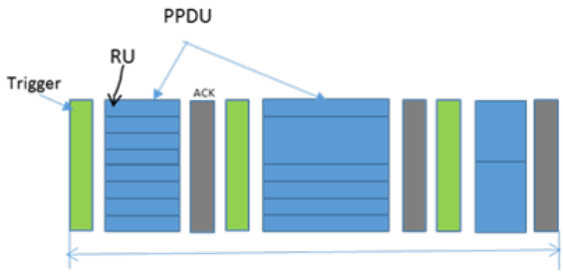


Figure 3.2: Multiplexing of frequencies (RU) and time (PPDU) in IEEE 802.11ax transmission window (formally referred as Transmission Opportunity or TXOP in the standard. The total transmission time is the time of all PPDUs, including the overhead of trigger frames (TF) and acknowledgments.

To conclude the development of the wireless control formulation, we define a quadratic Lyapunov function $L(\mathbf{x}) := \mathbf{x}^T\mathbf{P}\mathbf{x}$ for some positive definite $\mathbf{P} \in \mathbb{R}^{p\times p}$ that measures the performance of the system as a function of the state. Because the scheduler only has access to estimated state info, we consider the expected value of $L(\mathbf{x})$ given the state estimate, which can be found via (3.5) as

$$\begin{aligned}\mathbb{E}[L(\mathbf{x}_{i,k}) \,|\hat{\mathbf{x}}_{i,k}^{(l_i)}] \\ = (\hat{\mathbf{x}}_{i,k}^{(l_i)})^T\mathbf{P}(\hat{\mathbf{x}}_{i,k}^{(l_i)}) + \sum_{j=0}^{l_i-1} \mathrm{Tr}[(\mathbf{A}_i^T\mathbf{P}^{\frac{1}{j}}\mathbf{A}_i)^j\mathbf{W}_i].\end{aligned} \tag{3.6}$$

Thus, the control-specific goal is to keep $\mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}]$ within acceptable bounds for each system i. We now proceed to discuss the wireless communication model that determines the resource allocations necessary to close the loop.

3.2.1 IEEE 802.11ax communication model

We consider the communication model provided in the next-generation Wi-Fi standard IEEE 802.11ax. While 3GPP wireless systems such as LTE [107] or the next generation 5G [3] can also be considered as alternate communication models, most factory floors are already equipped with Wi-Fi connectivity and, moreover, Wi-Fi can operate in the unlicensed band. It is generally considered to be cost-effective to operate and maintain.

Traditional Wi-Fi systems rely only on contention-based channel access and may introduce high or variable latency in congested or dense deployment scenarios even in a fully managed Wi-Fi network, which is typically available in industrial control and automation scenarios. To address the problems with dense deployment, the draft 802.11ax amendment has defined scheduling capability for Wi-Fi access points (APs). Wi-Fi devices can now be scheduled for accessing the channel in addition to the traditional contention-based channel access. Such scheduled access enables more controlled and deterministic behavior in the Wi-Fi networks.

Within each transmission window (formally referred as transmission opportunity or TXOP in the standard), the AP may schedule devices through both frequency and time division multiplexing using the multi-user (MU) OFDMA technique. This to say that devices can be slotted in various frequency bands—formally called resource units (RUs)—and in different timed transmission slots—formally called PPDUs. An example of the multiplexing of devices across time and frequency is demonstrated in Figure 3.2. The AP additionally sends a trigger frame (TF) indicating which devices should transmit data in the current TXOP and the time/frequency resources these triggered devices should use in their transmissions.

To state this model formally, the scheduling parameters assigned by the AP to each device consist of a frequency-slotted RU, time-slotted PPDU, and an associated modulation and coding scheme (MCS) to determine the transmission format. The transmission power is assumed to be fixed and equally divided amongst all devices. We define the following notations to formulate these parameters. To specify an RU, we first notate by $f^1, f^2, \ldots, f^b$, where n is the number of discrete frequency bands of fixed bandwidth (typically 2 MHz) in which a device can transmit; in a 20MHz channel, for example, there are $n = 10$ such bands. For each device, we then define a set of binary variables $\varsigma_i^j \in \{0, 1\}$ if device i transmits in band f^j and collect all such variables for device i in $\boldsymbol{\varsigma}_i = [\varsigma_i^1; \ldots; \varsigma_i^b] \in \{0, 1\}^b$ and for all devices in $\boldsymbol{\Sigma} := [\boldsymbol{\varsigma}_i, \ldots, \boldsymbol{\varsigma}_m] \in \{0, 1\}^{b \times m}$. A device may transmit in bands in certain multiples of 2MHz as well, which would be notated as, e.g. $\boldsymbol{\varsigma}_i = [1; 1; 0; \ldots; 0]$ for transmission in an RU of size 4MHz. Note, however, that allowable RU's contain only sizes of certain multiples of 2MHz—namely, 2MHz, 4MHz, 8MHz, and 20MHz in the 802.11ax standard. Furthermore, it is only permissible to transmit in *adjacent* bands, e.g. f^j and f^{j+1}. We therefore define the set $\mathcal{S} \subset \{0, 1\}^b$ as the set of binary vectors that define permissible RUs and consider only $\boldsymbol{\varsigma}_i \in \mathcal{S}$ for all devices i. Finally, note that the RU assignment $\mathbf{0} \in \mathcal{S}$ signifies a device does not transmit in this particular transmission window.

To specify the PPDU of all scheduled devices, we define for device i a positive integer value $\alpha_i \in \mathbb{Z}_{++}$ that denotes the PPDU slot in which it transmits and collect such variables for all devices in $\boldsymbol{\alpha} = [\alpha_1; \ldots; \alpha_m] \in \mathbb{Z}_{++}^m$. Likewise, device i is given an MCS μ_i from the discrete space $\mathcal{M} = \{0, 1, 2, \ldots, 10\}$. The MCS in particular defines a pair of modulation scheme and coding rate that subsequently determine both the data rate and packet error rate of the transmission. The allowable MCS settings provided in 802.11ax are provided in Table 3.1. Finally, we notate by $\mathbf{h}_i := [h_i^1; h_i^2; \ldots; h_i^b] \in \mathbb{R}_+^b$ a set of channel states experienced by device i, where h_i^j is the gain of a wireless fading channel in frequency band f^j. We assume that channel conditions are constant within a single TXOP, i.e. do not vary across PPDUs.

We now proceed to define two functions that describe the wireless communications over the channel. Firstly, we define a function $q : \mathbb{R}_+^b \times \mathcal{M} \times \mathcal{S} \to [0, 1]$ which, given a set of channel conditions $\mathbf{h}$, MCS μ, and RU $\boldsymbol{\varsigma}$, returns the probability of successful transmission,

μ	Modulation type	Coding rate	Data rate (Mb/s)
0	BPSK	1/2	4
1	QPSK	1/2	16
2	QPSK	3/4	24
3	16-QAM	1/2	33
4	16-QAM	3/4	49
5	64-QAM	2/3	65
6	64-QAM	3/4	73
7	64-QAM	5/6	81
8	256-QAM	3/4	98
9	256-QAM	5/6	108
10	1024-QAM	3/4	122

Table 3.1: Data rates for MCS configurations in IEEE 802.11ax for 20MHz channel. The modulation type and coding rate in the first 2 columns together specify a PDR function $q(\boldsymbol{\mu}, \boldsymbol{\varsigma})$ for RU $\boldsymbol{\varsigma}$. The data rate in the third column specifies the associated transmission time $\tau(\mu, \boldsymbol{\varsigma})$.

otherwise called packet delivery rate (PDR). Furthermore, define by $\tau : \mathcal{M} \times \mathcal{S} \to \mathbb{R}_+$ a function that, given an MCS μ and RU $\boldsymbol{\varsigma}$, returns the maximum time taken for a single transmission attempt. Assuming a fixed packet size, such a function can be determined from the data rates associated with each MCS in Table 3.1. Observe that all functions just defined are determined independent of the PPDU slot the transmission takes place in, while transmission time is also independent of the channel state. Because a PPDU cannot finish until all transmissions within the PPDU have been completed, the total transmission time of a single PPDU s is the maximum transmission time taken by all devices within that time slot. We define the transmission time of PPDU slot s as

$$\hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) := \max_{i:\alpha_i = s} \tau(\mu_i, \boldsymbol{\varsigma}_i) + \tau_0(\boldsymbol{\alpha}, s), \tag{3.7}$$

where $\tau_0 : \mathbb{Z}_{++}^m \times \mathbb{Z}_{++} \to \mathbb{R}_+$ is a function that specifies the communication overhead of PPDU s. This overhead may consist of, e.g., the time required to send TFs to scheduled users, as seen in Figure 3.2.

3.3 Optimal Control Aware Scheduling

Using the communication model of 802.11ax just outlined and the control-based Lyapunov metric of (3.6), we can formulate an optimization problem that characterizes the exact optimal scheduling of transmissions with a transmission window to maximize control performance. The optimal scheduling and allocation selects the set of RUs $\boldsymbol{\Sigma}$. MCS $\boldsymbol{\mu}$, and PPDUs $\boldsymbol{\alpha}$ for all devices–which in effect fully determine the schedule—such to minimize a cost subject to scheduling design and feasibility constraints. In particular, we discuss two related,

alternative formulations of the low-latency scheduling problem.

3.3.1 Latency-constrained scheduling

In the latency-constrained formulation, we are interested in minimizing a common control cost subject to strict latency requirements. In particular, in the low-latency setting we set a bound $\tau_{\max}$ on the total transmission time across all PPDUs in a TXOP. This constraint is relevant in design of MAC-layer protocols that set strict limits on transmission times. In addition, the RU and PPDU allocation across devices must be feasible, i.e., two devices cannot be transmitting in the same frequency band in the same PPDU.

Recall the PDR function $q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$ and consider that this can alternatively be interpreted as the probability of closing the control loop under certain channel conditions and scheduling parameters. From there, we can now write the expected Lyapunov value for at time $k+1$ given its current state $\mathbf{x}_{i,k}$, channel state $\mathbf{h}_{i,k}$, MCS μ_i, and RU $\boldsymbol{\varsigma}_i$ using the expected cost in (3.6). By defining $\mathbf{x}^c_{i,k+1}$ and $\mathbf{x}^o_{i,k+1}$ as the closed loop and open loop states, respectively, as determined by the switched system in (3.3), this is written as

$$\begin{aligned} J_i(\hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) &:= \mathbb{E}(L(\mathbf{x}_{i,k+1}) \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \\ &= (1 - q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i))\mathbb{E}L(\mathbf{x}^o_{i,k+1} \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}) \\ &\quad + q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)\mathbb{E}L(\mathbf{x}^c_{i,k+1} \mid \hat{\mathbf{x}}^{(l_i)}_{i,k}). \end{aligned} \tag{3.8}$$

For notational convenience, we collect all current estimated control states at time k as $\hat{\mathbf{X}}_k := [\hat{\mathbf{x}}^{(l_1)}_{1,k}, \ldots, \hat{\mathbf{x}}^{(l_m)}_{m,k}]$ and channel states $\mathbf{H}_k = [\mathbf{h}_{1,k}, \ldots, \mathbf{h}_{m,k}]$. Now, define the total control cost, given the current states and scheduling parameters as some aggregation of the combined expected future Lyapunov costs across all devices, i.e.,

$$\begin{aligned} \tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) := \\ g(J_1(\hat{\mathbf{x}}^{(l_1)}_{1,k}, \mathbf{h}_{1,k}, \mu_1, \boldsymbol{\varsigma}_1), \ldots, J_m(\hat{\mathbf{x}}^{(l_m)}_{m,k}, \mathbf{h}_{m,k}, \mu_m, \boldsymbol{\varsigma}_m)). \end{aligned} \tag{3.9}$$

Natural choices of the aggregation function $g(\cdot)$ are, for example, either the sum or maximum of its arguments.

The optimal scheduling at transmission time k is formulated as the one which minimizes this cost $\tilde{J}$ while satisfying low-latency and feasibility requirements of the schedule, expressed

formally with the following optimization problem.

$$[\boldsymbol{\Sigma}_k^*, \boldsymbol{\mu}_k^*, \boldsymbol{\alpha}_k^*] := \underset{\boldsymbol{\Sigma},\boldsymbol{\mu},\boldsymbol{\alpha},S}{\operatorname{argmin}} \tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.10}$$

$$\text{s. t.} \sum_{i:\alpha_i=s} \varsigma_i^j \leq 1, \quad \forall j, s, \tag{3.11}$$

$$\sum_{s=1}^{S} \hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) \leq \tau_{\max}, \tag{3.12}$$

$$1 \leq \alpha_i \leq S, \quad \forall i, \tag{3.13}$$

$$\boldsymbol{\varsigma}_i \in \mathcal{S}, \ \forall i, \quad \boldsymbol{\mu} \in \mathcal{M}^m, \quad \boldsymbol{\alpha} \in \mathbb{Z}_+^m, \quad S \in \mathbb{Z}_+. \tag{3.14}$$

The optimization problem in (3.10) provides a precise and instantaneous selection of frequency allocations between devices given their current control states $\hat{\mathbf{X}}_k$ and communication states $\mathbf{H}_k$. The constraints in (3.11)-(3.14) encode the following scheduling conditions. The constraint (3.11) ensures that for every PPDU s, there is only one device transmitting on a frequency slot j. In (3.12), we set the low-latency transmission time constraint in terms of the sum of all transmission times for each PPDU s. The constraint in (3.13) bounds each transmission slot by the total number of PPDU's S while (3.14) constrains each variable to its respective feasible set. Note that S is itself treated as an optimization variable in the above problem, so that the number of PPDUs may vary as needed.

Observe in the objective in (3.10) that, by minimizing an aggregate of local control costs, the devices with the highest cost J_i as described by (3.8) will be given the most bandwidth or most favorable frequency bands to increase probability of successful transmission $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$. This in effect increases the chances those devices will close their control loops and be driven towards a more favorable state. Likewise, a device who is experiencing very adverse channel conditions may not be allocated prime transmission slots to reserve such resources who have more favorable channel conditions. In this way, we say this is *control-aware* scheduling, as it considers both the control and channel states of the devices to determine optimal scheduling. However, we stress that the optimization problem described in (3.10)-(3.14) is by no means easy to solve. In fact, the optimization over multiple discrete variables makes this problem combinatorial in nature. In the following section, we discuss a practical reformulation of the problem above and develop heuristic methods to approximate the solutions in realistic low-latency wireless applications.

3.3.2 Control-constrained scheduling

We reformulate the problem in (3.10)-(3.14) to an alternative formulation that more directly informs the control-aware, low-latency scheduling method to be developed. To do so, we

introduce a *control-constrained* formulation, in which the Lyapunov decrease goals are presented as explicit requirement, i.e. constraints in the optimization problem. We are interested, then, in constraint of the form

$$\tilde{J}(\hat{\mathbf{X}}_k, \mathbf{H}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \leq J_{\max}, \tag{3.15}$$

where $J_{\max}$ is a limiting term design to enforce desired system performance. Determining this constant is largely dependent on the particular application of interest, needs of the control systems, and also may be related to the choice aggregation function $g(\cdot)$ in (3.9). For example, $J_{\max}$ may represent a point at which control systems become volatile, unsafe, or unstable.

For the scheduling procedure developed in this chapter, we focus on a particular formulation of the control constraint in (3.15) that constrains the expected future Lyapunov value of each system by a rate decrease of its current value. In particular the following rate-decrease condition for each device i,

$$J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq \rho_i \mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}] + c_i, \tag{3.16}$$

where $\rho_i \in (0, 1]$ is a decrease rate and $c_i \geq 0$ is a constant. Recall the definition of $J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ in (3.8) as the expected Lyapunov value of time $k+1$ given its current estimate and scheduling $\mu_i, \boldsymbol{\varsigma}_i$. The constraint in (3.16) ensures the future Lyapunov cost will exhibit a decrease of at least a rate of ρ_i for device i in expectation. The constant c_i is included to ensure this condition is satisfied by default if the state $\hat{\mathbf{x}}_{i,k}^{(l_i)}$ is already sufficiently small.

We formulate the control-constrained scheduling problem by substituting the latency constraint with the control constraint in (3.16), i.e.,

$$[\boldsymbol{\Sigma}_k^*, \boldsymbol{\mu}_k^*, \boldsymbol{\alpha}_k^*] := \underset{\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, S}{\operatorname{argmin}} \sum_{s=1}^{S} \hat{\tau}(\boldsymbol{\Sigma}, \boldsymbol{\mu}, \boldsymbol{\alpha}, s) \tag{3.17}$$

$$\text{s. t.} \sum_{i:\alpha_i = s} \varsigma_i^j \leq 1, \quad \forall j, s, \tag{3.18}$$

$$J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq \rho \mathbb{E}[L(\mathbf{x}_{i,k}) \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}] + c_i \ \forall i, \tag{3.19}$$

$$1 \leq \alpha_i \leq S, \quad \forall i, \tag{3.20}$$

$$\boldsymbol{\varsigma}_i \in \mathcal{S}, \ \forall i, \quad \boldsymbol{\mu} \in \mathcal{M}^m, \quad \boldsymbol{\alpha} \in \mathbb{Z}_+^m, \quad S \in \mathbb{Z}_+. \tag{3.21}$$

Observe that the objective in (3.17) is now to minimize the total transmission time, rather than being forced as an explicit constraint. In this way, the optimization problem defined in (3.17)-(3.21) can be viewed as an alternative to the latency constrained problem in

(3.10)-(3.14). Because the scheduling algorithm we develop in this chapter requires the ability to quickly identify feasible solutions, we focus our attention on the control-constrained formulation in (3.17)-(3.21). Before presenting the details of the scheduling algorithm, we present a brief remark regarding the addition of "safety", or worst-case, constraints to either problem formulation.

Remark 7. The control constraint in (3.19) is formulated to guarantee an average decrease of expected Lyapunov value by a rate of ρ. This is of interest to ensure the system states are driven to zero over time. However, in practical systems we may also be interested in protecting against worst-case behavior, e.g. entering an unsafe or unstable region. Consider a vector $\mathbf{b}_i \in \mathbb{R}^p$ as the boundary of safe operation of system i. A constraint that protects against exceeding this boundary can be written as

$$\mathrm{P}\left[[] \, |\mathbf{x}_{i,k+1}| \geq \mathbf{b}_i \mid \hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \varsigma_i\right] \leq \delta, \tag{3.22}$$

where $\delta \in (0, 1)$ is small. The expression in (3.22) can be included as an additional constraint to either the latency-constrained or control-constrained scheduling problems previously discussed.

3.4 Control-Aware Low-Latency Scheduling (CALLS)

We develop a control-aware low-latency scheduling (CALLS) algorithm to approximately solve the control-constrained scheduling formulation in (3.17)-(3.21). Because this problem is combinatorial in nature, it is infeasible to solve exactly. Instead, we focus on a practical and efficient means of solving approximately. In particular, we identify sets of feasible points and use a heuristic approach towards minimizing the transmission time objective among the set of feasible points. Additionally, within the development of the CALLS method we identify and characterize new PDR requirements that are defined relative to the control system requirements; these are generally significantly less strict than the PDR requirements often considered in general high reliability communication systems without codesign. Overall, the CALLS method consists of (i) the derivation of adaptive control-aware PDR targets, (ii) a principled random selection of devices to schedule to reduce latency, and (iii) the use of assignment based methods to find a low-latency schedule. We discuss these three components in detail in the proceeding subsections.

3.4.1 Control adaptive PDR

Due to the complexity of the scheduling problem in (3.17)-(3.21), we first focus our attention on identifying scheduling parameters $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$ that are feasible, i.e. satisfy

the constraints in (3.18)-(3.21). In particular, the Lyapunov control constraint in (3.19) is of significant interest. Recall that the control cost function $J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ is itself determined by the PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$, as per (3.8). Thus, the constraint in (3.19) can be seen as indirectly placing a constraint on the required PDR necessary to achieve a ρ_i-rate decrease in expectation. The equivalent condition on PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ is presented in the following proposition.

Proposition 3. *Consider the Lyapunov control constraint in* (3.19) *and the definition of* $J_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}, \mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ *given in* (3.8). *Define the closed-loop state transition matrix* $\mathbf{A}_i^c := \mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i$ *and* j*-accumulated noise* $\omega_i^j := Tr[(\mathbf{A}_i^T\mathbf{P}^{1/j}\mathbf{A}_i)^j\mathbf{W}_i]$. *The control constraint in* (3.19) *is satisfied for device* i *if and only if the following condition on PDR* $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ *holds,*

$$q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \geq \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}) := \tag{3.23}$$
$$\frac{1}{\Delta_i}\left[\left\|(\mathbf{A}_i^c - \rho_i\mathbf{I})\hat{\mathbf{x}}_{i,k}^{(l_i)}\right\|_{\mathbf{P}^{\frac{1}{2}}}^2 + (1-\rho_i)\sum_{j=0}^{l_i-1}\omega_i^j + \omega_i^{l_i} - c_i\right],$$

where we have further defined the constant

$$\Delta_i := \sum_{j=0}^{l_i-1}[\omega_i^{j+1} - Tr(\mathbf{A}_i^{cT}(\mathbf{A}_i^T\mathbf{P}^{1/j}\mathbf{A}_i)^j\mathbf{A}_i^c\mathbf{W}_i)]. \tag{3.24}$$

Proof: Consider the Lyapunov decrease constraint as written in (3.19). As the same logic holds for all i and k, for ease of presentation we remove all subscripts when presenting the details of this proof. We further introduce the simpler notation $q := q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$. Now, we may expand the left hand side of (3.19) be rewriting the definition in (3.8) as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = q\mathbb{E}_{\mathbf{w}}[L(\mathbf{A}_c\mathbf{x} + \mathbf{w})] \tag{3.25}$$
$$+ (1-q)\mathbb{E}_{\mathbf{w}}[L(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\hat{\mathbf{x}} + \mathbf{w})].$$

Recall the definition of the quadratic Lyapunov function $L(\mathbf{x}) := \mathbf{x}^T\mathbf{P}\mathbf{x}$ for some positive definite $\mathbf{P}$. Further recall the relation $\mathbf{x} = \hat{\mathbf{x}} + \mathbf{e}$ as described by (3.5). Combining these, we expand the right hand size of (3.25) as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \tag{3.26}$$
$$q\mathbb{E}_{\mathbf{w}}\left[\mathbf{A}_c\left(\hat{\mathbf{x}} + \mathbf{e}\right) + \mathbf{w}\right]^T\mathbf{P}\left[\mathbf{A}_c\left(\hat{\mathbf{x}} + \mathbf{e}\right) + \mathbf{w}\right]$$
$$+ (1-q)\mathbb{E}_{\mathbf{w}}\left[\mathbf{A}_c\hat{\mathbf{x}} + \mathbf{A}\mathbf{e} + \mathbf{w}\right]^T\mathbf{P}\left[\mathbf{A}_c\hat{\mathbf{x}} + \mathbf{A}\mathbf{e} + \mathbf{w}\right].$$

To evaluate the expectations in (3.26), recall the random noise $\mathbf{w}$ follows a Gaussian distribution with zero mean and covariance $\mathbf{W}$. Thus, the expectation can be evaluated over $\mathbf{w}$ and expanded as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \tag{3.27}$$
$$q\left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \text{Tr}(\mathbf{PW}) + \sum_{j=0}^{l-1}\text{Tr}(\mathbf{A}_c(\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{A}_c\mathbf{W})\right] +$$
$$(1-q)\left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \text{Tr}(\mathbf{PW}) + \sum_{j=1}^{l}\text{Tr}((\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{W})\right].$$

From here, we rearrange terms and substitute the notation $\omega^j := \text{Tr}[(\mathbf{A}^T\mathbf{P}^{1/j}\mathbf{A})^j\mathbf{W}]$ to obtain that the control cost can be written as

$$J(\hat{\mathbf{x}}^{(l)}, \mathbf{h}, \mu, \boldsymbol{\varsigma}) = \left[\|\mathbf{A}_c\hat{\mathbf{x}}\|^2_{\mathbf{P}^{\frac{1}{2}}} + \text{Tr}(\mathbf{PW}) + \sum_{j=1}^{l}\omega^j\right] \tag{3.28}$$
$$+ q\sum_{j=0}^{l-1}[\text{Tr}(\mathbf{A}_c(\mathbf{A}^T\mathbf{P}^{\frac{1}{j}}\mathbf{A})^j\mathbf{A}_c\mathbf{W}) - \omega^{j+1}].$$

With (3.28), we have expanded the control cost in terms of the PDR q. Now, we return to the constraint in (3.19). Recall the expansion for $\mathbb{E}[L(\mathbf{x}) \mid \hat{\mathbf{x}}^{(l)}]$ via (3.6). By combining this with the expansion in (3.28), the terms in(3.19) can be rearranged to obtain the inequality in (3.23). ■

In Proposition 3 we establish a lower bound $\tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})$ on the PDR of device i that is dependent upon the current estimated state $\hat{\mathbf{x}}^{(l_i)}_{i,k}$ and system dynamics determined by $\mathbf{A}^c_i, \mathbf{A}_i$, and $\mathbf{W}^i$. We may note the following intuitions about the constraint in (3.23). The PDR condition naturally grows stricter as the bound $\tilde{q}_i(\hat{\mathbf{x}}^{(l_i)}_{i,k})$ defined on the right hand side of (3.23) gets larger. The first term on the right hand side reflects the current estimated channel state, and will become larger as the state gets larger. Similarly, the latter two terms on the right hand side together reflect the size of the noise that has accumulated by operating in open loop. When the noise variance $\mathbf{W}_i$ is high and when the last-update counter l_i is large, these latter two noise terms will both be large. Thus, both the current magnitude of the control state and the growing uncertainty from infrequent transmissions together determine how large is the PDR requirement in (3.23).

We stress the value of the PDR condition in (3.23) is both in its adaptability to the control system state and dynamics, as well as its identification of precise target delivery rates that are necessary to keep the control systems moving towards stability on average.

Depending on the particular system dynamics as described in (3.35), such PDR's may be, and often are considerably more lenient than the default target transmission success rates used in practical wireless systems (e.g. $q = 0.999$). Thus, through (3.23) we make a claim that, with knowledge of the control system dynamics and targeted *control performance*, we can effectively soften the targeted *communication performance*—or "reliability"— accordingly to something more easily obtained in low-latency constrained systems.

Remark 8. It is worthwhile to note that by placing a stricter Lyapunov decrease constraint with smaller rate ρ_i in (3.19), then the first term on the right hand side of (3.23) also grows larger and increases the necessary PDR. Generally, selecting a smaller ρ will result in a faster convergence to stability but will require stricter communication requirements. In fact, we may use the inherent bound on the probability $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \leq 1$ to find a lower bound on the Lyapunov decrease rate ρ_i that can be feasibly obtained based upon current control state and system dynamics. This bound, however, may not be obtainable in practice due to the scheduling constraints. In practice, we select ρ_i to be in the interval $[0.90, 0.1)$.

3.4.2 Selective scheduling

We now proceed to describe the procedure with which we can find a set of feasible scheduling decisions $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$. To begin, we first consider a stochastically *selective scheduling* protocol, whereby we do not attempt to schedule every device at each transmission cycle, but instead select a subset to schedule a principled random manner. Define by $\nu_{i,k} \in [0, 1]$ the probability that device i is included in the transmission schedule at time k and further recall by $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i)$ to be the packet delivery rate with which it transmits. Then, we may consider the *effective* packet delivery rate $\hat{q}$ as

$$\hat{q}(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) = \nu_{i,k} q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \tag{3.29}$$

Selective scheduling is motivated by the ultimate goal of minimizing total transmit time as described in the objective in (3.17). As we consider a large number of total devices m, scheduling all such devices will require a larger number of PPDU slots—a maximum of 9 devices can transmit within a single PPDU. Recall in (3.7) that each additional PPDU requires unavoidable overhead in τ_0, which in aggregation over multiple PPDUs may become a significant bottleneck in minimizing $\hat{\tau}$ or meeting a strict latency requirement τ_{max}. Thus, by decreasing the amount of scheduled devices, we may decrease the number of total PPDUs and the overhead that is added to the total transmission time.

Observe that by introducing the term ν_i to the evaluation of effective PDR $\tilde{q}_i$ in (3.29), we would thus need to transmit with higher PDR $q(\mathbf{h}_{i,k}, \mu_i, \boldsymbol{\varsigma}_i) \geq \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})/\nu_{i,k}$ to meet the condition in (3.23). While imposing a tighter PDR requirement will indeed require longer

transmission times, this added time cost is generally less than the transmission overhead of additional PPDUs. In this work, we use the determine scheduling probability of device i through its PDR requirement $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ as

$$\nu_{i,k} := e^{\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})-1}. \tag{3.30}$$

With (3.30), the probability of scheduling device i increases as the required PDR increases. Notice that, when a transmission is required, i.e. $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)}) = 1$, then device i is included in the scheduling with probability 1. In general, devices with very high PDR requirements, e.g. > 0.99, will be scheduled with very high probability. Thus, the transmission time gains that are provided through selective scheduling using (3.30) would be minimal, if non-existent, in high-reliability settings in which PDR requirements remain high at all times. However, with the lower PDR requirement obtained through the control-aware scheduling in (3.23), selective scheduling as the potential to create significant time savings, as will be later shown in Section 3.5 of this chapter.

3.4.3 Assignment-based scheduling

We now proceed to discuss how the PDR requirements previously derived are used to schedule the devices during a TXOP. Rather than employing a greedy method as is commonly done in wireless scheduling problems, in the proposed method we use assignment-type methods. In such assignment-type methods, we assign all scheduled devices to a PPDU and RU at the beginning of the TXOP rather than make scheduling decisions after each PPDU. To begin, we must determine a set of schedules that satisfy the constraints in (3.18)-(3.21). Recall each device i is selected to be scheduled at cycle k with probability $\nu_{i,k}$ and define the set of m_k devices to selected be scheduled as $\mathcal{I}_k \subseteq \{1, 2, \ldots, m\}$ where $|\mathcal{I}_k| = m_k$. To specify the sets of RUs that we consider in our scheduling, we first define some notation necessary in the description. We define $\hat{\mathcal{S}}_{(n)} \subset \mathcal{S}$ to be an arbitrary set of RUs that do not intersect over any frequency bands (i.e. satisfy the constraint in (3.18)) with exactly n elements. To accommodate the m_k devices to be scheduled, we consider a set of S_k such sets $\hat{\mathcal{S}}_{(n_s)}$ with size n_s, whose combined elements total $\sum_{s=1}^{S_k} n_s = m_k$. In other words, we identify a set S_k PPDUs in which the sth PPDU contains n_s non-intersecting PPDUs. We define this full set of assignable RUs at cycle k as

$$\mathcal{S}'_k := \hat{\mathcal{S}}^1_{(n_1)} \cup \hat{\mathcal{S}}^1_{(n_2)} \cup \ldots \cup \hat{\mathcal{S}}^{S_k}_{(n_{S_k})}. \tag{3.31}$$

Note that in (3.31) we further superindex each set by a PPDU index s, in order to stress that elements are distinct between sets. That is, an RU $\boldsymbol{\varsigma}$ present in sets $\hat{\mathcal{S}}^x_{(n_x)}$ and $\hat{\mathcal{S}}^y_{(n_y)}$ is considered as two distinct elements in $\mathcal{S}'_k$, denoted $\boldsymbol{\varsigma}^x$ and $\boldsymbol{\varsigma}^y$, respectively. In this way

PPDU 1	PPDU 2	PPDU 3
RU 1	RU 10	RU 13
RU 2		
RU 3	RU 11	
RU 4		
RU 5	RU 12	RU 14
RU 6		
RU 7		
RU 8		
RU 9		

Table 3.2: Example of RU selection with $m_k = 14$ devices. There are a total of $S_k = 3$ PPDUs, given $n_1 = 9$, $n_2 = 3$, $n_3 = 2$ RUs, respectively.

(3.31) defines a complete set of combinations of frequency-allocated RU and *time*-allocated PPDUs to assign devices during this cycle. We point out that there are numerous ways in which to define such sets of RUs in each PPDU that total m_k assignments. There are various heuristic methods that may be employed to quickly identify a permissible assignment pool $\mathcal{S}'_k$, and various simple heuristics may be developed to make this selection in a manner that reduces the overall latency of the transmission window. An example of the set $\mathcal{S}'_k$ for scheduling $m_k = 14$ devices is shown in Table 3.2.

For all $i \in \mathcal{I}_k$ and RU $\boldsymbol{\varsigma} \in \mathcal{S}'_k$, define the largest affordable MCS given the *modified* PDR requirement $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})/\nu_{i,k}$ by

$$\mu_{i,k}(\boldsymbol{\varsigma}) := \begin{cases} \max\{\mu \mid q(\mathbf{h}_{i,k}, \mu, \boldsymbol{\varsigma}) \geq \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})/\nu_{i,k}\} \\ 1, \quad \text{if } q(\mathbf{h}_{i,k}, \mu, \boldsymbol{\varsigma}) < \tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})/\nu_{i,k} \ \forall \mu \end{cases} \tag{3.32}$$

Observe in (3.32) that, when no MCS achieves the desired PDR in a particular RU, this value is set to $\mu = 1$ by default. The above adaptive MCS selection can be achieved based on channel conditions using the techniques outlined in [58]. This MCS selection subsequently then yields a corresponding time cost $\tau(\mu_{i,k}(\boldsymbol{\varsigma}), \boldsymbol{\varsigma})$ for assigning device i to RU $\boldsymbol{\varsigma}$. Further define an 3-D assignment tensor V—where $v_{ij}^s = 1$ when device i is assigned to RU $\boldsymbol{\varsigma}_j^s$ and 0 otherwise—and $\mathcal{V}$ as the set of all possible assignments. Recalling the form of the total transmission time given PPDU arrangements in (3.7), the assignment that minimizes total transmission time is given by

$$V^* = \underset{V \in \mathcal{V}}{\operatorname{argmin}} \sum_{s=1}^{S} \max_j \left[v_{ij}^s \tau(\mu_{i,k}(\boldsymbol{\varsigma}_j^s), \boldsymbol{\varsigma}_j^s) \right]. \tag{3.33}$$

The expression in (3.33) can be identified as a particular form of the *assignment problem*,

Algorithm 2 Control-Aware Low Latency Scheduling (CALLS) at cycle k

1: **Parameters:** Lyapunov decrease rate ρ
2: **Input:** Channel conditions $\mathbf{H}_k$ and estimated states $\hat{\mathbf{X}}_k$
3: Compute target PDR $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ for each device i [cf. (3.23)].
4: Determine selection probabilities $\nu_{i,k}$ for each device [cf. (3.30)].
5: Select devices $\mathcal{I}_k$ with probs. $\{\nu_{1,k}, \ldots, \nu_{m,k}\}$
6: Determine set of RUs/PPDUs $\mathcal{S}'_k$ [cf. (3.31)].
7: Determine maximum MCS for each device/RU assignment [cf. (3.32)].
8: Schedule selected devices via assignment method.
9: **Return:** Scheduling variables $\{\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k\}$

a common combinatorial optimization problem in which the selection of mutually exclusive assignment of agents to tasks incurs some cost. Here, the cost is the total transmission time across all PPDUs necessary for scheduled devices to meet the target PDRs. Assignment problems are generally very challenging to solve—there are $m_k!$ combinations—although polynomial-time algorithms exist for simple cases. The Hungarian method [67], for example, is a standard method for solving linear-cost assignment problems. While the cost we consider in (3.33) is nonlinear, the Hungarian method may be used as an approximation. Alternatively, other heuristic assignment approaches may be designed to approximate the solution to (3.33). We note that, for the simulations performed later in this chapter, we apply such a heuristic method, the details of which are left out for proprietary reasons.

