

Learning Optimal Resource Allocations in Wireless Systems

Mark Eisen* Clark Zhang* Luiz F. O. Chamon* Daniel D. Lee† Alejandro Ribeiro*

Abstract—This paper considers the design of optimal resource allocation policies in wireless communication systems which are generically modeled as a functional optimization problem with stochastic constraints. These optimization problems have the structure of a learning problem in which the statistical loss appears as a constraint, motivating the development of learning methodologies to attempt their solution. To handle stochastic constraints, training is undertaken in the dual domain. It is shown that this can be done with small loss of optimality when using near-universal learning parameterizations. In particular, since deep neural networks (DNN) are near-universal their use is advocated and explored. DNNs are trained here with a model-free primal-dual method that simultaneously learns a DNN parametrization of the resource allocation policy and optimizes the primal and dual variables. Numerical simulations demonstrate the strong performance of the proposed approach on a number of common wireless resource allocation problems.

Index Terms— wireless systems, deep learning, resource allocation, strong duality

I. 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 $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ 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 $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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 $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$. 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 [1], beamforming [2], [3], and random access [4], [5].

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

Supported by ARL DCIST CRA W911NF-17-2-0181 and Intel Science and Technology Center for Wireless Autonomous Systems. The authors are with the *Department of Electrical and Systems Engineering, University of Pennsylvania and †Department of Electrical and Computer Engineering, Cornell Tech. Email: maeisen@seas.upenn.edu, clarkz@seas.upenn.edu, luizf@seas.upenn.edu, ddl46@cornell.edu, aribeiro@seas.upenn.edu.

convexity of the constraint $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$. For resource allocation problems, such as interference management, heuristic methods have been developed [6]–[8]. 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 been known to be null for convex problems – e.g., the water level in water filling solutions is a dual variable – and has more recently been shown to be null under mild technical conditions despite the presence of the nonconvex constraint $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ [9], [10]. 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 [11]–[17].

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 [18], [19]. In some other cases heuristics can be used to find approximate solutions to construct a training set [20]–[22]. 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 $\mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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, $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ a loss function to be minimized, and the expectation $\mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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 $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ at sampled pairs $(\mathbf{h}, \mathbf{p}(\mathbf{h}))$. This setting is typical of, e.g., reinforcement learning problems [23], and is a learning approach that has been taken in several *unconstrained* problems in wireless optimization [24]–[27]. 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 II).

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 III). 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 I)—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 III-A). More formally, the duality gap is $\mathcal{O}(\epsilon)$ if the learning parametrization can approximate arbitrary functions with error $\mathcal{O}(\epsilon)$ (Theorem I).

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 III-B). 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 IV).

This model-free approach additionally includes the policy gradient method for efficiently estimating the gradients of a function of a policy (Section IV-A). 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 [28]. 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 V).

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 VI).

II. 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 $\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^u$. The components of the vector valued function $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ 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 $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] \in \mathbb{R}^u$, which, for formulating optimal wireless design problems, is relaxed to the inequality

$$\mathbf{x} \leq \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]. \quad (1)$$

In (1), we interpret $\mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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 $\mathbf{x} = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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 $\mathbf{g} : \mathbb{R}^u \rightarrow \mathbb{R}^r$ and a scalar utility function $g_0 : \mathbb{R}^u \rightarrow \mathbb{R}$, 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

$$\begin{aligned} P^* := \max_{\mathbf{p}(\mathbf{h}), \mathbf{x}} & g_0(\mathbf{x}), \\ \text{s. t.} & \mathbf{x} \leq \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})], \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \quad \mathbf{x} \in \mathcal{X}, \quad \mathbf{p} \in \mathcal{P}. \end{aligned} \quad (2)$$

In (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 $\mathbf{x} \leq \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ relates the instantaneous resource allocations with the long term average performances as per (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 $f(\cdot, \mathbf{h})$ 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 [10]. We present three examples below to clarify ideas and proceed to motivate and formulate learning approaches for solving (2).

Example 1 (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 (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 [10].

Example 2 (Multiple access interference channel). A set of m terminals communicates with associated receivers. The channel linking terminal i to 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 $\mathbf{h} = [h^{11}; h^{12}; \dots; h^{mm}]$. 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

$$x^i \leq \mathbb{E}_{\mathbf{h}} \left[\log \left(1 + \frac{h^{ii}p^i(\mathbf{h})}{1 + \sum_{j \neq i} h^{ji}p^j(\mathbf{h})} \right) \right]. \quad (3)$$

The constraint in (3) has the form of (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 (2).

Example 3 (Time division multiple access). In Example 2 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 (3) as

$$x^i \leq \mathbb{E}_{\mathbf{h}} \left[\alpha^i(\mathbf{h}) \log (1 + h^i p^i(\mathbf{h})) \right], \quad (4)$$

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 2, we not only allocate power but channel access as well.

A. Learning formulations

The problem in (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 (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 (2) a functional optimization problem, which, compounded with the fact that (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 (2) is known to have null duality gap [10]. This, however, does not generally make the problem easy to solve and moreover requires having model information.

This brings a third challenge in solving (2), namely the availability of the wireless system functions:

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

As we have seen in Examples 1-3, the function $f(\mathbf{p}(\mathbf{h}), \mathbf{h})$ 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 $g(\mathbf{x})$ 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 $\theta \in \mathbb{R}^q$ we make

$$\mathbf{p}(\mathbf{h}) = \phi(\mathbf{h}, \theta). \quad (5)$$

With this parametrization the ergodic constraint in (1) becomes

$$\mathbf{x} \leq \mathbb{E}[\mathbf{f}(\phi(\mathbf{h}, \theta), \mathbf{h})] \quad (6)$$

If we now define the set $\Theta := \{\boldsymbol{\theta} \mid \phi(\mathbf{h}, \boldsymbol{\theta}) \in \mathcal{P}\}$, the optimization problem in (2) becomes one in which the optimization is over \mathbf{x} and $\boldsymbol{\theta}$

$$\begin{aligned} P_\phi^* := \max_{\boldsymbol{\theta}, \mathbf{x}} & g_0(\mathbf{x}), \\ \text{s. t. } & \mathbf{x} \leq \mathbb{E}[\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})], \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \quad \mathbf{x} \in \mathcal{X}, \quad \boldsymbol{\theta} \in \Theta. \end{aligned} \quad (7)$$

