Online Deep Learning in Wireless Communication Systems

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

***Abstract*—We consider the problem of optimal power allocation in wireless fading channels. Due to interference, this problem is non-convex and challenging to solve exactly. The resemblance of this problem to a statistical loss problem motivates the use of a learning parameterization of the power allocation function. In particular, we use deep neural networks (DNNs) to represent the power allocation and develop a primal-dual learning method to train the weights of the DNN. Because the channel and capacity models may not be known in practice, we extend the learning algorithm to permit stochastic online operation, in which gradients are approximated by sampling. We demonstrate in a series of numerical simulations the performance of the proposed online primal-dual learning method in training a DNN-parameterization relative to a well-known heuristic benchmark.**

***Index Terms*—Wireless communications, deep learning, interference channel, power control**

I. Introduction

We consider the problem of power allocation across a wireless random fading channel. The goal is to design a power allocation policy across a series of transmitters that takes into account the instantaneous fading state. Optimal design requires the resources be allocated in such a way that maximizes a utility of the expected capacity across the random channel. Problems of this form range from the simple power allocation in wireless fading channels to the optimization of frequency division multiplexing [1], beamforming [2], [3], and random access [4], [5]. Perhaps the most prevalent of such resource allocation problems considers the problem of allocating power across multiple transmitter/receiver pairs in an interference channel.

The optimal power allocation problem when considering interference is generally very challenging to solve. Not only is it non-convex, but also requires optimization of an infinite dimensional variable. Moreover, more sophisticated formulations of this problem, such as those that consider fairness criteria, are subject to additional constraints in the optimization problem. Some simpler cases of this problem can be solved in the Lagrnagian dual domain due to an inherent lack of duality gap in the problem [6], [7]. This permits dual domain operation in a wide class of resource allocation problems [8]–[12]. For the power management problem in an interference channel, specific heuristic methods have been developed over the years to find approximate solutions [13]–[17]. The WMMSE [13] is, in particular, commonly used in practice.

As an alternative to model-based heuristics, machine learning techniques have been applied to this problem. Some approaches use existing heuristics to construct a training set of labelled solutions, against which they can train a deep neural network to approximate the heuristic method directly [19]–[21]. 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. Rather than construct a training set, one could parameterize the power allocation policy directly using a neural network or other sophisticated learning model and train with respect to a given performance criteria. This setting is typical of, e.g., reinforcement learning problems [22], and is a learning approach that has been taken in several *unconstrained* problems in wireless resource allocation [23]. However, such approaches are not directly applicable when constraints are added, but may be augmented by adding a simple penalty function to the reward objective [24].

We begin the paper by formulating the problem of optimal power allocation in a wireless fading channel with interference (Section II). We generalize the classical formulation by considering a potentially unknown capacity function as well as additional constraints on the average capacity. We transform this problem to a constrained regression problem by parameterizing the power allocation policy with a deep neural network (DNN) (Section II-A). We derive an unconstrained reformulation of this problem through the Lagrangian dual problem, with which we develop a primal-dual learning algorithm (Section III). Because capacity and channel models are not necessarily known, we present an online implementation that estimates the gradients through sampling the channel (Section III-A). Finally, we present numerical simulations that compare the performance of the online learning method relative to a benchmark heuristic method (Section IV).

II. Optimal Power Management

Consider a series of m transmitter/receiver pairs and let $\mathbf{h} \in \mathcal{H} \subseteq \mathbb{R}_+^{m^2}$ be a matrix of random variables representing the collection of m^2 stationary wireless fading channels drawn according to the probability distribution $m(\mathbf{h})$. Each element h_{ij} reflects the fading interference channel between transmitter i and receiver j, while h_{ii} represents the direct link channel between transmitter and receiver i. Associated with each fading channel realization, we have a power allocation variable $\mathbf{p}(\mathbf{h}) \in \mathbb{R}^m$ and a capacity function

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.