By combining these methods with the control-based PDR targets and selective scheduling procedure, we obtain the complete control-aware low-latency scheduling (CALLS) algorithm. The steps as performed by the centralized AP/controller are outlined in Algorithm 2. At each cycle k, the AP determines the scheduling parameters based on the current channel states $\mathbf{H}_k$ (obtained via pilot signals) and the current estimated control states $\hat{\mathbf{X}}_k$ (obtained via (3.2) for each device i). With the current state estimates, the AP computes target PDRs $\tilde{q}_i(\hat{\mathbf{x}}_{i,k}^{(l_i)})$ for each device via (3.23) in Step 3. In Step 4, the target PDRs are used to establish selection probabilities $\nu_{i,k}$ for each agent with (3.30). After randomly selecting devices $\mathcal{I}_k$ with their associated probabilities in Step 5, the set of RUs and PPDUs $\mathcal{S}'_k$ are determined in Step 6 as in (3.31) , based upon the number of devices selected to be scheduled $|\mathcal{I}_k|$. In Step 7, the associated MCS values are determined each possible assignment of device to RU via (3.32). Finally, in Step 8 the assignment is performed using either the Hungarian method [67] or other user-designed heuristic assignment method. The resulting assignment determines the scheduling parameters $\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k, \boldsymbol{\alpha}_k$ for the current cycle.

Remark 9. Observe that the CALLS method as outlined in Algorithm 2 seeks to minimize the total latency of the transmission but does not explicitly prevent latency from exceeding some specific threshold $\tau_{\max}$. In practical systems, this limit may need to be enforced. In

such a setting, the CALLS method can be modified so that all devices scheduled in PPDUs whose transmission end after $\tau_{\max}$ seconds do not transmit.

Remark 10. In practical systems, the channel state information $\mathbf{H}_k$ is often obtained with some estimation errors, which may impact the channel aware scheduling approach taken here. Observe, however, that the computation of adaptive PDR targets in (3.23) does not depend upon channel state information. Thus, the primary component of the CALLS method—namely Steps 3-6 in Algorithm 2— is unaffected by inaccurate channel estimation. Likewise, the assignment method-based scheduling in Step 8 does also not directly depend upon channel information—see the formulation of the assignment problem in (3.33). The only component that may be negatively impacted by estimation errors is Step 7, i.e. the maximum MCS selection in (3.32). Here, estimation errors may result in the selection of an MCS that cannot meet the target PDR targets. Such an effect can be mitigated by selecting a smaller, more conservative MCS to account for channel estimation errors. Such a conservative scheme would tradeoff latency to the benefit of reliability or robustness.

3.5 Simulation Results

In this section, we simulate the implementation of both the control-aware CALLS method and a standard "control-agnostic" scheduling methods for various low-latency control systems over a simulated wireless channel. We point out the low-latency based scheduling/assignment approaches of both methods being compared are identical, with the distinguishing features being the dynamic control-aware packet delivery rates incorporated in the CALLS method. In doing so, we may analyze the performance of the control-aware design outlined in the previous section relative to a standard latency-aware approach in terms of, e.g., number of users supported with fixed latency threshold or best latency achieved with fixed number of users. As we are interested primarily in low latency settings that tightly restrict the communication resources, we consider two standard control systems whose rapidly changing state requires high sampling rates, and consequently a communication latency on the order of milliseconds. The parameters for the simulation setup are provided in Table 3.3. The wireless fading channel modeled using IEEE Indoor Channel Model E. In our performance analysis, we use link layer abstractions commonly used for wireless system level simulations (SLS) to model the wireless physical layer. In this approach, the AWGN SINR-BLER curves are used to evaluate the packet delivery rate function $q(\mathbf{h}, \mu, \boldsymbol{\varsigma})$; note that the curves are evaluated at the effective SNR (ESINR) values that take into account the instantaneous fading channel conditions and selected MCS. The transmission time $\tau(\mu, \boldsymbol{\varsigma})$ is computed in the simulations using the associated data rates of an MCS in Table 3.1 for a 100 byte packet and overhead (e.g. TFs) of the 802.11ax specifications. The latency overhead for this setting

Channel model	IEEE Model E (indoor) [75]
Sensor to AP distances	Random (1 to 50 meters)
Transmit power	23 dbm
Channel bandwidth	20 MHz
RU sizes	2, 4, 8, 20 MHz
# of antennas at AP	2
# of antennas at sensors	1
MCS options	See Table 3.1
State sampling period	10 ms

Table 3.3: Simulation setting parameters.

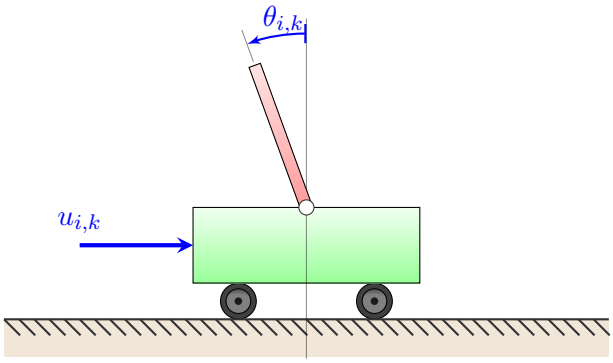


Figure 3.3: Inverted pendulum-cart system i. The state $\mathbf{x}_{i,k} = [x_{i,k}, \dot{x}_{i,k}, \theta_{i,k}, \dot{\theta}_{i,k}]$ contains specifies angle $\theta_{i,k}$ of the pendulum to the vertical, while the input $u_{i,k}$ reflects a horizontal force on the cart.

amounts to approximately $\tau_0 \approx 100\mu$s.

3.5.1 Inverted pendulum system

We perform an initial set of simulations on the well-studied problem of controlling a series of inverted pendulums on a horizontal cart. While conceptually simple, the highly unstable dynamics of the inverted pendulum make it a representative example of control system that requires fast control cycles, and subsequently low-latency communications when being controlled over a wireless medium. Consider a series of m identical inverted pendulums, as pictured in Figure 3.3. Each pendulum of length L is attached at one end to a cart that can move along a single, horizontal axis. The position of the pendulum changes by the effects of gravity and the force applied to the linear cart. For our experiments, we use the modeling of the inverted pendulum as provided by Quanser [99]. The state is $p = 4$ dimensional vector that maintains the position and velocity of the cart along the horizontal axis, and the angular position and velocity of the pendulum, i.e. $\mathbf{x}_{i,k} := [x_{i,k}, \dot{x}_{i,k}, \theta_{i,k}, \dot{\theta}_{i,k}]$. The system input $u_{i,k}$ reflects a horizontal force placed upon ith pendulum. By applying a zeroth order hold on the continuous dynamics with a state sampling rate of 0.01 seconds and linearizing,

we obtained the following discrete linear dynamic matrices of the pendulum system

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0.037 & 3.477 & 0.042 \\ 0 & 2.055 & -0.722 & 4.828 \\ 0 & 0.023 & 0.91 & 0.037 \\ 0 & 0.677 & -0.453 & 2.055 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} 0.034 \\ 0.168 \\ 0.019 \\ 0.105 \end{bmatrix}. \tag{3.34}$$

Because the state $\mathbf{x}_{i,k}$ measures the angle of the ith pendulum at time k, the goal is to keep this close to zero, signifying that the pendulum remains upright. The input matrix $\mathbf{K}$ is computed to be a standard LQR-controller.

We perform a set of simulations scheduling the transmissions to control a series of inverted pendulums, varying both the latency threshold $\tau_{\max}$ and number of devices m. We perform the scheduling using the proposed CALLS method for control-aware low latency scheduling an, as a point of comparison, consider scheduling using a fixed "high-reliability" PDR of 0.99 for all devices. Each simulation is run for a total of 1000 seconds and is deemed "successful" if all pendulums remain upright for the entire run. We perform 100 such simulations for each combination of latency threshold and number of devices to determine how many devices we can support at each latency threshold using both the CALLS and fixed-PDR methods for scheduling.

In Figure 3.4 we show the results of a representative simulation of the control of $m = 25$ pendulum systems with a latency bound of $\tau_{\max} = 10^{-3}$ seconds. In both graphs we show the average distance from the center vertical of each pendulum over the course of 1000 seconds. In the top figure, we see by using the control-aware CALLS method we are able to keep each of the 25 pendulums close to the vertical for the whole simulation. Meanwhile, using the standard fixed PDR, we are unable to meet the scheduling limitations imposed by the latency threshold, and many of the pendulums swing are unable to be kept upright, as signified by the large deviations from the origin. This is due to the fact that certain pendulums were not scheduled when most critical, and they subsequently became unstable.

We present in Figure 3.5 the final capacity results obtained over all the simulations. We say that a scheduling method was able to successfully serve m' devices if it keeps all devices within a $|\theta_{i,k}| \leq 0.05$ error region for 100 independent simulations. Observe that the proposed approach is able to increase the number of devices supported in each case, with up to 1.5 factor increase over the standard fixed PDR approach. Indeed, the proposed CALLS method is able to allocate the available resource in a more principled manner, which allows for the support of more devices simultaneously being controlled.

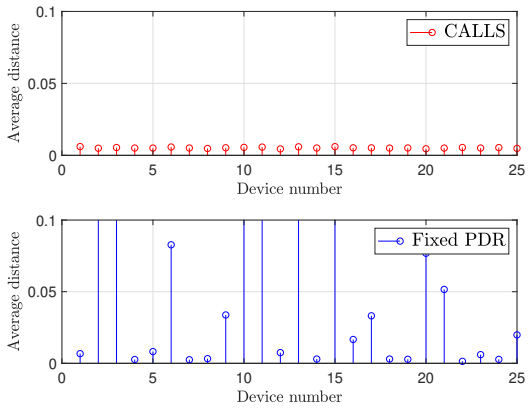


Figure 3.4: Average pendulum distance to center vertical for $m = 25$ devices using (top) CALLS and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The proposed control aware scheme keeps all pendulums close to the vertical, while fixed-PDR scheduling cannot.

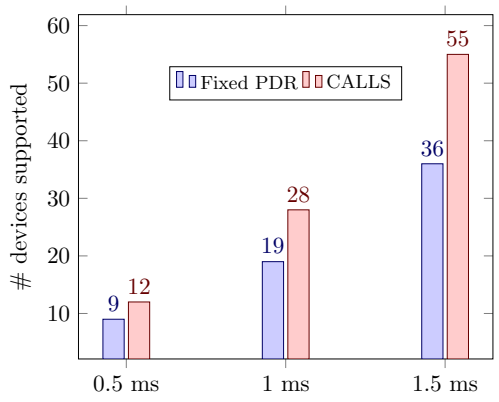


Figure 3.5: Total number of inverted pendulum devices that can be controlled using Fixed-PDR and CALLS scheduling for various latency thresholds.

3.5.2 Balancing board ball system

We perform another series of experiments on the wireless control of a series of balancing board ball systems developed by Acrome [2]. In such a system, a ball is kept on a rectangular board with a single point of stability in the center of the board. Two servo motors underneath the board are used to push the board in the horizontal and vertical directions, with the objective to keep the ball close to the center of the board. The state here reflects the position and velocity in the horizontal and vertical axes, i.e. $\mathbf{x}_{i,k} := [x_{i,k}, \dot{x}_{i,k}, y_{i,k}, \dot{y}_{i,k}]$. The input $\mathbf{u}_{i,k} = [v_x, v_y]$ reflects the voltage applied to the horizontal and vertical motors. As before, we apply a zeroth order hold on the continuous dynamics with a state sampling rate of 0.01 seconds and linearize, thus obtaining the following dynamic system matrices,

$$\mathbf{A}_i = \begin{bmatrix} 1 & 0.01 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.01 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} -0.0001 & 0 \\ -0.02 & 0 \\ 0 & -0.00008 \\ 0 & -0.01 \end{bmatrix}. \tag{3.35}$$

As before, we compute the control matrix $\mathbf{K}$ using standard LQR-control computation.

In the simulations performed with the balancing board system, in addition to making comparisons of the CALLS method to a fixed PDR low latency scheduling scheme, we perform additional comparisons to a standard control-aware scheduling approach—namely, the event-triggered scheduling approach [17, 80]. In event triggered scheduling, we schedule devices only when its estimated control state goes above some threshold value. When such an event occurs, this device is scheduled with a fixed high reliability PDR using a low-latency assignment based scheduling method. This, in effect, combines the selective scheduling approach of CALLS with fixed high reliability PDR targets commonly used in URLLC.

In Figure 3.6 we show the results of a representative simulation of the control of $m = 50$ balancing board ball systems with a latency bound of $\tau_{\max} = 10^{-3}$ seconds. Observe that, in this system, even with a large number of users, both the event-triggered scheduling and the CALLS method can keep all systems very close to the center of the board, while the fixed PDR scheduler loses a few of the balls due to the agnosticism of the scheduler.

To dive deeper into the benefits provided by control aware scheduling, we present in Figure 3.7 a histogram of the actual packet delivery rates each of the devices achieved over the representative simulation. It is interesting to observe that, for the CALLS method, the achieved PDRs are closely concentrated, ranging from 0.3 to 0.44. On the other hand, using either event-triggered or a fixed PDR scheduling scheme, the non-variable rates are too strict for the low-latency system to support, and without control-aware scheduling the achieved PDRs range wildly from close to 0 to close to 1. In this case, some devices are able

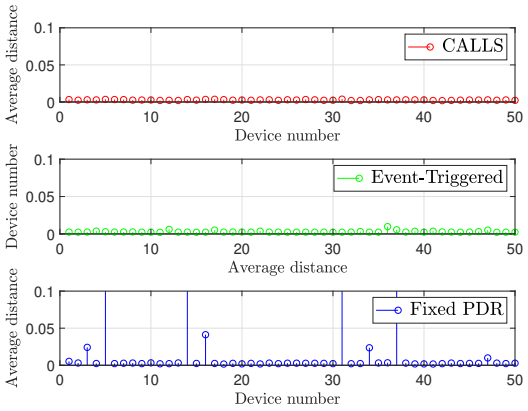


Figure 3.6: Average ball distance to center for $m = 50$ devices using (top) CALLS, (middle) event-triggered, and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The control aware schemes keeps all balancing balls close to center, while fixed-PDR scheduling cannot.

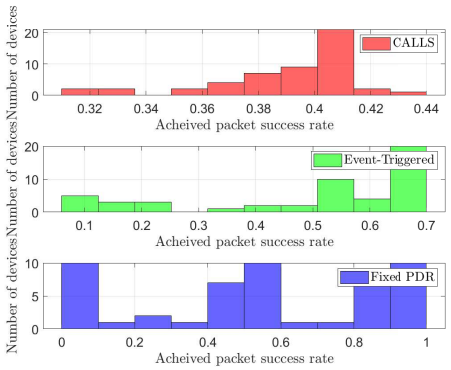


Figure 3.7: Histogram of achieved PDRs in $m = 50$ balancing board systems (top) CALLS, (middle) event-triggered, and (bottom) fixed-PDR scheduling with $\tau_{\max} = 1$ ms latency threshold. The proposed CALLS method achieves similar PDRs for all devices, while the fixed-PDR and event-triggered scheduling results in large variation in packet delivery rates.

to transmit almost every cycle while others are almost never able to successfully transmit their packets. This suggests that, by using control aware scheduling, we indirectly achieve a sense of fairness across users over the long term. Further note that the PDRs required to keep the balancing board ball stable, e.g. 0.4, are relatively small. This is due to the fact that the balancing board ball features relatively slow moving dynamics, making it easier to control with less frequent transmissions. This is comparison to the inverted pendulum system, in which the pendulums were kept stable with PDRs in the range 0.6-0.75.

We present in Figure 3.8 the final capacity results obtained over all the simulations for the balancing board ball system. Observe that proposed approach increases the number of supported devices by factor of 2 relative to the standard fixed PDR approach. The even greater improvement here relative to the inverted pendulum simulations can be attributed

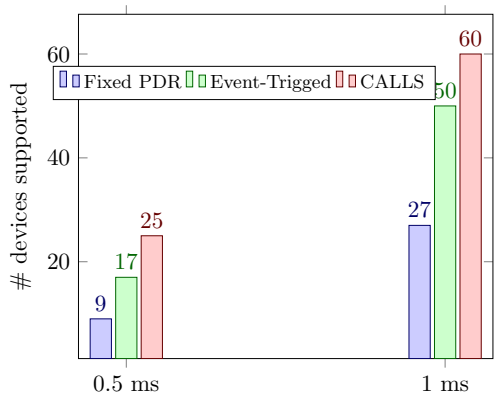


Figure 3.8: Total number of balancing ball board devices that can be controlled using Fixed-PDR, Event-Triggered, and CALLS scheduling for various latency thresholds.

to the slower dynamics of the balancing board ball, which allows for even more gains using control-aware PDRs due to the lower PDR requirements of the system. Likewise, the Lyapunov-based adaptive PDR requirements allow for even greater scalability than the more standard event-triggered approach, which can service only 17 and 50 users with 0.5 and 1 ms latency thresholds, respectively.

3.6 Discussion and Conclusions

In this chapter we proposed a novel control-communication co-design approach to solving the radio resource allocation problem for time-sensitive wireless control systems. Given a channel state and control state, we mathematically derive a minimum packet delivery rate a device must meet to maintain a control-orientated target, as defined by a stability-inducing Lyapunov function. By dynamically assigning variable packet delivery rate targets to each device based on its current conditions, we are able to more easily meet feasibility requirements of a latency-constrained wireless control problem and maintain stability and strong performance. We perform simulations on numerous well-studied low-latency control problems to demonstrate the benefits of using the control-aware approach, which can include a 2x gain on number of devices that can be supported. In future research, we aim to investigate how more sophisticated and realistic modeling, such as non-linear control or actuation over wireless links, may be used in this control-aware framework.

The results presented in this chapter suggest an interesting potential for control-aware resource allocation and scheduling, particularly in low-latency industrial systems. By considering the control-specific targets such as maintaining stability or an error margin, we observe that the standard high reliability targets considered in URLLC (e.g. packet delivery rates ≥ 0.999) can in some cases be substantially stricter than necessary for adequate performance.

Wireless control systems with sufficiently slow dynamics can be kept stable with much lower packet delivery rates, which in turn make low-latency communications more achievable. Furthermore, in realistic industrial systems there will be many heterogeneous devices being controlled, whose variation in communication needs is well-served by control-aware adaptivity proposed in this chapter. This suggests the potential for wireless communications to be adopted using a smart control-communication co-design approach even while ultra-reliable wireless system technology remains under development.

Chapter 4

Primal-Dual Learning for Wireless Systems

4.1 Introduction

The defining feature of wireless communication is fading and the role of optimal wireless system design is to allocate resources across fading states to optimize long term system properties. Mathematically, we have a random variable $\mathbf{h}$ that represents the instantaneous fading environment, a corresponding instantaneous allocation of resources $\mathbf{p}(\mathbf{h})$, and an instantaneous performance outcome $\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)$ resulting from the allocation of resources $\mathbf{p}(\mathbf{h})$ when the channel realization is $\mathbf{h}$. The instantaneous system performance tends to vary too rapidly from the perspective of end users for whom the long term average $\mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)\big]$ is a more meaningful metric. This interplay between instantaneous allocation of resources and long term performance results in distinctive formulations where we seek to maximize a utility of the long term average $\mathbf{x}$ subject to the constraint $\mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)\big]$. Problems of this form range from the simple power allocation in wireless fading channels – the solution of which is given by water filling – to the optimization of frequency division multiplexing [126], beamforming [8, 112], and random access [61, 62].

Optimal resource allocation problems are as widespread as they are challenging. This is because of the high dimensionality that stems from the variable $\mathbf{p}(\mathbf{h})$ being a function over a dense set of fading channel realizations and the lack of convexity of the constraint $\mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)\big]$. For resource allocation problems, such as interference management, heuristic methods have been developed [18, 111, 130]. Generic solution methods are often undertaken in the Lagrangian dual domain. This is motivated by the fact that the dual problem is not functional, as it has as many variables as constraints, and is always convex whether the original problem is convex or not. A key property that enables this solution is the lack of duality gap, which allows dual operation without loss of optimality. The duality

gap has long being known to be null for convex problems – e.g., the water level in water filling solutions is a dual variable – and has more recently being shown to be null under mild technical conditions despite the presence of the nonconvex constraint $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ [103, 136]. This permits dual domain operation in a wide class of problems and has lead to formulations that yield problems that are *more* tractable, although not necessarily tractable without resorting to heuristics [31, 42, 47, 76, 92, 124, 138].

The inherent difficulty of resource allocation problems makes the use of machine learning tools appealing. One may collect a training set composed of optimal resource allocations $\mathbf{p}^*(\mathbf{h}_k)$ for some particular instances $\mathbf{h}_k$ and utilize the learning parametrization to interpolate solutions for generic instances $\mathbf{h}$. The bottleneck step in this learning approach is the acquisition of the training set. In some cases this set is available by reverse engineering as it is possible to construct a problem having a given solution [43, 116]. In some other cases heuristics can be used to find approximate solutions to construct a training set [69, 70, 115]. This limits the performance of the learning solution to the performance of the heuristic, though the methodology has proven to work well at least in some particular problems.

Instead of acquiring a training set, one could exploit the fact that the expectation $\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ has a form that is typical of learning problems. Indeed, in the context of learning, $\mathbf{h}$ represents a feature vector, $\mathbf{p}(\mathbf{h})$ the regression function to be learned, $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ a loss function to be minimized, and the expectation $\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ the statistical loss over the distribution of the dataset. We may then learn without labeled training data by directly minimizing the statistical loss with stochastic optimization methods which merely observe the loss $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ at sampled pairs $(\mathbf{h}, \mathbf{p}(\mathbf{h}))$. This setting is typical of, e.g., reinforcement learning problems [117], and is a learning approach that has been taken in several *unconstrained* problems in wireless optimization [25, 93, 94, 134]. In general, wireless optimization problems *do* have constraints as we are invariably trying to balance capacity, power consumption, channel access, and interference. Still, the fact remains that wireless optimization problems have a structure that is inherently similar to learning problems. This realization is the first contribution of this paper:

(C1) Parametrizing the resource allocation function $\mathbf{p}(\mathbf{h})$ yields an optimization problem with the structure of a learning problem in which the statistical loss appears as a constraint (Section 4.2).

This observation is distinct from existing work in learning for wireless resource allocation. Whereby existing works apply machine learning methods to wireless resource allocation, such as via supervised training, here we identify that the wireless resource allocation is *itself* a statistical learning problem. This motivates the use of learning methods to directly solve the resulting optimization problems bypassing the acquisition of a training set. To do so, it

is natural to operate in the dual domain where constraints are linearly combined to create a weighted objective (Section 4.3). The first important question that arises in this context is the price we pay for learning in the dual domain. Our second contribution is to show that this question depends on the quality of the learning parametrization. In particular, if we use learning representations that are near universal—meaning that they can approximate any function up to a specified accuracy (Definition 1)—-we can show that dual training is close to optimal:

(C2) The duality gap of learning problems in wireless optimization is small if the learning parametrization is nearly universal (Section 4.3.1). More formally, the duality gap is $\mathcal{O}(\epsilon)$ if the learning parametrization can approximate arbitrary functions with error $\mathcal{O}(\epsilon)$ (Theorem 3).

A second question that we address is the design of training algorithms for optimal resource allocation in wireless systems. The reformulation in the dual domain gives natural rise to a gradient-based, primal-dual learning method (Section 4.3.2). The primal-dual method cannot be implemented directly, however, because computing gradients requires unavailable model knowledge. This motivates a third contribution:

(C3) We introduce a model-free learning approach, in which gradients are estimated by sampling the model functions and wireless channel (Section 4.4).

This model-free approach additionally includes the policy gradient method for efficiently estimating the gradients of a function of a policy (Section 4.4.1). We remark that since the optimization problem is not convex, the primal-dual method does not converge to the optimal solution of the learning problem but to a stationary point of the KKT conditions [14]. This is analogous to unconstrained learning where stochastic gradient descent is known to converge only to a local minima.

The quality of the learned solution inherently depends on the ability of the learning parametrization to approximate the optimal resource allocation function. In this paper we advocate for the use of neural networks:

(C4) We consider the use of deep neural networks (DNN) and conclude that since they are universal parameterizations, they can be trained in the dual domain without loss of optimality (Section 4.5).

Together, the Lagrangian dual formulation, model-free algorithm, and DNN parameterization provide a practical means of learning in resource allocation problems with near-optimality. We conclude with a series of simulation experiments on a set of common wireless resource allocation problems, in which we demonstrate the near-optimal performance of the proposed DNN learning approach (Section 4.6).

4.2 Optimal Resource Allocation in Wireless Communication Systems

Let $\mathbf{h} \in \mathcal{H} \subseteq \mathbb{R}^n_+$ be a random vector representing a collection of n stationary wireless fading channels drawn according to the probability distribution $m(\mathbf{h})$. Associated with each fading channel realization, we have a resource allocation vector $\mathbf{p}(\mathbf{h}) \in \mathbb{R}^m$ and a function $\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^u$. The components of the vector valued function $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ represent performance metrics that are associated with the allocation of resources $\mathbf{p}(\mathbf{h})$ when the channel realization is $\mathbf{h}$. In fast time varying fading channels, the system allocates resources instantaneously but users get to experience the average performance across fading channel realizations. This motivates considering the vector ergodic average $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big] \in \mathbb{R}^u$, which, for formulating optimal wireless design problems, is relaxed to the inequality

$$\mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]. \tag{4.1}$$

In (4.1), we interpret $\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ as the level of service that is available to users and $\mathbf{x}$ as the level of service utilized by users. In general we will have $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ at optimal operating points, but this is not required a priori.

The goal in optimally designed wireless communication systems is to find the instantaneous resource allocation $\mathbf{p}(\mathbf{h})$ that optimizes the performance metric $\mathbf{x}$ in some sense. To formulate this problem mathematically we introduce a vector utility function $\mathbf{g} : \mathbb{R}^u \to \mathbb{R}^r$ and a scalar utility function $g_0 : \mathbb{R}^u \to \mathbb{R}$, taking values $\mathbf{g}(\mathbf{x})$ and $g_0(\mathbf{x})$, that measure the value of the ergodic average $\mathbf{x}$. We further introduce the set $\mathcal{X} \subseteq \mathbb{R}^u$ and $\mathcal{P} \subseteq \mathcal{M}$, where $\mathcal{M}$ is the set of functions integrable with respect to $m(\mathbf{h})$, to constrain the values that can be taken by the ergodic average and the instantaneous resource allocation, respectively. We assume $\mathcal{P}$ contains bounded functions, i.e., that the resources being allocated are finite. With these definitions, we let the optimal resource allocation problem in wireless communication systems be a program of the form

$$\begin{aligned} P^* := \max_{\mathbf{p}(\mathbf{h}), \mathbf{x}} \; & g_0(\mathbf{x}), \\ \text{s.t.} \quad & \mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big], \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \; \mathbf{x} \in \mathcal{X}, \; \mathbf{p} \in \mathcal{P}. \end{aligned} \tag{4.2}$$

In (4.2) the utility $g_0(\mathbf{x})$ is the one we seek to maximize while the utilities $\mathbf{g}(\mathbf{x})$ are required to be nonnegative. The constraint $\mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big]$ relates the instantaneous resource allocations with the long term average performances as per (4.1). The constraints $\mathbf{x} \in \mathcal{X}$ and $\mathbf{p} \in \mathcal{P}$ are set restrictions on $\mathbf{x}$ and $\mathbf{p}$. The utilities $g_0(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are assumed to be

concave and the set $\mathcal{X}$ is assumed to be convex. However, the function $\mathbf{f}(\cdot, \mathbf{h})$ is not assumed convex or concave and the set $\mathcal{P}$ is not assumed to be convex either. In fact, the realities of wireless systems make it so that they are typically non-convex [103]. We present three examples below to clarify ideas and proceed to motivate and formulate learning approaches for solving (4.2).

Example 2 (Point-to-point wireless channel)**.** In a point-to-point channel we measure the channel state h and allocate power $p(h)$ to realize a rate $c(p(h); h) = \log(1 + hp(h))$ assuming the use of capacity achieving codes. The metrics of interest are the average rate $c = \mathbb{E}_h[c(p(h); h)] = \mathbb{E}_h[\log(1 + hp(h))]$ and the average power consumption $p = \mathbb{E}_h[p(h)]$. These two constraints are of the ergodic form in (4.1). We can formulate a rate maximization problem subject to power constraints with the utility $g_0(\mathbf{x}) = g_0(c, p) = c$ and the set $\mathcal{X} = \{p : 0 \leq p \leq p_0\}$. Observe that the utility is concave (linear) and the set $\mathcal{X}$ is convex (a segment). In this particular case the instantaneous performance functions $\log(1 + hp(h))$ and $p(h)$ are concave. A similar example in which the instantaneous performance functions are not concave is when we use a set of adaptive modulation and coding modes. In this case the rate function $c(p(h); h)$ is a step function [103].

Example 3 (Multiple access interference channel)**.** A set of m terminals communicates with associated receivers. The channel linking terminal i to the its receiver is h^{ii} and the interference channel to receiver j is given by h^{ji}. The power allocated in this channel is $p^i(\mathbf{h})$ where $\mathbf{h} = [h^{11}; h^{12}; \ldots; h^{mm}]$. The instantaneous rate achievable by terminal i depends on the signal to interference plus noise ratio (SINR) $c^i(\mathbf{p}(\mathbf{h}); \mathbf{h}) = h^{ii}p^i(\mathbf{h})/[1 + \sum_{j \neq i} h^{ji}p^j(\mathbf{h})]$. Again, the quantity of interest for each terminal is the long term rate which, assuming use of capacity achieving codes, is

$$x^i \leq \mathbb{E}_{\mathbf{h}}\left[\log\left(1 + \frac{h^{ii}p^i(\mathbf{h})}{1 + \sum_{j \neq i} h^{ji}p^j(\mathbf{h})}\right)\right]. \tag{4.3}$$

The constraint in (4.3) has the form of (4.1) as it relates instantaneous rates with long term rates. The problem formulation is completed with a set of average power constraints $p^i = \mathbb{E}_{\mathbf{h}}[p^i(\mathbf{h})]$. Power constraints can be enforced via the set $\mathcal{X} = \{p : 0 \leq p \leq p_0\}$ and the utility g_0 can be chosen to be the weighted sum rate $g_0(\mathbf{x}) = \sum_i w^i x^i$ or a proportional fair utility $g_0(\mathbf{x}) = \sum_i \log(x^i)$. Observe that the utility is concave but the instantaneous rate function $c^i(\mathbf{p}(\mathbf{h}); \mathbf{h})$ is not convex. A twist on this problem formulation is to make $\mathcal{P} = \{0, 1\}^m$ in which case individual terminals are either active or not for a given channel realization. Although this set $\mathcal{P}$ is not convex, it is allowed in (4.2).

Example 4 (Time division multiple access)**.** In Example 3 terminals are allowed to transmit simultaneously. Alternatively, we can request that only one terminal be active at any point

in time. This can be modeled by introducing the scheduling variable $\alpha^i(\mathbf{h}) \in \{0,1\}$ and rewriting the rate expression in (4.3) as

$$x^i \leq \mathbb{E}_{\mathbf{h}}\Big[\alpha^i(\mathbf{h}) \log\big(1 + h^i p^i(\mathbf{h})\big)\Big], \tag{4.4}$$

where the interference term does not appear because we restrict channel occupancy to a single terminal. To enforce this constraint we define the set $\mathcal{P} := \{\alpha^i(\mathbf{h}) : \alpha^i(\mathbf{h}) \in \{0,1\}, \sum_i \alpha^i(\mathbf{h}) \leq 1\}$. This is a problem formulation in which, different from Example 3, we not only allocate power but channel access as well.

4.2.1 Learning formulations

The problem in (4.2), which formally characterizes the optimal resource allocation policies for a diverse set of wireless problems, is generally a very difficult optimization problem to solve. In particular, two well known challenges in solving (4.2) directly are:

(i) The optimization variable $\mathbf{p}$ is a function.

(ii) The channel distribution $m(\mathbf{h})$ is unknown.

Challenge (ii) is of little concern as it can be addressed with stochastic optimization algorithms. Challenge (i) makes (4.2) a functional optimization problem, which, compounded with the fact that (4.1) defines a nonconvex constraint, entails large computational complexity. This is true even if we settle for a local minimum because we need to sample the n-dimensional space $\mathcal{H}$ of fading realizations $\mathbf{h}$. If each channel is discretized to d values the number of resource allocation variables to be determined is md^n. As it is germane to the ideas presented in this paper, we point that (4.2) is known to have null duality gap [103]. This, however, does not generally make the problem easy to solve and moreover requires having model information.

This brings a third challenge in solving (4.2), namely the availability of the wireless system functions:

(iii) The form of the instantaneous performance function $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$, utility $g_0(\mathbf{x})$, and constraint $\mathbf{g}(\mathbf{x})$ may not be known.

As we have seen in Examples 2-4, the function $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ models instantaneous achievable rates. Although these functions *may* be available in ideal settings, there are difficulties in measuring the radio environment that make them uncertain. This issue is often neglected but it can cause significant discrepancies between predicted and realized performances. Moreover, with less idealized channel models or performance rate functions—such as bit

error rate—reliable models may even not be available to begin with. While the functions $g_0(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ are sometimes known or designed by the user, we assume they are not here for complete generality.