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 (7) is $q + u$. This comes at a loss of optimality because (5) restricts resource allocation functions to adhere to the parametrization $\mathbf{p}(\mathbf{h}) = \phi(\mathbf{h}, \boldsymbol{\theta})$. E.g., if we use a linear parametrization $\mathbf{p}(\mathbf{h}) = \boldsymbol{\theta}^T \mathbf{h}$ it is unlikely that the solutions of (2) and (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 $\phi(\mathbf{h}, \boldsymbol{\theta})$ 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

$$\mathbb{E} \|\mathbf{p}(\mathbf{h}) - \phi(\mathbf{h}, \boldsymbol{\theta})\|_\infty \leq \epsilon. \quad (8)$$

A number of popular machine learning models are known to exhibit the universality property in Definition 1, such as radial basis function networks (RBFNs) [29] and reproducing kernel Hilbert spaces (RKHS) [30]. This work focuses in particular on deep neural networks (DNNs), which can be shown to exhibit a universal function approximation property [31] and are observed to work remarkably well in practical problems—see, e.g., [32], [33]. The specific details regarding the use of DNNs in the proposed learning framework of this paper are discussed in Section V.

While the reduction of the dimensionality of the optimization space is valuable, the most important advantage of (7) is that we can use training to bypass the need to estimate the distribution $m(\mathbf{h})$ and the functions $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$. 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 $\mathbf{p}_k(\mathbf{h}_k) = \phi(\mathbf{h}_k, \boldsymbol{\theta}_k)$. The resulting performance $\mathbf{f}(\mathbf{h}_k, \phi(\mathbf{h}_k, \boldsymbol{\theta}_k))$ is then observed and utilized to learn the optimal parametrized resource allocation as defined by (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 III). Our main result on learning for constrained optimization is to show that, its lack of convexity notwithstanding, the duality gap of (7) is small for near-universal parameterizations (Theorem 1). 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 (7) is then introduced in Section IV and neural network parameterizations are described in Section V.

III. LAGRANGIAN DUAL PROBLEM

Solving the optimization problem in (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 $\mathbf{x} \leq \mathbb{E}[\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})]$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. The Lagrangian of (7) is an average of objective and constraint values weighted by their respective multipliers:

$$\begin{aligned} \mathcal{L}_\phi(\boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := & g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) \\ & + \boldsymbol{\lambda}^T (\mathbb{E}[\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] - \mathbf{x}). \end{aligned} \quad (9)$$

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

$$D_\phi(\boldsymbol{\lambda}, \boldsymbol{\mu}) := \max_{\boldsymbol{\theta} \in \Theta, \mathbf{x} \in \mathcal{X}} \mathcal{L}_\phi(\boldsymbol{\theta}, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \quad (10)$$

We think of (10) as a penalized version of (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 $\boldsymbol{\lambda}^T (\mathbb{E}[\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] - \mathbf{x})$. This interpretation is important here because the problem in (10) is unconstrained except for the set restrictions $\boldsymbol{\theta} \in \Theta$ and $\mathbf{x} \in \mathcal{X}$. This renders (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 $D_\phi(\boldsymbol{\lambda}, \boldsymbol{\mu}) \geq P_\phi^*$. This motivates definition of the dual problem in which we search for the multipliers that make $D_\phi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ as small as possible

$$D_\phi^* := \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} D_\phi(\boldsymbol{\lambda}, \boldsymbol{\mu}). \quad (11)$$

The dual optimum D_ϕ^* is the best approximation we can have of P_ϕ^* when using (10) as a proxy for (7). It follows that the two concerns that are relevant in utilizing (10) as a proxy for (7) are: (i) evaluating the difference between D_ϕ^* and P_ϕ^* and (ii) designing a method for finding the optimal multipliers that attains the minimum in (11). We address (i) in Section III-A and (ii) in Section III-B.

A. Suboptimality of the dual problem

The duality gap is the difference $D_\phi^* - P_\phi^*$ 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 (10) without loss of optimality. The optimization problem in (7), however, is not convex as it incorporates the nonconvex constraint in (6). We will show here that despite the presence of this nonconvex constraint the duality gap $D_\phi^* - P_\phi^*$ is small when using parameterizations 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.

Assumption 1. 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, $0 < \mathbb{E}_h(\mathbb{I}(\mathcal{E}')) < \mathbb{E}_h(\mathbb{I}(\mathcal{E}))$.

Assumption 2. *Slater's condition hold for the unparameterized problem in (2) and for the parametrized problem in (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*

$$\mathbb{E}[\mathbf{f}(\mathbf{p}_0(\mathbf{h}), \mathbf{h})] - \mathbf{x}_0 \geq s\mathbf{1}. \quad (12)$$

Assumption 3. *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 $g_0(\mathbf{x}) \leq g_0(\mathbf{x}')$.*

Assumption 4. *The expected performance function $\mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})]$ 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*

$$\mathbb{E}\|\mathbf{f}(\mathbf{p}_1(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\mathbf{p}_2(\mathbf{h}), \mathbf{h})\|_\infty \leq L\mathbb{E}\|\mathbf{p}_1(\mathbf{h}) - \mathbf{p}_2(\mathbf{h})\|_\infty. \quad (13)$$

Although Assumptions 1-4 restrict the scope of problems (2) and (7), they still allow consideration of most problems of practical importance. Assumption 2 simply states that service demands can be provisioned with some slack. We point that an inequality analogous to (12) holds for the other constraints in (2) and (7). However, it is only the slack s that appears in the bounds we will derive. Assumption 3 is a restriction on the utilities $g_0(\mathbf{x})$, namely that increasing performance values result in increasing utility. Assumption 4 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 1-3, it is evident that they satisfy the monotonicity assumption in Assumption 3. Furthermore, the continuity assumption in Assumption 4 is immediately satisfied by the continuous capacity function in Examples 1 and 2, and is also satisfied by the binary problem in Example 3 due to the bounded expectation of the capacity function.

Assumption 1 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 (2) is known to be null – see Appendix A and [10]. Given the validity of Assumptions 1-4 and using a parametrization that is nearly universal in the sense of Definition 1, we show that the duality/parametrization gap $|D_\phi^* - P^*|$ between problems (2) and (11) is small as we formally state next.