$\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^{m^2} \to \mathbb{R}^m$. The power allocation is divided into m components $\mathbf{p}(\mathbf{h}) = [p_1(\mathbf{h}); \ldots; p_m(\mathbf{h})]$, where $p_i(\mathbf{h})$ is the power allocated to transmitter i. In the case of fast fading channels, the system allocates resources instantaneously but users experience the average performance across fading channel realizations. This motivates considering the vector ergodic average $\mathbf{x} = \mathbb{E}\big[\mathbf{f}\big(\mathbf{p}(\mathbf{h}), \mathbf{h}\big)\big] \in \mathbb{R}^m$, which, for formulating optimal wireless design problems, is relaxed to the inequality

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

The goal in optimal power management is to find the instantaneous power allocation $\mathbf{p}(\mathbf{h})$ that optimizes a weighted summation of the capacity experienced by all receivers. In general, it is known for a standard AWGN interference channel with noise variance σ^2, that the capacity of receiver i can be written as $f_i(\mathbf{p}(\mathbf{h}), \mathbf{h}) = \log(1 + \text{SNIR}_i)$, where SNIR_i represents the signal to noise plus interference ratio experienced by receiver i. This term is typically modeled as $\text{SNIR}_i = \frac{|h_{ii}|^2 p_i(\mathbf{h})}{\sigma^2 + \sum_{j \neq i} |h_{ji}|^2 p_j(\mathbf{h})}$. The constrained weighted sum-maximization problem can then be written as

$$\begin{aligned} P^* := \max_{\mathbf{p}(\mathbf{h}), \mathbf{x}} \ & \mathbf{w}^T \mathbf{x}, \\ \text{s.t.} \quad & x_i \leq \log\left(1 + \frac{|h_{ii}|^2 p_i(\mathbf{h})}{\sigma^2 + \sum_{j \neq i} |h_{ji}|^2 p_j(\mathbf{h})}\right) \forall i, \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \ \mathbf{p}(\mathbf{h}) \in [0, p_{\max}]^m. \end{aligned} \tag{2}$$

In (2), introduce the set of positive weights $\mathbf{w} \geq \mathbf{0}$ associated with each pair and a maximum transmit power $p_{\max}$. We further introduce a generic set of constraint $\mathbf{g}(\mathbf{x})$ that can be placed on the system. These constraints may reflect, for instance, a minimum average capacity for all users to achieve in the power allocation policy. The inclusion of such constraints is not standard in these problems, but we include them here for full generality so that the proposed learning method can be applied in such augmented problems. Finally, we note that while an analytic expression for the interference capacity is given in (2), this is only an idealistic model and the true value of the capacity $\mathbf{f}(\mathbf{p}(\mathbf{h}), \mathbf{h})$ or the channel distribution $m(\mathbf{h})$ may not be known in practice. Later in this paper we discuss how this term can be estimated within the online learning method.

A. Deep learning parameterization

The problem in (2), which formally characterizes the optimal power allocation policies in a constrained interference channel, is generally a very difficult optimization problem. The problem itself is non-convex and therefore cannot be solved exactly. Moreover, from a computational standpoint, the problem in (2) is challenging to operate on due to both the infinite dimensionality of the power allocation variable $\mathbf{p}(\mathbf{h})$, as well as the inclusion of constraints. The approach taken in this work is one of *learning*, or more precisely, recognizing that the power management problem in (2) takes the particular form of a statistical learning—or regression—problem. Thus, our first step to reduce the dimensionality of the problem is to introduce a parametrization of the power allocation function. For some parameter $\boldsymbol{\theta} \in \mathbb{R}^q$ we introduce a learning model $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$, i.e.

$$\mathbf{p}(\mathbf{h}) = \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}). \tag{3}$$

