LEARNING-BASED RESOURCE ALLOCATION WITH DYNAMIC DATA RATE CONSTRAINTS

Pourya Behmandpoor *Panagiotis Patrinos* *Marc Moonen*

KU Leuven, Department of Electrical Engineering (ESAT)
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics

ABSTRACT

In this paper, we address the problem of resource allocation (RA) in wireless communication networks, where each user has a dynamic data rate constraint. The objective of RA is to maximize the sum rate (SR) of the users while satisfying the data rate constraints in expectation. For a given set of data rate constraints, a suitable probability distribution for the activation of users is found iteratively with a stochastic gradient descent (SGD) approach to satisfy the data rate constraints in expectation. At each time instant, RA amongst the randomly activated users is performed noniteratively by a centralized deep neural network (DNN). Simulations show that the proposed approach is convergent and not only can consider dynamic data rate constraints accurately, but also that it achieves a SR higher than that of the conventional geometric programming (GP) method. The proposed approach can open up a direction of research for cross-layer RA in the current deep learning-based RA context.

Index Terms— resource allocation, cross-layer, deep learning, dynamic data rate constraints

1. INTRODUCTION

Resource allocation (RA) in wireless communication networks is a challenging task that has been studied for decades [1, 2, 3, 4] and has been upgraded by deep learning (DL)-based approaches in recent years [5]. These DL-based approaches try to find an optimal policy that maps time-varying parts of an RA problem, e.g. the channel coefficients, to the optimal allocation of resources, e.g. the transmit power, among the users. This policy can be approximated by a deep neural network (DNN).

Among the first attempts to make use of a DNN for RA, one can mention [6, 7, 8]. For instance, [8] aims to train a centralized DNN to generalize training data labeled by the conventional RA approaches, in a supervised manner. Another approach, however, consists in training a DNN whose cost function is a weighted sum rate (WSR) of the users in the network, in an unsupervised manner [6, 7, 9]. Moreover, the model-free and distributed extensions are also introduced in the literature [10, 11, 12, 13, 14, 15].

All the mentioned DL-based RA approaches in the literature consider the maximization of a utility function, e.g. WSR, with some fixed constraints, e.g. data rate and power constraints. However, while power constraints are mostly dictated by the physical limitations of the hardware that the users have, the data rate constraints, or the minimum demands for the data rate, are dependent on the applications the users are running for a period of time. Moreover, even in the case all users have fixed data rate demands if the number of active users changes –which is common in practice– the corresponding data rate constraints should be added or eliminated to have an optimal solution. Hence, it would be more efficient if the DL-based RA process can also take into account the dynamic nature of data rate constraints. Initial efforts have been made in [16] to consider the dynamic nature of the data rate constraints in wired communication networks. However, they cannot be directly applied to wireless communication networks, as in the wired communication networks the channel coefficients are fixed while the channel coefficients are dynamic in the wireless scenario.

While the proposed methods in the literature need to retrain their policy or DNN to address a new set of data rate constraints or active users, we propose an alternative approach that can allocate resources among the users with dynamic data rate constraints, or with a dynamic number of active users. RA is done noniteratively by a trained DNN while satisfying the data rate constraints is done iteratively with a stochastic gradient descent (SGD) approach on the activation pattern of the users.

2. SYSTEM MODEL

We consider N users, each user consisting of a transmitter and a receiver communicating with each other in a wireless communication network. The transmit power of user i is p_i, the ith element of the vector $\boldsymbol{p} \in \mathbb{R}_+^N$. The direct channel be-

This research work was carried out at the ESAT Laboratory of KU Leuven, in the frame of Research Project FWO nr. G.0B1818N 'Real-time adaptive cross-layer dynamic spectrum management for fifth generation broadband copper access networks' and Fonds de la Recherche Scientifique - FNRS and Fonds voor Wetenschappelijk Onderzoek - Vlaanderen EOS Project no 30452698 '(MUSE-WINET) MUlti-SErvice WIreless NETwork'. The scientific responsibility is assumed by its authors.