Theorem 1. *Consider the parameterized resource allocation*

problem in (7) and its Lagrangian dual in (11) in which the parametrization ϕ is ϵ -universal in the sense of Definition 1. If Assumptions 1-4 hold, then the dual value D_ϕ^* is bounded by

$$P^* - \|\boldsymbol{\lambda}^*\|_1 L \epsilon \leq D_\phi^* \leq P^*, \quad (14)$$

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

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}_0)}{s} < \infty, \quad (15)$$

in which \mathbf{x}_0 is the strictly feasible point of Assumption 2. ■

Proof: See Appendix A. ■

Given any near-universal parameterization that achieves ϵ -accuracy with respect to all resource allocation policies in \mathcal{P} , Theorem 1 establishes an upper and lower bound on the dual value in (11) relative to the optimal primal of the original problem in (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 (10) as a means of solving the (unparameterized) wireless resource allocation problem in (2). Theorem 1 shows that there exist a set of multipliers – those that attain the optimal dual value D_ϕ^* – that yield a problem that is within $\mathcal{O}(\epsilon)$ of optimal.

It is interesting to observe that the duality gap $P^* - D_\phi^* \leq \|\boldsymbol{\lambda}^*\|_1 L \epsilon$ has a very simple dependence on problem constants. The ϵ factor comes from the error of approximating arbitrary resource allocations $\mathbf{p}(\mathbf{h})$ with parametrized resource allocations $\phi(\mathbf{h}, \boldsymbol{\theta})$. The Lipschitz constant L translates this difference into a corresponding difference between the functions $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ and $\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$. 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 $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ and $\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$. This latter statement is clear from the bound in (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 s is not too small. For problems where constraints are difficult to satisfy, a small slack 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 (15) is a classical bound in optimization theory that we include here for completeness.

B. Primal-Dual learning

In order to train the parametrization $\phi(\mathbf{h}, \boldsymbol{\theta})$ on the problem (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 (9) to find a local stationary point of the KKT conditions of (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 (9), i.e. $\nabla_{\boldsymbol{\theta}}\mathcal{L}, \nabla_{\mathbf{x}}\mathcal{L}$, and projecting to the corresponding feasible set, i.e.,

$$\boldsymbol{\theta}_{k+1} = \mathbf{P}_\Theta[\boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta}, k} \nabla_{\boldsymbol{\theta}}\mathbb{E}\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h})\boldsymbol{\lambda}_k], \quad (16)$$

$$\mathbf{x}_{k+1} = \mathbf{P}_{\mathcal{X}}[\mathbf{x}_k + \gamma_{\mathbf{x}, k} (\nabla g_0(\mathbf{x}) + \nabla \mathbf{g}(\mathbf{x}_k)\boldsymbol{\mu}_k - \mathbf{x}_k)], \quad (17)$$

where we introduce $\gamma_{\theta,k}, \gamma_{\mathbf{x},k} > 0$ 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

$$\boldsymbol{\lambda}_{k+1} = [\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k} (\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_{k+1}), \mathbf{h}) - \mathbf{x}_{k+1})]_+, \quad (18)$$

$$\boldsymbol{\mu}_{k+1} = [\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k} \mathbf{g}(\mathbf{x}_{k+1})]_+, \quad (19)$$

with associated step sizes $\gamma_{\boldsymbol{\lambda},k}, \gamma_{\boldsymbol{\mu},k} > 0$. The gradient primal-dual updates in (16)-(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.

IV. 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 (16)-(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 (16)-(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 $\hat{\nabla}g_0$, $\hat{\nabla}\mathbf{g}$, and $\hat{\nabla}_{\boldsymbol{\theta}}\mathbb{E}\mathbf{f}$ can be constructed using function observations at given points $\{\mathbf{x}_0, \boldsymbol{\theta}_0\}$ and the

sampled perturbations as

$$\hat{\nabla}g_0(\mathbf{x}_0) := \frac{\hat{g}_0(\mathbf{x}_0 + \alpha_1 \hat{\mathbf{x}}_1) - \hat{g}_0(\mathbf{x}_0)}{\alpha_1} \hat{\mathbf{x}}_1, \quad (20)$$

$$\hat{\nabla}\mathbf{g}(\mathbf{x}_0) := \frac{\hat{\mathbf{g}}(\mathbf{x}_0 + \alpha_3 \hat{\mathbf{x}}_2) - \hat{\mathbf{g}}(\mathbf{x}_0)}{\alpha_3} \hat{\mathbf{x}}_2^T, \quad (21)$$

$$\begin{aligned} \hat{\nabla}_{\boldsymbol{\theta}}\mathbb{E}[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_0), \mathbf{h})] := & \hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_0 + \alpha_2 \hat{\boldsymbol{\theta}}), \hat{\mathbf{h}}) - \hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_0), \hat{\mathbf{h}}) \hat{\boldsymbol{\theta}}^T, \\ & \alpha_2 \end{aligned} \quad (22)$$

where we define scalar step sizes $\alpha_1, \alpha_2, \alpha_3 > 0$. The expressions in (20)-(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 they coincide with the true gradients in expectation—see, e.g., [34]. Note also in (22) that, by sampling both the function \mathbf{f} and a channel state \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 (20)-(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.

A. 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 $\mathbb{E}_{\mathbf{h}, \boldsymbol{\theta}}[\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})]$ with respect to $\boldsymbol{\theta}$ can be rewritten using this density function as

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \mathbb{E}_{\mathbf{h}, \mathbf{p}}[\mathbf{f}(\mathbf{p}, \mathbf{h}) \nabla_{\boldsymbol{\theta}} \log \pi_{\mathbf{h}, \boldsymbol{\theta}}(\mathbf{p})^T], \quad (23)$$

where \mathbf{p} is a random variable drawn from distribution $\pi_{\mathbf{h}, \boldsymbol{\theta}}(\mathbf{p})$ —see, e.g., [35]. Observe in (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 $\phi(\mathbf{h}, \boldsymbol{\theta})$, e.g., Gaussian distribution. If an analytic form for $\pi_{\mathbf{h}, \boldsymbol{\theta}}(\mathbf{p})$ is known, we can estimate $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$ by instead directly estimating the left-hand side of (23). In the context of reinforcement learning, this is called the REINFORCE method [35]. By using the previous function observations, we can obtain the following policy gradient estimate,

$$\widehat{\nabla}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \hat{\mathbf{f}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{h}}) \nabla_{\boldsymbol{\theta}} \log \pi_{\hat{\mathbf{h}}, \boldsymbol{\theta}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}})^T, \quad (24)$$

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 (24) can be taken as an alternative to the finite difference approach in (22) for estimating the gradient of the policy constraint function, provided the gradient of the density function π 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 π .