With this parametrization the ergodic constraint in (1) becomes

$$\mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}\big)\big] \tag{4}$$

The optimization problem in (2) becomes one in which the optimization is over $\mathbf{x}$ and $\boldsymbol{\theta}$

$$\begin{aligned} P^*_{\boldsymbol{\phi}} := \max_{\boldsymbol{\theta}, \mathbf{x}} \ & \mathbf{w}^T \mathbf{x}, \\ \text{s.t.} \quad & x_i \leq \log\left(1 + \frac{|h_{ii}|^2 \phi_i(\mathbf{h}, \boldsymbol{\theta})}{\sigma^2 + \sum_{j \neq i} |h_{ji}|^2 \phi_j(\mathbf{h}, \boldsymbol{\theta})}\right) \forall i, \\ & \mathbf{g}(\mathbf{x}) \geq \mathbf{0}, \ \boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in [0, p_{\max}]^m. \end{aligned} \tag{5}$$

In (5), we restrict our attention to power allocation policies that can be represented with the form $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ for some parameter vector $\boldsymbol{\theta}$. Since the optimization is now carried over the parameter $\boldsymbol{\theta} \in \mathbb{R}^q$ and the ergodic variable $\mathbf{x} \in \mathbb{R}^m$, the number of variables in (5) is $q + m$ rather than infinite dimensionality of the problem in (2).

While the parameterization in (5) makes the problem easier in term of dimensionality, this comes at a loss of of optimality because (3) restricts power allocation functions to adhere to a parametrization. The choice of parameterization and its ability to represent arbitrary functions is therefore important. In this work, we focus our attention on a widely-used learning parameterization called *deep neural networks* (DNN). The exact form of a particular DNN is described by what is commonly referred to as its *architecture*. The architecture consists of a prescribed number of layers, each of which consisting of a linear operation followed by a point-wise nonlinearity—also known as an activation function. In particular, consider a DNN with L layers, labelled $l = 1, \ldots, L$ and each with a corresponding dimension q_l. The layer l is defined by the linear operation $\mathbf{W}_l \in \mathbb{R}^{q_{l-1} \times q_l}$ followed by a non-linear activation function $\boldsymbol{\sigma}_l : \mathbb{R}^{q_l} \to \mathbb{R}^{q_l}$. Common choices of activation functions $\boldsymbol{\sigma}_l$ include a sigmoid function, a rectifier function (commonly referred to as ReLu), as well as a smooth approximation to the rectifier known as softplus. If layer l receives as an input from the $l-1$ layer $\mathbf{w}_{l-1} \in \mathbb{R}^{q_{l-1}}$, the resulting output $\mathbf{w}_l \in \mathbb{R}^{q_l}$ is then computed as $\mathbf{w}_l := \boldsymbol{\sigma}_l(\mathbf{W}_l \mathbf{w}_{l-1})$. The final output of the DNN, $\mathbf{w}_L$, is then related to the input $\mathbf{w}_0$ by propagating through each later of the DNN as $\mathbf{w}_L = \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{w}_0))))))$.

For the parameterized resource allocation problem in (5), the policy $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ can be defined by an L-layer DNN as

$$\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) := \boldsymbol{\sigma}_L(\mathbf{W}_L(\boldsymbol{\sigma}_{L-1}(\mathbf{W}_{L-1}(\ldots(\boldsymbol{\sigma}_1(\mathbf{W}_1 \mathbf{h})))))), \tag{6}$$

where $\boldsymbol{\theta} \in \mathbb{R}^q$ contains the entries of $\{\mathbf{W}_l\}_{l=1}^L$ with $q = \sum_{l=1}^{L-1} q_l q_{l+1}$. To enforce the box constraint $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in [0, p_{\max}]$, we may set the last, or output, layer $\boldsymbol{\sigma}_L$ to output values in this region.

Deep neural networks are powerful parameterizations due to their well-known property of universality—in other words, they have the power to approximate any continuous function arbitrarily well as the layer size grown large [25]. This is a useful property as it gives reason to suspect that a DNN may provide a solution $P^*_{\boldsymbol{\phi}}$ that is close to that of P^*. Additionally, they are popular learning models because of the empirical evidence that suggests that, despite their non-convexity, they contain many strong local optima that can easily be found using gradient descent methods. The constrained formulation of the learning problem in (5) makes it impossible to apply gradient descent algorithms directly. Thus, in the next section we consider a dual formulation of (5) that naturally leads to such a gradient-based learning method.