Challenges (i)-(iii) can all be overcome with the use of a learning formulation. This is accomplished by introducing a parametrization of the resource allocation function so that for some $\boldsymbol{\theta} \in \mathbb{R}^q$ we make

$$\mathbf{p}(\mathbf{h}) = \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}). \tag{4.5}$$

With this parametrization the ergodic constraint in (4.1) becomes

$$\mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}\big)\big] \tag{4.6}$$

If we now define the set $\Theta := \{\boldsymbol{\theta} \mid \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in \mathcal{P}\}$, the optimization problem in (4.2) becomes one in which the optimization is over $\mathbf{x}$ and $\boldsymbol{\theta}$

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := & \max_{\boldsymbol{\theta}, \mathbf{x}} \; g_0(\mathbf{x}), \\ & \text{s.t.} \;\; \mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}\big)\big], \\ & \quad \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \;\; \mathbf{x} \in \mathcal{X}, \;\; \boldsymbol{\theta} \in \Theta. \end{aligned} \tag{4.7}$$

Since the optimization is now carried over the parameter $\boldsymbol{\theta} \in \mathbb{R}^q$ and the ergodic variable $\mathbf{x} \in \mathbb{R}^u$, the number of variables in (4.7) is $q+u$. This comes at a loss of of optimality because (4.5) restricts resource allocation functions to adhere to the parametrization $\mathbf{p}(\mathbf{h}) = \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$. E.g., if we use a linear parametrization $\mathbf{p}(\mathbf{h}) = \boldsymbol{\theta}^T \mathbf{h}$ it is unlikely that the solutions of (4.2) and (4.7) are close. In this work, we focus our attention on a widely-used class of parameterizations we define as *near-universal*, which are able to model any function in $\mathcal{P}$ to within a stated accuracy. We present this formally in the following definition.

Definition 1. *A parameterization $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ is an ϵ-universal parameterization of functions in $\mathcal{P}$ if, for some $\epsilon > 0$, there exists for any $\mathbf{p} \in \mathcal{P}$ a parameter $\boldsymbol{\theta} \in \Theta$ such that*

$$\mathbb{E} \left\| \mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \right\|_\infty \leq \epsilon. \tag{4.8}$$

A number of popular machine learning models are known to exhibit the universality property in Definition 1, such as radial basis function networks (RBFNs) [96] and reproducing kernel Hilbert spaces (RKHS) [114]. This work focuses in particular on deep neural networks (DNNs), which can be shown to exhibit a universal function approximation property [59] and are observed to work remarkably well in practical problems—see, e.g, [66, 77]. The specific details regarding the use of DNNs in the proposed learning framework of this paper

are discussed in Section 4.5.

While the reduction of the dimensionality of the optimization space is valuable, the most important advantage of (4.7) is that we can use training to bypass the need to estimate the distribution $m(\mathbf{h})$ and the functions $\mathbf{f}\big(\mathbf{p}(\mathbf{h}),\mathbf{h}\big)$. The idea is to learn over a time index k across observed channel realizations $\mathbf{h}_k$ and probe the channel with tentative resource allocations $\mathbf{p}_k(\mathbf{h}_k)=\boldsymbol{\phi}(\mathbf{h}_k,\boldsymbol{\theta}_k)$. The resulting performance $\mathbf{f}\big(\mathbf{h}_k,\boldsymbol{\phi}(\mathbf{h}_k,\boldsymbol{\theta}_k)\big)$ is then observed and utilized to learn the optimal parametrized resource allocation as defined by (4.7). The major challenge to realize this idea is that existing learning methods operate in unconstrained optimization problems. We will overcome this limitation by operating in the dual domain where the problem is unconstrained (Section 4.3). Our main result on learning for constrained optimization is to show that, its lack of convexity notwithstanding, the duality gap of (4.7) is small for near-universal parameterizations (Theorem 3). This result justifies operating in the dual domain as it does not entail a significant loss of optimality. A model-free primal-dual method to train (4.7) is then introduced in Section 4.4 and neural network parameterizations are described in Section 4.5.

4.3 Lagrangian Dual Problem

Solving the optimization problem in (4.7) requires learning both the parameter $\boldsymbol{\theta}$ and the ergodic average variables $\mathbf{x}$ over a set of both convex and non-convex constraints. This can be done by formulating and solving the Lagrangian dual problem. To do so, introduce the nonnegative multiplier dual variables $\boldsymbol{\lambda}\in\mathbb{R}^p_+$ and $\boldsymbol{\mu}\in\mathbb{R}^r_+$, respectively associated with the constraints $\mathbf{x}\leq\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big]$ and $\mathbf{g}(\mathbf{x})\leq\mathbf{0}$. The Lagrangian of (4.7) is an average of objective and constraint values weighted by their respective multipliers:

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta},\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) := g_0(\mathbf{x})+\boldsymbol{\mu}^T\mathbf{g}(\mathbf{x}) \\ +\boldsymbol{\lambda}^T\Big(\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big]-\mathbf{x}\Big).\end{aligned} \tag{4.9}$$

With the Lagrangian so defined, we introduce the dual function $D_{\boldsymbol{\phi}}(\boldsymbol{\lambda},\boldsymbol{\mu})$ as the maximum Lagrangian value attained over all $\mathbf{x}\in\mathcal{X}$ and $\boldsymbol{\theta}\in\Theta$

$$D_{\boldsymbol{\phi}}(\boldsymbol{\lambda},\boldsymbol{\mu}) := \max_{\boldsymbol{\theta}\in\Theta,\mathbf{x}\in\mathcal{X}} \mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta},\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{4.10}$$

We think of (4.10) as a penalized version of (4.7) in which the constraints are not enforced but their violation is penalized by the Lagrangian terms $\boldsymbol{\mu}^T\mathbf{g}(\mathbf{x})$ and $\boldsymbol{\lambda}^T(\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big]-\mathbf{x})$. This interpretation is important here because the problem in (4.10) is unconstrained except for the set restrictions $\boldsymbol{\theta}\in\Theta$ and $\mathbf{x}\in\mathcal{X}$. This renders (4.10) analogous to conventional learning objectives and, as such, a problem that we can solve with conventional learning

algorithms.

It is easy to verify and well-known that for any choice of $\boldsymbol{\lambda} \geq \mathbf{0}$ and $\boldsymbol{\mu} \geq \mathbf{0}$ we have $D_{\boldsymbol{\phi}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) \geq P^*_{\boldsymbol{\phi}}$. This motivates definition of the dual problem in which we search for the multipliers that make $D_{\boldsymbol{\phi}}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ as small as possible

$$D^*_{\boldsymbol{\phi}} := \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} D_{\boldsymbol{\phi}}(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{4.11}$$

The dual optimum $D^*_{\boldsymbol{\phi}}$ is the best approximation we can have of $P^*_{\boldsymbol{\phi}}$ when using (4.10) as a proxy for (4.7). It follows that the two concerns that are relevant in utilizing (4.10) as a proxy for (4.7) are: (i) evaluating the difference between $D^*_{\boldsymbol{\phi}}$ and $P^*_{\boldsymbol{\phi}}$ and (ii) designing a method for finding the optimal multipliers that attains the minimum in (4.11). We address (i) in Section 4.3.1 and (ii) in Section 4.3.2.

4.3.1 Suboptimality of the dual problem

The duality gap is the difference $D^*_{\boldsymbol{\phi}} - P^*_{\boldsymbol{\phi}}$ between the dual and primal optima. For convex optimization problems this gap is null, which implies that one can work with the Lagrangian as in (4.10) without loss of optimality. The optimization problem in (4.7), however, is not convex as it incorporates the nonconvex constraint in (4.6). We will show here that despite the presence of this nonconvex constraint the duality gap $D^*_{\boldsymbol{\phi}} - P^*_{\boldsymbol{\phi}}$ is small when using parametrizations that are near universal in the sense of Definition 1. In proving this result we need to introduce some restrictions to the problem formulation that we state as assumptions next.

AS6. *The probability distribution $m(\mathbf{h})$ is nonatomic in $\mathcal{H}$. I.e., for any set $\mathcal{E} \subseteq \mathcal{H}$ of nonzero probability there exists a nonzero probability strict subset $\mathcal{E}' \subset \mathcal{E}$ of lower probability, $0 < \mathbb{E}_{\mathbf{h}}(\mathbb{I}(\mathcal{E}')) < \mathbb{E}_{\mathbf{h}}(\mathbb{I}(\mathcal{E}))$.*

AS7. *Slater's condition hold for the unparameterized problem in* (4.2) *and for the parametrized problem in* (4.7). *In particular, there exists variables $\mathbf{x}_0$ and $\mathbf{p}_0(\mathbf{h})$ and a strictly positive scalar constant $s > 0$ such that*

$$\mathbb{E}\big[\mathbf{f}\big(\mathbf{p}_0(\mathbf{h}), \mathbf{h}\big)\big] - \mathbf{x}_0 \geq s\mathbf{1}. \tag{4.12}$$

AS8. *The objective utility function $g_0(\mathbf{x})$ is monotonically non-decreasing in each component. I.e., for any $\mathbf{x} \leq \mathbf{x}'$ it holds $g_0(\mathbf{x}) \leq g_0(\mathbf{x}')$.*

AS9. *The expected performance function $\mathbb{E}\left[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\right]$ is expectation-wise Lipschitz on $\mathbf{p}(\mathbf{h})$ for all fading realizations $\mathbf{h} \in \mathcal{H}$. Specifically, for any pair of resource allocations*

$\mathbf{p}_1(\mathbf{h}) \in \mathcal{P}$ *and* $\mathbf{p}_2(\mathbf{h}) \in \mathcal{P}$ *there is a constant* L *such that*

$$\mathbb{E}\|\mathbf{f}(\mathbf{p}_1(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\mathbf{p}_2(\mathbf{h}), \mathbf{h})\|_\infty \leq L\mathbb{E}\|\mathbf{p}_1(\mathbf{h}) - \mathbf{p}_2(\mathbf{h})\|_\infty. \tag{4.13}$$

Although Assumptions 6-9 restrict the scope of problems (4.2) and (4.7), they still allow consideration of most problems of practical importance. Assumption 7 simply states that service demands can be provisioned with some slack. We point that an inequality analogous to (4.12) holds for the other constraints in (4.2) and (4.7). However, it is only the slack s that appears in the bounds we will derive. Assumption 8 is a restriction on the utilities $g_0(\mathbf{x})$, namely that increasing performance values result in increasing utility. Assumption 9 is a continuity statement on each of the dimensions of the expectation of the constraint function $\mathbf{f}$—we point out this is weaker than general Lipschitz continuity. Referring back to the problems discussed in Examples 2-4, it is evident that they satisfy the monotonicity assumption in Assumption 8. Furthermore, the continuity assumption in Assumption 9 is immediatley satisfied by the continuous capacity function in Examples 2 and 3, and is also satisfied by the binary problem in Example 4 due to the bounded expectation of the capacity function.

Assumption 6 states that there are no points of strictly positive probability in the distributions $m(\mathbf{h})$. This requires that the fading state $\mathbf{h}$ take values in a dense set with a proper probability density – no distributions with delta functions are allowed. This is the most restrictive assumption in principle if we consider systems with a finite number of fading states. We observe that in reality fading does take on a continuum of values, though the channel estimation algorithms may quantize estimates to a finite number of fading states. We stress, however, that the learning algorithm we develop in the proceeding sections does not depend upon this property, and may be directly applied to channels with discrete states.

The duality gap of the original (unparameterized) problem in (4.2) is known to be null – see Appendix 4.8 and [103]. Given the validity of Assumptions 6 - 9 and using a parametrization that is nearly universal in the sense of Definition 1, we show that the duality/parametrization gap $|D^*_{\boldsymbol{\phi}} - P^*|$ between problems (4.2) and (4.11) is small as we formally state next.

Theorem 3. *Consider the parameterized resource allocation problem in* (4.7) *and its Lagrangian dual in* (4.11) *in which the parametrization* $\boldsymbol{\phi}$ *is* ϵ*-universal in the sense of Definition 1. If Assumptions 6–9 hold, then the dual value* $D^*_{\boldsymbol{\phi}}$ *is bounded by*

$$P^* - \|\boldsymbol{\lambda}^*\|_1 L\epsilon \;\leq\; D^*_{\boldsymbol{\phi}} \;\leq\; P^*, \tag{4.14}$$

where the multiplier norm $\|\boldsymbol{\lambda}^*\|_1$ *can be bounded as*

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}_0)}{s} < \infty, \tag{4.15}$$

in which $\mathbf{x}_0$ *is the strictly feasible point of Assumption 7.*

Proof: See Appendix 4.8. ■

Given any near-universal parameterization that achieves ϵ-accuracy with respect to all resource allocation policies in $\mathcal{P}$, Theorem 3 establishes an upper and lower bound on the dual value in (4.11) relative to the optimal primal of the original problem in (4.2). The dual value is not greater than P^* and, more importantly, not worse than a bias on the order of ϵ. These bounds justify the use of the parametrized dual function in (4.10) as a means of solving the (unparameterized) wireless resource allocation problem in (4.2). Theorem 3 shows that there exist a set of multipliers – those that attain the optimal dual value $D^*_{\boldsymbol{\phi}}$ – that yield a problem that is within $\mathcal{O}(\epsilon)$ of optimal.

It is interesting to observe that the duality gap $P^* - D^*_{\boldsymbol{\phi}} \leq \|\boldsymbol{\lambda}^*\|_1 L\epsilon$ has a very simple dependance on problem constants. The ϵ factor comes from the error of approximating arbitrary resource allocations $\mathbf{p}(\mathbf{h})$ with parametrized resource allocations $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$. The Lipschitz constant L translates this difference into a corresponding difference between the functions $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ and $\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}\big)$. The norm of the Lagrange multiplier $\|\boldsymbol{\lambda}^*\|_1$ captures the sensibility of the optimization problem with respect to perturbations, which in this case comes from the difference between $\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)$ and $\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}\big)$. This latter statement is clear from the bound in (4.15). For problems in which the constraints are easy to satisfy, we can find feasible points close the optimum so that $P^* - g_0(\mathbf{x}_0) \approx 0$ and $\mathbf{s}$ is not too small. For problems where constraints are difficult to satisfy, a small slack $\mathbf{s}$ results in a meaningful variation in $P^* - g_0(\mathbf{x}_0)$ and a large value for the ratio $[P^* - g_0(\mathbf{x}_0)]/s$. We point out that (4.15) is a classical bound in optimization theory that we include here for completeness.

4.3.2 Primal-Dual learning

In order to train the parametrization $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ on the problem (4.7) we propose a *primal-dual* optimization method. A primal-dual method performs gradient updates directly on both the primal and dual variables of the Lagrangian function in (4.9) to find a local stationary point of the KKT conditions of (4.7). In particular, consider that we successively update both the primal variables $\boldsymbol{\theta}, \mathbf{x}$ and dual variables $\boldsymbol{\lambda}, \boldsymbol{\mu}$ over an iteration index k. At each index k of the primal-dual method, we update the current primal iterates $\boldsymbol{\theta}_k, \mathbf{x}_k$ by adding the corresponding partial gradients of the Lagrangian in (4.9), i.e. $\nabla_{\boldsymbol{\theta}}\mathcal{L}, \nabla_{\mathbf{x}}\mathcal{L}$, and projecting to

the corresponding feasible set, i.e.,

$$\boldsymbol{\theta}_{k+1} = \mathsf{P}_{\Theta}\left[\boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k}\nabla_{\boldsymbol{\theta}}\mathbb{E}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}_k),\mathbf{h})\boldsymbol{\lambda}_k\right], \tag{4.16}$$

$$\mathbf{x}_{k+1} = \mathsf{P}_{\mathcal{X}}\left[\mathbf{x}_k + \gamma_{\mathbf{x},k}(\nabla g_0(\mathbf{x}) + \nabla\mathbf{g}(\mathbf{x}_k)\boldsymbol{\mu}_k - \mathbf{x}_k)\right], \tag{4.17}$$

where we introduce $\gamma_{\boldsymbol{\theta},k}, \gamma_{\mathbf{x},k} > 0$ as scalar step sizes. Likewise, we perform a gradient update on current dual iterates $\boldsymbol{\lambda}_k, \boldsymbol{\mu}_k$ in a similar manner—by *subtracting* the partial stochastic gradients $\nabla_{\boldsymbol{\lambda}}\mathcal{L}, \nabla_{\boldsymbol{\mu}}\mathcal{L}$ and projecting onto the positive orthant to obtain

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k}\left(\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}_{k+1}),\mathbf{h}) - \mathbf{x}_{k+1}\right)\right]_+, \tag{4.18}$$

$$\boldsymbol{\mu}_{k+1} = \left[\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k}\mathbf{g}(\mathbf{x}_{k+1})\right]_+, \tag{4.19}$$

with associated step sizes $\gamma_{\boldsymbol{\lambda},k}, \boldsymbol{\gamma}_{\boldsymbol{\mu},k} > 0$. The gradient primal-dual updates in (4.16)-(4.19) successively move the primal and dual variables towards maximum and minimum points of the Lagrangian function, respectively.

The above gradient-based updates provide a natural manner by which to search for the optimal point of the dual function $D_{\boldsymbol{\phi}}$. However, direct evaluation of these updates requires both the knowledge of the functions $g_0, g, \mathbf{f}$, as well as the wireless channel distribution $m(\mathbf{h})$. We cannot always assume this knowledge is available in practice. Indeed, existing models for, e.g., capacity functions, do not always capture the true physical performance in practice. The primal-dual learning method presented is thus considered here only as a baseline method upon which we can develop a completely model-free algorithm. The details of model-free learning are discussed further in the following section.

4.4 Model-Free Learning

In this section, we consider that often in practice, we do not have access to explicit knowledge of the functions g_0, $\mathbf{g}$, and $\mathbf{f}$, along with the distribution $m(\mathbf{h})$, but rather observe noisy estimates of their values at given operating points. While this renders the direct implementation of the standard primal-dual updates in (4.16)-(4.19) impossible, given their reliance on gradients that cannot be evaluated, we can use these updates to develop a *model-free* approximation. Consider that given any set of iterates and channel realization $\{\tilde{\boldsymbol{\theta}}, \tilde{\mathbf{x}}, \tilde{\mathbf{h}}\}$, we can observe stochastic function values $\hat{g}_0(\tilde{\mathbf{x}})$, $\hat{\mathbf{g}}(\tilde{\mathbf{x}})$, and $\hat{\mathbf{f}}(\tilde{\mathbf{h}}, \boldsymbol{\phi}(\tilde{\mathbf{h}}, \tilde{\boldsymbol{\theta}}))$. For example, we may pass test signals through the channel at a given power or bandwidth to measure its capacity or packet error rate. These observations are, generally, unbiased estimates of the true function values.

We can then replace the updates in (4.16)-(4.19) with so-called zeroth-ordered updates, in which we construct estimates of the function gradients using observed function values. Zeroth-

ordered gradient estimation can be done naturally with the method of finite differences, in which unbiased gradient estimators at a given point are constructed through random perturbations. Consider that we draw random perturbations $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2 \in \mathbb{R}^u$ and $\hat{\boldsymbol{\theta}} \in \mathbb{R}^q$ from a standard Gaussian distribution and a random channel state $\hat{\mathbf{h}}$ from $m(\mathbf{h})$. Finite-difference gradients estimates $\widehat{\nabla} g_0$, $\widehat{\nabla} \mathbf{g}$, and $\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}\mathbf{f}$ can be constructed using function observations at given points $\{\mathbf{x}_0, \boldsymbol{\theta}_0\}$ and the sampled perturbations as

$$\widehat{\nabla} g_0(\mathbf{x}_0) := \frac{\hat{g}_0(\mathbf{x}_0 + \alpha_1 \hat{\mathbf{x}}_1) - \hat{g}_0(\mathbf{x}_0)}{\alpha_1} \hat{\mathbf{x}}_1, \tag{4.20}$$

$$\widehat{\nabla} \mathbf{g}(\mathbf{x}_0) := \frac{\hat{\mathbf{g}}(\mathbf{x}_0 + \alpha_3 \hat{\mathbf{x}}_2) - \hat{\mathbf{g}}(\mathbf{x}_0)}{\alpha_3} \hat{\mathbf{x}}_2^T, \tag{4.21}$$

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_0), \mathbf{h})] := \frac{\hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_0 + \alpha_2 \hat{\boldsymbol{\theta}}), \hat{\mathbf{h}}) - \hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_0), \hat{\mathbf{h}})}{\alpha_2} \hat{\boldsymbol{\theta}}^T, \tag{4.22}$$

where we define scalar step sizes $\alpha_1, \alpha_2, \alpha_3 > 0$. The expressions in (4.20)-(4.22) provide estimates of the gradients that can be computed using only two function evaluations. Indeed, the finite difference estimators can be shown to be unbiased, meaning that that they coincide with the true gradients in expectation—see, e.g., [90]. Note also in (4.22) that, by sampling both the function $\mathbf{f}$ and a channel state $\hat{\mathbf{h}}$, we directly estimate the expectation $\mathbb{E}_{\mathbf{h}}\mathbf{f}$. We point out that these estimates can be further improved by using batches of B samples, $\{\hat{\mathbf{x}}_1^{(b)}, \hat{\mathbf{x}}_2^{(b)}, \hat{\boldsymbol{\theta}}^{(b)}, \hat{\mathbf{h}}^{(b)}\}_{b=1}^B$, and averaging over the batch. We focus on the simple stochastic estimates in (4.20)-(4.22), however, for clarity of presentation.

Note that, while using the finite difference method to estimate the gradients of the deterministic function $g_0(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ is relatively simple, estimating the stochastic policy function $\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$ is often a computational burden in practice when the parameter dimension q is very large—indeed, this is often the case in, e.g., deep neural network models. An additional complication arises in that the function must be observed multiple times for the same sample channel state $\hat{\mathbf{h}}$ to obtain the perturbed value. This might be impossible to do in practice if the channel state changes rapidly. There indeed exists, however, an alternative model free approach for estimating the gradient of a policy function, which we discuss in the next subsection.

4.4.1 Policy gradient estimation

The ubiquity of computing the gradients of policy functions such as $\nabla_{\boldsymbol{\theta}} \mathbb{E}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$ in machine learning problems has motivated the development of a more practical estimation method. The so-called *policy gradient* method exploits a likelihood ratio property found in such functions to allow for an alternative zeroth ordered gradient estimate. To derive

the details of the policy gradient method, consider that a deterministic policy $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ can be reinterpreted as a *stochastic* policy drawn from a distribution with density function $\pi(\mathbf{p})$ defined with a delta function, i.e., $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p}) = \delta(\mathbf{p} - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}))$. It can be shown that the Jacobian of the policy constraint function $\mathbb{E}_{\mathbf{h},\boldsymbol{\phi}}[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})]$ with respect to $\boldsymbol{\theta}$ can be rewritten using this density function as

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \mathbb{E}_{\mathbf{h},\mathbf{p}}[\mathbf{f}(\mathbf{p}, \mathbf{h}) \nabla_{\boldsymbol{\theta}} \log \pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})^T], \tag{4.23}$$

where $\mathbf{p}$ is a random variable drawn from distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$—see, e.g., [118]. Observe in (4.23) that the computation of the Jacobian reduces to a function evaluation multiplied by the gradient of the policy distribution $\nabla_{\boldsymbol{\theta}} \log \pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$. Indeed, in the deterministic case where the distribution is a delta function, the gradient cannot be evaluated without knowledge of $m(\mathbf{h})$ and $\mathbf{f}$. However, we may approximate the delta function with a known density function centered around $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$, e.g., Gaussian distribution. If an analytic form for $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$ is known, we can estimate $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$ by instead directly estimating the left-hand side of (4.23). In the context of reinforcement learning, this is called the REINFORCE method [118]. By using the previous function observations, we can obtain the following policy gradient estimate,

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \hat{\mathbf{f}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{h}}) \nabla_{\boldsymbol{\theta}} \log \pi_{\hat{\mathbf{h}},\boldsymbol{\theta}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}})^T, \tag{4.24}$$

where $\hat{\mathbf{p}}_{\boldsymbol{\theta}}$ is a sample drawn from the distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$.

The policy gradient estimator in (4.24) can be taken as an alternative to the finite difference approach in (4.22) for estimating the gradient of the policy constraint function, provided the gradient of the density function $\boldsymbol{\pi}$ can itself be evaluated. Observe in the above expression that the policy gradient approach replaces a sampling of the parameter $\boldsymbol{\theta} \in \mathbb{R}^q$ with a sampling of a resource allocation $\mathbf{p} \in \mathbb{R}^m$. This is indeed preferable for many sophisticated learning models in which $q \gg m$. We stress that while policy gradient methods are preferable in terms of sampling complexity, they come at the cost of placing an additional approximation through the use of a stochastic policy analytical density functions $\boldsymbol{\pi}$.

4.4.2 Model-free primal-dual method

Using the gradient estimates in (4.20)-(4.22)—or (4.24)—we can derive a model-free, or zeroth-ordered, stochastic updates to replace those in (4.16)-(4.19). By replacing all function evaluations with the function observations and all gradient evaluations with the finite

difference estimates, we can perform the following stochastic updates

$$\boldsymbol{\theta}_{k+1} = \mathsf{P}_{\Theta}\left[\boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k}\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}_k),\mathbf{h})\boldsymbol{\lambda}_k\right], \tag{4.25}$$

$$\mathbf{x}_{k+1} = \mathsf{P}_{\mathcal{X}}\left[\mathbf{x}_k + \gamma_{\mathbf{x},k}(\widehat{\nabla g_0}(\mathbf{x}) + \widehat{\nabla \mathbf{g}}(\mathbf{x}_k)\boldsymbol{\mu}_k - \mathbf{x}_k)\right], \tag{4.26}$$

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k}\left(\hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}_k,\boldsymbol{\theta}_{k+1}),\hat{\mathbf{h}}_k) - \mathbf{x}_{k+1}\right)\right]_+ \tag{4.27}$$

$$\boldsymbol{\mu}_{k+1} = \left[\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k}\hat{\mathbf{g}}(\mathbf{x}_{k+1})\right]_+. \tag{4.28}$$

The expressions in (4.25)-(4.28) provides means of updating both the primal and dual variables in a primal-dual manner without requiring any explicit knowledge of the functions or channel distribution through observing function realizations at the current iterates. We may say this method is *model-free* because all gradients used in the updates are constructed entirely from *measurements*, rather than analytic computation done via model knowledge. The complete model-free primal-dual learning method can be summarized in Algorithm 3. The method is initialized in Step 1 through the selection of parameterization model $\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta})$ and form of the stochastic policy distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}$ and in Step 2 through the initialization of the primal and dual variables. For every step k, the algorithm begins in Step 4 by drawing random samples (or batches) of the primal and dual variables. In Step 5, the model functions are sampled at both the current primal and dual iterates and at the sampled points. These function observations are then used in Step 6 to form gradient estimates via finite difference (or policy gradient). Finally, in Step 7 the model-free gradient estimates are used to update both the primal and dual iterates.

We briefly comment on the known convergence properties of the model-free learning method in (4.25)-(4.28). Due to the non-convexity of the Lagrangian defined in (4.9), the stochastic primal-dual descent method will converge only to a local optima and is not guaranteed to converge to a point that achieves $D^*_{\boldsymbol{\theta}}$. These are indeed the same convergence properties of general *unconstrained* non-convex learning problems as well. We instead demonstrate through numerical simulations the performance of the proposed learning method in practical wireless resource allocation problems in the proceeding section.

Remark 11. The algorithm presented in Algorithm 3 is generic in nature and can be supplemented with more sophisticated learning techniques that can improve the learning process. Some examples include the use of entropy regularization to improve policy optimization in non-convex problems [52]. Policy optimization can also be improved using actor-critic methods [117], while the use of a model function estimate to obtain "supervised" training signals can be used to initialize the parameterization vector $\boldsymbol{\theta}$. The use of such techniques in optimal wireless design are not explored in detail here and left as the study of future work.

Algorithm 3 Model-Free Primal-Dual Learning

Parameters: Policy model $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ and distribution form $\pi_{\mathbf{h},\boldsymbol{\theta}}$

Input: Initial states $\boldsymbol{\theta}_0, \mathbf{x}_0, \boldsymbol{\lambda}_0 \boldsymbol{\mu}_0$

for [**do**main loop]$k = 0, 1, 2, \ldots$

Draw samples $\{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}_k\}$, or in batches of size B

Obtain random observation of function values $\hat{g}_0, \hat{\mathbf{f}}\ \hat{\mathbf{g}}$ at current and sampled iterates

Compute gradient estimates $\widehat{\nabla} g_0(\mathbf{x})$, $\widehat{\nabla} \mathbf{g}(\mathbf{x})$, $\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h},\boldsymbol{\phi}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$, [cf. (4.20)-(4.22) or (4.24)]

Update primal and dual variables [cf. (4.25)-(4.28)]

$$\boldsymbol{\theta}_{k+1} = \mathsf{P}_{\Theta} \left[\boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k} \widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\hat{\mathbf{h}}_k, \boldsymbol{\theta}_k), \hat{\mathbf{h}}_k) \boldsymbol{\lambda}_k \right],$$

$$\mathbf{x}_{k+1} = \mathsf{P}_{\mathcal{X}} \left[\mathbf{x}_k + \gamma_{\mathbf{x},k} (\widehat{\nabla g_0}(\mathbf{x}) + \widehat{\nabla} \mathbf{g}(\mathbf{x}_k) \boldsymbol{\mu}_k - \mathbf{x}_k) \right],$$

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k} \left(\hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}_k, \boldsymbol{\theta}_{k+1}), \hat{\mathbf{h}}_k) - \mathbf{x}_{k+1} \right) \right]_+$$

$$\boldsymbol{\mu}_{k+1} = \left[\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k} \hat{\mathbf{g}}(\mathbf{x}_{k+1}) \right]_+ .$$

end for

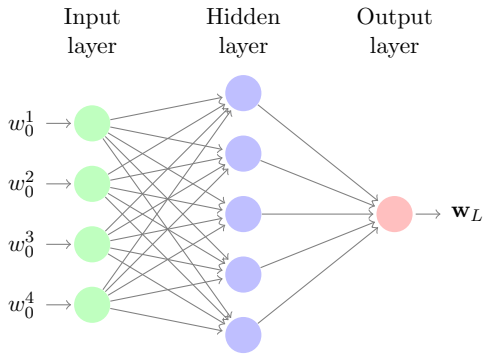


Figure 4.1: Typical architecture of fully-connected deep neural network.

4.5 Deep Neural Networks

We have so far discussed a theoretical and algorithm means of learning in wireless systems by employing any near universal parametrization as defined in Definition 1. In this section, we restrict our attention to the increasingly popular set of parameterizations known as *deep neural networks* (DNNs), which are often observed in practice to exhibit strong performance in function approximation. In particular, we discuss the details of the DNN parametrization model and both the theoretical and practical implications within our constrained learning framework.

The exact form of a particular DNN is described by what is commonly referred to as its *architecture*. The architecture consists of a prescribed number of layers, each of which consisting of a linear operation followed by a point-wise nonlinearity—also known

as an activation function. In particular, consider a DNN with L layers, labelled $l = 1, \ldots, L$ and each with a corresponding dimension q_l. The layer l is defined by the linear operation $\mathbf{W}_l \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\boldsymbol{\sigma}_l : \mathbb{R}^{q_l} \to \mathbb{R}^{q_l}$. If layer l receives as an input from the $l-1$ layer $\mathbf{w}_{l-1} \in \mathbb{R}^{q_{l-1}}$, the resulting output $\mathbf{w}_l \in \mathbb{R}^{q_l}$ is then computed as $\mathbf{w}_l := \boldsymbol{\sigma}_l(\mathbf{W}_l \mathbf{w}_{l-1})$. The final output of the DNN, $\mathbf{w}_L$, is then related to the input $\mathbf{w}_0$ by propagating through each later of the DNN as $\mathbf{w}_L = \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{w}_0))))))$.

An illustration of a fully-connected example DNN architecture is given in Figure 4.1. In this example, the inputs $\mathbf{w}$ are passed through a single hidden layer, following which is an output layer. The grey lines between layers reflect the linear transformation $\mathbf{W}_l$, while each node contains an additional element-wise activation function $\boldsymbol{\sigma}_l$. This general DNN structure has been observed to have remarkable generalization and approximation properties in a variety of functional parameterization problems.

The goal in learning DNNs in general then reduces to learning the linear weight functions $\mathbf{W}_1, \ldots, \mathbf{W}_L$. Common choices of activation functions $\boldsymbol{\sigma}_l$ include a sigmoid function, a rectifier function (commonly referred to as ReLu), as well as a smooth approximation to the rectifier known as softplus. For the parameterized resource allocation problem in (4.7), the policy $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ can be defined by an L-layer DNN as

$$\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) := \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{h}))))))), \tag{4.29}$$

where $\boldsymbol{\theta} \in \mathbb{R}^q$ contains the entries of $\{\mathbf{W}_l\}_{l=1}^L$ with $q = \sum_{l=1}^{L-1} q_l q_{l+1}$. Note that $q_1 = n$ by construction.

To contextualize the primal-dual algorithm in (4.16)-(4.19) with respect to traditional neural network training, observe that the update in (4.16) requires computation of the gradient $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\mathbf{h}, \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}))$. Using the chain rule, this can be expanded as

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h}) = \\ \nabla_{\boldsymbol{\phi}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h}) \nabla_{\boldsymbol{\theta}} \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_k). \end{aligned} \tag{4.30}$$

Thus, the computation of the full gradient requires evaluating the gradient of the policy function $\mathbf{f}$ as well as the gradient of the DNN model $\boldsymbol{\phi}$. For the DNN structure in (4.29), the evaluation of $\nabla_{\boldsymbol{\theta}} \boldsymbol{\phi}$ may itself also require a chain rule expansion to compute partial derivatives at each layer of the network. This process of performing gradient descent to find the optimal weights in the DNN is commonly referred to as backpropogation.

We further take note how our learning approach differs from a more traditional, *supervised* training of DNNs. As in (4.30), the backpropogation is performed with respect to the given policy constraint function $\mathbf{f}$, rather than with respect to a Euclidean loss function over a set

of given training data. Furthermore, due to the constraints, the backpropogation step in (4.16) is performed in sequence with the more standard primal and dual variable updates in (4.17)-(4.19). In this way, the DNN is trained indirectly *within* the broader optimization algorithm used to solve (4.7). This is in contrast with other approaches of training DNNs in constrained wireless resource allocation problems—see, e.g. [69, 70, 115]—which train a DNN to approximate the complete constrained maximization function in (4.2) directly. Doing so requires the ability to solve (4.2) either exactly or approximately enough times to acquire a labeled training set. The primal-dual learning approach taken here is preferable in that it does not require the use of training data. The dual problem can be seen as a simplified reinforcement learning problem—one in which the actions do not affect the next state.