B. Model-free primal-dual method

Using the gradient estimates in (20)-(22)—or (24)—we can derive a model-free, or zeroth-ordered, stochastic updates to replace those in (16)-(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

$$\boldsymbol{\theta}_{k+1} = \mathbb{P}_{\Theta} \left[\boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta}, k} \widehat{\nabla}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h}) \boldsymbol{\lambda}_k \right], \quad (25)$$

$$\mathbf{x}_{k+1} = \mathbb{P}_{\mathcal{X}} \left[\mathbf{x}_k + \gamma_{\mathbf{x}, k} (\widehat{\nabla} g_0(\mathbf{x}) + \widehat{\nabla} \mathbf{g}(\mathbf{x}_k) \boldsymbol{\mu}_k - \mathbf{x}_k) \right], \quad (26)$$

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda}, k} (\hat{\mathbf{f}}(\phi(\hat{\mathbf{h}}_k, \boldsymbol{\theta}_{k+1}), \hat{\mathbf{h}}_k) - \mathbf{x}_{k+1}) \right]_+ \quad (27)$$

$$\boldsymbol{\mu}_{k+1} = [\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu}, k} \hat{\mathbf{g}}(\mathbf{x}_{k+1})]_+. \quad (28)$$

The expressions in (25)-(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 1. The method is initialized in Step 1 through the selection of parameterization model $\phi(\mathbf{h}, \boldsymbol{\theta})$ 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

Algorithm 1 Model-Free Primal-Dual Learning

1: **Parameters:** Policy model $\phi(\mathbf{h}, \boldsymbol{\theta})$ and distribution form $\pi_{\mathbf{h}, \boldsymbol{\theta}}$
2: **Input:** Initial states $\boldsymbol{\theta}_0, \mathbf{x}_0, \boldsymbol{\lambda}_0, \boldsymbol{\mu}_0$
3: **for** $k = 0, 1, 2, \dots$ **do** {main loop}
4: Draw samples $\{\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}_k\}$, or in batches of size B
5: Obtain random observation of function values $\hat{g}_0, \hat{\mathbf{f}}, \hat{\mathbf{g}}$ at current and sampled iterates
6: Compute gradient estimates $\widehat{\nabla} g_0(\mathbf{x}), \widehat{\nabla} \mathbf{g}(\mathbf{x}), \widehat{\nabla}_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}, \boldsymbol{\theta}} \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$, [cf. (20)-(22) or (24)]
7: Update primal and dual variables [cf. (25)-(28)]

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

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

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

$$\boldsymbol{\mu}_{k+1} = [\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu}, k} \hat{\mathbf{g}}(\mathbf{x}_{k+1})]_+.$$
8: **end for**

(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 (25)-(28). Due to the non-convexity of the Lagrangian defined in (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 1. The algorithm presented in Algorithm 1 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 [36]. Policy optimization can also be improved using actor-critic methods [23], 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.

V. 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.

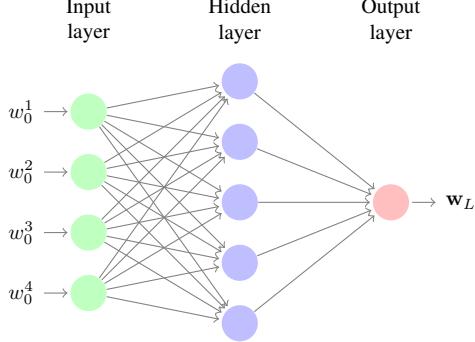


Fig. 1: Typical architecture of fully-connected deep neural network.

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 $l = 1, \dots, L$ 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 $\sigma_l : \mathbb{R}^{q_l} \rightarrow \mathbb{R}^{q_l}$. 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 $\mathbf{w}_l := \sigma_l(\mathbf{W}_l \mathbf{w}_{l-1})$. The final output of the DNN, \mathbf{w}_L , is then related to the input \mathbf{w}_0 by propagating through each layer of the DNN as $\mathbf{w}_L = \sigma_L(\mathbf{W}_L(\sigma_{L-1}(\mathbf{W}_{L-1}(\dots(\sigma_1(\mathbf{W}_1 \mathbf{w}_0))))))$.

An illustration of a fully-connected example DNN architecture is given in Figure 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 σ_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 $\mathbf{W}_1, \dots, \mathbf{W}_L$. Common choices of activation functions σ_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 (7), the policy $\phi(\mathbf{h}, \theta)$ can be defined by an L -layer DNN as

$$\phi(\mathbf{h}, \theta) := \sigma_L(\mathbf{W}_L(\sigma_{L-1}(\mathbf{W}_{L-1}(\dots(\sigma_1(\mathbf{W}_1 \mathbf{h}))))), \quad (29)$$

where $\theta \in \mathbb{R}^q$ 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 (16)–(19) with respect to traditional neural network training, observe that the update in (16) requires computation of the gradient $\nabla_\theta \mathbb{E}_\mathbf{h} \mathbf{f}(\mathbf{h}, \phi(\mathbf{h}, \theta))$. Using the chain rule, this can be expanded as

$$\begin{aligned} \nabla_\theta \mathbb{E}_\mathbf{h} \mathbf{f}(\phi(\mathbf{h}, \theta_k), \mathbf{h}) &= \\ \nabla_\phi \mathbb{E}_\mathbf{h} \mathbf{f}(\phi(\mathbf{h}, \theta_k), \mathbf{h}) \nabla_\theta \phi(\mathbf{h}, \theta_k). \end{aligned} \quad (30)$$

Thus, the computation of the full gradient requires evaluating the gradient of the policy function ϕ as well as the gradient of the DNN model ϕ . For the DNN structure in (29), the evaluation of $\nabla_\theta \phi$ 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 backpropagation.

We further take note how our learning approach differs from a more traditional, *supervised* training of DNNs. As in (30), the backpropagation 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 backpropagation step in (16) is performed in sequence with the more standard primal and dual variable updates in (17)–(19). In this way, the DNN is trained indirectly *within* the broader optimization algorithm used to solve (7). This is in contrast with other approaches of training DNNs in constrained wireless resource allocation problems—see, e.g. [20]–[22]—which train a DNN to approximate the complete constrained maximization function in (2) directly. Doing so requires the ability to solve (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 1, 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 (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 [31], [37]. 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 (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 1 in the special case of DNNs.