III. Primal-Dual Learning

To solve (5), we must learn both the parameter $\boldsymbol{\theta}$ and the ergodic average variables $\mathbf{x}$ over a set of both convex and non-convex constraints. Doing so requires we reformulate (5) as an *unconstrained* optimization problem. This can be done by formulating and solving the Lagrangian dual problem. We introduce the nonnegative multiplier dual variables $\boldsymbol{\lambda} \in \mathbb{R}^m_+$ and $\boldsymbol{\mu} \in \mathbb{R}^r_+$, respectively associated with the constraints $\mathbf{x} \leq \mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big]$ and $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$. The Lagrangian of (5) is then written as

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta},\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) := \mathbf{w}^T\mathbf{x} + \boldsymbol{\mu}^T\mathbf{g}(\mathbf{x}) \\ + \boldsymbol{\lambda}^T\Big(\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big] - \mathbf{x}\Big).\end{aligned} \tag{7}$$

With the Lagrangian so defined, we introduce the dual function $D_{\boldsymbol{\phi}}(\boldsymbol{\lambda},\boldsymbol{\mu})$ as the maximum Lagrangian value attained over the so-called *primal* variables $\mathbf{x},\boldsymbol{\theta}$, i.e.

$$D_{\boldsymbol{\phi}}(\boldsymbol{\lambda},\boldsymbol{\mu}) := \max_{\boldsymbol{\theta},\mathbf{x}} \mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta},\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{8}$$

The dual function in (8) is a penalized version of (5) in which the constraints are not enforced but their violation is penalized by the Lagrangian terms $\boldsymbol{\mu}^T\mathbf{g}(\mathbf{x})$ and $\boldsymbol{\lambda}^T(\mathbb{E}\big[\mathbf{f}\big(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}),\mathbf{h}\big)\big] - \mathbf{x})$. This is distinct, however, from general penalty methods in that the dual weights $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are themselves treated as variables to optimize. This unconstrained formulation in (8) is analogous to conventional learning objectives and, as such, a problem that we can solve with conventional learning algorithms.

In particular, the dual problem is one in which we minimize the dual function in (8) over the dual variables, thus rending a min-max, or saddle point, optimization problem over the Lagrangian, i.e.

$$D^*_{\boldsymbol{\phi}} := \min_{\boldsymbol{\lambda},\boldsymbol{\mu}\geq\mathbf{0}} \max_{\boldsymbol{\theta},\mathbf{x}} \mathcal{L}_{\boldsymbol{\phi}}(\boldsymbol{\theta},\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{9}$$

The dual optimum $D^*_{\boldsymbol{\phi}}$ is the best approximation we can have of $P^*_{\boldsymbol{\phi}}$ when using (8) as a proxy for (5). In fact, under some standard assumptions on the problem and assuming a sufficiently dense DNN architecture, we can formally bound the difference between $D^*_{\boldsymbol{\phi}}$ and P^*—see [26] for details on this result.

With the unconstrained saddle point problem in (9), we may perform standard gradient-based optimization methods to obtain solutions. In particular, we present a *primal-dual* optimization method, in which we perform gradient updates directly on both the primal and dual variables of the Lagrangian function in (7) to find a local stationary point of the KKT conditions of (5). 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 (7), i.e. $\nabla_{\boldsymbol{\theta}}\mathcal{L},\nabla_{\mathbf{x}}\mathcal{L}$, i.e.,

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k}\nabla_{\boldsymbol{\theta}}\mathbb{E}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}_k),\mathbf{h})\boldsymbol{\lambda}_k, \tag{10}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_{\mathbf{x},k}(\mathbf{w} + \nabla\mathbf{g}(\mathbf{x}_k)\boldsymbol{\mu}_k - \mathbf{x}_k), \tag{11}$$