For DNNs to be valid parametrization with respect to the result in Theorem 3, we must first verify that they satisfy the near-universality property in Definition 1. Indeed, deep neural networks are popular parameterizations for arbitrary functions precisely due to the richness inherent in (4.29), which in general grows richer with number of layers L and associated layer sizes q_l. This richness property of DNNs has been the subject of mathematical study and formally referred to as a complete universal function approximation [24, 59]. In words, this property implies that a large class of functions $\mathbf{p}(\mathbf{h})$ can be approximated with *arbitrarily* small accuracy ϵ using a DNN parameterization of the form in (4.29) with only a single layer of arbitrarily large size. With this property in mind, we can present the following theorem that extends the result in Theorem 3 in the special case of DNNs.

Theorem 4. *Consider the DNN parametrization $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ in (4.29) with non-constant, continuous activation functions $\boldsymbol{\sigma}_l$ for $l = 1, \ldots, L$. Define the vector of layer lengths $\mathbf{q} = [q_1; q_2; \ldots; q_L]$ and a DNN defined in (4.29) with lengths $\mathbf{q}$ as $\boldsymbol{\phi}_{\mathbf{q}}(\mathbf{h}, \boldsymbol{\theta})$. Now consider the set of possible L-layer DNN parameterization functions $\Phi := \{\boldsymbol{\phi}_{\mathbf{q}}(\mathbf{h}, \boldsymbol{\theta}) \mid \mathbf{q} \in \mathbb{N}^L\}$. If Assumptions 6–9 hold, then the optimal dual value of the parameterized problem satisfies*

$$\inf_{\boldsymbol{\phi} \in \Phi} \mathbf{D}^*_{\boldsymbol{\phi}} = P^*. \tag{4.31}$$

Proof: See Appendix 4.9. ■

With Theorem 4 we establish the null duality gap property of a resource allocation problem of the form in (4.7) given a DNN parameterization that achieves arbitrarily small function approximation accuracy as the dimension of the DNN parameter—i.e. the number of hidden nodes—grows to infinity. While such a parametrization is indeed guaranteed to exist through the universal function approximation theorem, one would require a DNN with arbitrarily large size to obtain such a network in practice. As such, the suboptimality bounds presented in Theorem 3, which require only an DNN-approximation of given accuracy ϵ provide the more practical characterization of (4.7), while the result in Theorem 4 suggests

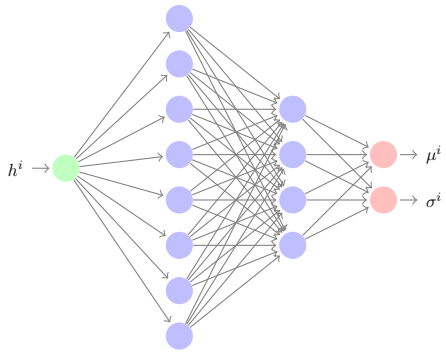


Figure 4.2: Neural network architecture used for simple AWGN channel. Each channel state h^i is fed into an independent SISO network with two hidden layers of size 8 and 4, respectively. The DNN outputs a mean μ^i and standard deviation σ^i for a truncated Gaussian distribution.

DNNs can be used find parameterizations of arbitrarily strong accuracy.

4.6 Simulation Results

In this section, we provide simulation results on using the proposed primal-dual learning method to solve for DNN-parameterizations of resource allocation in a number of common problems in wireless communications that take the form in (4.2). For the simulations performed, we employ a stochastic policy and implement the REINFORCE-style policy gradient described in Section 4.4.1. In particular, we select the policy distribution $\pi_{\boldsymbol{\theta},\mathbf{h}}$ as a truncated Gaussian distribution. The truncated Gaussian distribution has fixed support on the domain $[0, p_{\max}]$. The output layer of the DNN $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in \mathbb{R}^{2m}$ is the set of m means and standard deviations to specify the respective truncated Gaussian distributions, i.e. $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) := [\mu^1; \sigma^1; \mu^2; \sigma^2; \ldots; \mu^m; \sigma^m]$. Furthermore, to represent policies that are bounded on the support interval, the output of the last layer is fed into a scaled sigmoid function such that the mean lies in the area of support and the variance is no more than the square root of the support region. In the following experiments, this interval is [0, 10].

For updating the primal and dual variables, we use a batch size of 32. The primal dual method is performed with an exponentially decaying step size for dual updates and the ADAM optimizer [65] for the DNN parameter update. Both updates start with a learning rate of 0.0005, while random channel conditions are generated with an exponential distribution with parameter $\lambda = 2$ (to represent the square of a unit variance Rayleigh fading channel state).

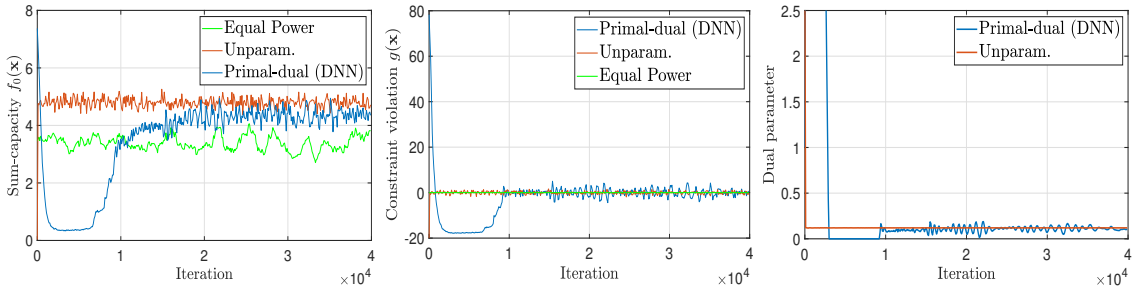


Figure 4.3: Convergence of (left) objective function value, (center) constraint value, and (right) dual parameter for simple capacity problem in (4.32) using proposed DNN method with policy gradients, the exact unparameterized solution, and an equal power allocation amongst users. The DNN parameterization obtains near-optimal performance relative to the exact solution and outperforms the equal power allocation heuristic.

4.6.1 Simple AWGN channel

To begin, we simulate the learning of a DNN to solve the problem of maximizing total capacity over a set of simple AWGN wireless fading channel. In this case, each user is given a dedicated channel to communicate, and we wish to allocate resources between users within a total expected power budget $p_{\max}$. In this case, the capacity over the channel can be modeled as $\log(1+\text{SNR}^i)$, where $\text{SNR}^i := h^i p^i(h^i)/v^i$ is the signal-to-noise ratio experienced by user i and $v^i > 0$ is the noise variance. The capacity function for the ith user is thus given by $f^i(p^i(h^i), h^i) := \log(1 + h^i p^i(h^i)/v^i)$. We are interested in maximizing the weighted aggregate throughput across all users, with user i weighted by $w^i \geq 0$. The total capacity problem can be written as

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta},\mathbf{x}} \sum_{i=1}^{m} w^i x^i \qquad & (4.32) \\ \text{s.t. } x^i \leq \mathbb{E}_{h^i}\left[\log(1 + h^i \phi^i(h^i, \boldsymbol{\theta})/v^i)\right], \ \forall i \\ \mathbb{E}_{\mathbf{h}}\left[\sum_{i=1}^{m} \phi^i(h^i, \boldsymbol{\theta})\right] \leq p_{\max}. \end{aligned}$$

Note that, despite the non-convex structure of the problem in (4.32), the loose coupling over the resource allocation variables allows for this problem to be solved exactly without any DNN parametrization using a simple dual stochastic gradient (SGD) method—see, e.g., [125]. Nonetheless, this is an instructive example with which to validate our approach by seeing if the DNN is capable of learning resource allocation policies that closely match the exact optimal solutions found without any parametrization. Furthermore, the model-free learning capabilities of the DNN parametrization make the proposed learning method applicable in cases in which the, e.g., capacity function is not known.

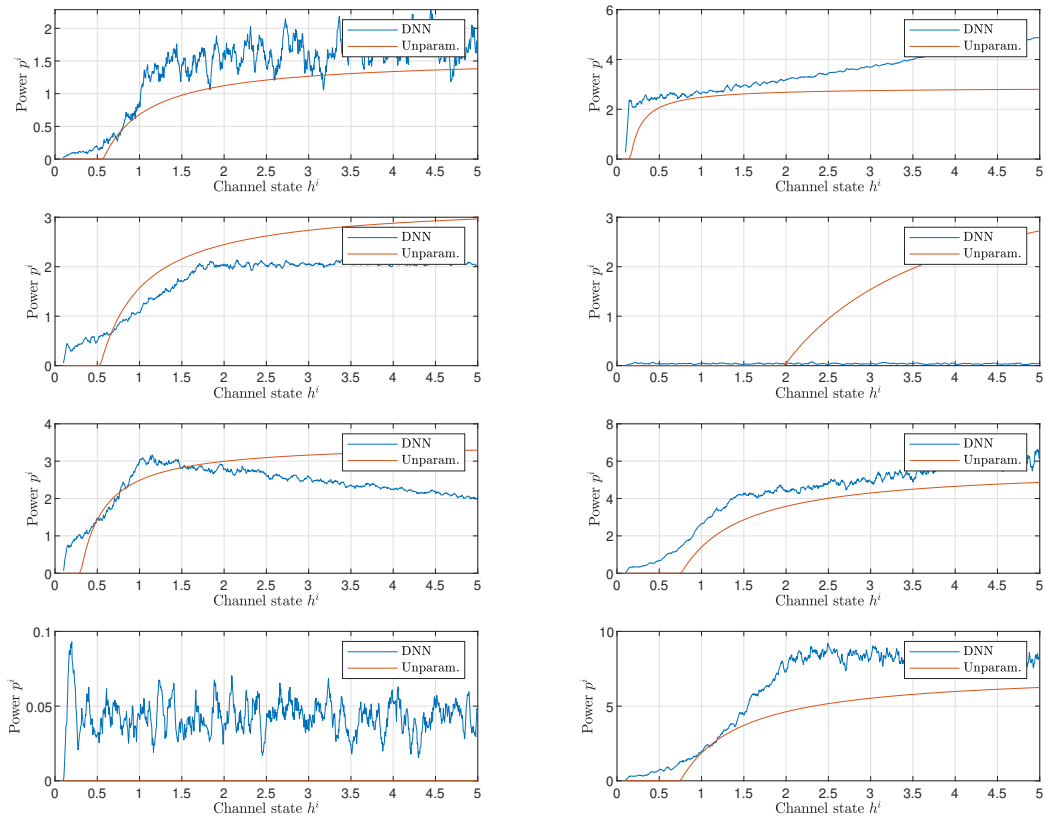


Figure 4.4: Example of 8 representative resource allocation policy functions found through DNN parameterization and unparameterized solution. Although the policies differ from the analytic solution, many contain similar shapes. Overall, the DNN method learns variations on the optimal policies that nonetheless achieve similar performance.

To have the the outputs of the DNN match the same form as the analytic solution (4.32), we construct m independent, uncoupled DNNs for each user. Each channel gain h^i is provided as input to a single-input-single-output (SISO) DNN, which outputs a power allocation $p^i(h^i)$. In particular, each DNN is constructed with two hidden layers, of size 8 and 4, respectively. In addition, each layer is given a ReLU activation function, i.e. $\gtrapprox(\mathbf{z}) = [\mathbf{z}]_+$; see Figure 4.2 for the architecture.

The results of a simple experiment with $m = 20$ users with random weights w^i and variances v^i is shown in Figure 4.3. We further set the maximum power as $p_{\max} = 20$. In this plot we compare the performance of the DNN primal dual learning method with the exact, unparameterized solution and an equal power allocation policy. In the equal power allocation policy, we allocate a power of $p^i = p_{\max}/m$ for all users. Here we see in the left figure that the the total capacity achieved by the DNN primal-dual method converges to roughly the same value as the exact solution found by SGD. Likewise, in the center figure, we plot the value of the constraint function. Here, we see that primal-dual converges to 0, thus implying feasibility of the learned policy. Finally, in the left figure we see that the dual variable obtained by the DNN matches that of the unparameterized.

Remark 12. Observe that the learning process may take many iterations to converge than the unparameterized solution due to the many parameters that need to be learned and the model-free nature of the learning process. It is generally the case that the training is done offline before implementation, in which the case the learning rate does not play a significant factor. In the case in which the weights w^i and channel noise power v^i may change over time, we may use the existing as a "warm-start" to quickly adapt to the changes in the model. For problem parameters that are changing fast, there have been higher order optimization methods that have been proposed to adapt to changing conditions of the problem [31].

In Figure 4.4 we show the actual learned policies from both methods for 8 example users. Here, comparing the optimal unparameterized policies to those learned with DNNs, we see in some cases the policies learned with the DNN match the shape and function, while others differ. For instance, the fourth user shown in Fig 4.4 is not assigned any resources by the DNN-based policy, while the seventh user is likewise not given any resources by the unparameterized policy. In any case, the overall performance achieved matches that of the the exact solution. We further note that this phenomenon of certain users not being given any resources in both policies occurs because our only goal in (4.32) is to maximize the sum-capacity, which does not necessitate that every user gets to transmit. To impose this condition, we may add a constraint to (4.32) that specifies a maximum average capacity for all users to achieve.

For a more thorough comparison of the DNN approach to the exact solution to (4.32), we perform multiple experiments for varying number of users and with different DNN layer

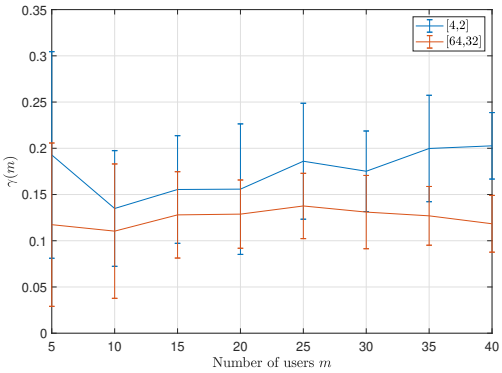


Figure 4.5: Optimality gap between optimal objective value and learned policy for the simple capacity problem in (4.32) for different number of users m and DNN architectures. The results are obtained across 10 randomly initialized simulations. The mean is plotted as the solid lines, while the one standard deviation above and below the mean is show with error bars.

sizes. In Figure 4.5, we plot the normalized optimality gap between the precise solution P^* and the parameterized solution $\hat{P}^*_{\boldsymbol{\phi}}$ found after convergence of the primal-dual method. If we define $P^*(m)$ and $\hat{P}^*_{\boldsymbol{\phi}}(m)$ to be the sum capacities achieved by the optimal policy and DNN-based policy found after 40,000 learning iterations, respectively, with m users, the normalized optimality gap can be computed as

$$\gamma(m) := \left| \frac{\hat{P}^*_{\boldsymbol{\phi}}(m) - P^*(m)}{P^*(m)} \right|. \tag{4.33}$$

The blue line shows the results for small DNNs with layer sizes 4 and 2, while the red line shows results for networks with hidden layers of size 32 and 16. Observe that, as the number of channels grows, the DNNs of fixed size achieve the same optimality. Further note that, while the blue line shows that even small DNNs can find near-optimal policies for even large networks, increasing the DNN size increases the expressive power of the DNN, thereby improving upon the suboptimality that can be obtained.

4.6.2 Interference channel

We provide further experiments on the use of neural networks in maximizing capacity over the more complex problem of allocating power over an (IC) interference channel. We first consider the problem of m transmitters communicating with a common receiver, or base station. Given the fading channels $\mathbf{h} := [h^1; \ldots; h^m]$, the capacity is determined using the signal-to-noise-plus-interference ratio (SNIR), which for transmission i is given as $\text{SNIR}^i := h^i p^i(\mathbf{h}))/(v^i + \sum_{j \neq i} h^j p^j(\mathbf{h}))$. The resulting capacity function observed by the receiver from user i is then given by $f^i(p^i(\mathbf{h}), \mathbf{h}) := \log(1 + h^{ii} p^i(\mathbf{h})/(v^i + \sum_{j \neq i} h^{ji} p^j(\mathbf{h})))$

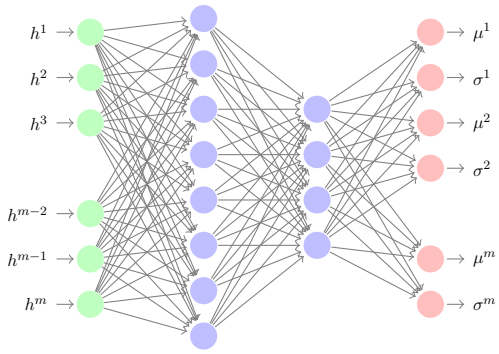


Figure 4.6: Neural network architecture used for interference channel problem in (4.34). All channel states $\mathbf{h}$ are fed into a MIMO network with two hidden layers of size 32 and 16, respectively (each circle in hidden layers represents 4 neurons). The DNN outputs means μ^i and standard deviations σ^i for i truncated Gaussian distributions.

and the DNN-parameterized problem is written as

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta},\mathbf{x}} & \sum_{i=1}^{m} w^i x^i \\ \text{s.t. } & x^i \leq \mathbb{E}_{\mathbf{h}}\left[\log\left(1+\frac{h^i\phi^i(\mathbf{h},\boldsymbol{\theta})}{v^i+\sum_{j\neq i}h^i\phi^j(\mathbf{h},\boldsymbol{\theta})}\right)\right],\ \forall i \\ & \mathbb{E}_{\mathbf{h}}\left[\sum_{i=1}^{m}\phi^i(\mathbf{h},\boldsymbol{\theta})\right] \leq p_{\max}. \end{aligned} \tag{4.34}$$

Here, the coupling of the resource policies in the capacity constraint make the problem in (4.34) very challenging to solve. Existing dual method approaches are ineffective here because the non-convex capacity function cannot be minimized exactly. This makes the primal-dual approach with the DNN parametrization a feasible alternative. However, this means that we cannot provide comparison to the analytic solution, but instead compare against the performance of some standard, model-free heuristic approaches.

As the power allocation of user i will depend on the channel conditions of all users, due to the coupling in the interference channel, rather than the m SISO networks used in the previous example, we construct a single multiple-input-multiple-output (MIMO) DNN architecture, shown in Figure 4.6, with a two layers of size 32 and 16 hidden nodes. In this architecture, all channel conditions $\mathbf{h}$ are fed as inputs to the DNN, which outputs the truncated Gaussian distribution parameters for every user's policy. In Figure 4.7 we plot the convergence of the objective value and constraint value learned using the DNN parameterization and those obtained by three model-free heuristics for a system with $m = 20$ users. These include (i) an equal division of power $p_{\max} = 20$ across all m users, (ii) randomly selecting 4 users to transmit with power $p = 5$, and (iii) randomly selecting 3 users to

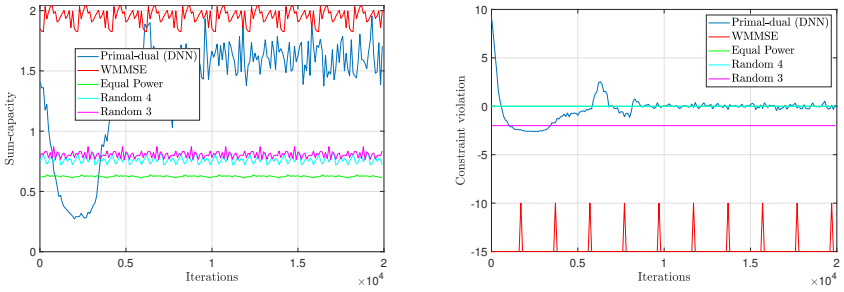


Figure 4.7: Convergence of (left) objective function value and (right) constraint value for interference capacity problem in (4.34) using proposed DNN method, WMMSE, and simple model free heuristic power allocation strategies $m = 20$ users. The DNN-based primal dual method learns a policy that achieves close performance to WMMSE, better performance than the other model free heuristics, and moreover converges to a feasible solution.

transmit with power $p = 6$. While not a model free method, we also compare performance against the well-known heuristic method WMMSE [111]. Here, we observe that, as in the previous example, all values converge to stationary points, suggesting that the method converges to a local optimum. We can also confirm in the right plot of the constraint value that the learned policy is indeed feasible. It can be observed that the performance of the DNN-based policy leaned with the primal dual method is superior to that of the other model free heuristic methods, while obtaining close performance to that of WMMSE, which we stress does indeed require model information to implement.

To study another key aspect of the parameters of the DNN—namely the output distribution $\pi_{\boldsymbol{\theta},\mathbf{h}}$—we make a comparison against the performance achieved using two natural choices for distribution in the power allocation problem in (4.34). The primary motivation behind using a truncated Gaussian is that the parameters, namely mean and variance, are easy to interpret and learn. The Gamma distribution, alternatively, has parameters that are less interpretable in this scenario and the outputs may vary as the parameters change. In Figure 4.8 we demonstrate the comparison of performance between using a Gamma distribution and the truncated Gaussian distribution. Here, we observe that the performance of the method does indeed rely on proper choice of output distribution, as it can be seen that the truncated Gaussian distribution induces stronger performance relative to a Gamma distribution.

Our last series of experiment concerns the classical problem in interference management in which there are m transmitter/receiver pairs sending information to each other. The allocation policy for each user is given as a binary decision $\alpha^i \in \{0, 1\}$ of whether or not to transmit with power p_0—a variation of this problem is described further detail in Example 3. In this case, the SNIR for transmission i can be given as $\text{SNIR}^i := h^{ii} p_0 \alpha^i(\mathbf{h}))/(v^i +$

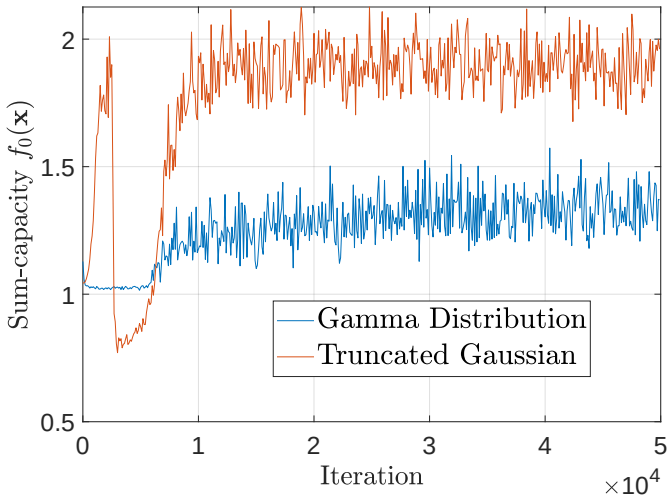


Figure 4.8: Comparison of performance using Gamma and truncated Gaussian distributions in output layer of a DNN.

$p_0 \sum_{j \neq i} h^{ji} \alpha^j(\mathbf{h})$). The DNN-parameterized problem is written as

$$
\begin{aligned}
P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta}, \mathbf{x}} \; & \sum_{i=1}^{m} w^i x^i \\
\text{s.t.} \;\; & x^i \leq \mathbb{E}_{\mathbf{h}} \left[\log \left(1 + \frac{h^{ii} p_0 \phi^i(\mathbf{h}, \boldsymbol{\theta})}{v^i + p_0 \sum_{j \neq i} h^{ji} \phi^j(\mathbf{h}, \boldsymbol{\theta})} \right) \right], \;\; \forall i \\
& \mathbb{E}_{\mathbf{h}} \left[\sum_{i=1}^{m} \phi^i(\mathbf{h}, \boldsymbol{\theta}) \right] \leq p_{\max}, \quad \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in \{0,1\}^m.
\end{aligned}
\tag{4.35}
$$

As the case in (4.34), this problem cannot be solved exactly. We instead compare the performance against that of the random selection heuristic considered in previous examples.

We plot in Figure 4.9 the performance achieved during the learning process for the DNN against the performance of WMMSE and heuristic that randomly selects 2 users to transmit. These simulations are performed on a system of size $m = 5$ with a maximum power of $p_{\max} = 20$ and unit weights and variances $w^i = v^i = 1$. The DNN has the same fully-connected architecture as used in the previous example. However, given the binary nature of the allocation policies, we employ a Bernoulli distribution as the output policy distribution. In Figure 4.9, we observe that using a DNN learning model, we in fact learn a policy that is close to matching the performance that can be obtained using the WMMSE heuristic and outperforms other model-free heuristics. We further note in the right figure that the learned policy is indeed feasible. This demonstrates the ability of the generic primal-dual learning method to either match or exceed the performance given by heuristic

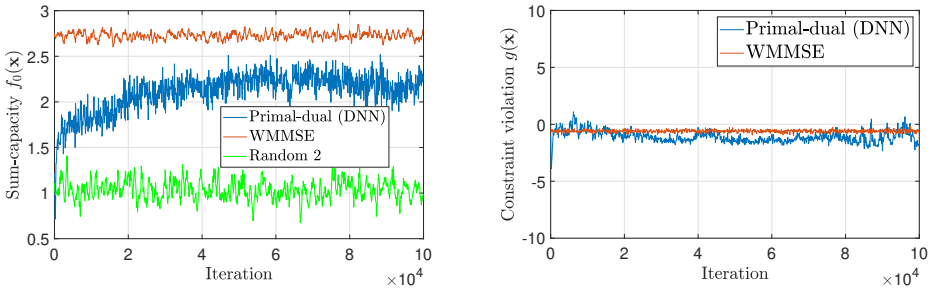


Figure 4.9: Convergence of (left) objective function value and (right) constraint value for interference capacity problem in (4.35) using proposed DNN method, heuristic WMMSE method, and the equal power allocation heuristic for $m = 5$ users. The DNN-based primal dual method learns a policy that is feasible and almost matches the WMMSE method in terms of achieved sum-capacity, without having access to capacity model.

methods that are specifically designed to solve certain problems when applied to problems that do not have a known exact solution. We also stress that the proposed learning method learned such a policy using the model-free learning, thereby not having access to the model for the capacity function, which is necessary in the WMMSE method.

4.7 Conclusion

In this paper, we studied a generic formulation for resource allocation in wireless systems. The functional-optimization, non-convex constraints, and lack of model knowledge makes these problems challenging, if not impossible, to solve directly. We used the concept of universal function approximation of deep neural networks and the theory of Lagrangian duality to show that, despite the non-convex nature of these problems, they can be formulated with a finite-dimensional, unconstrained optimization problem in the dual domain with either bounded suboptimality, or in the case of arbitrarily large DNNs, precise optimality with respect to the original problem. The dual domain formulation motivates solving via the use of primal-dual descent methods, which can furthermore be replaced with zeroth-ordered equivalents that estimate gradients without explicit model knowledge. We additionally perform a variety of simulations on common resource allocation problems that demonstrate the effectiveness in DNN-parameterizations to find accurate solutions.

4.8 Proof of Theorem 3

To inform the analysis of the suboptimality of D^*_ϕ from (4.11), we first present an established result previously referenced, namely the null duality gap property of the original problem in (4.7). We proceed by presenting the associated Lagrangian function and dual problem for

the constrained optimization problem in (4.2):

$$\mathcal{L}(\mathbf{p}(\mathbf{h}), \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) := g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}^T \left(\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) \right] - \mathbf{x} \right), \tag{4.36}$$

$$D^* := \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} \max_{\mathbf{p} \in \mathcal{P}, \mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{p}(\mathbf{h}), \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}). \tag{4.37}$$

Despite non-convexity of (4.2), a known result established in [103] demonstrates that problems of this form indeed satisfy a null duality gap property given the technical conditions previously presented. Due to the central role it plays in the proceeding analysis of (4.11), we present this theorem here for reference.

Theorem 5. *[103, Theorem 1] Consider the optimization problem in* (4.2) *and its Lagrangian dual in* (4.37)*. Provided that Assumptions 6 and 7 hold, then the problem in* (4.2) *exhibits null duality gap, i.e.,* $P^* = D^*$*.*

With this result in mind, we begin to establish the result in (4.14) by considering the upper bound. First, note that the dual problem of (4.7) defined in (4.11) can be written as

$$D^*_{\boldsymbol{\phi}} = \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} \left\{ \max_{\mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} + \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) \right] \right\}. \tag{4.38}$$

Focusing on the second term, observe then that for any solution $\mathbf{p}^*(\mathbf{h})$ of (4.2), it holds that

$$\max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) \right] = \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h}) \right] + \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) - \mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h}) \right]. \tag{4.39}$$

Since $\mathcal{P}_{\boldsymbol{\phi}} \subseteq \mathcal{P}$, it must be that $g_0(\mathbf{x}^\star_\theta) \leq g_0(\mathbf{x}^\star)$, where $\mathbf{x}^\star$ and $\mathbf{x}^\star_\theta$ are the maximizers of (4.2) and (4.7) respectively. Because g_0 is monotonically non-decreasing, the ergodic constraint holds with equality and $g_0(\mathbf{x}^\star_\theta) \leq g_0(\mathbf{x}^\star)$ implies that

$$\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{h}, \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}^\star)) \right] = \mathbf{x}^\star_\theta \leq \mathbf{x}^\star = \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h}) \right],$$

where $\boldsymbol{\theta}^\star$ is a solution of (4.7). By optimality, it holds that $\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) - \mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h}) \right] \leq 0$ for all $\boldsymbol{\theta} \in \Theta$. Since $\boldsymbol{\lambda} \geq \mathbf{0}$, (4.39) yields

$$\max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) \right] \leq \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h}) \right]. \tag{4.40}$$

Substituting (4.40) back into (4.38) and using the strong duality result from Theorem 5, we

obtain

$$D^*_{\boldsymbol{\phi}} \leq \min_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq \mathbf{0}} \max_{\mathbf{x}\in\mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ + \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})\right] = D^* = P^*, \quad (4.41)$$

where we used the fact that the right-hand side of the inequality in (4.41) is the optimal dual value of problem (4.2) as defined in (4.37).

We prove the lower bound in (4.14) by proceeding in a similar manner, i.e., by manipulating the expression of the dual value in (4.38). In contrast to the previous bound, however, we obtain a perturbed version of (4.2) which leads to the desired bound. Explicitly, notice that for all $\mathbf{p} \in \mathcal{P}$ it holds that

$$\max_{\boldsymbol{\theta}\in\Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] = \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right] \\ - \min_{\boldsymbol{\theta}\in\Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right], \quad (4.42)$$

where we used the fact that for any f_0 and $\mathcal{Y}$, it holds that $\max_{y\in\mathcal{Y}} f_0(y) = -\min_{y\in\mathcal{Y}} -f_0(y)$. Then, apply Hölder's inequality to bound the second term in (4.42) as

$$\min_{\boldsymbol{\theta}\in\Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] \\ \leq \|\boldsymbol{\lambda}\|_1 \left[\min_{\boldsymbol{\theta}\in\Theta} \|\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] \|_\infty\right]. \quad (4.43)$$

To upper bound the minimization in (4.43), start by using the convexity of the infinity norm and the continuity of $\mathbb{E}_{\mathbf{h}}\mathbf{f}(\mathbf{h}, \cdot)$ to obtain

$$\min_{\boldsymbol{\theta}\in\Theta} \|\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] \|_\infty \\ \leq \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\mathbf{h}} \left[\|\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\|_\infty\right] \\ \leq \min_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\mathbf{h}} \left[L\|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta})\|_\infty\right].$$

The definition in (4.8) then readily gives

$$\min_{\boldsymbol{\theta}\in\Theta} \|\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] \|_\infty \leq L\epsilon. \quad (4.44)$$

Substituting (4.43) and (4.44) into (4.42) yields

$$\max_{\boldsymbol{\theta}\in\Theta} \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}), \mathbf{h})\right] \geq \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right] - \|\boldsymbol{\lambda}\|_1 L\epsilon,$$

which we can then use in the definition of the dual value (4.38) to obtain

$$D^*_{\boldsymbol{\phi}} \geq \min_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq \mathbf{0}} \max_{\mathbf{x}\in\mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ + \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right] - \|\boldsymbol{\lambda}\|_1 L\epsilon. \quad (4.45)$$

We are now ready to derive the perturbed version of (4.2) in order to obtain our lower bound. To do so, notice that $\boldsymbol{\lambda} \geq 0$ implies that $\|\boldsymbol{\lambda}\|_1 = \boldsymbol{\lambda}^T \mathbf{1}$, where $\mathbf{1}$ is a column vector of ones. Since (4.45) holds for all $\mathbf{p} \in \mathcal{P}$, we get

$$D^*_{\boldsymbol{\phi}} \geq \min_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq \mathbf{0}} \max_{\mathbf{x}\in\mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ + \max_{\mathbf{p}\in\mathcal{P}} \boldsymbol{\lambda}^T \left\{\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right] - L\epsilon \mathbf{1}\right\}. \quad (4.46)$$

Now, observe that the right-hand side of (4.46) is the dual value of an $(L\epsilon)$-perturbed version of (4.2)

$$\begin{aligned} P^*_{L\epsilon} := & \max_{\mathbf{p},\mathbf{x}} C(\mathbf{x}) \\ & \text{s.t. } L\epsilon\mathbf{1} + \mathbf{x} \leq \mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right], \quad \mathbf{0} \leq \mathbf{g}(\mathbf{x}), \\ & \mathbf{x} \in \mathcal{X}, \quad \mathbf{p} \in \mathcal{P} \end{aligned} \quad (4.47)$$

Naturally, (4.47) has the same strong duality property as (4.2) from Theorem 5, which implies that $D^*_{\boldsymbol{\phi}} \geq D^*_{L\epsilon} = P^*_{L\epsilon}$. A well-known perturbation inequalitiy, e.g., [14, Eq. (5.57)], relates $P^*_{L\epsilon}$ to P^* as

$$P^*_{L\epsilon} \geq P^* - \|\boldsymbol{\lambda}^*\|_1 L\epsilon. \quad (4.48)$$

Combining (4.48) with $D^*_{\boldsymbol{\phi}} \geq P^*_{L\epsilon}$, we obtain (4.14).

We proceed to prove the bound in (4.15). Note that the strong duality result in Theorem 5 implies that

$$\begin{aligned} P^* = D^* = & \max_{\mathbf{p}\in\mathcal{P},\mathbf{x}\in\mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^{*T} \mathbf{g}(\mathbf{x}) \\ & + \boldsymbol{\lambda}^{*T} \left(\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})\right] - \mathbf{x}\right) \\ & \geq g_0(\mathbf{x}') + \boldsymbol{\mu}^{*T} \mathbf{g}(\mathbf{x}') \\ & + \boldsymbol{\lambda}^{*T} \left(\mathbb{E}_{\mathbf{h}} \left[\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))\right] - \mathbf{x}'\right), \end{aligned} \quad (4.49)$$

where $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ are the minimizers of (4.37) and $(\mathbf{x}', \mathbf{p}')$ are arbitrary feasible points of (4.2). Since Slater's condition holds, we can choose $(\mathbf{x}', \mathbf{p}')$ to be strictly feasible, i.e., such

that $\mathbf{g}(\mathbf{x}') > \mathbf{0}$ and $\mathbb{E}_{\mathbf{h}}\left[\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))\right] > \mathbf{x}'$, to obtain

$$\begin{aligned} P^* &\geq g_0(\mathbf{x}') + \boldsymbol{\lambda}^{*T}\left(\mathbb{E}_{\mathbf{h}}\left[\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))\right] - \mathbf{x}'\right) \\ &\geq g_0(\mathbf{x}') + \boldsymbol{\lambda}^{*T}\mathbf{1} \cdot s, \end{aligned} \tag{4.50}$$

where $s = \min_i \mathbb{E}_{\mathbf{h}}\left[f_i(\mathbf{h}, \mathbf{p}'(\mathbf{h}))\right] - x_i'$ as defined in Assumption 7, with $\mathbf{f} = [f_i]$ and $\mathbf{x}' = [x_i]$. Note that $s > 0$ since $\mathbf{x}'$ is strictly feasible. Finally, since $\boldsymbol{\lambda}^* \geq 0$, we can rearrange (4.50) to obtain

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}')}{s}. \tag{4.51}$$

4.9 Proof of Theorem 4

We start by presenting the well-established result that a DNN of arbitrarily large size is a universal parameterization for measurable functions in probability.