tween the transmitter and receiver of user i is denoted by h_{ii}, while the interference channel between the transmitter of user j and the receiver of user i is denoted by h_{ij}. All the channel coefficients define the full channel matrix $\boldsymbol{H} \in \mathbb{C}^{N \times N}$ with h_{ij} in its ith row and jth column. The additive white Gaussian noise is assumed to be iid with power σ^2 where the power is constant and the same in all the receivers. The achievable data rate of user i can then be written as,

$$R_i(\boldsymbol{H}, \boldsymbol{p}) = \log_2 \left(1 + \frac{|h_{ii}|^2 p_i}{\sigma^2 + \sum_{j \neq i} |h_{ij}|^2 p_j}\right). \qquad (1)$$

Unlike the conventional RA methods such as [3, 2], to avoid solving an optimization problem at each *time instant* for each new *channel realization*, we consider finding a policy that maps each channel realization to the optimal user powers. By this approach, the process of RA can be reduced to only the evaluation of the policy function. We denote the output of the policy by $\boldsymbol{p}(\boldsymbol{H})$ indicating the optimum allocated power vector given the channel matrix $\boldsymbol{H}$. This policy can be found by the following optimization problem,

$$\begin{aligned} \boldsymbol{p}(\boldsymbol{H}) :=& \underset{\boldsymbol{p}}{\operatorname{argmax}} \sum_{i=1}^{N} R_i(\boldsymbol{H}, \boldsymbol{p}) \\ s.t. \quad & R_i(\boldsymbol{H}, \boldsymbol{p}) \geq r_i^{min}, \quad \forall i \in [N] \\ & 0 \leq \boldsymbol{p} \leq \boldsymbol{p}^{max}, \end{aligned} \qquad (2)$$

where $[N] := \{1, \ldots, N\}$, r_i^{min} is the data rate constraint of user i, and $\boldsymbol{p}^{max} \in \mathbb{R}_+^N$ is a vector containing power constraints. In practice, we may consider expectation of data rates over a distribution of the channel $\mathcal{D}$. This expectation may be included as we need data rates to meet the constraints over a period of time only. In practice, for some channel realizations the optimization problem may become infeasible. However, this can be tolerable for users and their running applications as long as their data rate constraints are met on average over a period of time. This approach is also employed by [10, 7] to optimize the policy.

As the above optimization is a parametric optimization with an infinite number of dimensions, we limit the policy to be a member of a set $\boldsymbol{\Phi} \subset \mathbb{R}^n$ with a finite number of dimensions. Hence, we approximate the policy $\boldsymbol{p}(\boldsymbol{H})$ with a DNN denoted by the function $\phi(\boldsymbol{H}, \boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} \in \boldsymbol{\Phi}$.

In theory, by the universal function approximation capability of DNNs [17, 18], we can train a DNN to approximate the optimal policy for dynamic data rate constraints. However, in practice, we would need a very large DNN to do so. Although a DNN with a reasonable size can be trained for policies without data rate constraints or with fixed data rate constraints [7, 6], finding such a DNN to satisfy dynamic constraints is prohibitive in practice, as also tested by [19]. In the next section, we propose a remedy to this problem.

3. RESOURCE ALLOCATION WITH DYNAMIC DATA RATE CONSTRAINTS

To be able to satisfy dynamic data rate constraints, while maximizing the SR of the users, we propose to detach the data rate constraints in (2) and treat the constrained optimization problem with only power constraints, and the data rate constraints, separately. In this way, the optimal policy parameter $\boldsymbol{\theta}^*$ can be found by training the DNN to maximize the power-constrained SR of users. Having found $\boldsymbol{\theta}^*$ in this way, the data rate constraints can be met by an iterative procedure in a time window in which the data rate constraints are assumed to be fixed.

To do so, we propose the selection vector $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_N]$ with random variables $\xi_i \in \{0, 1\}$ that take value 1 with probability κ_i meaning that user i is selected to be active, and take value 0 otherwise, meaning that user i is selected to be inactive. By detaching the data rate constraints and denoting the vector $\boldsymbol{R} = [R_1, R_2, \ldots, R_N]^T$, the policy optimization of (2) becomes,

$$\begin{aligned} \boldsymbol{\theta}^* =& \operatorname{argmax} \mathbb{E}_{\boldsymbol{H},\boldsymbol{\xi}} \{\boldsymbol{\xi}^T \boldsymbol{R}(\boldsymbol{H}, \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}))\} \\ s.t. \quad & 0 \leq \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}) \leq \boldsymbol{p}^{max}, \end{aligned} \qquad (3)$$

i.e. maximization of the SR of the active users averaged over $\boldsymbol{H}$ and $\boldsymbol{\xi}$ for a fixed set of κ_is. Here, $\boldsymbol{H}_\xi$ is the channel matrix $\boldsymbol{H}$ with rows and columns equal to zero corresponding to inactive users in $\boldsymbol{\xi}$. By the optimization problem (3), we find a policy that maps the channel realization $\boldsymbol{H}_\xi$ to the optimum powers for the active users, while the powers for the inactive users are equal to zero.