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

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k}\left(\mathbb{E}_{\mathbf{h}}\mathbf{f}(\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta}_{k+1}),\mathbf{h}) - \mathbf{x}_{k+1}\right)\right]_+, \tag{12}$$

$$\boldsymbol{\mu}_{k+1} = \left[\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k}\mathbf{g}(\mathbf{x}_{k+1})\right]_+, \tag{13}$$

with associated step sizes $\gamma_{\boldsymbol{\lambda},k},\gamma_{\boldsymbol{\mu},k} > 0$. The gradient primal-dual updates in (10)-(13) successively move the primal and dual variables towards maximum and minimum points of the Lagrangian function, respectively.

The primal-dual learning method presented above is a straightforward approach for learning the DNN parameter $\boldsymbol{\theta}$ simultaneously with the dual parameters $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. However, observe that the updates in (10) and (12) require knowledge of either the capacity function $\mathbf{f}$ or the channel distribution $m(\mathbf{h})$, or both. While capacity models exist, such as the expression given in (2) and (5), they do not necessarily hold in practical channels. Moreover, models for $m(\mathbf{h})$ do not always hold well in practical channels. With that in mind, we may extend the primal-dual methods in (10) and (12) to handle stochastic *online* operation by sampling the achieved capacity and channel conditions throughout the training process.

A. Online learning

To develop the online learning method, consider that given any set of iterates and channel realization $\{\tilde{\boldsymbol{\theta}},\tilde{\mathbf{h}}\}$, we can observe a noisy function values $\hat{\mathbf{f}}(\tilde{\mathbf{h}},\boldsymbol{\phi}(\tilde{\mathbf{h}},\tilde{\boldsymbol{\theta}}))$ by, say, passing test signal through the channel and measuring the capacity. These observations can be interpreted as unbiased estimates of the true function values. We can then replace the updates in (10) and (12) with gradient estimates using these on-line observations. The so-called *policy gradient* method exploits a likelihood ratio property found in such functions to allow for a simple gradient estimate. To derive the details of the policy gradient method, consider that a deterministic policy $\boldsymbol{\phi}(\mathbf{h},\boldsymbol{\theta})$ can be approximated with a *stochastic* policy drawn from a distribution with a known density function $\pi(\mathbf{p})$ defined with a delta function. It can be shown that the Jacobian of the policy

Algorithm 1 Online Primal-Dual Learning

1: **Parameters:** Policy model $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta})$ 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** $k = 0, 1, 2, \ldots$ **do** {main loop}
4: Draw samples $\{\hat{\boldsymbol{\theta}}, \hat{\mathbf{h}}\}$, or in batches of size B
5: Obtain random observation of function values $\hat{\mathbf{f}}$ at current and sampled iterates
6: Compute policy gradient estimate $\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h},\boldsymbol{\phi}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h})$ [c.f. (15)]
7: Update primal and dual variables

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k} \widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h}) \boldsymbol{\lambda}_k, [cf.(16)]$$
$$\mathbf{x}_{k+1} = \mathbf{x}_k + \gamma_{\mathbf{x},k}(\mathbf{w} + \nabla \mathbf{g}(\mathbf{x}_k) \boldsymbol{\mu}_k - \mathbf{x}_k), [cf.(11)]$$
$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k} \left(\hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_{k+1}), \hat{\mathbf{h}}) - \mathbf{x}_{k+1}\right)\right]_+ [cf.(17)]$$
$$\boldsymbol{\mu}_{k+1} = [\boldsymbol{\mu}_k - \gamma_{\boldsymbol{\mu},k} \mathbf{g}(\mathbf{x}_{k+1})]_+ . [cf.(13)]$$