Theorem 2. Consider the DNN parametrization $\phi(\mathbf{h}, \theta)$ in (29) with non-constant, continuous activation functions σ_l for $l = 1, \dots, L$. Define the vector of layer lengths $\mathbf{q} = [q_1; q_2; \dots; q_L]$ and a DNN defined in (29) with lengths \mathbf{q} as $\phi_{\mathbf{q}}(\mathbf{h}, \theta)$. Now consider the set of possible L -layer DNN parameterization functions $\Phi := \{\phi_{\mathbf{q}}(\mathbf{h}, \theta) \mid \mathbf{q} \in \mathbb{N}^L\}$. If Assumptions 1–4 hold, then the optimal dual value of the parameterized problem satisfies

$$\inf_{\phi \in \Phi} \mathbf{D}_\phi^* = P^*. \quad (31)$$

Proof: See Appendix B. ■

With Theorem 2 we establish the null duality gap property of a resource allocation problem of the form in (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 1, which require only an DNN-approximation of given accuracy ϵ provide the more practical characterization of (7), while the result in Theorem 2 suggests DNNs can be used find parameterizations of arbitrarily strong accuracy.

VI. 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 (2). For the simulations performed, we employ a stochastic policy and implement the REINFORCE-style policy gradient described in Section IV-A. In particular, we select the policy distribution $\pi_{\theta, h}$ as a truncated Gaussian distribution. The truncated Gaussian distribution has fixed support on the domain $[0, p_{\max}]$. The output layer of the DNN $\phi(h, \theta) \in \mathbb{R}^{2m}$ is the set of m means and standard deviations to specify the respective truncated Gaussian distributions, i.e. $\phi(h, \theta) := [\mu^1; \sigma^1; \mu^2; \sigma^2; \dots; \mu^m; \sigma^m]$. 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 [38] 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).

A. 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 i th 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

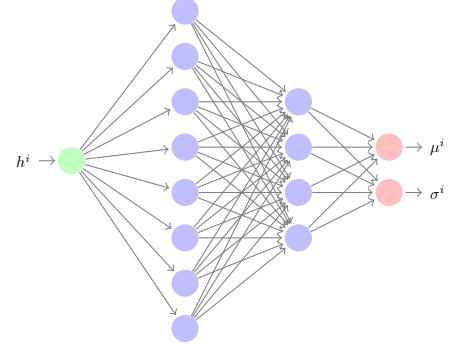


Fig. 2: Neural network architecture used for simple AWGN channel. Each channel state h^i is fed into an independent SISO network with two hidden layers of size 8 and 4, respectively. The DNN outputs a mean μ^i and standard deviation σ^i for a truncated Gaussian distribution.

across all users, with user i weighted by $w^i \geq 0$. The total capacity problem can be written as

$$\begin{aligned} P_{\phi}^* := \max_{\theta, x} & \sum_{i=1}^m w^i x^i \\ \text{s. t. } & x^i \leq \mathbb{E}_{h^i} [\log(1 + h^i \phi^i(h^i, \theta)/v^i)], \forall i \\ & \mathbb{E}_h \left[\sum_{i=1}^m \phi^i(h^i, \theta) \right] \leq p_{\max}. \end{aligned} \quad (32)$$

Note that, despite the non-convex structure of the problem in (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., [39]. 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.

To have the the outputs of the DNN match the same form as the analytic solution (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. $\sigma(z) = [z]_+$; see Figure 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 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

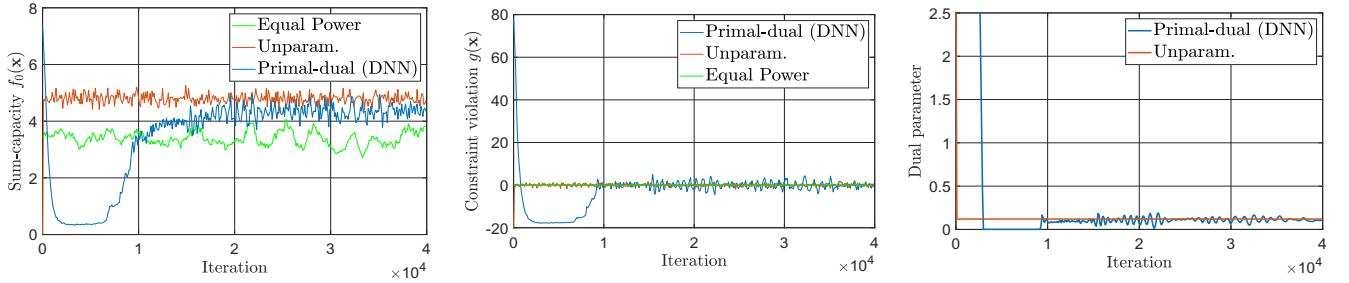


Fig. 3: Convergence of (left) objective function value, (center) constraint value, and (right) dual parameter for simple capacity problem in (32) using proposed DNN method with policy gradients, the exact unparameterized solution, and an equal power allocation amongst users. The DNN parameterization obtains near-optimal performance relative to the exact solution and outperforms the equal power allocation heuristic.

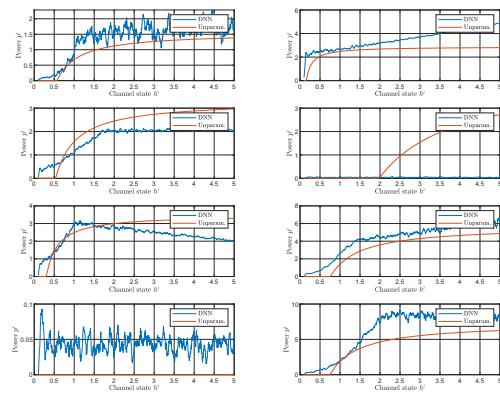


Fig. 4: Example of 8 representative resource allocation policy functions found through DNN parameterization and unparameterized solution. Although the policies differ from the analytic solution, many contain similar shapes. Overall, the DNN method learns variations on the optimal policies that nonetheless achieve similar performance.

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 2. 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 [14].