3.1. Training the DNN

With a similar approach employed by related works in DL-based RA, a DNN can be trained in an unsupervised manner to maximize the SR of randomly activated users (3). The loss function is $\mathbb{E}_{\boldsymbol{H},\boldsymbol{\xi}}\{\boldsymbol{\xi}^T \boldsymbol{R}\}$ maximized by iteratively updating the parameter vector $\boldsymbol{\theta}$ for the input matrices $\boldsymbol{H}_\xi$. In this step, $\kappa_i = 0.5$ is assumed for $i \in [N]$, to make the DNN able to address different sets of data rate constraints in the next section. The power constraints can also be considered by the output activation function in the output layer of DNN, or a penalization approach described in [6].

The selection vector $\boldsymbol{\xi}$ is similar to the weight vector in weighted sum rate (WSR) maximization problems where a DNN with a moderate size cannot generalize the problem when the weights are dynamic [19]. In the proposed approach though, the selection vector $\boldsymbol{\xi}$ can be seen as a discretized weight vector with only two values of zero and one for its elements, resulting in the shrinkage of the search space to be generalized by the DNN. The capability of the DNN to generalize this problem for different possible realizations of $\boldsymbol{\xi}$ is shown by the numerical simulations.

3.2. Satisfying the data rate constraints

Thanks to the slower dynamics of the data rate constraints compared to the dynamics of the channel coefficients, we assume the data rate constraints are fixed in a given time window. The objective now is to optimize the probability vector $\boldsymbol{\kappa} = [\kappa_1, \ldots, \kappa_N]$ to satisfy the data rate constraints in expectation. Recall that κ_i is the probability of user i to be active.

In the vanilla primal-dual updates for constrained optimization problems, the Lagrangian function is formed as

$$L(\boldsymbol{\lambda}, \boldsymbol{H}, \boldsymbol{p}) = \sum_{i=1}^{N} (1 + \lambda_i) R_i(\boldsymbol{H}, \boldsymbol{p}) - \lambda_i r_i^{min} \tag{4}$$

for each realization $\boldsymbol{H}$ and with the dual variables λ_i corresponding to the data rate constraints. Also, we have the following updates until convergence,

$$\boldsymbol{p}^+(\boldsymbol{H}, \boldsymbol{\lambda}) = \operatorname*{argmax}_{\boldsymbol{p}} L(\boldsymbol{\lambda}, \boldsymbol{H}, \boldsymbol{p}) \tag{5a}$$

$$\boldsymbol{\lambda}^+ = [\boldsymbol{\lambda} + \gamma\big(\boldsymbol{r}^{min} - \mathbb{E}_{\boldsymbol{H}}\{\boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p}^+(\boldsymbol{H}, \boldsymbol{\lambda}))\}\big)]_+, \tag{5b}$$

where $[.]_+ = \max\{0, .\}$ is a pointwise operator, the superscript $+$ indicates the next iteration, and γ is the stepsize for this update.

Using the same concept, we iteratively adjust the distribution of the selection vector $\boldsymbol{\xi}$ to satisfy the data rate constraints. We set

$$\boldsymbol{\kappa} = \mathbb{E}\{\boldsymbol{\xi}\} = \mathbf{1} + \boldsymbol{\lambda} \tag{6}$$

and approximate $\boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p}^+(\boldsymbol{H}, \boldsymbol{\lambda}))$ by $\mathbb{E}_{\boldsymbol{\xi}}\{\boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p}_\xi)\}$ where $\boldsymbol{p}_\xi = \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}^*)$ as the primal update for each realization $\boldsymbol{H}$ and $\boldsymbol{\xi}$. By this approximation, we incur a suboptimality as we have approximated $\max_{\boldsymbol{p}} (\mathbf{1} + \boldsymbol{\lambda})^T \boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p}) = \max_{\boldsymbol{p}} \mathbb{E}\{\boldsymbol{\xi}\}^T \boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p})$ by $\mathbb{E}_{\boldsymbol{\xi}}\{\max_{\boldsymbol{p}} \boldsymbol{\xi}^T \boldsymbol{R}(\boldsymbol{H}, \boldsymbol{p})\}$ for each realization $\boldsymbol{H}$ with changing the order of the operators $\mathbb{E}\{.\}$ and $\max\{.\}$.