8: **end for**

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

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \mathbb{E}_{\mathbf{h},\mathbf{p}}[\mathbf{f}(\mathbf{p}, \mathbf{h}) \nabla_{\boldsymbol{\theta}} \log \pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})^T], \quad (14)$$

where $\mathbf{p}$ is a random variable drawn from distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$—see, e.g., [27]. Observe in (14) that the computation of the Jacobian reduces to a function evaluation multiplied by the gradient of the policy distribution $\nabla_{\boldsymbol{\theta}} \log \pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$. By using the previous function observations, we can obtain the following policy gradient estimate,

$$\widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}), \mathbf{h}) = \hat{\mathbf{f}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}}, \hat{\mathbf{h}}) \nabla_{\boldsymbol{\theta}} \log \pi_{\hat{\mathbf{h}},\boldsymbol{\theta}}(\hat{\mathbf{p}}_{\boldsymbol{\theta}})^T, \quad (15)$$

where $\hat{\mathbf{p}}_{\boldsymbol{\theta}}$ is a sample drawn from the distribution $\pi_{\mathbf{h},\boldsymbol{\theta}}(\mathbf{p})$. Thus, we can replace the updates in (10) and (12) with the stochastic online updates

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \gamma_{\boldsymbol{\theta},k} \widehat{\nabla_{\boldsymbol{\theta}}} \mathbb{E}_{\mathbf{h}} \mathbf{f}(\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}_k), \mathbf{h}) \boldsymbol{\lambda}_k, \quad (16)$$
$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \gamma_{\boldsymbol{\lambda},k} \left(\hat{\mathbf{f}}(\boldsymbol{\phi}(\hat{\mathbf{h}}, \boldsymbol{\theta}_{k+1}), \hat{\mathbf{h}}) - \mathbf{x}_{k+1}\right)\right]_+ . \quad (17)$$

The complete online primal-dual learning algorithm is summarized in Algorithm 1.

IV. Numerical Results

In this section, we perform simulation experiments to compare the performance of the policies learned via the primal-dual method with DNNs against the performance of the popular WMMSE heuristic [13] for solving the power management problem in (2). For the simulations performed, we employ a stochastic policy and implement the policy gradient described in Section III-A. In particular, we select the policy distribution $\pi_{\boldsymbol{\theta},\mathbf{h}}$ as a truncated Gaussian distribution. The truncated Gaussian distribution has fixed support on the domain $[0, p_{\max}]$. The output layer of the DNN $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) \in \mathbb{R}^{2m}$ is the set of m means and standard deviations to specify the respective truncated Gaussian distributions, i.e. $\boldsymbol{\phi}(\mathbf{h}, \boldsymbol{\theta}) := [\mu^1; \sigma^1; \mu^2; \sigma^2; \ldots; \mu^m; \sigma^m]$. Furthermore, to represent policies that are bounded on the support interval, the output of the last layer is fed into a scaled sigmoid function such that the mean lies in the area of support and the variance is no more than the square root of the support region. For updating the primal and dual variables, we sample channel conditions and capacity values with batch sizes of 64 samples per training iteration. The primal dual method is performed with an exponentially decaying step size for dual updates and the ADAM optimizer [28] for the DNN parameter update. Both updates start with a learning rate of 0.0005, while random channel conditions are generated with a Rayleigh distribution.

We learn a power allocation policy for a set of $m = 8$ pairs. For this setting, we construct a DNN architecture with two hidden layers, of size 32 and 16, respectively. In addition, each layer is given a ReLU activation function, i.e. $\sigma(\mathbf{z}) = [\mathbf{z}]_+$. In Figure 1a we show the performance, or sum-capacity, achieved throughout the learning process of the DNN in comparison with the performance of the WMMSE heuristic. Observe that the DNN parameterization is indeed able to outperform WMMSE, achieving a sum-rate capacity of 1.6 after 3×10^4 learning iterations. We point out that the number of learning iterations is only displayed here to demonstrate the learning time, but this does not effect the computational or sampling complexity of using the DNN in runtime.