In Figure 4 we show the actual learned policies from both

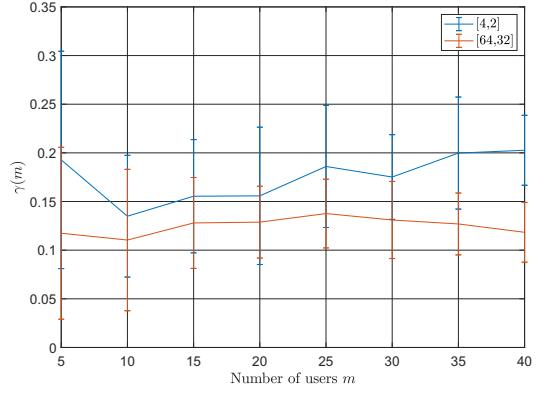


Fig. 5: Optimality gap between optimal objective value and learned policy for the simple capacity problem in (32) for different number of users m and DNN architectures. The results are obtained across 10 randomly initialized simulations. The mean is plotted as the solid lines, while the one standard deviation above and below the mean is show with error bars.

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 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 (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 (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 (32), we perform multiple experiments for varying number of users and with different DNN layer sizes. In Figure 5, we plot the normalized optimality gap between the precise solution P^* and the parameterized solution \hat{P}_ϕ^* found after convergence of the primal-dual method. If we

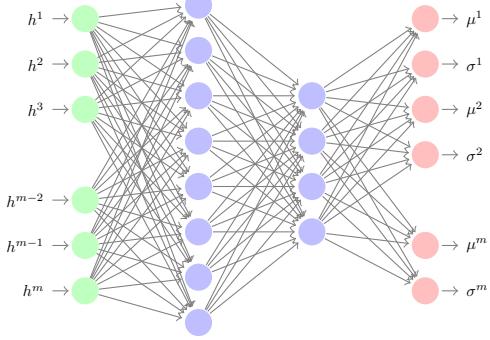


Fig. 6: Neural network architecture used for interference channel problem in (34). All channel states \mathbf{h} are fed into a MIMO network with two hidden layers of size 32 and 16, respectively (each circle in hidden layers represents 4 neurons). The DNN outputs means μ^i and standard deviations σ^i for i truncated Gaussian distributions.

define $P^*(m)$ and $\hat{P}_\phi^*(m)$ 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

$$\gamma(m) := \left| \frac{\hat{P}_\phi^*(m) - P^*(m)}{P^*(m)} \right|. \quad (33)$$

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.

B. 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 $\mathbf{h} := [h^1; \dots; h^m]$, 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 $f^i(p^i(\mathbf{h}), \mathbf{h}) := \log(1 + h^{ii} p^i(\mathbf{h}) / (v^i + \sum_{j \neq i} h^{ji} p^j(\mathbf{h})))$ and the DNN-parameterized problem is writ-

ten as

$$\begin{aligned} P_\phi^* := \max_{\boldsymbol{\theta}, \mathbf{x}} \sum_{i=1}^m w^i x^i \\ \text{s. t. } x^i \leq \mathbb{E}_{\mathbf{h}} \left[\log \left(1 + \frac{h^i \phi^i(\mathbf{h}, \boldsymbol{\theta})}{v^i + \sum_{j \neq i} h^j \phi^j(\mathbf{h}, \boldsymbol{\theta})} \right) \right], \forall i \\ \mathbb{E}_{\mathbf{h}} \left[\sum_{i=1}^m \phi^i(\mathbf{h}, \boldsymbol{\theta}) \right] \leq p_{\max}. \end{aligned} \quad (34)$$

Here, the coupling of the resource policies in the capacity constraint make the problem in (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 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 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 transmit with power $p = 6$. While not a model free method, we also compare performance against the well-known heuristic method WMMSE [6]. 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 (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 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

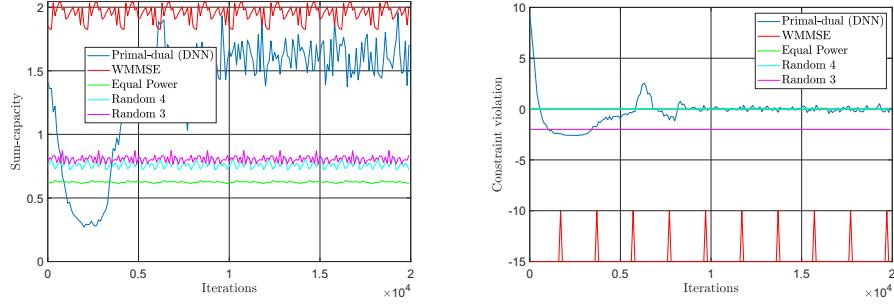


Fig. 7: Convergence of (left) objective function value and (right) constraint value for interference capacity problem in (34) using proposed DNN method, WMMSE, and simple model free heuristic power allocation strategies $m = 20$ users. The DNN-based primal dual method learns a policy that achieves close performance to WMMSE, better performance than the other model free heuristics, and moreover converges to a feasible solution.

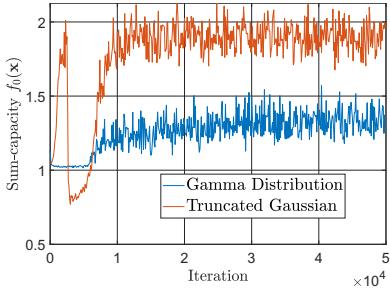


Fig. 8: Comparison of performance using Gamma and truncated Gaussian distributions in output layer of a DNN.

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 2. In this case, the SNIR for transmission i can be given as $\text{SNIR}^i := h^{ii} p_0 \alpha^i(\mathbf{h}) / (v^i + p_0 \sum_{j \neq i} h^{ji} \alpha^j(\mathbf{h}))$. The DNN-parameterized problem is written as

$$\begin{aligned} P_{\phi}^* := \max_{\theta, \mathbf{x}} & \sum_{i=1}^m w^i x^i \\ \text{s. t. } & x^i \leq \mathbb{E}_{\mathbf{h}} \left[\log \left(1 + \frac{h^{ii} p_0 \phi^i(\mathbf{h}, \theta)}{v^i + p_0 \sum_{j \neq i} h^{ji} \phi^j(\mathbf{h}, \theta)} \right) \right], \forall i \\ & \mathbb{E}_{\mathbf{h}} \left[\sum_{i=1}^m \phi^i(\mathbf{h}, \theta) \right] \leq p_{\max}, \quad \phi(\mathbf{h}, \theta) \in \{0, 1\}^m. \end{aligned} \quad (35)$$

As the case in (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 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 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 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.