Theorem 6. *[59, Theorem 2.2] Define $\mathcal{D} = \{\boldsymbol{\phi}(\cdot, \boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathbb{R}^q\}$ to be the set of all functions described by the DNN in (4.29) with $\boldsymbol{\sigma}_l$ non-constant and continuous for all $l = 1, \ldots, L$. Then, for an arbitrarily large number q of hidden nodes, $\mathcal{D}$ is dense in probability in the set of measurable functions $\mathcal{M}$, i.e., for every function $\hat{\mathbf{p}}(\mathbf{h}) \in \mathcal{M}$ and all $\tilde{\epsilon} > 0$, there exists a $q > 0$ and $\boldsymbol{\theta} \in \mathbb{R}^q$ such that*

$$m\left(\{\mathbf{h} \in \mathcal{H} : \|\hat{\mathbf{p}}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\|_\infty > \tilde{\epsilon}\}\right) < \tilde{\epsilon}. \tag{4.52}$$

Using Theorem 6, we can show that DNNs satisfy the ϵ-universality condition from Definition 1 for all $\epsilon > 0$.

Lemma 4. *The entire class of DNN parameterizations $\boldsymbol{\phi} \in \Phi$, where $\boldsymbol{\phi}$ is defined in (4.29) with non-constant, continuous activation $\boldsymbol{\sigma}_l$ for some layer size $\mathbf{q} > \mathbf{0}$, is an ϵ-universal parametrization as in Definition 1 for all $\epsilon > 0$.*

Proof. Let $\mathcal{K}_{\epsilon'} = \{\mathbf{h} \in \mathcal{H} : \|\hat{\mathbf{p}}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\|_\infty > \epsilon'\} \subseteq \mathcal{H}$ and observe that (4.8) can be written as

$$\begin{aligned} \mathbb{E}\left\|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\right\|_\infty = &\int_{\mathcal{H}\setminus\mathcal{K}_{\epsilon'}} \left\|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\right\|_\infty dm(\mathbf{h}) \\ &+ \int_{\mathcal{K}_{\epsilon'}} \left\|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\right\|_\infty dm(\mathbf{h}), \end{aligned}$$

where m is the probability measure from which the channel state is drawn. It is ready that the first integral is upper bounded by $\epsilon' \cdot m(\mathcal{H} \setminus \mathcal{K}_{\epsilon'}) < \epsilon' \cdot m(\mathcal{H}) < \epsilon'$ for all $\epsilon' > 0$. To bound the second integral, recall that the set of feasible policies $\mathcal{P}$ is bounded and

let $\Gamma = \sup\{\|\hat{\mathbf{p}}(\mathbf{h})\|_\infty : \hat{\mathbf{p}} \in \mathcal{P} \text{ and } \mathbf{h} \in \mathcal{H}\} < \infty$. Then, from (4.52), we obtain the following bound over all DNNs $\boldsymbol{\phi}$, i.e.

$$\inf_{\boldsymbol{\phi}\in\Phi} \int_{\mathcal{K}_{\epsilon'}} \|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\|_\infty \, dm(\mathbf{h}) < 2\Gamma \cdot m(\mathcal{K}_{\epsilon'}) < 2\Gamma\epsilon'.$$

Thus, for all $\epsilon' > 0$,

$$\inf_{\boldsymbol{\phi}\in\Phi} \mathbb{E}\, \|\mathbf{p}(\mathbf{h}) - \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})\|_\infty < (1 + 2\Gamma)\epsilon'. \tag{4.53}$$

Taking $\epsilon' = \epsilon/(1 + 2\Gamma)$ in (4.53) yields (4.8). □

Lemma 4 implies that the dual value bound (4.14) from Theorem 3 holds for all $\epsilon > 0$ if we consider the entire class of DNN functions $\boldsymbol{\phi} \in \Phi$. Since the Lipschitz constant $L < \infty$, the only obstacle to completing a continuity argument is if $\|\boldsymbol{\lambda}^*\|_1$ is unbounded. However, recall from (4.15) that

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}_0)}{s} < \infty. \tag{4.54}$$

Hence, we obtain that

$$P^* - \delta \leq \inf_{\boldsymbol{\phi}\in\Phi} D^*_{\boldsymbol{\phi}} \leq P^*$$

for all $\delta > 0$ (simply take $\epsilon = \delta \, \|\boldsymbol{\lambda}^*\|_1^{-1} \, L^{-1} > 0$). Then, there would exist $\delta' > 0$ such that $P^* > D^*_{\boldsymbol{\phi}} + \delta'$ (e.g., take δ' to be the midpoint between P^* and $D^*_{\boldsymbol{\phi}}$), which would contradict Theorem 3. Hence, $\inf_{\boldsymbol{\phi}} D^*_{\boldsymbol{\phi}} = P^*$.

Chapter 5

Wireless Resource Allocation with Graph Neural Networks

5.1 Introduction

The design of wireless systems has become an integral part of recent developments in large scale intelligent systems, from robotics to the Internet of Things. Such a design requires the balancing of the numerous utilities and constraints of large networks of wireless connected devices. At a high level, the problem of optimal design can be viewed as the allocation of a finite set of resources to achieve strong average performance over the randomly noisy wireless channel. Problems of this form range from power control to the optimization of frequency division multiplexing [126], beamforming [8, 112], and random access [61, 62].

The optimal resource allocation can, in general, be formulated as an optimization problem that maximizes the expected capacity over all devices subject to a set of system constraints. While this problem can be easily formulated, both the non-convexity and infinite dimensionality inherent in the problem makes it generally challenging to solve. Simpler wireless systems of this form can be solved in the Lagrangian dual domain [103, 136] and subsequently solved using dual descent methods—see, e.g. [47,124,138] for applications of this approach. All such approaches invariably require accurate system models and may require prohibitivley large computational complexity for each allocation decision. Alternatively, heuristic optimization and scheduling methods have been developed for the more canonical resource allocation problems [18, 87, 111, 130].

In contrast to such model-based heuristics, more recent work has applied machine learning and regression techniques to solve resource allocation problems. Machine learning methods train a generic learning model, such as a deep neural network (DNN), to approximate the behavior of resource allocation strategies for a wide variety of problems. One such approach follows the tenants of supervised learning, or in other words fitting a neural network to

a training set of solutions obtained using a specific algorithm [70, 115, 121, 131]. These techniques are useful in their simplicity and relative effectiveness—neural networks are well suited for finding good local minima in loss functions typically used in supervised learning, e.g. Euclidean loss. However, supervised learning techniques are limited by both the availability of solutions needed to built a training set as well as the accuracy of such solutions. The former limitation implies supervised learning can only be used in resource allocation problems with existing heuristic solutions, while the latter limitation implies that the learning model will only meet the performance of such heuristics but never exceed them.

A more promising approach in learning for resource allocation uses a learning model to directly parameterize the resource allocation policy in the optimization problem [**?**, 25, 69, 73, 83, 132]. This can be considered unsupervised in that such techniques can train neural networks with respect to an abstract performance measure and thus does not require the acquisition of a training set—or, in other words, reinforcement learning. These techniques are further beneficial in that they can be applied to any arbitrary resource allocation problem and have the potential to exceed performance of existing heuristics. Previous work in [**?**] formally draws an equivalence between resource allocation problems and constrained statistical learning—or constrained regression—to develop a theoretical and algorithmic framework for learning resource allocation policies for a generic class of resource allocation problems. The universal approximation properties of fully connected NNs (FCNNs) is also used to recover the duality results of [103, 136], making them an attractive policy parameterization.

Just as FCNNs are made a naturally viable choice for resource allocation parameterizations from their universality property [**?**, 115], so too are they limited. The practical challenge of training FCNNs to parameterize strong performing policies is well documented in empirical study, as their expressive power inherently necessitates performing optimization over a very high dimensional space. Moreover, the completely generic structure of FCNN contains no intrinsic invariance to input scaling or variation; any change in wireless network size or shuffling of the network labels renders the current FCNN-based policy ineffective. Convolutional neural networks architectures (CNNs), on the other hand, have proved a solution to this problem for many learning domains such as image classification and recommender systems by preserving invariances in the architecture itself. While some existing work has used CNNs for wireless resource allocation [69, 121, 131], they utilize standard temporal or spatial CNNs and thus do not leverage the true invariances present in wireless networks. In particular, the structure—and subsequent invariances—of wireless networks comes from the links between transmitters and receivers that result from fading. The work in [23] uses spatial CNNs that utilize the geometric structure of the network, but in doing so does not incorporate the fading link structure. In this work, we incorporate a recent

development in CNNs that perform convolutions on arbitrarily structured data—called graph neural networks [45, 55]—to fully utilize the network of fading links in parameterizing a resource allocation policy. When such learning architectures are used in conjunction with the model-free algorithmic learning approach developed in [38], we obtain a unified framework for learning effective resource allocation policies in wireless systems.

We begin the paper by introducing a generic formulation of wireless resource allocation problems in which we seek a instantaneous resource allocation policy given a set of random fading states and a set of transmitter states (Section 5.2). Such a formulation has many applications, ranging from multiple access to wireless control systems. By observing that the resource allocation problem takes the form of a statistical learning problem, we proceed by parameterizing the resource allocation policy with a learning model (Section 5.2.1). The choice of this parameterization, however, is key in finding good policies. A FCNN, despite its universality property, is ineffective for large scale systems and is thus not appropriate for modeling complex and practical wireless systems. We instead opt to utilize a parameterization that retains a key structural property of the optimal resource allocation policy—namely, its permutation equivariance (Section 5.2.2).

We proceed to discuss the details of the so-called graph neural network (GNN), a recently developed architecture that generalizes the popular convolutional neural networks for graph structured data (Section 5.3). Such an architecture is perfectly suited for resource allocation policies, as the graph structure naturally occurs though the fading interference graph on the links between transmitters and receivers. However, as such a graph will itself randomly vary with the fading states, we consider the resulting random edge graph neural networks (REGNNs) as our policy parameterizations (Section 5.3.1). The REGNN retains the permutation equivariance property, thus making it easier to scale and generalize to varying networks. To train the filters weights of the REGNN, we utilize a model-free, primal-dual learning algorithm that optimizes the objective while learning to satisfy constraints (Section 5.4). We perform a comprehensive set of numerical simulations to evaluate the REGNN-based policies on three different resource allocation problems (Section 5.5).

5.2 Optimal Resource Allocation

Consider a large scale wireless system with a set m transmitter in $\mathcal{M} := \{1, 2, \dots, m\}$ and a set of n receivers in $\mathcal{N} := \{1, 2, \dots, n\}$, as pictured in Figure 5.1. Each transmitter $i \in \mathcal{M}$ is paired with a single receiver $k_i \in \mathcal{N}$—denote the set of transmitters paired with receiver k as $\mathcal{M}_k := \{i \mid k_i = k\}$ such that $\bigcup_{k=1}^{n} \mathcal{M}_k = \mathcal{M}$ and $\mathcal{M}_k \cap \mathcal{M}_{k^*} = \emptyset$ for all $k \neq k^*$. The system experiences random states both on the links between transmitters—as a result of their interference in the wireless fading channel—and on the transmitters themselves—some state of the system associated with the transmitter. We denote by $\mathbf{H} \in \mathcal{H} \subseteq \mathbb{R}^{m \times m}$ the

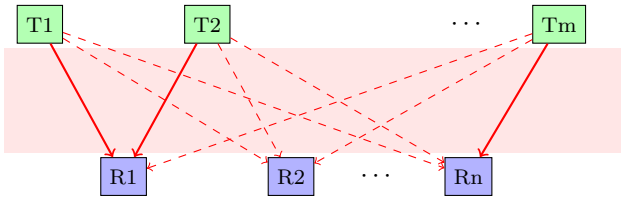


Figure 5.1: A wireless network with m transmitters (green) and n receivers (blue). The transmitters send information over a wireless fading channel, with direct links between transmitter and receiver shown in solid lines and the interference links shown in dashed lines.

random link states drawn from a distribution $m(\mathbf{H})$ and by $\boldsymbol{\eta} \in \mathbb{R}^m$ the random transmitter states drawn from a distribution $\mu(\boldsymbol{\eta})$. The components h_{ij} reflect the state of the link between transmitter i and j, while η_i reflects the state of transmitter i. Typically, the diagonal terms h_{ii} denote the fading state of the direct link between transmitter i and its associated receiver k_i, while of the off-diagonal terms h_{ji} denote the fading state of the interference link between transmitter j and receiver k_i.

For each fading channel realization $\mathbf{H}$ and transmitter state $\boldsymbol{\eta}$, we define a resource allocation policy $\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}) \in \mathbb{R}^m$. Furthermore, given a channel realization and a resource allocation, the system experiences a performance level $\mathbf{f}(\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}), \mathbf{H})$ that is some measure of channel utilization, e.g. channel capacity, bit error rate—see Examples 5-7. Likewise, we define a cost function $\mathbf{g}(\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}), \boldsymbol{\eta})$ that represents the cost of being in transmitter state $\boldsymbol{\eta}$ with power allocation $\mathbf{p}(\mathbf{H}, \boldsymbol{\eta})$. In fast fading channels, the system allocates resources instantaneously but the policy is designed with respect to long term or average behavior. For this, we consider the vector $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}), \mathbf{H}\big)\big] \in \mathbb{R}^m$ as the average level of performance experienced by users and the vector $\mathbf{y} = \mathbb{E}\big[\mathbf{g}\big(\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}), \boldsymbol{\eta}\big)\big] \in \mathbb{R}^m$ as the average cost experienced by users.

The goal in optimal design of wireless communication systems is to find the instantaneous resource allocation policy $\mathbf{p}(\mathbf{H}, \boldsymbol{\eta})$ that optimizes a utility over the performance metric $\mathbf{x}$ and cost metric $\mathbf{y}$. To formulate this problem mathematically we introduce a scalar utility $w : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ and a vector constraint function $\mathbf{c} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^u$. We further introduce the sets $\mathcal{P} \subseteq \mathcal{M}$, where $\mathcal{M}$ is the set of functions integrable with respect to $m(\mathbf{H})$ and $\mu(\boldsymbol{\eta})$, to constrain the values that can be output by the instantaneous resource allocation policy. We assume $\mathcal{P}$ contains bounded functions, i.e., that the resources being allocated are finite. With these definitions, we let the optimal resource allocation problem in wireless

communication systems be a program of the form

$$\begin{aligned} P^* := \max_{\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\mathbf{x},\mathbf{y}} \; & w(\mathbf{x},\mathbf{y}), \\ \text{s.t. } & \mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\mathbf{H}\big)\big],\ \mathbf{y} = \mathbb{E}\big[\mathbf{g}\big(\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\boldsymbol{\eta}\big)\big] \\ & \mathbf{c}(\mathbf{x},\mathbf{y}) \geq \mathbf{0}, \quad \mathbf{p}(\mathbf{H},\boldsymbol{\eta}) \in \mathcal{P}. \end{aligned} \tag{5.1}$$

In (5.1) the utility $w(\mathbf{x},\mathbf{y})$ is the one we seek to maximize while the constraints $\mathbf{c}(\mathbf{x},\mathbf{y})$ are required to be nonnegative. Note that the utilities $w(\mathbf{x},\mathbf{y})$ and $\mathbf{c}(\mathbf{x},\mathbf{y})$ are assumed to be concave with respect to $\mathbf{x}$ and $\mathbf{y}$. However, the functions $\mathbf{f}(\cdot,\mathbf{H})$ and $\mathbf{g}(\cdot,\boldsymbol{\eta})$ are not assumed convex or concave, nor the set $\mathcal{P}$. For many resource allocation problems of interest, these functions are indeed non-convex so it is thus essential that this no such modeling assumption holds here—see [103]. To elaborate further on the resource allocation formulation in 5.1, we present practical cases that take this form.

Example 5 (Multiple access AWGN channel)**.** Consider a set of m terminals communicating with associated receivers on a shared channel. A standard instantaneous performance metric of interest here is the capacity experience by each user, which is generally obtained as $c = \log(1 + \text{SINR})$, where SINR denotes the signal to interference plus noise ratio. The ith element of $\mathbf{f}(\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\mathbf{H})$ may then denote the instantaneous capacity achieved by transmitter i. In a channel subject to additive white Gaussian noise (AWGN) and multi-user interference and assuming the use of capacity achieving codes, this is written as

$$f_i(\mathbf{p},\mathbf{H}) := \log\left(1 + \frac{h_{ii}p_i(\mathbf{H},\boldsymbol{\eta})}{1 + \sum_{j\neq i} h_{ji}p_j(\mathbf{H},\boldsymbol{\eta})}\right). \tag{5.2}$$

Defining the performance as in (5.2) in the constraint in (5.1) reflects the a maximization with respect to long term or average capacity experienced by the users. A constraint of the form $\mathbf{c}(\mathbf{x},\mathbf{y})$ can be a minimum average capacity $\mathbf{c}(\mathbf{x},\mathbf{y}) := \mathbf{x} - \mathbf{c}_{\min}$ for all users. Power constraints can be enforced via the set $\mathcal{P} = \{\mathbf{p} : \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}_0\}$ and the utility w can be chosen to be the weighted sum rate $w(\mathbf{x},\mathbf{y}) = \sum_i w_i x_i$ or a proportional fair utility $w(\mathbf{x},\mathbf{y}) = \sum_i \log(x_i)$. Observe that, in this problem, it is assumed that there is no transmitter state $\boldsymbol{\eta}$ or associated cost function $\mathbf{g}(\mathbf{p},\boldsymbol{\eta})$ that play a role in the system design.

Example 6 (Multiple access with data collection)**.** Example 5 can be further augmented to consider transmitter states when the transmitters are co-located with local sensors that are collecting data to be transmitted. Here, the state η_i reflects the collection or arrival rate at the i transmitter. A simple cost metric is then given by the arrival rate, i.e. $g_i(\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\boldsymbol{\eta}) := \eta_i$, and a necessary constraint of the system is that average capacity

exceeds the average collection rate, i.e.

$$\mathbf{c}(\mathbf{x}, \mathbf{y}) := \mathbf{x} - \mathbf{y} \geq \mathbf{0}. \tag{5.3}$$

This is a problem formulation that makes full use of the generality in (5.1) by containing both the fading channel states and associated performance metric in the capacity function given in (5.2), as well as a transmitter state given by data collection and associated coupled constraint given by (5.3).

Example 7 (Random access wireless control systems)**.** A more complex example that takes the form of (5.1) is the modeling of a wireless control system. Consider that transmitters are sending plant state information to a single receiver/base station to compute control inputs over a shared random access channel that is subject to potential packet collisions. Given the direct and interference channel states and transmission powers, we define a function $q(p_i, h_{ii}, p_j, h_{ij}) \to [0, 1]$ that gives a probability of collision between nodes i and j. We are interested in a performance metric $f_i(\mathbf{p}, \mathbf{H})$ that measures the probability of successful transmission of transmitter i as

$$f_i(\mathbf{p}, \mathbf{H}) := \prod_{j \neq i} \left(1 - q(p_i, h_{ii}, p_j, h_{ij})\right). \tag{5.4}$$

Likewise, the transmitter state η_i denotes the state of the plant at the ith transmitter. If power is applied, the system state evolves with gain $\gamma_c > 0$; otherwise, it evolves with gain $\gamma_o > \gamma_c$. We are often concerned with a quadratic cost that measures the one future step distance from the origin of the plant state, which can be written as

$$g_i(\mathbf{p}, \boldsymbol{\eta}) := \mathbb{1}[p_i \geq 0](\gamma_c \eta_i)^2 + \mathbb{1}[p_i = 0](\gamma_o \eta_i)^2. \tag{5.5}$$

From here, we wish to minimize a long-term objective that scales the long term quadratic cost in (5.5) with the long-term packet success rate in (5.4), i. e. $w(\mathbf{x}, \mathbf{y}) := \sum_{i=1}^{m} x_i y_i$ and constraints that impose a minimum long-term cost $\kappa_{\max}$ for each plant, i. e. $c_i(x_i, y_i) := x_i y_i - \kappa_{\max} \leq 0$.

As exemplified in Examples 5-7, the resource allocation problem in (5.1) generalizes a wide variety of problems of interest in optimal design of wireless systems. We can thus proceed to develop a means of obtaining solutions to (5.1) for large scale systems, i.e. when m or n are large. Finding policies that solve (5.1) directly is inherently challenging, due to both the functional optimization form it takes, as well as the non-convex constraints it includes. To handle these complexities, we follow an interpretation of (5.1) originally developed in [**?**], in which we identify it as a constrained statistical learning, or regression, problem. Observe that the constraints the channel performance function $\mathbf{f}(\mathbf{p}(\mathbf{H}, \boldsymbol{\eta}), \mathbf{H})$

and state cost function $\mathbf{g}(\mathbf{p}(\mathbf{H},\boldsymbol{\eta}),\boldsymbol{\eta})$ are statistical in nature in that they are functions of random quantities $\mathbf{H}$ and $\boldsymbol{\eta}$. In this way, seek a function that optimizes over statistical losses or utilities, precisely the goal of standard statistical regression. The resource allocation problem in (5.1) is distinct from standard regression, however, due to the statistical losses appearing as constraints. We present a manner in which to handle the constraints in Section 5.4, but may nonetheless proceed in a manner similar to regression problems—by replacing the functional optimization with a finite dimensional policy parameterization.

5.2.1 Parameterization of resource allocation policy

We reduce the dimensionality of the problem in (5.1) by parameterizing the resource allocation function $\mathbf{p}(\mathbf{H},\boldsymbol{\eta})$ with a policy of a pre-specified form. That is, we define some function $\boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta})$ that has a closed, analytic from for any given parameter vector $\boldsymbol{\theta}\in\mathbb{R}^q$. For example, we may select a simple linear-quadratic parameterization, i.e. $\boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta}) = \boldsymbol{\theta}^T\mathbf{H}\boldsymbol{\theta}+\boldsymbol{\theta}^T\boldsymbol{\eta}$. With a given parametrization, the optimization problem in (5.1) becomes one in which the optimization is over $\mathbf{p}(\mathbf{H},\boldsymbol{\eta})$ is replaced with an optimization over the set of parameter vectors $\boldsymbol{\theta}$,

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta},\mathbf{x},\mathbf{y}}\ & w(\mathbf{x},\mathbf{y}), \\ \text{s.t. }\ & \mathbf{x}=\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta}),\mathbf{H}\big)\big], \mathbf{y}=\mathbb{E}\big[\mathbf{g}\big(\boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta}),\boldsymbol{\eta}\big)\big] \\ & \mathbf{c}(\mathbf{x},\mathbf{y})\leq\mathbf{0}, \quad \boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta})\in\Theta, \end{aligned} \tag{5.6}$$

where we have defined the set $\Theta := \{\boldsymbol{\theta}\in\mathbb{R}^p \mid \boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta})\in\mathcal{P}\}$ as the set of policies of the parameterized form that are in $\mathcal{P}$. We further define the optimal policy $\mathbf{p}^*(\mathbf{H},\boldsymbol{\eta})$ as the policy that solves (5.1).

Observe in (5.6) that by restricting our attention to policies of the parameterized form, we remove the functional optimization in (5.1) with optimization over a parameter vector $\boldsymbol{\theta}$ with finite dimension q. While the problem in (5.6) can be formed with any choice of parameterization, this choice is critical in far from arbitrary and plays a central role in the design of learning policies for wireless resource allocation systems. Naturally, the difference between $P^*_{\boldsymbol{\phi}}$ and P^* depends upon the representative power of the $\boldsymbol{\phi}(\cdot,\boldsymbol{\theta})$ relative to the optimal power allocation policy $\mathbf{p}^*(\mathbf{H},\boldsymbol{\eta})$. This is to say that proper selection of a parameterization can be done with two approaches:

(a) Select $\boldsymbol{\phi}(\cdot,\boldsymbol{\theta})$ that is sufficiently dense that it can represent a large class of arbitrary functions, or

(b) Select $\boldsymbol{\phi}(\cdot,\boldsymbol{\theta})$ that retains specific structural properties held by $\mathbf{p}^*(\mathbf{H},\boldsymbol{\eta})$.

For the former approach (a) of parameterization design, it reasons to select one among those known to exhibit a property of universality, or near-universality. Such parameterizations have theoretical properties that allow them to approximate any arbitrary or integrable function within small error. Well-known cases of this include deep neural networks, radial basis function networks, and reproducing kernel Hilbert spaces. Fully connected deep neural networks (FCNNs), in particular, are commonly used in many function approximation applications and was the focus of previous work in wireless resource allocation; see, e.g. [**?**,115]. It can be shown that the optimality gap between solutions can be bounded by a small constant when using near-universal parameterizations [**?**]. However, for large scale wireless systems that are necessary for design of modern IoT applications and considered in this work, near-universal parameterizations are not necessarily a proper design choice. The mere theoretical existence of a parameter that achieves small error does not imply that such a parameter is easy to find.

To see this, consider the case of a FCNN with L layers labeled $l = 1, \ldots, L$ and each with a corresponding dimension q_l. In the FCNN architecture, each layer l is defined by the linear operation $\mathbf{W}_l \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\boldsymbol{\sigma}_l : \mathbb{R}^{q_l} \to \mathbb{R}^{q_l}$. If layer l receives as an input from the $l-1$ layer $\mathbf{z}_{l-1} \in \mathbb{R}^{q_{l-1}}$, the resulting output $\mathbf{z}_l \in \mathbb{R}^{q_l}$ is then computed as $\mathbf{z}_l := \boldsymbol{\sigma}_l(\mathbf{W}_l \mathbf{z}_{l-1})$. The final output of the FCNN, is then related to the input $\mathbf{z}_0 := [\mathbf{H}, \boldsymbol{\eta}]$ by propagating through each later of the DNN as $\boldsymbol{\phi}(\mathbf{H}, \boldsymbol{\eta}, \boldsymbol{\theta}) = \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{z}_0))))))$. Observe that the parameter vector $\boldsymbol{\theta}$ here contains the entries of $\{\mathbf{W}_l\}_{l=1}^{L}$ and $q = \sum_{l=1}^{L-1} q_l q_{l+1}$. By construction, the input dimension will equal $q_1 = m(n+1)$ as it includes the channel and transmitter states. Future layer dimensions $q_l, l > 1$ will often be even larger. As can be seen, the full parameter dimension q grows quickly with wireless network size and is challenging to learning in practice. Furthermore, achieving the near-universality property inherently requires considering very deep or wide FCNN architectures, and thus q will necessarily be large in practice. Computational challenges aside, this further complicates the learning process by increasing the likelihood of getting stuck in poor local minima of the non-convex FCNN. These complications make near-universal parameterizations impractical for large scale systems.

These complications motivate the adoption of the latter approach (b), in which we employ a lower-dimensional parametrization that retains some of the overall structure of $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$, thus significantly facilitating the learning process in practice. Such a structure preserving parameterization has the additional benefit of having better generalization properties with respect to small variations in the $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$ that come from perturbing the problem in (5.1)—FCNNs, on the other hand, are known for being prone to overfitting. We proceed by establishing a key structural property of $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$ called permutation equivariance.

5.2.2 Permutation equivariance of optimal resource allocation

A function or policy that demonstrates permutation equivariance is one such that a permutation of inputs results in an equally permuted output. We define a permutation matrix $\boldsymbol{\Pi} \in \{0,1\}^{m \times m}$ as a binary matrix that satisfies

$$\boldsymbol{\Pi}\mathbf{1} = \mathbf{1}, \qquad \boldsymbol{\Pi}^T\mathbf{1} = \mathbf{1}. \tag{5.7}$$

A permutation can alternatively be thought of as a re-labeling or reordering of the coordinates in a vector or matrix. This is an intuitive property, as it implies that the output of the policy should not rely on the ordering of the input, and naturally holds for many regression or classification functions. More specifically, this property can be established for the optimal policy $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$, i.e. the argument that solves (5.1), given a couple commonly held assumptions on the functions given in (5.1). We present the following assumptions.

AS10. *The function $w(\mathbf{x}, \mathbf{y})$ is permutation invariant, i.e. $w(\boldsymbol{\Pi}^T\mathbf{x}, \boldsymbol{\Pi}^T\mathbf{y}) = w(\mathbf{x}, \mathbf{y})$.*

AS11. *The constraint functions $\mathbf{f}(\mathbf{p}, \mathbf{H})$, $\mathbf{g}(\mathbf{p}, \boldsymbol{\eta})$, and $\mathbf{c}(\mathbf{x}, \mathbf{y})$ are permutation equivariant, i.e.*

$$\begin{gathered} \mathbf{f}(\boldsymbol{\Pi}^T\mathbf{p}, \boldsymbol{\Pi}^T\mathbf{H}\boldsymbol{\Pi}) = \boldsymbol{\Pi}^T\mathbf{f}(\mathbf{p}, \mathbf{H}), \\ \mathbf{g}(\boldsymbol{\Pi}^T\mathbf{p}, \boldsymbol{\Pi}^T\boldsymbol{\eta}) = \boldsymbol{\Pi}^T\mathbf{g}(\mathbf{p}, \boldsymbol{\eta}), \quad \mathbf{c}(\boldsymbol{\Pi}^T\mathbf{x}, \boldsymbol{\Pi}^T\mathbf{x}) = \boldsymbol{\Pi}^T\mathbf{c}(\mathbf{x}, \mathbf{y}). \end{gathered} \tag{5.8}$$

Assumptions 10-11 establish permutation equivariances (or invariances) for the component functions that make up the resource allocation problem in (5.1). We point out that most common cases, such as those described in Examples 5-7, can be easily shown to satisfy such properties. In the case of the utility function $w(\mathbf{x}, \mathbf{y})$, common choices that satisfy Assumption 10 include the unweighted sum-rate $\sum x_i$ and sum-log-rate $\sum \log(x_i)$. Likewise, both the capacity function in (5.2) and the packet success rate in (5.4) satisfy the permutation equivariant property for $\mathbf{f}(\mathbf{p}, \mathbf{H})$ in Assumption 11. Given these assumptions, we present a proposition on the permutation equivariance of the optimal resource allocation policy.

Proposition 4. *Denote by $\mathbf{p}^*(\mathbf{H})$ as the argument that solves the power control problem (5.1) with wireless fading channels $\mathbf{H} \sim m(\mathbf{H})$ and $\boldsymbol{\eta} \sim \mu(\boldsymbol{\eta})$. Further assume that Assumptions 10 and 11 hold. There exists an optimal policy $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$ of (5.1) that is permutation equivariant, i.e. for any permutation matrix $\boldsymbol{\Pi}$, it holds that*

$$\mathbf{p}^*(\boldsymbol{\Pi}^T\mathbf{H}\boldsymbol{\Pi}, \boldsymbol{\Pi}^T\boldsymbol{\eta}) = \boldsymbol{\Pi}^T\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta}). \tag{5.9}$$

Proof:

For notational convenience throughout this proof, we use the hat notation to define permutations of vectors and matrices by an arbitrary permutation matrix $\boldsymbol{\Pi}$, e.g. $\hat{\mathbf{x}} = \boldsymbol{\Pi}^T\mathbf{x}$ and $\hat{\mathbf{H}} = \boldsymbol{\Pi}^T\mathbf{H}\boldsymbol{\Pi}$. To demonstrate the permutation equivariance of $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$ with respect to $\mathbf{H}$ and $\boldsymbol{\eta}$, first consider the relation between the optimal policy $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$ and associated ergodic variables $\mathbf{x}^*$ and $\mathbf{y}^*$ that solve (5.1)

$$\mathbf{x}^* = \mathbb{E}\left[\mathbf{f}(\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta}), \mathbf{H})\right], \tag{5.10}$$

$$\mathbf{y}^* = \mathbb{E}\left[\mathbf{g}(\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta}), \boldsymbol{\eta})\right]. \tag{5.11}$$

Consider the following policy $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ that outputs permutations of the optimal policy given permuted states, i.e. $\mathbf{p}'(\hat{\mathbf{H}}, \hat{\boldsymbol{\eta}}) = \hat{\mathbf{p}}^*(\mathbf{H}, \boldsymbol{\eta})$. Further consider the ergodic performances $\mathbf{x}'$ and $\mathbf{y}'$ obtained by the resource allocation $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ when that states $\mathbf{H}$ and $\boldsymbol{\eta}$ are permuted by $\boldsymbol{\Pi}$ as $\hat{\mathbf{H}}$ and $\hat{\boldsymbol{\eta}}$, respectively, i.e.

$$\mathbf{x}' = \mathbb{E}\left[\mathbf{f}(\mathbf{p}'(\hat{\mathbf{H}}, \hat{\boldsymbol{\eta}}), \hat{\mathbf{H}})\right] = \int_{\mathbf{H}} \mathbf{f}(\mathbf{p}'(\hat{\mathbf{H}}, \hat{\boldsymbol{\eta}}), \hat{\mathbf{H}}) m(\mathbf{H}) d\mathbf{H}. \tag{5.12}$$

$$\mathbf{y}' = \mathbb{E}\left[\mathbf{g}(\mathbf{p}'(\hat{\mathbf{H}}, \hat{\boldsymbol{\eta}}), \hat{\boldsymbol{\eta}})\right] = \int_{\boldsymbol{\eta}} \mathbf{g}(\mathbf{p}'(\hat{\mathbf{H}}, \hat{\boldsymbol{\eta}}), \hat{\mathbf{H}}) \mu(\boldsymbol{\eta}) d\boldsymbol{\eta}. \tag{5.13}$$

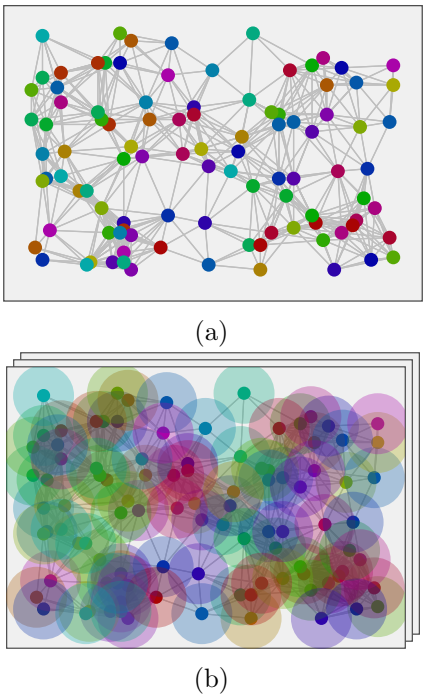
Given the definition of $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ and further considering the permutation equivariance of $\mathbf{f}$ and $\mathbf{g}$ from Assumption 11, it follows that

$$\mathbf{x}' = \int_{\mathbf{H}} \hat{\mathbf{f}}(\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta}), \mathbf{H}) m(\mathbf{H}) d\mathbf{H} = \hat{\mathbf{x}}^*, \tag{5.14}$$

$$\mathbf{y}' = \int_{\boldsymbol{\eta}} \hat{\mathbf{g}}(\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta}), \boldsymbol{\eta}) \mu(\boldsymbol{\eta}) d\boldsymbol{\eta} = \hat{\mathbf{y}}^*, \tag{5.15}$$

where the right hand sides $\hat{\mathbf{x}}^*$ and $\hat{\mathbf{y}}^*$ are the respective permutations of $\mathbf{x}^*$ and $\mathbf{y}^*$ in (5.10)-(5.11). From here, we recall from Assumption 10 the permutation invariance of the objective $w(\hat{\mathbf{x}}^*, \hat{\mathbf{y}}^*) = w(\mathbf{x}^*, \mathbf{y}^*) = P^*$. Likewise, the permutation equivariance of the constraint function $\mathbf{c}(\mathbf{x}, \mathbf{y})$ from Assumption 11, the policy $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ satisfies $\mathbf{c}(\mathbf{x}', \mathbf{y}') \geq \mathbf{0}$. Given that $\mathbf{p}' \in \mathcal{P}$ by its definition, the policy $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ is both feasible and achieves the optimal objective value $\mathbf{P}^*$, implying it is an optimal solution to (5.1). As the definition of $\mathbf{p}'(\mathbf{H}, \boldsymbol{\eta})$ is permutation equivariant by construction, this concludes the proof. ■

In Proposition 4, we establish that there exists a solution to (5.1) that is permutation equivariant. We point out that this a very intuitive property to be held by the optimal power allocation, as the labeling of the nodes is generally arbitrary and the structure of the policy should indeed reflect that. Such a structural property not only aids in the learning process by restricting the policy search to a significantly lower dimensional subspace, but is essential for the practical execution of a learned resource allocation policy in wireless systems.