At each time instant, a random vector $\boldsymbol{\xi}$ is generated each of whose elements is drawn from a Bernoulli distribution with the probability κ_i. Afterwards, using the trained policy from the previous section, the RA problem corresponding to the vector $\boldsymbol{\xi}$ is solved to obtain $\boldsymbol{p}_\xi = \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}^*)$ to address the primal update (5a). After w time instants, the probability vector $\boldsymbol{\kappa}$ is updated by the SGD approach with the dual update (5b) as,

$$\begin{aligned} \boldsymbol{\lambda}^+ &= [\boldsymbol{\lambda} + \gamma\big(\boldsymbol{r}^{min} - \mathbb{E}^w\{\boldsymbol{R}(\boldsymbol{H}, \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}^*))\}\big)]_+ \\ \kappa_i^+ &= \frac{1 + \lambda_i^+}{\max_{\ell \in [N]}\{1 + \lambda_\ell^+\}} \quad \forall i \in [N], \end{aligned} \tag{7}$$

where we used (6) and normalized $\boldsymbol{\kappa}$ to have maximum of 1, as we treat this vector as a probability vector. Note that normalization does not change the maximization of Lagrangian function (5a). Moreover, $\mathbb{E}^w\{.\}$ is the expectation over $\boldsymbol{H}$ and $\boldsymbol{\xi}$ approximated by averaging over w time instants, making the optimization nature stochastic. The whole procedure is depicted in Algorithm 1.

Algorithm 1: Proposed RA approach

Training stage:
train the DNN using (3)
Output: the trained DNN as $\phi(., \boldsymbol{\theta}^*)$

Inference stage:
Initialize: $\gamma > 0, w > 0, \lambda_i = 0, \kappa_i = 1, \ \forall i \in [N]$
for $t = 1, 2, \ldots, T$ **do**
  for $k = 1, 2, \ldots, w$ **do**
    randomly generate $\boldsymbol{\xi}$ according to $\boldsymbol{\kappa}$
    find the optimum power: $\boldsymbol{p}_\xi = \phi(\boldsymbol{H}_\xi, \boldsymbol{\theta}^*)$
    find the corresponding rate: $\boldsymbol{R}^k(\boldsymbol{H}_\xi, \boldsymbol{p}_\xi)$
  end
  find the average rate: $\bar{\boldsymbol{R}} = \sum_{\ell=1}^{w} \boldsymbol{R}^\ell$
  update $\boldsymbol{r}^{min}$ (assumed fixed for some while)
  update the dual: $\boldsymbol{\lambda} \leftarrow [\boldsymbol{\lambda} + \gamma(\boldsymbol{r}^{min} - \bar{\boldsymbol{R}})]_+$
  update $\kappa_i = \frac{1+\lambda_i}{max_{\ell \in [N]}\{1+\lambda_\ell\}} \ \ \forall i \in [N]$
end

Selecting a subset of active users before performing DL-based RA is also addressed in [19] in a different communication setting to ensure fairness. It is not obvious how it can be extended to satisfy dynamic data rate constraints, and also direct application of the update rule (7) on the weights of the users in [19] is not trivial. Moreover, the approach in [19] needs to repeat a one-dimensional search at each time instant, incurring a high computational complexity.

4. SIMULATIONS

In this section, we try to answer the following questions: 1) Can the trained DNN, as a policy, generalize the RA problem with a different subset of active users? 2) Is the proposed approach convergent to a fixed solution? 3) How fast can the proposed approach satisfy the constraints? 4) What is the performance compared with the conventional methods?

We assume N users are present in a wireless communication network. The noise variable σ^2 is set to ensure a SNR of $10 \log\left(\frac{P_{max}}{\sigma^2}\right) = 10\ dB$ with the maximum power $P_{max} = 10$. The channel coefficients are randomly and independently generated from a normal distribution $\mathcal{N}(0, 1)$, and their feasibility in the RA problem is not checked for training of the DNN. The number of neurons per layer in our DNN is $\{N^2, 50, 50, N\}$. The *relu* activation function is used for hidden layers while the *sigmoid* activation function is used for the output layer to satisfy the power constraints. Finally, the stepsize γ and the expectation approximation length w are set to 0.5 and 100 respectively in all simulations.