VII. 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.

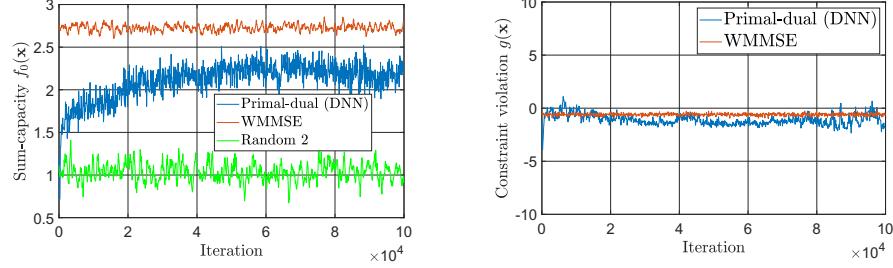


Fig. 9: Convergence of (left) objective function value and (right) constraint value for interference capacity problem in (35) using proposed DNN method, heuristic WMMSE method, and the equal power allocation heuristic for $m = 5$ users. The DNN-based primal dual method learns a policy that is feasible and almost matches the WMMSE method in terms of achieved sum-capacity, without having access to capacity model.

APPENDIX A PROOF OF THEOREM 1

To inform the analysis of the suboptimality of D_ϕ^* from (11), we first present an established result previously referenced, namely the null duality gap property of the original problem in (7). We proceed by presenting the associated Lagrangian function and dual problem for the constrained optimization problem in (2):

$$\mathcal{L}(\mathbf{p}(\mathbf{h}), \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}) := g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) + \boldsymbol{\lambda}^T (\mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] - \mathbf{x}), \quad (36)$$

$$D^* := \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \max_{\mathbf{p} \in \mathcal{P}, \mathbf{x} \in \mathcal{X}} \mathcal{L}(\mathbf{p}(\mathbf{h}), \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\lambda}). \quad (37)$$

Despite non-convexity of (2), a known result established in [10] 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 (11), we present this theorem here for reference.

Theorem 3. [10, Theorem 1] Consider the optimization problem in (2) and its Lagrangian dual in (37). Provided that Assumptions 1 and 2 hold, then the problem in (2) exhibits null duality gap, i.e., $P^* = D^*$.

With this result in mind, we begin to establish the result in (14) by considering the upper bound. First, note that the dual problem of (7) defined in (11) can be written as

$$D_\phi^* = \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \left\{ \max_{\mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} + \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] \right\}. \quad (38)$$

Focusing on the second term, observe then that for any solution $\mathbf{p}^*(\mathbf{h})$ of (2), it holds that

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] &= \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})] \\ &+ \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) - \mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})]. \end{aligned} \quad (39)$$

Since $\mathcal{P}_\phi \subseteq \mathcal{P}$, it must be that $g_0(\mathbf{x}_\theta^*) \leq g_0(\mathbf{x}^*)$, where \mathbf{x}^* and \mathbf{x}_θ^* are the maximizers of (2) and (7) respectively. Be-

cause g_0 is monotonically non-decreasing, the ergodic constraint holds with equality and $g_0(\mathbf{x}_\theta^*) \leq g_0(\mathbf{x}^*)$ implies that

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

where $\boldsymbol{\theta}^*$ is a solution of (7). By optimality, it holds that $\mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) - \mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})] \leq 0$ for all $\boldsymbol{\theta} \in \Theta$. Since $\boldsymbol{\lambda} \geq 0$, (39) yields

$$\max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] \leq \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})]. \quad (40)$$

Substituting (40) back into (38) and using the strong duality result from Theorem 3, we obtain

$$\begin{aligned} D_\phi^* &\leq \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq 0} \max_{\mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ &\quad + \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}^*(\mathbf{h}), \mathbf{h})] = D^* = P^*, \end{aligned} \quad (41)$$

where we used the fact that the right-hand side of the inequality in (41) is the optimal dual value of problem (2) as defined in (37).

We prove the lower bound in (14) by proceeding in a similar manner, i.e., by manipulating the expression of the dual value in (38). In contrast to the previous bound, however, we obtain a perturbed version of (2) which leads to the desired bound. Explicitly, notice that for all $\mathbf{p} \in \mathcal{P}$ it holds that

$$\begin{aligned} \max_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] &= \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] \\ &\quad - \min_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})], \end{aligned} \quad (42)$$

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 (42) as

$$\begin{aligned} \min_{\boldsymbol{\theta} \in \Theta} \boldsymbol{\lambda}^T \mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})] \\ \leq \|\boldsymbol{\lambda}\|_1 \left[\min_{\boldsymbol{\theta} \in \Theta} \|\mathbb{E}_\mathbf{h} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\phi(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})]\|_\infty \right]. \end{aligned} \quad (43)$$

To upper bound the minimization in (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

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

The definition in (8) then readily gives

$$\min_{\theta \in \Theta} \|\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h}) - \mathbf{f}(\phi(\mathbf{h}, \theta), \mathbf{h})]\|_{\infty} \leq L\epsilon. \quad (44)$$

Substituting (43) and (44) into (42) yields

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

which we can then use in the definition of the dual value (38) to obtain

$$\begin{aligned} D_{\phi}^* \geq \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} \max_{\mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ + \boldsymbol{\lambda}^T \mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] - \|\boldsymbol{\lambda}\|_1 L\epsilon. \end{aligned} \quad (45)$$

We are now ready to derive the perturbed version of (2) in order to obtain our lower bound. To do so, notice that $\boldsymbol{\lambda} \geq \mathbf{0}$ implies that $\|\boldsymbol{\lambda}\|_1 = \boldsymbol{\lambda}^T \mathbf{1}$, where $\mathbf{1}$ is a column vector of ones. Since (45) holds for all $\mathbf{p} \in \mathcal{P}$, we get

$$\begin{aligned} D_{\phi}^* \geq \min_{\boldsymbol{\lambda}, \boldsymbol{\mu} \geq \mathbf{0}} \max_{\mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{x} \\ + \max_{\mathbf{p} \in \mathcal{P}} \{\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] - L\epsilon \mathbf{1}\}. \end{aligned} \quad (46)$$

Now, observe that the right-hand side of (46) is the dual value of an $(L\epsilon)$ -perturbed version of (2)

$$\begin{aligned} P_{L\epsilon}^* := \max_{\mathbf{p}, \mathbf{x}} C(\mathbf{x}) \\ \text{s. t. } L\epsilon \mathbf{1} + \mathbf{x} \leq \mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})], \quad \mathbf{0} \leq \mathbf{g}(\mathbf{x}), \\ \mathbf{x} \in \mathcal{X}, \quad \mathbf{p} \in \mathcal{P} \end{aligned} \quad (47)$$

Naturally, (47) has the same strong duality property as (2) from Theorem 3, which implies that $D_{\phi}^* \geq D_{L\epsilon}^* = P_{L\epsilon}^*$. A well-known perturbation inequality, e.g., [28, Eq. (5.57)], relates $P_{L\epsilon}^*$ to P^* as

$$P_{L\epsilon}^* \geq P^* - \|\boldsymbol{\lambda}^*\|_1 L\epsilon. \quad (48)$$

Combining (48) with $D_{\phi}^* \geq P_{L\epsilon}^*$, we obtain (14).