(a)

(b)

Figure 5.2: An illustration of (a) a graph signal colored onto the nodes of graph with grey-colored edges and (b) a graph convolution operation performed on the graph signal.

Indeed, the geometric configuration of the wireless system is not fixed over time, so an effective resource allocation policy should not need to be retrained as the system undergoes small changes. We point out that while FCNNs have the expressive power to represent a permutation equivariant function, they do not hold this property by default. The difficulty of training FCNNs thus lies, in part, in the difficulty of training the layer weights such that a permutation equivariant policy is found. We proceed to discuss a more appropriate neural network architecture for application in wireless resource allocation problems.

Remark 13. The permutation equivariance in Proposition 4 relies on a permutation invariance of the objective $w(\mathbf{x}, \mathbf{y})$ as stated in Assumption 10. It is natural to consider objectives where this assumption does not hold, such as the commonly used weighted sum-rate $w(\mathbf{x}) = \sum_i w_i x_i$. For many of these cases, although $w(\mathbf{x})$ is not permutation invariant with respect to $\mathbf{x}$, it may nonetheless by permutation invariant with respect to some transformation $\omega(\mathbf{x})$. In the case of the weighted sum-rate, such a transformation can be immediately found as $\omega(x_i) := w_i x_i$.

5.3 Graph Neural Networks

Convolutional neural networks (CNNs) are perhaps the most common alternative to FCNNs that have structure-preserving properties for many learning problems. CNNs are relatively low-dimensional NN architectures that replace the full weight matrices of FCNNs with sparse matrices populated by convolutional filters. They are empirically observed to be strikingly effective in many learning tasks ranging from image classification [66] to recommender systems [21]. This is in large part due to the fact that the optimal classification functions in these tasks with certain permutation invariances that are retained by the convolution operations that make up any CNN architecture. The parameter dimension furthermore does not grow with change in input dimension and is significantly less prone to overfitting. Thus, convolutional architectures are a promising direction to pursue for designing policy parameterizations for large scale wireless resource allocation. Standard CNNs perform either spatial or time-series convolutions, making them naturally suited for, e.g., image or video input data, respectively. In this work, we consider convolutional neural network architectures that are better suited to solve problems in wireless networks with arbitrary relational, or graph, structure.

A convolutional neural network with arbitrarily structured data is referred to as a graph neural network (GNN), which perform generalized *graph* convolutional operations at each layer [45, 55]. The GNN architecture is built around a the structure of a given graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with m vertices, or nodes, $\mathcal{V} = \{1, 2, \ldots, m\}$ and the e weighted edges $\mathcal{E} = \{e_{i,j} \mid i, j \text{ connected}\}$ that connect them. The network structure can be compactly encoded in the graph shift operator (GSO) matrix $\mathbf{S} \in \mathbb{R}_{+}^{m \times m}$ whose (i, j)th component $s_{ij} = e_{ij}$ for all edges in $\mathcal{E}$ and 0 otherwise. The graph represents a relationship between elements of a graph signal $\mathbf{z} \in \mathbb{R}^m$, whose component z_i represents the value of the signal at node i. In Figure 5.2(a) we show an example of such a graph. The colors of the vertex nodes represent the value of the signal $\mathbf{z}$ and the grey edges represent the edges connecting nodes. Observe that this graph structure is a generalization of 1-dimensional time signal or 2-d spatial grid signal, i.e. image. For example, a time signal is equivalent to a graph signal on a so-called cycle graph with a GSO matrix

$$\mathbf{S}_{cycle} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{5.16}$$

The graph is a highly relevant structure in the context of wireless systems as it can naturally

encode wireless network. Recall the transmitter/receiver network considered in this work and shown in Figure 5.1. The set of fading links $\mathbf{H}$ between the transmitters and receivers can be used to define a GSO for a graph that represents the wireless network. E.g., a stronger link h_{ij} between transmitter i and receiver j will reflect a higher weighted edge e_{ij} in the graph. Likewise, the transmitter state $\boldsymbol{\eta}$ can be used to represent a corresponding graph signal $\mathbf{z}$ for the graph created by the fading channels. It is thus natural to proceed by considering these graph structures as representations of the wireless network on which we can perform operations and develop learning parameterizations.

As the graph structure generalizes time series signal, so too can the convolution, or filtering, operation be generalized for this arbitrary domain. The graph convolution of input signal $\mathbf{z} \in \mathbb{R}^m$ and graph filter $\boldsymbol{\alpha} \in \mathbb{R}^K$ with respect to the GSO $\mathbf{S}$ is a vector $\hat{\mathbf{w}} \in \mathbb{R}^m$ whose jth component is

$$\hat{w}_j := [\boldsymbol{\alpha} *_{\mathbf{S}} \mathbf{z}]_j := \sum_{k=0}^{K} \alpha_k [\mathbf{S}^k \mathbf{z}]_j. \tag{5.17}$$

In (5.17), the term $\mathbf{S}^k$ shifts the elements of $\mathbf{z}$ in k turns according to the structure defined in $\mathbf{S}$. Observe that the the graph convolution applied to $\mathbf{S}_{cycle}$ indeed recovers the standard cyclic time-series convolution. We illustrate the graph convolution in Figure 5.2(b), where the larger colored circles reflect the summing operations over the neighboring node signals. Note that the size of the larger signals reflects the filter size K—larger circles sum over larger neighborhoods—and the shade reflects the filter coefficients α_k—darker circles sum neighbor signals with higher weight.

Given the graph definition and convolution operation in (5.17), we can define the resulting GNN architecture. A GNN is constructed with a sequence of L so-called hidden layers, each of which contains a set of F_l graph filters of size K_l. The output of layer l is some accumulation of the outputs of its F_l filters, which is subsequently used as an input to layer $l+1$. Denote by $\mathbf{z}_l := [\mathbf{z}_l^1; \mathbf{z}_l^2; \ldots; \mathbf{z}_l^{F_l}] \in \mathbb{R}^{q_l}$as the input to layer l—a concatenation of F_l features of dimension $q_l = mF_l$. At layer l, each feature in $\mathbf{z}_l$ is passed through a graph filter $\boldsymbol{\alpha}_l^{ij} \in \mathbb{R}^{K_l}$ that generates an intermediate vector $\mathbf{u}_{l+1}^{ij}$ from the feature $\mathbf{z}_l^i$ to next layer feature $\mathbf{z}_{l+1}^j$. The complete jth feature at the subsequent layer $l+1$ aggregates all such intermediate inputs from the previous layer and applies the non-linear activation $\boldsymbol{\sigma}_l(\cdot)$as

$$\mathbf{z}_{l+1}^j := \boldsymbol{\sigma}_l \left(\sum_{i=1}^{F_l} \boldsymbol{\alpha}_l^{ij} *_{\mathbf{S}} \mathbf{z}_l^i \right). \tag{5.18}$$

The full signal output from layer l and input to $l+1$ is then constructed as the concatenation of its F_{l+1} layers $\mathbf{z}_{l+1} := [\mathbf{z}_{l+1}^1; \ldots; \mathbf{z}^{F_{l+1}}] \in \mathbb{R}^{q_{l+1}}$, with its dimension $q_{l+1} = mF_{l+1}$.

The full GNN architecture composes the filter operations for each layer $l = 1, \ldots, L$ as in (5.18) to produce an output $\mathbf{z}_L$. Observe that parameters to be learned for a GNN consists of the filter coeffcients at each layer. At layer l, this requires the learning of the filter coefficients $\alpha_l^{ij} \in \mathbb{R}^{K_l}$ for all feature combinations (i, j), for a total of $K_l \times F_l \times F_{l+1}$ coefficients. The total number of parameters then amounts to $q := \sum_{l=1}^{L} K_l \times F_l \times F_{l+1}$. This number may appear large but is often in practice significantly smaller than that of a FCNN in practice, as K_l, F_l, F_{l+1} are themselves small relative to the input dimension. We proceed to discuss the architectural considerations made to utilize the GNN parameterization in wireless resource allocation problems.

5.3.1 Random edge graph neural networks (REGNNs)

The GNN architecture presented in the previous section provides a parameterization with which we may utilize the various inputs $\mathbf{H}$ and $\boldsymbol{\eta}$ of the resource allocation function. In particular, the fading channel state matrix $\mathbf{H}$ can itself be used to represent an underlying graph $\mathcal{G}$ that encodes the state of the wireless network. The nodes $\mathcal{V}$ of such a graph represent each of the m direct transmitter/receiver pairings, while the edge $(i, j) \in \mathcal{E}$ represent the fading channel state between transmitter i and receiver n_j paired with transmitter j. Note that such a graph is both directed and necessarily contains self loops, i.e. the direct fading channel between a transmitter and its paired receiver. The resulting adjacency matrix, or GSO, for graph $\mathcal{G}$ is then given precisely by the link state matrix, i.e. $\mathbf{S} := \mathbf{H}$. As this link state invariably changes randomly over time as a result of wireless fading, so does the edges of the underlying graph $\mathcal{G}$—we refer to this as a *random edge (RE) graph.*

Just as the link states $\mathbf{H}$ are encoded into the graph topology itself via the GSO matrix $\mathbf{S}$, the transmitter states $\boldsymbol{\eta}$ can be subsequently defined as a graph signal whose component for node i is the transmitter state η_i for the ith transmitter. For given states $\mathbf{H}$ and $\boldsymbol{\eta}$, we may define a parametrization in (5.6) as the resulting GNN, i.e.

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{H},\boldsymbol{\eta},\boldsymbol{\theta}) := & \\ & \boldsymbol{\sigma}_L(\boldsymbol{\alpha}_L *_{\mathbf{H}} (\boldsymbol{\sigma}_{L-1}(\boldsymbol{\alpha}_{L-1} *_{\mathbf{H}} (\ldots (\boldsymbol{\sigma}_1(\boldsymbol{\alpha}_1 *_{\mathbf{H}} \boldsymbol{\eta}) \ldots))))), \end{aligned} \tag{5.19}$$

where the parameter $\boldsymbol{\theta}$ contains the L sets of filter weights, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\alpha}_l\}_{l=1}^{L}$ and has dimension $q = \sum_{l=1}^{L} \sum_{l=1}^{L} K_l \times F_l \times F_{l+1}$. Note that in (5.19) only for clarity of presentation we assume all layers have a single feature, i.e. $F_l = 1, l = 1, \ldots, L$. As the GNN used for the parameterization in (5.19) is defined over a random edge graph defined via the fading network, we refer to the resulting learning architecture as a *random edge graph neural network (REGNN).* An REGNN is noted to be distinct from traditional GNN architectures, which receive a random input from a statistical distribution but whose underlying architecture is

fixed. Here, however, the REGNN receives both a random graph signal input $\boldsymbol{\eta} \sim \mu(\boldsymbol{\eta})$ and a random underlying graph GSO $\mathbf{H} \sim m(\mathbf{H})$. The filter weights $\boldsymbol{\theta}$ are trained relative to the statistics of the both the random inputs and the random underlying graphs.

There are a number of immediate advantages of using REGNNs as a parameterization of resource allocation policies. First, as previously noted the parameter dimension q for a REGNN is typically significantly smaller than that of a FCNN, which helps facilitate learning. This is, in part, due to the fact that the REGNN incorporates the channel link inputs $\mathbf{H}$ as part of its underlying architecture rather than as distinct input to the neural network. Thus, the effective input dimension for the REGNN is m rather than $m(m+1)$ as for the FCNN. Second, observe that the parameterization, as an application of graph filters, does not depend upon input dimension—the convolution operation with filters $\boldsymbol{\alpha}_l$ can be performed on signal of any dimension. Thus, same GNN can be be applied to varying size networks. This is not the case in FCNN, whose first layer weights $\mathbf{W}_1$ must have the dimension of the input. The third, and perhaps most important, advantage of REGNN in the context of resource allocation policies concerns its structural properties that match those of $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$

As previously alluded to, the REGNN defined in (5.19) does not retain the universality property of its fully-connected counterpart. This is due to the fact that the dense weight matrices $\mathbf{W}_l$ used to define the lth layer of a FCNN are here replaced with a matrix with a sparse, convolutional structure. However, what we lose in universality we gain in *structure*. That is, in learning the weights of a NN we restrict our attention to a class of graph convolutional matrices that contain certain structural properties. Recall in Proposition 4, we establish a permutation equivariance property help by the optimal allocation policy $\mathbf{p}^*(\mathbf{H}, \boldsymbol{\eta})$. The convolutional structure of the GNN—and its RE counterpart—allows us to establish the same equivariance property. That is, a permutation of the underlying graph and input signal of a GNN will produce an equally permuted output. We present this result formally in the following proposition from [104, Proposition 1].

Proposition 5 ([104]). *Furthermore, consider the REGNN defined in* (5.19). *For any choice of filter weights $\boldsymbol{\theta}$, the function $\boldsymbol{\phi}(\mathbf{H}, \boldsymbol{\theta})$ is also permutation equivariant w.r.t the channel network $\mathbf{H}$ and transmitter state $\boldsymbol{\eta}$, i.e.*

$$\boldsymbol{\phi}(\mathbf{H}, \boldsymbol{\eta}, \boldsymbol{\theta}) = \boldsymbol{\Pi}^T \boldsymbol{\phi}(\boldsymbol{\Pi}^T \mathbf{H} \boldsymbol{\Pi}, \boldsymbol{\theta}). \tag{5.20}$$

Proposition 5 establishes the permutation equivariance of REGNNs. In the context of wireless networks, this implies that a relabelling or reordering of the transmitters in the network will produce an appropriately permutation of the power allocation *without any permutation of the filter weights*. This essential structural property is not satisfied by general FCNNs, in which a restructuring of the network would require an equivalent permutation of

the interlayer weights. This is equivariance implies a certain degree of robustness towards shifts in the input data and is indeed a primary reason that general CNNs have proven significantly more valuable in speech and imaging applications than FCNNs. We proceed with a brief regarding the generalization implications of permutation equivariance.

Remark 14. Permutation equivariance alone doesn't necessarily suggest any robustness towards varying network topologies or increasing network size. However, recent results in [44] demonstrate a *stability* property of GNNs. That is, the distance between the output of GNN defined on some graph $\mathbf{S}$ and the output of the same GNN defined on some graph $\mathbf{S}'$ will be proportionally to the distance between the graphs $\mathbf{S}$ and $\mathbf{S}'$. As wireless networks increase in size, it becomes more likely that various random physical configurations of the transmitters and receivers in an equally dense area will be close to permutations of one another. This suggests a possibility that a GNN trained for one network will work well for another network of similar density. We explore this potential in extensive numerical simulations in Section 5.5.

5.4 Primal-Dual Learning

To find the optimal REGNN weights $\boldsymbol{\theta}$ and associated ergodic variables $\mathbf{x}$ and $\mathbf{y}$ in (5.6), we employ the model-free primal-dual learning method previously developed in Chapter 4. The primal-dual algorithm is a means of applying traditional gradient descent approaches to constrained optimization problems. To derive such an algorithm, consider that we are first interested in transforming the constrained problem in (5.6) to one that is unconstrained. This can be done naively by adding some penalty term to the objective function $w(\mathbf{x}, \mathbf{y})$ that penalizes violation of the constraints. However, such a penalty would require hyperparameter tuning and is thus not ideal. Rather, we form a Lagrangian function via the introduction of so-called dual variables. To form the Lagrangian, we first define for notational convenience the stacked ergodic variables $\boldsymbol{\chi} := [\mathbf{x}; \mathbf{y}] \in \mathbb{R}^{2m}$ and the stacked policy constraint functions, i.e.

$$\boldsymbol{\varphi}(\boldsymbol{\theta}, \boldsymbol{\chi}) := \mathbb{E}\begin{bmatrix}\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{H}, \boldsymbol{\eta}, \boldsymbol{\theta}), \mathbf{H}\big) \\ \mathbf{g}\big(\boldsymbol{\phi}(\mathbf{H}, \boldsymbol{\eta}, \boldsymbol{\theta}), \boldsymbol{\eta}\big)\end{bmatrix} - \boldsymbol{\chi}. \tag{5.21}$$

We introduce the multiplier dual variables $\boldsymbol{\lambda} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^u$ and write the Lagrangian dual problem as

$$\begin{aligned} D(\boldsymbol{\theta},&\boldsymbol{\chi}, \boldsymbol{\lambda}, \mu) := \\ &\min_{\boldsymbol{\lambda}, \mu} \max_{\boldsymbol{\theta}, \boldsymbol{\chi}} \left[w(\boldsymbol{\chi}) + \boldsymbol{\lambda}^T \boldsymbol{\varphi}(\boldsymbol{\theta}, \boldsymbol{\chi}) + \boldsymbol{\mu}^T \mathbf{c}(\boldsymbol{\chi})\right]. \end{aligned} \tag{5.22}$$

The dual problem in (5.22) is an unconstrained saddle point problem that simultaneously maximizes the primal variables $\boldsymbol{\theta}$ and $\boldsymbol{\chi}$ and minimizes the dual variables $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. We may then perform standard gradient-based optimization methods directly on (5.22) to obtain solutions via a so-called *primal-dual* optimization method to find a local stationary point to (5.6). In particular, consider that we successively update both the primal variables $\boldsymbol{\theta}, \boldsymbol{\chi}$ and dual variables $\boldsymbol{\lambda}, \boldsymbol{\mu}$ over an iteration index k. At each iteration we update the current primal iterates $\boldsymbol{\theta}_k, \boldsymbol{\chi}_k$ by adding the corresponding partial gradients of the saddle point problem in (5.22), i.e.,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \nabla_{\boldsymbol{\theta}} \boldsymbol{\varphi}(\boldsymbol{\theta}_k, \boldsymbol{\chi}_k) \boldsymbol{\lambda}_k, \tag{5.23}$$

$$\boldsymbol{\chi}_{k+1} = \boldsymbol{\chi}_k + \beta(\nabla w(\boldsymbol{\chi}_k) - \boldsymbol{\lambda}_k + \nabla \mathbf{c}(\boldsymbol{\chi}_k) \boldsymbol{\mu}_k), \tag{5.24}$$

where we introduce $\alpha, \beta > 0$ as scalar step sizes. In the same manner we descend on the dual variables using the partial gradient of saddle point problem in (5.22), i.e.,

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \gamma \left(\boldsymbol{\varphi}(\boldsymbol{\theta}_{k+1}, \boldsymbol{\chi}_{k+1}) \right), \tag{5.25}$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - \gamma \mathbf{c}(\boldsymbol{\chi}_{k+1}). \tag{5.26}$$

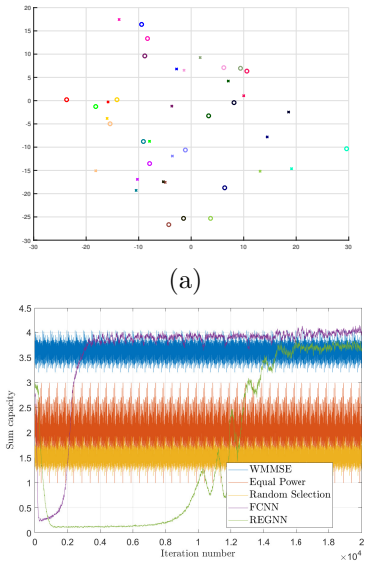
with associated step size $\gamma > 0$. The gradient primal-dual updates in (5.23)-(5.26) successively move the primal and dual variables towards maximum and minimum points of the Lagrangian dual function, respectively.

The combined updates in (5.23)-(5.26) form the primal-dual learning method for finding the parameters of the REGNN while learning the optimal dual parameter $\boldsymbol{\lambda}$ to enforce constraint satisfaction. Recall that this method is *unsupervised* in the sense that, contrary to standard neural network training methods, does not require a supervised training set of solutions to (5.6). Rather, we optimize directly with respect to the capacity function in a manner similar to, e.g., reinforcement learning.

Remark 15. We further point out that evaluating the updates in (5.23)-(5.26) require computing potentially challenging gradients and expectations—in practice, the channel may not even be known explicitly. The gradients in these updates can be replaced with well-known *model free* gradient estimation methods, such as policy gradient [118], that can be obtained with function and evaluations and channel sampling—see Chapter 4 or [38] for details on these approaches.

5.5 Numerical Results

In this section, we provide a numerical study of the performance of resource allocation policies that parameterized with REGNNs and trained with the model-free primal-dual learning



(a)

(b)

Figure 5.3: Performance comparison during training of REGNN for $m = 20$ pairs. With only $q = 40$ parameters, the REGNN strongly outperforms the WMMSE algorithm.The network is plotted in top figure, and the achieved sum-rate over the learning process is shown in the bottom figure.

method. We simulate the performance of the policy on a number of canonical resource allocation functions that take the form of (5.1) and compare against existing heuristic approaches. Where applicable, we point out the compared heuristics that rely on accurate model knowledge to be implemented, which as discussed in Section **??**, is *not* required to implement the primal-dual learning method. For all sets of simulations, we construct the wireless network illustrated in Figure 5.1 as follows. For a set of m pairs, we construct a random geometric graph by dropping transmitter i uniformly at random at location $\mathbf{t}_i \in [-m, m]^2$, with its paired receiver at location $\mathbf{r}_i \in [\mathbf{t}_i - m/4, \mathbf{t}_i + m/4]^2$ around its paired transmitter—see, e.g., Figure 5.3(a) for an example. Given the geometric placements, the complete fading channel state between transmitter i and receiver j is composed of $h_{ij} = h^p_{ij} h^f_{ij}$, where h^p_{ij} is a constant path-loss gain and h^f_{ij} is the time varying fast fading. The path loss is related to the geometric distance as $h^p_{ij} = \|\mathbf{t}_i - \mathbf{r}_j\|^{-2.2}$ and the fast fading h^f_{ij} is drawn randomly from a standard Rayleigh distribution at each scheduling cycle.

5.5.1 Binary power control

The first problem we study is the canonical problem of binary power control between m transmitter/receiver pairs over an AWGN channel with interference—see Example 5 for a discussion of this problem. In addition to maximizing the sum-rate capacity, a practical constraint of interest is a maximum average power budget $P_{\max}$ to be shared between

transmitters connected to a common power supply. The complete resource allocation problem can be written as

$$P^* := \max_{\mathbf{p}(\mathbf{H}),\mathbf{x}} \sum_{i=1}^{m} x_i, \tag{5.27}$$
$$\text{s.t. } x_i \leq \log\left(1 + \frac{|h_{ii}|^2 p_i(\mathbf{H})}{1 + \sum_{j\neq i} |h_{ji}|^2 p_j(\mathbf{H})}\right),$$
$$\mathbb{E}_{\mathbf{H}}\left[\mathbf{1}^T \mathbf{p}(\mathbf{H})\right] \leq P_{\max}, \quad \mathbf{p}(\mathbf{H}, \boldsymbol{\eta}) \in \{0, p_0\}^m.$$

As discussed in Example 5, the problem in (5.27) does not utilize any transmitter state $\boldsymbol{\eta}$ and associated cost constraint. This is however an instructive problem to study, as it is well studied and has numerous developed heuristic solutions with which the compare as baselines. Observe also that the power allocation is a binary selection of transmitting with power p_0 or not transmitting.

In employing the primal dual learning method in (5.23)-(5.26), we consider the model free version in which gradients are estimated via the policy gradient approximation. We construct a REGNN architecture with $L = 8$ hidden layers, each with $F_l = 1$ graph filters of length $K_l = 5$ and a standard ReLu non-linear activation function i.e. $\boldsymbol{\sigma}(\mathbf{z}) = [\mathbf{z}]_+$. The final layer is passed through a sigmoid function to normalize the outputs, which are then used as the parameter of a Bernoulli policy distribution (random policies are a necessary component of policy gradient computation—see [38]). The primal dual method is performed with a geometrically decaying step size for dual updates and the ADAM optimizer [65] for the primal updates.

In general, we make our comparisons against existing heuristic methods for solving (5.27). We primarily consider (i) the popular WMMSE heuristic [111] as a baseline, while also making comparisons against naive heuristics that either (ii) assign equal power $\bar{P}/m$ to all users or (iii) randomly select $\bar{P}/p_{\max}$ users to transmit with full power. Furthermore, we simulate the learning and performance of the convolutional REGNN architecture to a fully connected neural network (FCNN) for medium scale networks. In the top of Fig. 5.3, we show the geometric configuration of the network. The paired transmitter and receiver are shown in a cross and circle, respectively, with a matching color. In the bottom figure, we show the performance, or sum-capacity, achieved throughout the learning process of the REGNN and FCNN trained with the primal-dual learning method and the performance of the three heuristic baselines for a medium scale wireless system with $m = 20$ pairs. It can be observed that both the REGNN and FCNN narrowly outperform the performance of WMMSE for the medium scale system. We stress that this matching performance was obtained by the NNs using the model-free gradients, meaning that knowledge of the capacity function was not assumed. Explicit knowledge of capacity functions is needed, however, for

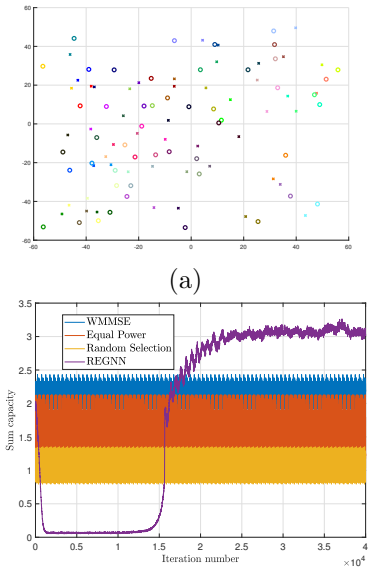
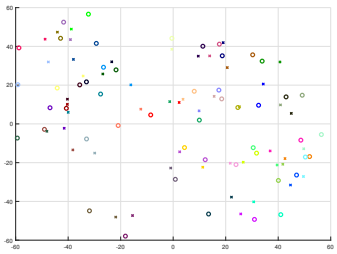


Figure 5.4: Performance comparison during training of REGNN for $m = 50$ pairs. With only $q = 40$ parameters, the REGNN strongly outperforms the WMMSE algorithm. The network is plotted in top figure, and the achieved sum-rate over the learning process is shown in the bottom figure.

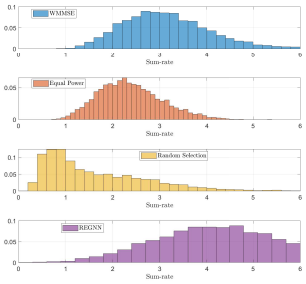
the WMMSE algorithm. We also point out that the REGNN, with only $q = 40$ parameters, matched the performance of the FCNN with two fully connected layers of size 64 and 32 for a total of $q = 20 \times 64 + 64 \times 32 + 32 \times 20 \approx 4000$ parameters—a 100 factor increase than that used by the REGNN.

In Figure 5.4, we show the performance while learning a REGNN in a larger scale system with $m = 50$ transmitter/receiver pairs. At this scale, the parameter dimension of the FCNN makes it challenging to train; the input dimension of channel states is 2500. Here, we see that the learned REGNN substantially outperforms all three heuristics, including WMMSE. Observe that while WMMSE achieves a sum-capacity of roughly 3.7 in the medium scale system, the algorithm performs even worse when more transmitters are added, obtaining a sum-rate of only 2.3. The REGNN, meanwhile, is able to still achieve a sum-rate of 3.1, all while only learning 40 parameters.

As previously alluded to in Remark 14, we are interested in exploring the generalization abilities of an REGNN learned over some fixed network. Recall that the filter-bank structure of an REGNN in (5.19) allows the same neural network to receive inputs of varying input dimension, or network size. Consider the REGNN leaned in the previous experiment in Figure 5.4. As an instructive example, consider another randomly drawn network of 50 pairs as shown in Figure 5.5. The performance of the REGNN trained in Figure 5.4 over many random iterations is shown here compared to the heuristics as an empirical histogram of sum-rates over all random iterations. We see that the same parameterization learned for



(a)



(b)

Figure 5.5: Performance comparison of an REGNN that was trained in Figure 5.4 in another randomly drawn network of equal size. The top figure plots the geometric configuration of the new network. The bottom figure shows the empirical distribution of the sum-rate achieved over many random iterations for all heuristic methods.

one network performs well with another network. Intuitively speaking, this relates to the stability and permutation equivariance of GNNs because random networks of size 50 may be close to each other in expectation.

Another comparison of interest here is the relative performance of a REGNN trained on a network of size $m = 50$ with an REGNN trained on a network of $m' > m$. In Figure 5.6 we show a histogram of the sum-rate performance over 50 randomly generated networks of size $m' = 75$ and $m' = 100$. For a set of m' pairs, we construct a random geometric graph by dropping transmitter i uniformly at random at location $\mathbf{t}_i \in [-m\sqrt{m'/m}, m\sqrt{m'/m}]^2$, with its paired receiver at location $\mathbf{r}_i \in [\mathbf{t}_i - m/4, \mathbf{t}_i + m/4]^2$ around its paired transmitter. This is done to keep the density of the network constant as the number of transceiver pairs grows.

The performance for each random network is itself evaluated over 100 separate fast fading samples. As can be seen, the performance of the REGNN trained on the smaller network of size $m = 50$ almost matches the performance of an REGNN trained on a network of size 75. The same procedure is performed for networks of size $m' = 100$. In Figure 5.7, we show the performance of the REGNN trained on a network of size 50 against the performance of a network trained on a network of size 100 on random networks of size 100. Again, the performance of the REGNN trained on the smaller network only slightly degrades relative

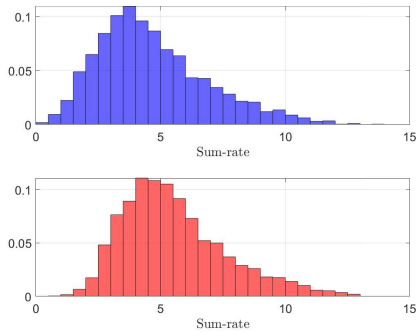


Figure 5.6: Empirical histogram of sum rates obtained by (top, blue) REGNN trained on network of size $m = 50$ and (bottom, red) REGNN trained on network of size $m' = 75$ on 50 randomly drawn networks of size of $m' = 75$. The REGNN trained on the smaller network closely matches the performance of the REGNN trained on the larger network

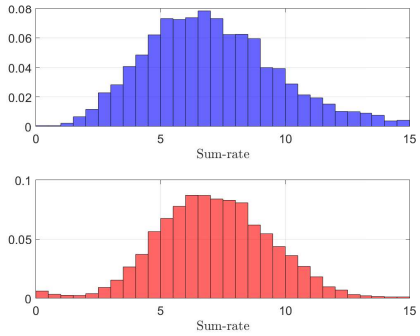


Figure 5.7: Empirical histogram of sum rates obtained by (top, blue) REGNN trained on network of size $m = 50$ and (bottom, red) REGNN trained on network of size $m' = 100$ on 50 randomly drawn networks of size of $m' = 100$. The REGNN trained on the smaller network closely matches the performance of the REGNN trained on the larger network

to the REGNN trained on the larger network. This highlights a potential to train REGNNs on smaller networks to later be implemented on larger networks. We point out that this is a powerful property for practical learning for such systems, as we can potentially train our neural networks on smaller systems when larger networks are either unavailable during training or when computational expense is prohibitive.

To fully explore these capabilities for increasingly large networks, we again use the REGNN trained in Figure 5.4 in random wireless networks of increasing size. Note that, as we increase the size of the networks, the density of the network remains constant so that the statistics of the channel conditions are the same. In Figure 5.8, we show the average sum-rate achieved by the REGNN over many random iterations for networks of increasing size m', where the geometric configurations generated using the fixed-density random geometric graph as done previously. We observe that, even as the network size increases, the same

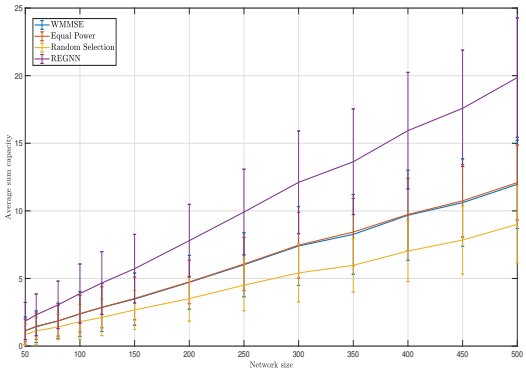


Figure 5.8: Performance of REGNN trained in Figure 5.4 in randomly drawn networks of varying size. From networks of size $m' = 50$ to 500, the REGNN is able to outperform the heuristic methods.