In the first set of simulations, we assume $N = 5$ users have a data rate constraint $\boldsymbol{r}^{min} = [1, 1, 2, 0.4, 2]$ and after some time they change their data rate constraints to $\boldsymbol{r}^{min} = [2, 1, 0.4, 2, 1]$. These data rate constraints are selected to be

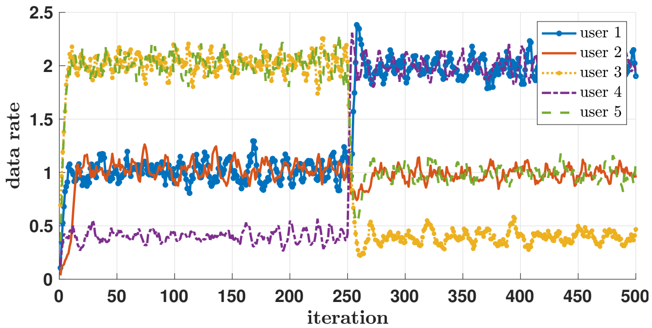

Fig. 1. Average data rates of the users with two different data rate constraints changed at iteration 250.

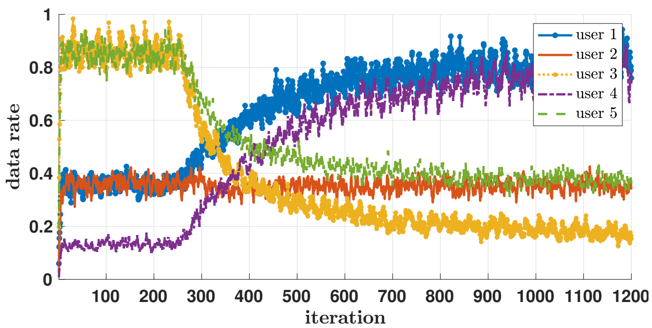

Fig. 2. Average data rates of the users with the same data rate constraints as fig.1 when no DNN is used.

tight enough, so all the data rate constraints are active after the RA. Moreover, in this simulation, we do not check the feasibility of the test samples, the same scenario as we have in practice. The user data rates are shown in expectation in Fig. 1. It can be concluded that the proposed approach can satisfy the dynamic data rate constraints in expectation successfully, and the convergence rate is fast enough, so the method can converge to a new optimal point after a few iterations.

The proposed approach can be seen as a time-sharing problem, when in each time instant only a subset of users are allowed to be active, mitigating the interference among the users and maximizing the SR. This time-sharing problem is considered using DNNs in recent works such as [11]. However, in the proposed approach, time-sharing and RA problems are combined to maximize constrained RA problems. A question is if it is possible to satisfy the constraints with the same update rule of (7), but without the use of a DNN, so the selected active users communicate with the maximum power p^{max}. As it can be seen in Fig. 2, not only the data rate constraints cannot be met in this scenario, but also the SR of the users is smaller than in the previous simulation that is using a DNN. This result is justifiable, as at each time instant a subset of users is selected, and there is no further RA among them.

We also compare the performance of the proposed approach in terms of SR and constraint satisfaction, with the geometric programming (GP) method [2]. To perform the comparison, we only include the feasible channel matrices for the RA problem (2). In practice the channel matrix can be infeasible, which conventional methods like GP cannot provide reasonable solutions for. However, as shown in Fig. 1, the proposed method can handle infeasible samples, as another advantage over the conventional methods. For the 5-user scenario, we assume $r^{min} = 0.5$ for all users, the same constraint [6] used to study the ability of the DNN to satisfy the constraints. Moreover, $r^{min} = 0.25$ and $r^{min} = 0.1$ are assumed for all users when $N = 10$ and $N = 15$, respectively. We report the constraint violation for the active constraints as $\max_i \left\{ \max\{0, r_i^{min} - \mathbb{E}^w\{R_i\}\} \right\}$ and its percentage as $100 \times \max_i \left\{ \max\{0, r_i^{min} - \mathbb{E}^w\{R_i\}\}/r_i^{min} \right\}$ to quantify how well the proposed approach can satisfy the constraints.