We proceed to prove the bound in (15). Note that the strong duality result in Theorem 3 implies that

$$\begin{aligned} P^* = D^* = \max_{\mathbf{p} \in \mathcal{P}, \mathbf{x} \in \mathcal{X}} g_0(\mathbf{x}) + \boldsymbol{\mu}^{*T} \mathbf{g}(\mathbf{x}) \\ + \boldsymbol{\lambda}^{*T} (\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})] - \mathbf{x}) \\ \geq g_0(\mathbf{x}') + \boldsymbol{\mu}^{*T} \mathbf{g}(\mathbf{x}') \\ + \boldsymbol{\lambda}^{*T} (\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))] - \mathbf{x}'), \end{aligned} \quad (49)$$

where $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ are the minimizers of (37) and $(\mathbf{x}', \mathbf{p}')$ are arbitrary feasible points of (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 $\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))] > \mathbf{x}'$, to obtain

$$\begin{aligned} P^* &\geq g_0(\mathbf{x}') + \boldsymbol{\lambda}^{*T} (\mathbb{E}_{\mathbf{h}} [\mathbf{f}(\mathbf{h}, \mathbf{p}'(\mathbf{h}))] - \mathbf{x}') \\ &\geq g_0(\mathbf{x}') + \boldsymbol{\lambda}^{*T} \mathbf{1} \cdot s, \end{aligned} \quad (50)$$

where $s = \min_i \mathbb{E}_{\mathbf{h}} [f_i(\mathbf{h}, \mathbf{p}'(\mathbf{h}))] - x'_i$ as defined in Assumption 2 with $\mathbf{f} = [f_i]$ and $\mathbf{x}' = [x'_i]$. Note that $s > 0$ since \mathbf{x}' is strictly feasible. Finally, since $\boldsymbol{\lambda}^* \geq \mathbf{0}$, we can rearrange (50) to obtain

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}')}{s}. \quad (51)$$

APPENDIX B PROOF OF THEOREM 2

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 4. [31, Theorem 2.2] Define $\mathcal{D} = \{\phi(\cdot, \theta) : \theta \in \mathbb{R}^q\}$ to be the set of all functions described by the DNN in (29) with σ_l non-constant and continuous for all $l = 1, \dots, L$. 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 $\theta \in \mathbb{R}^q$ such that

$$m(\{\mathbf{h} \in \mathcal{H} : \|\hat{\mathbf{p}}(\mathbf{h}) - \phi(\mathbf{h}, \theta)\|_{\infty} > \tilde{\epsilon}\}) < \tilde{\epsilon}. \quad (52)$$

Using Theorem 4, we can show that DNNs satisfy the ϵ -universality condition from Definition 1 for all $\epsilon > 0$.

Lemma 1. The entire class of DNN parameterizations $\phi \in \Phi$, where ϕ is defined in (29) with non-constant, continuous activation σ_l for some layer size $q > 0$, is an ϵ -universal parameterization as in Definition 1 for all $\epsilon > 0$.

Proof. Let $\mathcal{K}_{\epsilon'} = \{\mathbf{h} \in \mathcal{H} : \|\hat{\mathbf{p}}(\mathbf{h}) - \phi(\mathbf{h}, \theta)\|_{\infty} > \epsilon'\} \subseteq \mathcal{H}$ and observe that (8) can be written as

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

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 (52), we obtain the following bound over all DNNs ϕ , i.e.

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

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

$$\inf_{\phi \in \Phi} \mathbb{E} \|\mathbf{p}(\mathbf{h}) - \phi(\mathbf{h}, \theta)\|_{\infty} < (1 + 2\Gamma)\epsilon'. \quad (53)$$

Taking $\epsilon' = \epsilon/(1 + 2\Gamma)$ in (53) yields (8). \square

Lemma 1 implies that the dual value bound (14) from Theorem 1 holds for all $\epsilon > 0$ if we consider the entire class of DNN functions $\phi \in \Phi$. Since the Lipschitz constant $L < \infty$, the only obstacle to completing a continuity argument is if $\|\boldsymbol{\lambda}^*\|_1$ is unbounded. However, recall from (15) that

$$\|\boldsymbol{\lambda}^*\|_1 \leq \frac{P^* - g_0(\mathbf{x}_0)}{s} < \infty. \quad (54)$$

Hence, we obtain that

$$P^* - \delta \leq \inf_{\phi \in \Phi} D_\phi^* \leq P^*$$

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

REFERENCES

- [1] X. Wang and G. B. Giannakis, "Resource allocation for wireless multiuser OFDM networks," *IEEE Transactions on Information Theory*, vol. 54, no. 6-1, pp. 2239–2251, 2006.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] ———, "Optimal wireless networks based on local channel state information," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4913–4929, 2012.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] A. Ribeiro, "Optimal resource allocation in wireless communication and networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 272, 2012.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] M. Eisen, K. Gatsis, G. J. Pappas, and A. Ribeiro, "Learning in wireless control systems over nonstationary channels," *IEEE Transactions on Signal Processing*, vol. 67, no. 5, pp. 1123–1137, 2018.
- [15] 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.
- [16] 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.
- [17] V. Ntranos, N. D. Sidiropoulos, and L. Tassiulas, "On multicast beamforming for minimum outage," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, 2009.
- [18] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.
- [24] P. de Kerret, D. Gesbert, and M. Filippone, "Decentralized deep scheduling for interference channels," *arXiv preprint arXiv:1711.00625*, 2017.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [29] J. Park and I. W. Sandberg, "Universal approximation using radial-basis-function networks," *Neural computation*, vol. 3, no. 2, pp. 246–257, 1991.
- [30] 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.
- [31] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [39] 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.