REGNN is able to outperform the heuristic methods.

As a final numerical study for the pairwise network, we compare the performance of the REGNN trained on a fixed network of size $m = 50$ in new random networks of equivalent number of pairs but varying density. In these experiments, we draw random geometric graphs with some density factor r by dropping transmitter i uniformly at random at location $\mathbf{t}_i \in [-r^{-1}m\sqrt{m'/m}, r^{-1}m\sqrt{m'/m}]^2$, with its paired receiver at location $\mathbf{r}_i \in [\mathbf{t}_i - m/4, \mathbf{t}_i + m/4]^2$ around its paired transmitter. In this manner, as the density factor r increases, the physical space of the network gets smaller and thus more dense. In Figure 5.11, we show the average sum-rate achieved by the REGNN over many random iterations for networks of increasing densities r. We observe that, for wireless networks of equal or less density than the one used for training, the REGNN has strong performance relative to the heuristics. However, as the networks more dense, the REGNN is unable to match the performance of WMMSE. This results follows from the fact that the statistics of data seen in training begins to vary more and more from that seen in execution time as the networks increase in density. Indeed, as the transmitters become closer together, the path-loss component of the fading state decreases and the interference grows.

5.5.2 Multi-cell interference network

In this section, we consider a variation of the network architecture previously considered known as a multi-cell interference network. In contrast to the pair-wise setting, in the mutli-cell network there exist n receivers—or cellular base stations—who service the transmissions of a total m cellular users, which we assume are distributed evenly amongst the base stations. An example of such a multi-cell configuration is provided in the top of Figure 5.10 for $n = 5$ base stations, marked with circles, covering $m = 50$ cellular users, marked with crosses. The settings here is instructive not only in the real world practicality of its setting, but in its tendency to scale largely as the number of base stations or number of users grows.

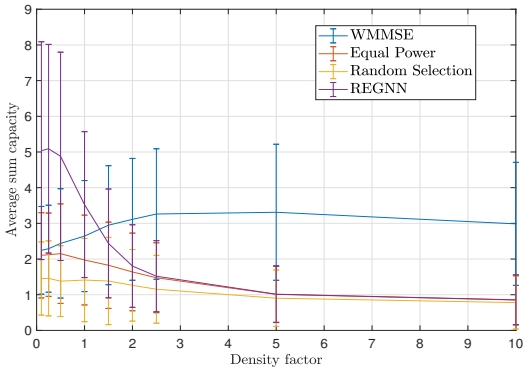
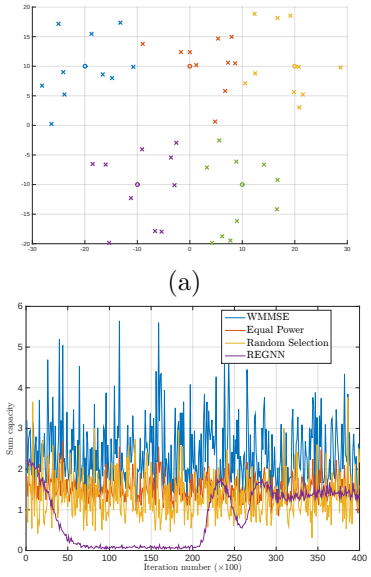


Figure 5.9: Performance of REGNN trained in Figure 5.4 in randomly drawn networks of varying densities from factors ranging from $r = 0.1$ to 10. As the density of the network increases, the REGNN is unable to match the performance of the WMMSE algorithm.



(a)

(b)

Figure 5.10: Performance comparison during training of REGNN for multi-cell interference network with $m = 50$ users and $n = 5$ base stations. With $q = 40$ parameters, the REGNN outperforms the model-free heuristics but does not quite meet the performance of WMMSE, which utilizes model information in its implementation. The top figure plots the geometric configuration of the network, and the bottom figure plots the sum-rate over learning iterations.

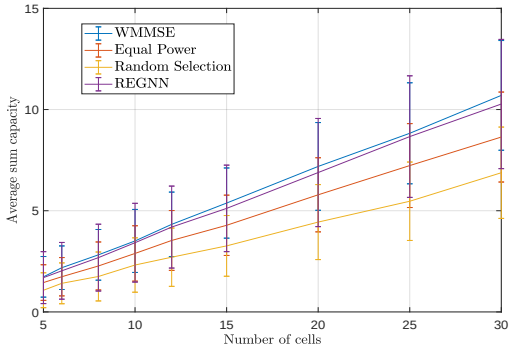


Figure 5.11: Performance of REGNN trained in Figure 5.10 in randomly drawn multo-cell networks of varying size. From networks of size 5 to 50 cells, the REGNN matches the performance of the best performing heuristic method.

In the bottom of Figure 5.10, we show the performance obtained by the REGNN during training compared to the heuristic methods. Here we see that the REGNN almost meets the performance of WMMSE. It is interesting to note that the performance of the REGNN degrades in the multi-cell network relative to its performance in the pairwise network previously considered. This may, in part, be attributed to the fact that the underlying graph for this problem is not as distinctive or informative in the pairwise network. That is, there are only $n = 5$ unique receivers, and thus much of the interference patterns will be the same for different transmitters. Another way to say this is that the GSO matrix $\mathbf{S} = \mathbf{H}$ will contain repeated rows, and is subsequently low-rank $rank(\mathbf{S}) = n < m$.

5.5.3 Wireless sensor networks

We proceed to perform simulations an extension to the binary control problem in the pairwise network studied in the previous section—namely binary power control with data collection. The problem was previously discussed in Example 6. For this setting, we assume that each transmitter additionally maintains a local state η_i that reflects an arrival or collection rate of data to be transmitted. Such a problem is highly relevant in, e.g. sensor networks and robotics applications. The resource allocation problem is closely related to that in (5.27), but with an additional constraint the ergodic capacity achieved by each transmitter must exceed the average collection rate of its associated sensor. We may write this problem as

$$\begin{aligned} P^* := \max_{\mathbf{p}(\mathbf{H}),\mathbf{x}} \; & \sum_{i=1}^{m} x_i, \\ \text{s.t. } \; & x_i \leq \log\left(1 + \frac{|h_{ii}|^2 p_i(\mathbf{H})}{1 + \sum_{j \neq i} |h_{ji}|^2 p_j(\mathbf{H})}\right), \\ & \mathbf{x} \geq \mathbb{E}[\boldsymbol{\eta}], \quad \mathbf{p}(\mathbf{H}, \boldsymbol{\eta}) \in \{0, p_0\}^m. \end{aligned} \tag{5.28}$$

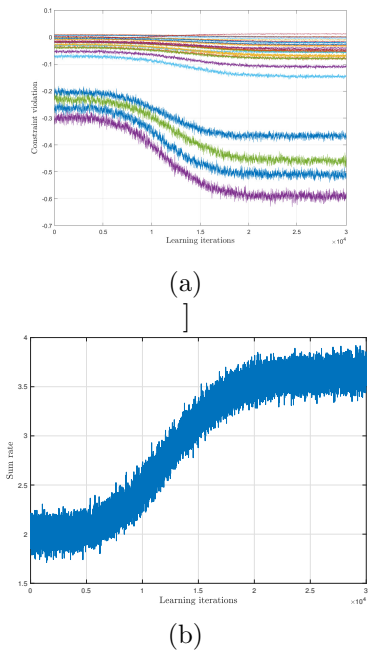


Figure 5.12: Convergence of training of an REGNN for the wireless sensor network problem with $m = 30$ sensors/transmitters. In the top figure, we show the constraint violation for each of the transmitters converges to a feasible solution. In the bottom figure, we show the objective function converge to a local maximum.

The problem in (5.28) contains an additional complexity in the ergodic constraint, which must be independently satisfied by each transmitter. The optimal resource allocation policy is which obtains sufficient capacity is achieved by each transceiver pair in expectation, while then maximizing the sum-rate achieved over the network.

We perform the primal-dual learning method to train a REGNN in a system with $m = 30$ transmitter/receiver pairs, who are placed randomly as in previous simulations. The collection rates $\boldsymbol{\eta}$ are drawn from a exponential distribution with mean 0.05. We train a REGNN with $L = 10$ layers, each with $F_l = 1$ filters of length $K_l = 5$. In Figure 5.12 we show the performance of the learning procedure. In Figure 5.12(a), we show the constraint violation for all 30 sensors. with a negative value of $\mathbb{E}[\boldsymbol{\eta}] - \mathbf{x}$ signifying a satisfaction of the capacity constraint, i.e. the sensor is transmitting data faster that it collects. While we may observe a wide variance in the capacities achieved by different sensors, a close examination of Figure 5.12(a) shows that all but 1 sensor achieves constraint satisfaction (constraint satisfaction can be seen with a negative constraint violation value). Likewise, in Figure 5.12(b), we show the REGNN-based resource allocation policy achieve an overall performance converges to local optimum as the rate constraints are being satisfied.

5.6 Conclusion

We considered the problem of optimal power allocation in a large scale wireless fading channel with interference. We parameterize the power allocation with a deep neural network. Because fully connected architectures are unsuited for large scale problems, we consider a graph convolutional architecture with only a small number of parameters. In wireless problems, the graph used to define the architecture is randomly varying due to the fast fading phenomenon, leading us to consider random edge graph neural networks (REGNNs). Such architectures retain an important property of permutation equivariance that is held by the optimal power allocation policy. Using a model free primal dual learning algorithm to train the parameters, numerical simulations show the advantages of the proposed learning method and NN architecture in comparison to existing baseline heuristics.

Chapter 6

Low-Latency Wireless Control via Primal-Dual Learning

6.1 Introduction

The recent advances in wireless technology and automation have given rise to efforts in integrating wireless communications in autonomous environments, particularly in industrial control and Tactile Internet settings where the scale of wired networks is proving increasingly costly [7]. The analysis of control systems operating over wireless communication links is thus an integral apart in enabling these wireless industrial automation applications. However, the performance specifications of Tactile Internet applications demands the design of wireless networks that can meet both the high reliability and low latency demands of the system [7, 10, 97]. Ultra reliable low latency communications (URLLC) is inherently challenging as the physical medium of wireless communication trades off reliability and latency, making it hard to meet both demands.

One promising direction in enabling low latency communications involves specific developments in radio resource allocation, or scheduling. For low latency applications, traditional delay-aware schedulers [4, 78, 129] have been employed, in addition to more recent URLLC techniques based on various forms of diversity [6, 97, 119]—all of which are agnostic to the control system. However, due to the physical limitations of the wireless channel, it is often necessary to use information from the control system to make proper use of scheduling resources in meeting latency requirements. While there exist numerous ways in which control system information is incorporated into "control-aware" scheduling methods [17,27,47,53,71], these are agnostic to latency requirements of the system. More recent work [36] looks at heuristic based scheduling methods that are both control and latency aware, but whose practical use in low latency systems is limited both by its computational complexity at every scheduling cycle and reliance on explicit knowledge of the communication model and control

dynamics.

Aside from traditional heuristic based scheduling methods, machine learning approaches can be incorporated into making intelligent scheduling and resource allocation decisions in wireless control systems without requiring model knowledge. The work in [38] builds a framework for solving a generic set of resource allocation problems by interpreting resource allocation as a constrained statistical learning problem. This leads to a natural use of learning models, such as deep neural networks (DNNs), for designing schedulers. Recent advancements apply techniques from both reinforcement learning and deep learning for control-aware scheduling in simple systems [27, 71]. Learning-based scheduling policies are well suited for URLLC as the computational complexity at each scheduling round is very low and can furthermore be implemented model-free. Our contributions namely consist of

1. formulating a statistical learning problem for control-aware, low latency scheduling,
2. parameterizing the scheduling policy with a deep neural network (DNN), and
3. utilizing the *model-free*, primal-dual learning framework of [38] to find control-aware scheduling policies.

This paper is organized as follows. We discuss the wireless control system in which state information is communicated to the control over a wireless channel as a switched dynamical system (Section 6.2). We formulate the optimal scheduling problem that minimizes a control cost under latency constraints (Section 6.2.1) and parameterize the optimal policy with a deep neural network (Section 6.2.2). The constrained learning problem is solved using a so-called primal-dual learning method (Section 6.3). We further discuss ways in which the primal-dual method can be approximated without explicit model knowledge (Section 6.3.1). The performance of the learned control-aware scheduling method is analyzed in a numerical simulation and compared other heuristic scheduling methods (Section 6.4).

6.2 Wireless Control Systems

We consider a series of m control systems—each a wireless device or plant—operating over a shared wireless channel as shown in Figure 6.1. The state of system i at control cycle index k is given by the variable $\mathbf{x}_i^k \in \mathbb{R}^p$. At each control/scheduling cycle, the sensor measures the state $\mathbf{x}_i^k$ and transmits it over a wireless channel to a common base station (BS) that is co-located with the controller. Given the state information, the controller determines the necessary control input which is fed back to the system. This is referred to as the closed-loop configuration of the control cycle. Given the noisy nature of the wireless channel, there is the potential for the communications packet containing the state information to be

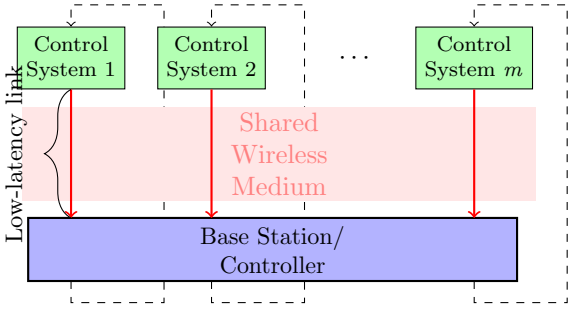


Figure 6.1: A series of independent wireless control systems send state information over a shared wireless medium to a base station, where control information is fed back to the systems. The uplink transmissions (red arrow) is subject to latency constraint $t_{\max}$.

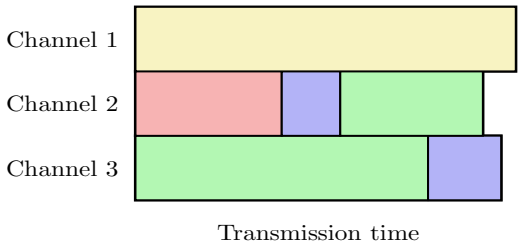


Figure 6.2: Scheduling architecture of $m = 4$ users—colored in green, blue, red, and yellow—across $n = 3$ channels. The total transmission length varies across channels. The channels need not be located on consecutive frequency bands, and are placed as such here only for the purposes of illustration.

dropped, resulting in an open-loop configuration of the control cycle. We may model the linear dynamics of the wireless control system for system i as

$$\mathbf{x}_i^{k+1} = \begin{cases} \hat{\mathbf{A}}_i \mathbf{x}_i^k + \mathbf{w}^k & \text{if packet received} \\ \mathring{\mathbf{A}}_i \mathbf{x}_i^k + \mathbf{w}^k & \text{otherwise} \end{cases}, \tag{6.1}$$

where $\hat{\mathbf{A}}_i \in \mathbb{R}^{p \times p}$ is the closed loop gain, $\mathring{\mathbf{A}}_i \in \mathbb{R}^{p \times p}$ is the open loop gain, and $\mathbf{w}^k \in \mathbb{R}^p$ is zero-mean i.i.d. disturbance process with covariance $\mathbf{W}$. The closed loop and open loop gains may reflect, e.g., controlled dynamics using accurate and estimated state information, respectively. We assume that the closed loop gains are preferable to the open loop gain, i.e. $\lambda_{\max}(\hat{\mathbf{A}}_i) < \lambda_{\max}(\mathring{\mathbf{A}}_i)$. Further note this model restricts its attention to wireless connections in uplink of the control loop, while downlink is assumed to occur over an ideal channel—i.e. no packet drops.

Given this dynamical model of the wireless control systems, the communications goal is to allocate radio resources among the various systems to maintain strong performance across all the systems. To do so, we present a generic frequency and time division multiplexing scheduling architecture with which the BS allocates scheduling resources to the systems. A

scheduling window occupies the uplink of a single cycle in the control loop in which each system has a single packet containing state information to transmit. For URLLC systems, the total length of this scheduling window is subject to a tight low-latency bound $t_{\max}$.

We assume that transmissions are scheduled by the BS across n available channels occupying different (possibly non-consecutive) frequency bands. Each channel is subject to continuous time division multiple access (TDMA), meaning that multiple transmissions in the same channel will occur in sequence. For full generality, we assume that a single device may be scheduled in multiple channels in a single cycle to add redundancy and improve chance of success. Denote by $\boldsymbol{\varsigma}_i \in \{0,1\}^n$ a binary vector whose jth element $\varsigma_{i,j}$ is 1 if the ith device transmits in the jth channel, and 0 otherwise. Further denote for each device a data rate selection $\mu_i \in [\mu_{\min}, \mu_{\max}]$. These two scheduling parameters together define the scheduling decision made for the ith system. An illustration of $m = 4$ users making multiple transmission across $n = 3$ channels is shown in Figure 6.2.

The achieved communications performance by a given scheduling decision can be formulated as follows. We first define $\mathbf{h}_i^k \in \mathbb{R}_+^n$ to be the set of fading channel states experienced by device i at cycle k, where the j element $h_{i,j}^k$ is the fading channel gain in channel j. We assume that these channel conditions do not change over the course of a scheduling window. In any given channel with fading state $\mathbf{h}$, we define a function $q(h, \mu)$ that returns the packet delivery rate (PDR), or the probability of successful transmission of the packet, when transmitting with data rate μ. Likewise, we define a function $\tau(\mu)$ that returns the transmission time to transmit a packet of fixed length with data rate μ. These two functions play a critical role in designing low-latency wireless control systems, as they allow us to explore the trade-off between PDR and transmission time and the resulting effect on control system performance. We may consider that the functions $q(h, \mu)$ and $\tau(\mu)$ both get smaller as we increase data rate μ, i.e.

$$\mu' > \mu \implies q(h, \mu) \leq q(h, \mu'), \quad \tau(\mu') \leq \tau(\mu). \tag{6.2}$$

Thus, by increasing the data rate we may reduce the transmission time to satisfy latency constraints, but at the cost of control system performance, as illustrated by the switched dynamics in (6.1).

Remark 16. *The communication architecture utilized here has a generic form that assumes both continuous time division and simultaneous transmission in independent, unsynchronized channels. We present the architecture in this form both for the purposes of a more tractable mathematical model as well as its generalization of the architectures used in, e.g., Bluetooth or centralized scheduled WiFi. Note that common OFDMA architectures, such as 5G [3] and next-generation WiFi IEEE 802.11ax [75], do not conform precisely to this architecture although it can be adapted as such with slight modifications. We leave the consideration of a*

synchronized, OFDMA architecture as a point of future work.

6.2.1 Optimal scheduling design

We are interested in designing scheduling policies that optimize control performance, subject to the strict low latency constraints of the system. To do so, we first formulate the global control-based performance given a scheduling decision. Collect in the matrix $\boldsymbol{\Sigma} \in \{0,1\}^{n\times m}$ all of the channel transmission vectors $\boldsymbol{\varsigma}_i$ for $i = 1,\dots,m$ and collect in the vector $\boldsymbol{\mu} \in [\mu_{\min}, \mu_{\max}]^m$ the data rates μ_i for $i = 1,\dots,m$. Given that a device may transmit in multiple channels within a single scheduling cycle, the probability of successful transmission can be given as the probability that the transmission was successful in at least one channel, i.e.

$$\tilde{q}(\mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i) := 1 - \prod_{j=1}^{n} \left(1 - \varsigma_{i,j} q(h_{i,j}, \mu_i)\right). \tag{6.3}$$

The total delivery rate in (6.3) can be viewed as the probability of receiving the packet and experiencing the closed loop dynamics in (6.1). Now, to evaluate the performance of a given system at a particular state $\mathbf{x}$, define a quadratic Lyapunov function $L_i(\mathbf{x}) := \mathbf{x}^T\mathbf{P}_i\mathbf{x}$ with some positive definite matrix $\mathbf{P}_i \in \mathbb{R}^{p\times p}$. Such a function can be used to evaluate performance or stability of the control system. Because the control system evolves in a random manner, the cost of a given scheduling decision $\{\boldsymbol{\varsigma}_i, \mu_i\}$ for the ith system can be formulated as the *expected future Lyapunov cost* under such a schedule. As the probability of closing the loop in (6.1) is given by $\tilde{q}(\mathbf{h}_i^k, \boldsymbol{\varsigma}_i, \mu_i)$, we may write this expected future cost as

$$\begin{aligned} J_i(\mathbf{x}_i, \mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i) := \;& \mathbb{E}\left[L_i(\mathbf{x}_i^{k+1}) \mid \mathbf{x}_i^k = \mathbf{x}_i, \mathbf{h}_i^k = \mathbf{h}_i\right] \\ =\;& \tilde{q}(\mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i)(\hat{\mathbf{A}}_i\mathbf{x}_i)^T\mathbf{P}_i(\hat{\mathbf{A}}_i\mathbf{x}_i) \quad + \\ & (1 - \tilde{q}(\mathbf{h}_i, \boldsymbol{\varsigma}_i, \mu_i))(\mathring{\mathbf{A}}_i\boldsymbol{\xi})^T\mathbf{P}_i(\mathring{\mathbf{A}}_i\mathbf{x}_i) \\ +\;& \operatorname{Tr}(\mathbf{P}_i\mathbf{W}_i). \end{aligned} \tag{6.4}$$

Observe that the local control cost for the ith system $J_i(\mathbf{x}_i^k, \mathbf{h}_i^k, \boldsymbol{\varsigma}_i, \mu_i)$ is a function of both the system *states*—the fading channel $\mathbf{h}_i^k$ and control state $\mathbf{x}_i^k$—and the scheduler *actions*—channel selection $\boldsymbol{\varsigma}_i$ and data rate μ_i. The objective is to choose the actions $\boldsymbol{\varsigma}_i$ and μ_i that minimizes the cost relative to states $\mathbf{h}_i^k$ and $\mathbf{x}_i^k$.

In addition to minimizing a control cost, we must make scheduling decisions that respect the low-latency requirements of the system. To formulate this constraint, consider the *total* time of a global scheduling decision $\boldsymbol{\Sigma}, \boldsymbol{\mu}$ of channel j as the sum of all active transmissions, i.e.

$$\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) := \sum_{i=1}^{m} \varsigma_{i,j}\tau(\mu_i). \tag{6.5}$$

Combining all the local costs for systems $i = 1, \ldots, m$ in (6.4) with the a constraint on the latency costs for all channels $j = 1, \ldots, n$ in (6.5), we may define the optimal scheduling design problem. Because we are interested in long-term, or average, performance across random channels and control states, we optimize with respect to expected costs and probabilistic constraints. Collect all channel vectors $\mathbf{h}_i$ in a matrix $\mathbf{H} \in \mathbb{R}_+^{n \times m}$ and states $\mathbf{x}_i$ in a matrix $\mathbf{X} \in \mathbb{R}^{p \times n}$. Consider a scheduling policy $\mathbf{p}(\mathbf{H}, \mathbf{X}) := \{\boldsymbol{\Sigma}, \boldsymbol{\mu}\}$ that, given a set of channel states $\mathbf{H}$ and control states $\mathbf{X}$, returns a schedule defined by the channel selection matrix $\boldsymbol{\Sigma}$ and data rate selection vector $\boldsymbol{\mu}$. The optimal low-latency constrained scheduling policy for the wireless control systems is the one which solves the program

$$\begin{aligned} J^* := \min_{\mathbf{p}(\mathbf{H},\mathbf{X})} \mathbb{E}_{\mathbf{H},\mathbf{X}} \left[\sum_{i=1}^{m} J_i(\mathbf{x}_i, \mathbf{h}_i, \varsigma_i, \mu_i) \right], \\ \text{s. t.} \quad \mathbf{P}_{\mathbf{H},\mathbf{X}} \left(\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \leq t_{\max} \right) \geq 1 - \delta \quad j = 1 \ldots, n, \\ \mathbf{p}(\mathbf{H}, \mathbf{X}) := \{\boldsymbol{\Sigma} \in \{0,1\}^{n \times m}, \boldsymbol{\mu} \in [\mu_{\min}, \mu_{\max}]^m\}. \end{aligned} \tag{6.6}$$

In (6.6), we minimize the average cost over the distribution of channel and control states, subject to the condition that the probability of violating the latency constraint over the distribution of states is less than some small value δ. Because each channel's transmission time varies, we impose this constraint independently for *each channel*. The above scheduling problem can be viewed as a constrained statistical learning problem—a connection made for a more generic class of resource allocation problems in [38]. While such a problem characterizes the optimal scheduling decision for the latency-constraint wireless control system, finding solutions to such a problem is a significant challenge. This is due to a number of complexities in (6.6), namely: (i) it requires functional optimization, (ii) it contains explicit constraints, and (iii) we typically do not have analytic forms for the functions and distributions in (6.6). The first of these complexities can be resolved using a standard technique in statistical learning, discussed next in Section 6.2.2. The latter two of these complexities are discussed and resolved later in Sections 6.3 and 6.3.1, respectively.

6.2.2 Deep learning parameterization

The scheduling problem in (6.6) is computationally challenging because it requires finding a policy—or *function*—$\mathbf{p}(\mathbf{H}, \mathbf{X})$. In statistical learning, or regression, problems the regression function is replaced by some given parameterization $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$ that is defined with some finite dimensional parameter $\boldsymbol{\theta} \in \mathbb{R}^q$. There exist a wide variety of choices of this parameterization, but in modern machine learning problems the *deep neural network (DNN)* is commonly employed. This is due to the fact the DNN can be shown both empirically and analytically to contain strong representative power and generalization ability, meaning that

it can approximate almost any function well. A DNN is defined as a composition of L layers, each of which consisting of a linear operation followed by a point-wise nonlinearity—also known as an activation function. More specifically, the layer l is defined by the linear operation $\mathbf{W}_l \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\boldsymbol{\sigma}_l : \mathbb{R}^{q_l} \to \mathbb{R}^{q_l}$. Common choices of activation functions $\boldsymbol{\sigma}_l$ include a sigmoid function or a rectifier function (commonly referred to as ReLu). Given an input from the $l-1$ layer $\mathbf{w}_{l-1} \in \mathbb{R}^{q_{l-1}}$, the resulting output $\mathbf{w}_l \in \mathbb{R}^{q_l}$ is then computed as $\mathbf{w}_l := \boldsymbol{\sigma}_l(\mathbf{W}_l \mathbf{w}_{l-1})$. The full DNN-parameterization of the scheduling policy is then defined as an L-layer DNN whose input at the initial layer is the concatenation of states $\mathbf{w}_0 := [\mathrm{vec}(\mathbf{H}); \mathrm{vec}(\mathbf{X})]$, i.e.

$$\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta}) := \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{w}_0)))))). \tag{6.7}$$

The parameter vector $\boldsymbol{\theta} \in \mathbb{R}^q$ that defines the DNN is then the entries of $\{\mathbf{W}_l\}_{l=1}^L$ with $q = \sum_{l=1}^{L-1} q_l q_{l+1}$. Further note that we can easily construct an activation function at the final layer $\boldsymbol{\sigma}_L$—or the *output layer*—such that the outputs $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$ are in the space $\{0,1\}^{n \times m} \times [\mu_{\min}, \mu_{\max}]$ that contains possible schedules. With this DNN parameterization, the control-aware scheduling problem can be rewritten as

$$\begin{aligned} J^*_{\boldsymbol{\phi}} := \min_{\boldsymbol{\theta} \in \mathbb{R}^q} \mathbb{E}_{\mathbf{H},\mathbf{X}} \left[\sum_{i=1}^m J_i(\mathbf{x}_i, \mathbf{h}_i, \boldsymbol{\sigma}_i, \mu_i) \right], \\ \text{s.t.} \quad \mathbf{P}_{\mathbf{H},\mathbf{X}} \left(\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \leq t_{\max} \right) \geq 1 - \delta \qquad \forall j, \\ \boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta}) := \{ \boldsymbol{\Sigma} \in \{0,1\}^{n \times m}, \boldsymbol{\mu} \in [\mu_{\min}, \mu_{\max}]^m \}. \end{aligned} \tag{6.8}$$

Observe in (6.8) that the optimization is performed over $\boldsymbol{\theta}$ rather than the scheduling policy directly. In other words, we look for the interlayer weights that define a DNN that minimizers the total control cost while satisfying the latency constraints. We proceed then to discuss a learning method that can find solutions to the constrained optimization problem in (6.8).

6.2.3 Graph Neural Network

As discussed in the previous chapter, the fully connected neural networks used in (6.7) may not be the best in practice. Observe that the input dimension for this problem is $mp+mn$, as it incorporates both the channel control states for each plant. An alternative approach here is to utilize the random edge graph neural network (REGNN) parameterization discussed in the previous chapter. To recap these details, recall that the REGNN is a convolutional architecture that, at each layer, performs convolutions over some graph. In the previously considered work of interference networks, the so-called graph shift operator (GSO) could be represented with the fading interference graph. In the problem considered here, we are interested in fading states over n independent channels. Therefore, the graph structure is

not immediately present for this setting as it was for those considered in Chapter 5.

We can, however, construct a graph of interest using the fading states $\mathbf{H} \in \mathbb{R}_+^{m \times n}$ over each of the channels. Consider that $\mathbf{H}$ considers the fading state that each of the m devices experiences in each of the n channels. We may then consider a GSO matrix defined as $\mathbf{S} := \mathbf{H}\mathbf{H}^T \in \mathbb{R}_+^{m \times m}$. Observe that the edges of the graph encoded by S will represent a degree of similarity between the channel states of two systems. In particular, the element $s_{ij} = \mathbf{h}_i^T \mathbf{h}_j$ will be largest when the channel state is favorable in similar channels between systems i and j. We may also interpret this as some degree of competition between these systems, as they will be interested in transmitting in the same channels.

To utilize the REGNN architecture then, we may input the control and channel states $\mathbf{w}_0$ into an L layer REGNN defined over the GSO matrix $\mathbf{S}$. Note that, as $\mathbf{S}$ is constructed using a random matrix $\mathbf{H}$, the GSO is also a random matrix. For given states $\mathbf{w}_0$, we may define a parametrization in (6.8) as the resulting GNN, i.e.

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{w}_0, \boldsymbol{\theta}) := & \\ & \boldsymbol{\sigma}_L(\boldsymbol{\alpha}_L *_{\mathbf{S}} (\boldsymbol{\sigma}_{L-1}(\boldsymbol{\alpha}_{L-1} *_{\mathbf{S}} (\ldots (\boldsymbol{\sigma}_1(\boldsymbol{\alpha}_1 *_{\mathbf{S}} \mathbf{w}_0) \ldots))))), \end{aligned} \tag{6.9}$$

where the parameter $\boldsymbol{\theta}$ contains the L sets of filter weights, i.e. $\boldsymbol{\theta} = \{\boldsymbol{\alpha}_l\}_{l=1}^L$ and has dimension $q = \sum_{l=1}^L \sum_{l=1}^L K_l \times F_l \times F_{l+1}$—recall K_l and F_l as the filter length and number of features at layer l, respectively. Again for only for clarity of presentation we assume all layers have a single feature, i.e. $F_l = 1, l = 1, \ldots, L$ in the definition in (6.9). Because the GSO is randomly varying, the filter weights $\boldsymbol{\theta}$ are trained relative to the statistics of the both the random inputs and the random underlying graphs.

6.3 Primal-Dual Learning

Finding the DNN layer weights $\boldsymbol{\theta}$ that provide good solutions to (6.8) requires the solving of a constraint learning problem. The standard approach of gradient-based optimization methods cannot be applied directly here due to the presence of the latency constraints. To proceed then, we must formulate an unconstrained problem that captures the form of (6.8). A naive penalty-based reformulation will introduce a similar but fundamentally different problem, so we thus opt for constructing a Lagrangian dual problem. For notational convenience, moving forward we employ the following shorthands for the state variables,

aggregate Lyapunov function, latency constraint functions, respectively:

$$\mathbf{w} := [\text{vec}(\mathbf{H}); \text{vec}(\mathbf{X})], \tag{6.10}$$

$$f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) : = \sum_{i=1}^{m} J_i(\mathbf{x}_i, \mathbf{h}_i, \varsigma_i, \mu_i), \tag{6.11}$$

$$g_j(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) := \mathbb{I}\left[\tilde{\tau}_j(\boldsymbol{\Sigma}, \boldsymbol{\mu}) \leq t_{\max}\right] - (1 - \delta) \tag{6.12}$$

We introduce the nonnegative dual variables $\boldsymbol{\lambda} \in \mathbb{R}_+^n$ associated with the vector of constraint functions $\mathbf{g}(\mathbf{p}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) := [g_1(\cdot); \ldots; g_n(\cdot)]$, and form the Lagrangian as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}) := \mathbb{E}_{\mathbf{w}} \left[f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) - \boldsymbol{\lambda}^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) \right]. \tag{6.13}$$

The Lagrangian in (6.13) penalizes constraint violation through the second term. Note, however, that the penalty is scaled by the dual parameter $\boldsymbol{\lambda}$. The so-called Lagranian dual problem is one in which both the primal variable $\boldsymbol{\theta}$ is simultaneously minimized while the dual parameter $\boldsymbol{\lambda}$ is maximized. Such a problem can be written with the saddle point formulation

$$D_{\boldsymbol{\phi}}^* := \max_{\boldsymbol{\lambda} \geq \mathbf{0}} \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}). \tag{6.14}$$

The dual optimum $D_{\boldsymbol{\phi}}^*$ is the best approximation of the form in (6.13) we can have of $J_{\boldsymbol{\phi}}^*$. In fact, under some standard assumptions on the problem and assuming a sufficiently dense DNN architecture, we can formally bound the difference between $D_{\boldsymbol{\phi}}^*$ and J^* to be proportional to the approximation capacity of the DNN $\boldsymbol{\phi}(\mathbf{H}, \mathbf{X}, \boldsymbol{\theta})$—see [38] for details on this result. Thus, we may say that, up to some approximation, solving the unconstrained problem in (6.14) is equivalent to solving the constrained problem in (6.8).