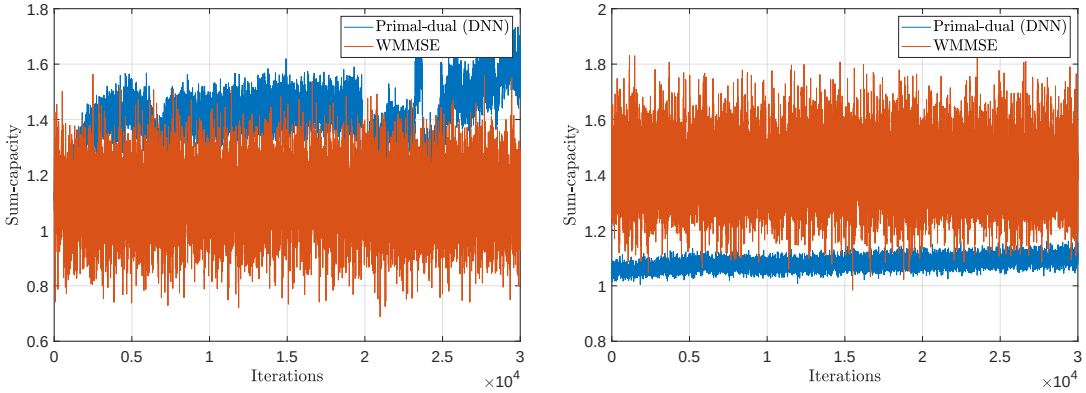
The results of of the simulations with $m = 20$ users is shown in Figure 1b. In this case, we construct a DNN architecture with two hidden layers, of size 64 and 32, respectively. Here, the DNN that is learned with the primal-dual method slightly outperforms the WMMSE algorithm. We note that, despite the small underperformance, the proposed method nonetheless does not presume any knowledge of the true capacity while also providing capabilities to add additional constraints on individual capacities. The WMMSE algorithm is not able to handle such cases. Overall, our simulation results demonstrate cases in which the DNN either outperforms, or get close to the performance of the existing heuristic approaches for solving the power management problem in wireless interference channels.

V. Conclusion

We considered the problem of optimal power allocation in a wireless fading channel with interference. We parameterize the power allocation with a deep neural network. The DNN is trained in the dual domain via a primal-dual learning method. When the system models are unknown, we extend the algorithm to permit online operation by using channel and capacity samples to construct policy gradient estimates. We perform numerical simulations that shows that the DNN learning method can achieve performance similar or better to that of the WMMSE heuristic method for solving power allocation problems.

References

[1] X. Wang and G. B. Giannakis, "Resource allocation for wireless multiuser OFDM networks," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4359–4372, 2011.



(a) (b)

Fig. 1: Performance of the power allocation policy that is parameterized by a DNN and learned with the online primal-dual method in comparison to performance of the popular WMMSE algorithm with (left) $m = 8$ users and (right) $m = 20$ users. In the first case, the DNN parameterization outperforms WMMSE while slightly outperforms in the second case.

[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] 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.

[7] A. Ribeiro, "Optimal resource allocation in wireless communication and networking," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 272, 2012.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] V. Ntranos, N. D. Sidiropoulos, and L. Tassiulas, "On multicast beamforming for minimum outage," *IEEE Transactions on Wireless Communications*, vol. 8, no. 6, 2009.

[13] 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.

[14] 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.

[15] 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.

[16] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE journal on selected areas in communications*, vol. 32, no. 6, pp. 1139–1151, 2014.

[17] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *Information Theory (ISIT), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 2323–2327.

[18] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," *arXiv preprint arXiv:1705.08044*, 2017.

[19] 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," *arXiv preprint arXiv:1705.09412*, 2017.

[20] 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.

[21] 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.

[22] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press, 1998.

[23] P. de Kerret, D. Gesbert, and M. Filippone, "Decentralized deep scheduling for interference channels," *arXiv preprint arXiv:1711.00625*, 2017.

[24] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *arXiv preprint arXiv:1807.10025*, 2018.

[25] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[26] M. Eisen, C. Zhang, L. F. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *arXiv preprint arXiv:1807.08088*, 2018.

[27] 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.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.