Table 1. Performance comparisons of the proposed RA approach and GP over 1000 test samples.

Sum Data Rates (bits/s/Hz)			
# of users	**SR**	**Constraint viol.**	**SR by GP**
5	8.06	6×10^{-4} (0.12%)	6.14
10	7.29	5×10^{-5} (0.02%)	5.42
15	6.97	8.4×10^{-4} (0.84%)	4.42

Table 1 shows that not only the data rate constraints are met in expectation very accurately, but also that we can achieve a higher SR with the proposed approach compared with GP. Recall the constraints are met in expectation in the proposed approach, allowing the approach to violate the constraints for the inactivated users and even some activated users at some time instants. This constraint violation allows the approach to achieve a higher SR while satisfying the constraints *over a period of time*. Note that users and their upper-layer applications usually care about the data rate over a period of time rather than at each time instant [7, 10].

5. CONCLUSION

In this paper, we have addressed the problem of DL-based RA when the users have dynamic data rate constraints to be met in expectation, i.e. over time. A subset of users at each time instant is selected at random, with an iteratively tuned distribution of the random selection, and the RA is done with a trained DNN among the selected users. The proposed approach achieves even higher SR compared to the conventional methods. The proposed approach opens up the possibility to allocate resources in cross-layer RA scenarios when the upper application layer dynamically can change its data rate demand. Studying the theoretical convergence properties, the suboptimality of the method mentioned in Section 3.2, and more advanced optimization algorithms to speed up satisfying the constraints are topics of future research activities.

6. REFERENCES

[1] Vivek P Mhatre, Konstantina Papagiannaki, and Francois Baccelli, "Interference mitigation through power control in high density 802.11 wlans," in *26th IEEE International Conference on Computer Communications*, 2007, pp. 535–543.

[2] Mung Chiang, Chee Wei Tan, Daniel P Palomar, Daniel O'neill, and David Julian, "Power control by geometric programming," *IEEE transactions on wireless communications*, vol. 6, no. 7, pp. 2640–2651, 2007.

[3] Qingjiang Shi, Meisam Razaviyayn, Zhi-Quan Luo, and Chen He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[4] Kaiming Shen and Wei Yu, "Fractional programming for communication systems—part i: Power control and beamforming," *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2616–2630, 2018.

[5] Fuhui Zhou, Guanyue Lu, Miaowen Wen, Ying-Chang Liang, Zheng Chu, and Yuhao Wang, "Dynamic spectrum management via machine learning: state of the art, taxonomy, challenges, and open research issues," *IEEE Network*, vol. 33, no. 4, pp. 54–62, 2019.

[6] Fei Liang, Cong Shen, Wei Yu, and Feng Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1760–1776, 2019.

[7] Mark Eisen, Clark Zhang, Luiz FO Chamon, Daniel D Lee, and Alejandro Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, 2019.

[8] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2017, pp. 1–6.

[9] Woongsup Lee, "Resource allocation for multi-channel underlay cognitive radio network based on deep neural network," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1942–1945, 2018.

[10] Dionysios S Kalogerias, Mark Eisen, George J Pappas, and Alejandro Ribeiro, "Model-free learning of optimal ergodic policies in wireless systems," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6272–6286, 2020.

[11] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019.

[12] Yasar Sinan Nasir and Dongning Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[13] Zhiyang Wang, Mark Eisen, and Alejandro Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *arXiv preprint arXiv:2107.01489*, 2021.

[14] Zhiyang Wang, Mark Eisen, and Alejandro Ribeiro, "Unsupervised learning for asynchronous resource allocation in ad-hoc wireless networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 8143–8147.

[15] Le Liang, Hao Ye, and Geoffrey Ye Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

[16] Pourya Behmandpoor, Jeroen Verdyck, and Marc Moonen, "Deep learning-based cross-layer resource allocation for wired communication systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 4120–4124.

[17] Sho Sonoda and Noboru Murata, "Neural network with unbounded activation functions is universal approximator," *Applied and Computational Harmonic Analysis*, vol. 43, no. 2, pp. 233–268, 2017.

[18] Tianping Chen and Hong Chen, "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Transactions on Neural Networks*, vol. 6, no. 4, pp. 911–917, 1995.

[19] Wei Cui, Kaiming Shen, and Wei Yu, "Spatial deep learning for wireless scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, 2019.