With the unconstrained saddle point problem in (6.14), we may perform standard gradient-based optimization methods to obtain solutions. The max-min structure necessitates the use of a *primal-dual* learning method, in which we iteratively update both the primal and dual variable in (6.13) to find a local stationary point of the KKT conditions of (6.8). Consider a learning iteration index $t = 0, 1, \ldots$ over which we define a sequence of primal variables $\{\boldsymbol{\theta}_t\}$ and dual variables $\{\boldsymbol{\lambda}_t\}$. At index t, we determine the value of next primal iterate $\mathbf{x}_{t+1}$ by adding to the current iterates the corresponding partial gradients of the Lagrangian in (6.13) $\nabla_{\boldsymbol{\theta}} \mathcal{L}$, i.e.,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{w}} \left[f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) - \boldsymbol{\lambda}_t^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) \right], \tag{6.15}$$

where we introduce $\alpha_t > 0$ as a scalar step size. We subsequently perform a corresponding

partial gradient update to compute the dual iterate $\boldsymbol{\lambda}_{t+1}$, i.e.

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t - \beta_t \mathbb{E}_{\mathbf{w}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_{t+1}), \mathbf{w})\right]_+, \tag{6.16}$$

with associated step size $\beta_t > 0$. Observe in (6.16) that we additionally project onto the positive orthant to maintain the nonnegative constraint on $\boldsymbol{\lambda}$. The gradient primal-dual updates in (6.15) and (6.16) successively move the primal and dual variables towards maximum and minimum points of the Lagrangian function, respectively.

6.3.1 Model-free updates

The updates in (6.15)-(6.16) cannot, in general, be applied exactly. To see this, observe that computing the gradients in (6.15) requires computing the gradient of $J_i(\cdot)$—which depends on PDR function $\tilde{q}(\cdot)$ and system dynamics—and the gradient of an indicator of transmission length function $\tilde{\tau}(\cdot)$. In practical systems, we do not typically have easily available analytic forms for these functions to take gradients. Furthermore, both the updates in (6.15) and (6.16) require to take the expectation over the distribution of states $\mathbf{x}$ and $\mathbf{h}$. These, too, are often unknown in practice. However, there exist standard ways of approximating the updates with stochastic, *model-free* updates that do not require such knowledge. Most popular among these is the policy gradient approximation [118].

To compute a policy gradient update, we consider the scheduling parameters $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$ are drawn stochastically from a distribution with given form $\pi_{\boldsymbol{\phi}(\mathbf{w},\boldsymbol{\theta})}$ whose parameters are given by the output of the DNN $\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta})$—e.g. the mean and variance of a normal distribution. Using such a stochastic policy, it can be shown that an unbiased estimators of the gradients in (6.15) and (6.16) can be formed as,

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbf{E}_{\mathbf{w}} f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) = f(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{w}}) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}) \tag{6.17}$$

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbf{E}_{\mathbf{w}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) = \mathbf{g}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{w}}) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}(\hat{\mathbf{p}}_{\boldsymbol{\theta}})^T \tag{6.18}$$

$$\widehat{\mathbf{E}_{\mathbf{w}}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}), \mathbf{w}) = \mathbf{g}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{w}}), \tag{6.19}$$

where $\hat{\mathbf{w}}$ is a sampled state and $\hat{\mathbf{p}}_{\boldsymbol{\theta}}$ is a sample drawn from the distribution $\pi_{\boldsymbol{\phi}(\hat{\mathbf{w}},\boldsymbol{\theta})}$. In practice, we may reduce the variance of these unbiased estimates by taking B samples and averaging. Note that the updates here only require taking gradients of the log likelihoods rather than of the functions themselves. This implies we can perform the learning process without explicitly knowing, e.g., system dynamics, performance metrics, state distributions. Thus, we can replace the updates in (6.15) and (6.16) with their model free counterparts by substituting the gradient estimates in (6.17)-(6.19). The complete primal-dual learning algorithm is summarized in Algorithm 4. We conclude with a brief remark on state sampling.

Algorithm 4 Model-Free Primal-Dual Learning

1: **Parameters:** Policy model $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ and distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}$
2: **Input:** Initial states $\boldsymbol{\theta}_0, \boldsymbol{\lambda}_0$
3: **for** [**do**main loop]$t = 0, 1, 2, \ldots$
4: Draw samples $\{\hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}\}$, or in batches of size B
5: Compute policy gradients [c.f. (6.17)-(6.19)]
6: Update primal and dual variables

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{w}} \left[f(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) - \boldsymbol{\lambda}_t^T \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_t), \mathbf{w}) \right], [cf.(6.15)]$$

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t - \beta_t \widehat{\mathbb{E}_{\mathbf{w}}} \mathbf{g}(\boldsymbol{\phi}(\mathbf{w}, \boldsymbol{\theta}_{t+1}), \mathbf{w}) \right]_+ [cf.(6.16)]$$

7: **end for**

Remark 17. In the gradient estimations in, e.g. (6.17), we sample both the control states $\mathbf{x}$ and channel states $\mathbf{h}$. This assumes that such samples can be drawn i.i.d. While this may generally be true for the channel states $\mathbf{h}$, it will not be generally be true for the control states $\mathbf{x}$ in practice, due to the fact that the states evolve based on the switched dynamics in (6.1), which itself depends on the scheduling actions taken. A more precise way to model the statistics of the control states would be with a Markov decision process (MDP). The generalization of the presented techniques for this setting make up what is known as *reinforcement learning* algorithms. In this work, we nonetheless assume that $\mathbf{x}$ can also be drawn i.i.d. from an approximate distribution and leave the full MDP formulation as the study of future work.

6.4 Simulation Results

We perform a series of simulations on latency-constrained wireless control systems to evaluate the performance the learning method in and the resulting control-aware scheduling policies. We generate a series $m = 9$ systems with closed-loop gains $\hat{\mathbf{A}}_i \sim \text{Uniform}(0.85, 0.95)$ and open-loop gains $\mathring{\mathbf{A}}_i \sim \text{Uniform}(1.01, 1.2)$. The variance for all system noise $\mathbf{w}_i$ is set to be $W = 1$. All such systems send their state information over a shared wireless channel with $n = 2$ independent channels with a total latency constraint of $t_{\max} = 0.5$ ms. A latency bound of this order is typical of industrial control systems such as printing machines and presses [7]. We further assume that the states of the systems are confined to the box $[-10, 10]$. In simulations, we substitute the fully-connected DNN with the more practically efficient REGNN discussed in Section 6.2.3.

With the scheduling architecture given in Figure 6.2 for 5 channels and 20 systems, at each control scheduling interval each system is given a data rate μ_i and a set of channels to transmit on. In our simulations, we use the modulation and coding schemes (MCS)

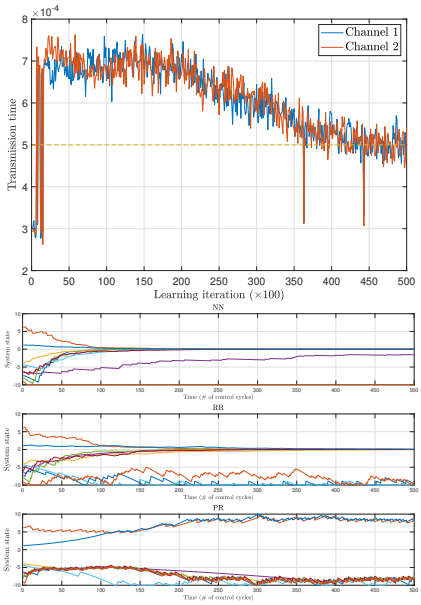


Figure 6.3: Convergence of (left) transmission time for a low-latency, control aware scheduling policy over the learning process. The DNN parameterized scheduling policy obtains feasible latency-contained schedules ($t_{\max} = 5 \times 10^{-4}$ shown in dashed red line) on both channels. In the (right) image, we simulate the control system using the learned scheduler and two baseline model-free heuristic scheduler. The NN policy keeps 8 systems stable, while RR and PR keep 6 and 0, respectively.

of the next-generation IEEE 802.11ax Wi-Fi protocol as a representative architecture for data rate selection and packet error rate computation. As such, the continuous data rates μ_i are selected in an interval of $[1.6, 13]$ and rounded down to the nearest discrete MCS selection given in 802.11ax—see [75] for details on the MCS tables given in this protocol. The corresponding transmission time $\tau(\mu)$ is then calculated assuming a fixed packet size of 100 bytes and the packet delivery rate $q(h, \mu)$ is computed using the associated AWGN error curve (scaled by the effective SNR given channel conditions).

In Figure 6.3 we show the training process for the control-aware scheduler using the primal-dual scheduler given in Algorithm 4. In the left figure, we show the transmission time utilized over the 2 channels by the NN-based scheduling policy over the course of $50,000$ learning iterations. As can be seen the in the left figure, the policies converge to scheduling decisions that respect latency requirements for both channels after $50,000$ iterations.

We proceed to compare the performance of the learned policy in terms of the control metric in (6.4) against other scheduling heuristics. We compares against a standard, control-agnostic round-robin scheduling policy (RR) and a control-aware priority ranking (PR) heuristic in which transmissions are prioritized for systems with largest states. We point out that both of the scheduling policies used fixed PDRs of 0.95 to determine data rate selection. In the right image of Figure 6.3 we show evolution of the 9 systems when using each of the

schedulers. It can be observed that the performance of the DNN scheduling policy learned with primal-dual learning outperforms both heuristics. The NN-based scheduler is able to keep 8 out of 9 systems stable, while the RR and PR schedulers keep only 6 and 0 systems stable, respectively. This can be attributed to the fact that DNN has been model-free trained to adapt to both changing channel conditions and the individual dynamics of each of the systems, which allows it to make more efficient scheduling policies with regards to the varying system dynamics and latency constraints.

6.5 Conclusion

We consider the setting of scheduling for low-latency wireless control systems. To handle the challenge of achieving high reliability performance with limited scheduling resources, we formulate a control-aware scheduling problem in which reliability is adapted to control and channel states. This problem takes the form of a constrained statistical learning problem, in which solutions can be found by parameterized the scheduling policy with a deep neural network and finding optimal weights with a primal-dual learning algorithm that can be implemented without system or dynamical models. Numerical simulations showcase DNN-based scheduling policies that outperform baseline scheduling procedures.

Chapter 7

Conclusions and Future Work

In this dissertation, we consider the problem of developing learning techniques for solving resource allocation problems in wireless systems. Learning is expected to play a crucial role in next-generation wireless networks, e.g. 6G internet [19, 46], to enable new applications in large scale autonomous networks. Learning techniques are advantageous in this context both due to their reliance on adapting to data over relying on models and their potential to generalize and scale as the problems grow in size and complexity. After studying some motivating examples of ongoing research problems in wireless control systems, we develop a learning framework to solve generic resource allocation problems. The framework firstly consists of a constrained learning formulation of wireless resource allocation that leads naturally to a primal-dual learning algorithm to train the parameters of the learning model. Secondly, we discuss the particular learning architectures or parameterizations, namely graph neural networks, that are well-suited for approximating resource allocation policies.

While the proposed framework provides a basis of which we may learn resource allocation policies for a wide variety of problems of practical interest in wireless autonomous systems, there nonetheless exists many opportunities to expand both the scope and practical deployment of such systems. In this concluding chapter, we outline just a few different extensions we believe are vital in future work. To do so, we represent the organizational flowchart in Figure 7.1, in which we include additional blocks with dashed edges to reflect future advances.

The first primary thrust of future work concerns the formulation of resource allocation problem itself, as well as the subsequent learning algorithms that follow. The formulation we study is generic in that it captures all problems in which the expected performance function can be sampled with i.i.d. samples of the state. In other words, we can sample states independently during training because future states are not correlated on previous actions or state. There do, however, exist problems of interest where this does not hold. Many wireless channel models, for instance, utilize a Markov model in which channel quality

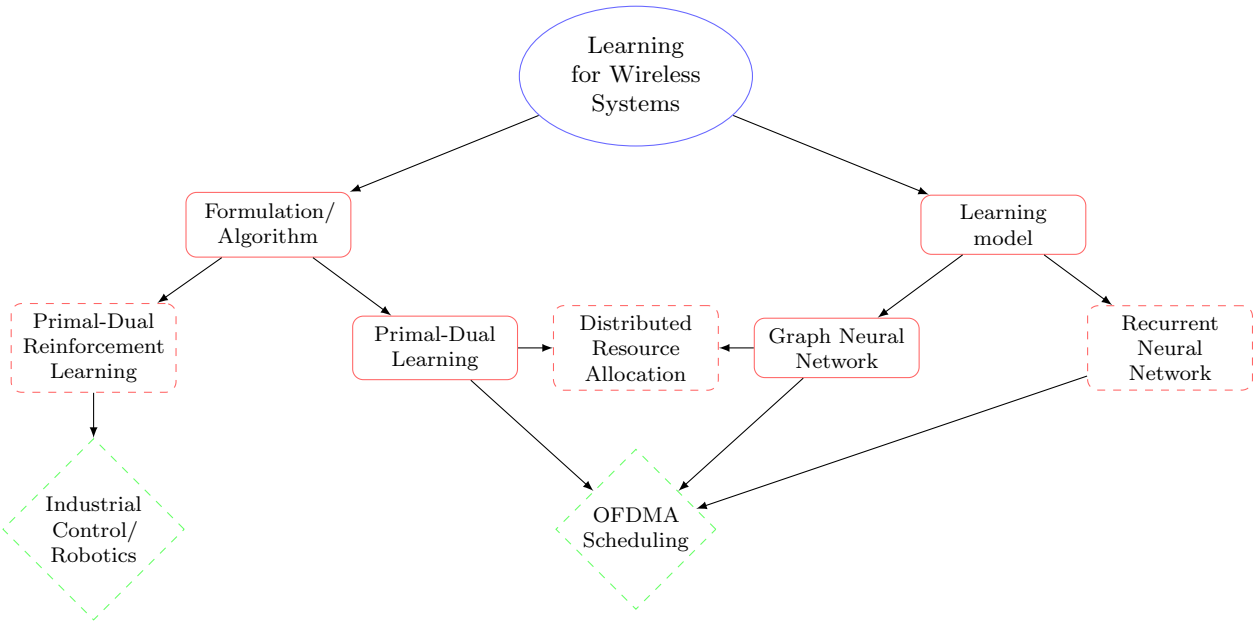


Figure 7.1: Outline of future extensions to learning framework for wireless resource allocation problems. Future thrusts are outlined in dashed borders.

at one time step is related to the quality at previous time steps. Similarly, problems in proportional fairness for wireless systems consider an additional state that tracks relative power history of each transmitter [109]. Perhaps most importantly, the plant state used in the design of wireless control systems will necessarily have dependence upon resource allocation decisions previously taken. To formulate such problems, consider that given an state/action pair at time t, denoted $\{\mathbf{h}_t, \boldsymbol{\phi}(\mathbf{h}_t)\}$, the state $\mathbf{h}_{t+1}$ at time $t+1$ belongs to some distribution $\psi(\mathbf{h}_t, \boldsymbol{\phi}(\mathbf{h}_t))$, parameterized by the previous state and action. We may then consider the statistical cost over some time horizon of length T, and rewrite the original learning formulation from Chapter 4 as

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta}, \mathbf{x}} \; & g_0(\mathbf{x}), \\ \text{s.t.} \;\; & \mathbf{x} \leq \mathbb{E}_{\psi(\mathbf{h}, \boldsymbol{\phi}(\mathbf{h}))} \sum_{t=0}^{T} \mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}_t, \boldsymbol{\theta}), \mathbf{h}_t\big), \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \;\; \mathbf{x} \in \mathcal{X}, \;\; \boldsymbol{\theta} \in \Theta. \end{aligned} \tag{7.1}$$

The statistical constraint in (7.1) is evaluated over a time horizon and the expectation is taken with respect to the distributions parameterized by the policy. Such a formulation is called a Markov decision process (MDP). The algorithms employed in this context are known as reinforcement learning (RL) methods [117]. It is necessary then to formulate such

problems as constrained MDPs and employ a primal-dual RL algorithm to solve. There has been some work studying constrained policy optimization, e.g. [1], but none focused on the challenges specifically brought on by wireless systems. We include this thrust in dashed, red line in the left side of the revised flow chart in Figure 7.1.

The second primary thrust of future work concerns the learning architectures we employ to parameterize the resource allocation policy. In Chapter 5 we focus our attention of graph neural networks, as many problems of interest will naturally contain an underlying graph structure to the data. Moreover, we can identify in many cases a permutation equivariance of the optimal resource allocation function, making the GNN a suitable architecture to solve. There are, however, numerous other key problems in wireless autonomous systems that do not readily contain a graph structure to exploit or a permutation equivariance property. For instance, wireless scheduling systems that feature discrete scheduling resource blocks, such as in OFDMA, may not be describable with a fully connected graph, but do contain structure that is worth incorporating into the learning architecture. Such a resource block assignment may be described with a weighted bipartite graph, or can alternatively be represented with data sequences. This latter representation suggests a potential use of recurrent neural network (RNN) architectures to parameterize the scheduling policy; see, e.g. recent work in using GNNs [98] or RNNs [84] for solving assignment-type problems. We include the development of RNN architectures in the dashed red line in the right side of the revised flow chart in Figure 7.1.

The third primary thrust considers another extension to be made to the deep learning architectures we employ. Distributed resource allocation remains an ongoing problem of interest in the study of autonomous wireless systems. Many applications of interest do not feature a centralized coordinator or scheduler to make resource allocation decisions for all transmitters. As an alternative, we may learn resource allocation polices that can be implemented locally at each transmitter, using only state information it can collect from neighboring systems. Some existing work considers learning distributed schedulers for simpler unconstrained resource allocation problems [25]. A promising direction to incorporate distributed policies into the constrained learning framework developed here, on the other hand, involves the use of a special architecture of GNNs called *aggregation* GNNs [45]. Such an architecture naturally leads to distributed implementation and, when combined with the primal-dual learning process, provides a manner in which we can learn distributed policies for complex resource allocation problems. We include the development of distributed policy learning in the dashed red line in the center column of the revised flow chart in Figure 7.1.

The final primary thrust of future work focuses on the new applications that may be enabled by the previous future thrusts. First and foremost is the continued study of wireless control systems necessary for solving problems in industrial control and robotics.

As previously mentioned, the extension of the formulation of the statistical learning problem to include MDPs will allow us to more accurately model both control system utilities and control system constraints. Particular problems of interest for Industry 4.0 include the previously considered low-latency wireless control systems, as well as safe reinforcement learning in wireless industrial systems. Another application domain that is highly relevant in next-generation wireless systems is learning for OFDMA-type scheduling architectures such as 5G/6G and IEEE 802.11ax. The combinatorial size of the action space requires some consideration into the learning architectures used. The previous future thrust in recurrent neural networks can be used to devise sequential architectures for scheduling policies. These two new applications are included in the bottom row of the revised flow chart in Figure 7.1.

Bibliography

[1] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 22–31.

[2] ACROME. [Online]. Available: https://www.acrome.net/ball-balancing-table

[3] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[4] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications magazine*, vol. 39, no. 2, pp. 150–154, 2001.

[5] J. Araújo, M. Mazo, A. Anta, P. Tabuada, and K. H. Johansson, "System architectures, protocols and algorithms for aperiodic wireless control systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 175–184, 2014.

[6] M. I. Ashraf, C.-F. Liu, M. Bennis, W. Saad, and C. S. Hong, "Dynamic resource allocation for optimized latency and reliability in vehicular networks," *IEEE Access*, vol. 6, pp. 63 843–63 858, 2018.

[7] S. A. Ashraf, I. Aktas, E. Eriksson, K. W. Helmersson, and J. Ansari, "Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G," in *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*. IEEE, 2016, pp. 1–8.

[8] J.-A. Bazerque and G. B. Giannakis, "Distributed scheduling and resource allocation for cognitive OFDMA radios," *Mobile Networks and Applications*, vol. 13, no. 5, pp. 452–462, 2008.

[9] B. Bellalta, "Ieee 802.11 ax: High-efficiency wlans," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[10] M. Bennis, M. Debbah, and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale," *arXiv preprint arXiv:1801.01270*, 2018.

[11] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[12] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[13] O. Bousquet and L. Bottou, "The tradeoffs of large scale learning," in *Advances in neural information processing systems*, 2008, pp. 161–168.

[14] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[15] N. Brahmi, O. N. Yilmaz, K. W. Helmersson, S. A. Ashraf, and J. Torsner, "Deployment strategies for ultra-reliable and low-latency communication in factory automation," in *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[16] M. S. Branicky, S. M. Phillips, and W. Zhang, "Scheduling and feedback co-design for networked control systems," in *Proc.of the 41st IEEE Conf. on Dec. and Control (CDC)*, vol. 2, 2002, pp. 1211–1217.

[17] A. Cervin and T. Henningsson, "Scheduling of event-triggered controllers on a shared network," in *Proc. of the 47th IEEE Conf. on Dec. and Control (CDC)*, 2008, pp. 3601–3606.

[18] C. S. Chen, K. W. Shum, and C. W. Sung, "Round-robin power control for the weighted sum rate maximisation of wireless networks over multiple interfering links," *European Transactions on Telecommunications*, vol. 22, no. 8, pp. 458–470, 2011.

[19] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[20] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, "Stochastic averaging for constrained optimization with application to online resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3078–3093, 2017.

[21] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 2016, pp. 7–10.

[22] M. B. Cloosterman, L. Hetel, N. Van de Wouw, W. Heemels, J. Daafouz, and H. Nijmeijer, "Controller synthesis for networked control systems," *Automatica*, vol. 46, no. 10, pp. 1584–1594, 2010.

[23] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *arXiv preprint arXiv:1808.01486*, 2018.

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[25] P. de Kerret, D. Gesbert, and M. Filippone, "Team deep neural networks for interference channels," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2018, pp. 1–6.

[26] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems*, 2014, pp. 1646–1654.

[27] B. Demirel, A. Ramaswamy, D. E. Quevedo, and H. Karl, "Deepcas: A deep reinforcement learning algorithm for control-aware scheduling," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 737–742, 2018.

[28] M. Donkers, W. Heemels, N. Van De Wouw, and L. Hetel, "Stability analysis of networked control systems using a switched linear systems approach," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2101–2115, 2011.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[30] M. Eisen, K. Gatsis, G. J. Pappas, and A. Ribeiro, "Learning in non-stationary wireless control systems via newton's method," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 1410–1417.

[31] ——, "Learning in wireless control systems over nonstationary channels," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1123–1137, 2018.

[32] ——, "Learning statistically accurate resource allocations in non-stationary wireless systems," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3559–3563.

[33] ——, "Optimization of switched linear systems over non-stationary wireless channels," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[34] M. Eisen, A. Mokhtari, and A. Ribeiro, "Large scale empirical risk minimization via truncated adaptive newton method," *arXiv preprint arXiv:1705.07957*, 2017.

[35] M. Eisen, M. M. Rashid, K. Gatsis, D. Cavalcanti, N. Himayat, and A. Ribeiro, "Control aware communication design for time wireless systems," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4584–4588.

[36] ——, "Control aware radio resource allocation in low latency wireless control systems," *IEEE Internet of Things Journal*, 2019.

[37] M. Eisen and A. Ribeiro, "Large scale wireless resource allocation with graph neural networks," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, p. (to appear).

[38] M. Eisen, C. Zhang, L. Chamon, D. D. Lee, and A. R. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, 2019.

[39] M. Eisen, C. Zhang, L. F. Chamon, D. D. Lee, and A. Ribeiro, "Online deep learning in wireless communication systems," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 1289–1293.

[40] ——, "Dual domain learning of optimal resource allocations in wireless systems," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 4729–4733.

[41] N. Elia, "Remote stabilization over fading channels," *Systems & Control Letters*, vol. 54, no. 3, pp. 237–249, 2005.

[42] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 6, pp. 1333–1344, 2007.

[43] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.

[44] F. Gama, J. Bruno, and A. Ribeiro, "Stability properties of graph neural networks," *arXiv preprint arXiv:1905.04497*, 2019.

[45] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2019.

[46] A. Gatherer, "What will 6G be?" *IEEE ComSoc Technol. News (June 2018)*, 2018.

[47] K. Gatsis, M. Pajic, A. Ribeiro, and G. J. Pappas, "Opportunistic control over shared wireless channels," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3140–3155, December 2015.

[48] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Optimal power management in wireless control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1495–1510, 2014.

[49] ——, "Random access design for wireless control systems," *Automatica*, vol. 91, pp. 1 – 9, May 2018.

[50] A. Goldsmith, *Wireless communications*. Cambr. Univ. Press, 2005.

[51] V. Gupta, B. Hassibi, and R. M. Murray, "Optimal LQG control across packet-dropping links," *Systems & Control Letters*, vol. 56, no. 6, pp. 439–446, 2007.

[52] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement learning with deep energy-based policies," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1352–1361.

[53] D. Han, J. Wu, H. Zhang, and L. Shi, "Optimal sensor scheduling for multiple linear dynamical systems," *Automatica*, vol. 75, pp. 260–270, 2017.

[54] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *51st Annual Conference on Decision and Control*, 2012, pp. 3270–3285.

[55] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[56] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[57] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[58] R. P. F. Hoefel and O. Bejarano, "On application of phy layer abstraction techniques for system level simulation and adaptive modulation in IEEE 802.11 ac/ax systems," *Journal of Communication and Information Systems*, vol. 31, no. 1, 2016.

[59] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[60] D. Hristu-Varsakelis, "Feedback control systems as users of a shared network: Communication sequences that guarantee stability," in *Proc. of the 40th IEEE Conf. on Dec. and Control (CDC)*, vol. 4, 2001, pp. 3631–3636.

[61] Y. Hu and A. Ribeiro, "Adaptive distributed algorithms for optimal random access channels," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2703–2715, 2011.

[62] ——, "Optimal wireless networks based on local channel state information," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4913–4929, 2012.

[63] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.

[64] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in neural information processing systems*, 2013, pp. 315–323.

[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[67] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[68] J. Le Ny, E. Feron, and G. J. Pappas, "Resource constrained lqr control under fast sampling," in *Proc. of the 14th International Conference on Hybrid Systems: Computation and Control*. ACM, 2011, pp. 271–280.

[69] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1276–1279, 2018.

[70] L. Lei, L. You, G. Dai, T. X. Vu, D. Yuan, and S. Chatzinotas, "A deep learning approach for optimizing content delivering in cache-enabled HetNet," in *Wireless Communication Systems (ISWCS), 2017 International Symposium on*. IEEE, 2017, pp. 449–453.

[71] A. S. Leong, A. Ramaswamy, D. E. Quevedo, H. Karl, and L. Shi, "Deep reinforcement learning for wireless sensor scheduling in cyber-physical systems," *arXiv preprint arXiv:1809.05149*, 2018.

[72] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of industry 4.0," *Wireless networks*, vol. 23, no. 1, pp. 23–41, 2017.

[73] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *arXiv preprint arXiv:1807.10025*, 2018.

[74] J. W. Liu, *Real-Time Systems*. Prentice-Hall, Inc, 2000.

[75] J. Liu, R. Porat, N. Jindal *et al.*, "Ieee 802.11 ax channel model document," *Wireless LANs, Rep. IEEE 802.11–14/0882r3*, 2014.

[76] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2064, 2001.

[77] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[78] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on networking*, vol. 7, no. 4, pp. 473–489, 1999.

[79] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[80] M. H. Mamduhi, D. Tolić, A. Molin, and S. Hirche, "Event-triggered scheduling for stochastic multi-loop networked control systems with packet dropouts," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2776–2782.

[81] M. Mazo and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[82] L. Meier, J. Peschon, and R. M. Dressler, "Optimal control of measurement subsystems," *IEEE Transactions on Automatic Control*, vol. 12, no. 5, pp. 528–536, 1967.

[83] F. Meng, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: Deep reinforcement learning approaches," *arXiv preprint arXiv:1901.07159*, 2019.

[84] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[85] A. Mokhtari, H. Daneshmand, A. Lucchi, T. Hofmann, and A. Ribeiro, "Adaptive Newton method for empirical risk minimization to statistical accuracy," in *Advances in Neural Information Processing Systems*, 2016, pp. 4062–4070.

[86] A. Mokkadem and M. Pelletier, "A generalization of the averaging procedure: The use of two-time-scale algorithms," *SIAM Journal on Control and Optimization*, vol. 49, no. 4, pp. 1523–1543, 2011.

[87] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE journal on selected areas in communications*, vol. 32, no. 6, pp. 1139–1151, 2014.

[88] I. Necoara and J. Suykens, "Interior-point lagrangian decomposition method for separable convex optimization," *Journal of Optimization Theory and Applications*, vol. 143, no. 3, p. 567, 2009.

[89] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[90] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), Tech. Rep., 2011.

[91] J. J. Nielsen, R. Liu, and P. Popovski, "Ultra-reliable low latency communication using interface diversity," *IEEE Transactions on Communications*, vol. 66, no. 3, pp. 1322–1334, 2018.

[92] V. Ntranos, N. D. Sidiropoulos, and L. Tassiulas, "On multicast beamforming for minimum outage," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, 2009.

[93] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[94] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the mimo fading channel," in *Communication, Control, and Computing (Allerton), 2017 55th Annual Allerton Conference on*. IEEE, 2017, pp. 76–80.

[95] A. B. Owen, "Self-concordance for empirical likelihood," *Canadian Journal of Statistics*, vol. 41, no. 3, pp. 387–397, 2013.

[96] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.

[97] P. Popovski, J. J. Nielsen, C. Stefanovic, E. de Carvalho, E. Strom, K. F. Trillingsgaard, A.-S. Bana, D. M. Kim, R. Kotaba, J. Park *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Network*, vol. 32, no. 2, pp. 16–23, 2018.

[98] M. O. Prates, P. H. Avelar, H. Lemos, L. Lamb, and M. Vardi, "Learning to solve NP-complete problems-a graph neural network for the decision TSP," *arXiv preprint arXiv:1809.02721*, 2018.

[99] Quanser, "Linear inverted pendulum user manual." [Online]. Available: https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum/

[100] D. E. Quevedo, A. Ahlén, A. S. Leong, and S. Dey, "On Kalman filtering over fading wireless channels with controlled transmission powers," *Automatica*, vol. 48, no. 7, pp. 1306–1316, 2012.

[101] M. Rabi and K. H. Johansson, "Scheduling packets for event-triggered control," in *European Control Conference (ECC)*, 2009, pp. 3779–3784.

[102] H. Rehbinder and M. Sanfridson, "Scheduling of a limited communication channel for optimal control," *Automatica*, vol. 40, no. 3, pp. 491–500, 2004.

[103] A. Ribeiro, "Optimal resource allocation in wireless communication and networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 272, 2012.

[104] L. Ruiz, F. Gama, A. G. Marques, and A. Ribeiro, "Invariance-preserving localized activation functions for graph neural networks," *arXiv preprint arXiv:1903.12575*, 2019.

[105] A. Sahai and S. Mitter, "The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link&# 8212; Part I: Scalar systems," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3369–3395, 2006.

[106] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007.

[107] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[108] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Learnability, stability and uniform convergence," *Journal of Machine Learning Research*, vol. 11, no. Oct, pp. 2635–2670, 2010.

[109] Z. Shen, J. G. Andrews, and B. L. Evans, "Adaptive resource allocation in multiuser ofdm systems with proportional rate constraints," *IEEE transactions on wireless communications*, vol. 4, no. 6, pp. 2726–2737, 2005.

[110] L. Shi, P. Cheng, and J. Chen, "Optimal periodic sensor scheduling with limited resources," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2190–2195, 2011.

[111] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3060–3063.

[112] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Processing*, vol. 54, no. 6-1, pp. 2239–2251, 2006.

[113] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[114] B. Sriperumbudur, K. Fukumizu, and G. Lanckriet, "On the relation between universality, characteristic kernels and rkhs embedding of measures," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 773–780.

[115] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[116] H. Sun, Z. Zhao, X. Fu, and M. Hong, "Limited feedback double directional massive MIMO channel estimation: From low-rank modeling to deep learning," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[117] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.

[118] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[119] V. N. Swamy, S. Suri, P. Rigge, M. Weiner, G. Ranade, A. Sahai, and B. Nikolić, "Cooperative communication for high-reliability low-latency wireless control," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4380–4386.

[120] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on Automatic Control*, vol. 49, no. 7, pp. 1056–1068, 2004.

[121] T. Van Chien, E. Björnson, and E. G. Larsson, "Sum spectral efficiency maximization in massive mimo systems: Benefits from deep learning," *arXiv preprint arXiv:1903.08163*, 2019.

[122] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[123] A. Varghese and D. Tandur, "Wireless requirements and challenges in industry 4.0," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*. IEEE, 2014, pp. 634–638.

[124] X. Wang, T. Chen, X. Chen, X. Zhou, and G. B. Giannakis, "Dynamic resource allocation for smart-grid powered mimo downlink transmissions," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3354–3365, 2016.

[125] X. Wang and N. Gao, "Stochastic resource allocation over fading multiple access and broadcast channels," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2382–2391, 2010.

[126] X. Wang and G. B. Giannakis, "Resource allocation for wireless multiuser OFDM networks," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4359–4372, 2011.

[127] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikolié, "Design of a low-latency, high-reliability wireless communication system for control applications," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3829–3835.

[128] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.

[129] C. Wu, M. Sha, D. Gunatilaka, A. Saifullah, C. Lu, and Y. Chen, "Analysis of edf scheduling for wireless sensor-actuator networks," in *Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of*. IEEE, 2014, pp. 31–40.

[130] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 21, no. 4, pp. 1215–1228, 2013.

[131] D. Xu, X. Che, C. Wu, S. Zhang, S. Xu, and S. Cao, "Energy-efficient subchannel and power allocation for hetnets based on convolutional neural network," *arXiv preprint arXiv:1903.00165*, 2019.

[132] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[133] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in ofdma wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 322–337, 2012.

[134] H. Ye and G. Y. Li, "Deep reinforcement learning for resource allocation in V2V communications," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[135] O. N. Yilmaz, Y.-P. E. Wang, N. A. Johansson, N. Brahmi, S. A. Ashraf, and J. Sachs, "Analysis of ultra-reliable and low-latency 5g communication for a factory automation use case," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1190–1195.

[136] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Transactions on Communications*, vol. 54, no. 7, pp. 1310–1322, 2006.

[137] P. Zand, S. Chatterjea, K. Das, and P. Havinga, "Wireless industrial monitoring and control networks: The journey so far and the road ahead," *Journal of sensor and actuator networks*, vol. 1, no. 2, pp. 123–152, 2012.

[138] J. Zhang and D. Zheng, "A stochastic primal-dual algorithm for joint flow control and mac design in multi-hop wireless networks," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 339–344.

[139] L. Zhang and D. Hristu-Varsakelis, "Communication and control co-design for networked control systems," *Automatica*, vol. 42, no. 6, pp. 953–958, 2006.

[140] W. Zhang, M. Branicky, and S. Phillips, "Stability of networked control systems," *IEEE Control Systems Mag.*, vol. 21, no. 1, pp. 84–99, 2001.