On the Transferability of Graph Neural Networks for Resource Allocation in Wireless Networks

Santiago Fernández*, Romina García Camargo*, Mark Eisen†, Alejandro Ribeiro‡ and Federico Larroca*
*Facultad de Ingeniería, Universidad de la República, Uruguay.
†Johns Hopkins University, Applied Physics Lab, Laurel, MD
‡Dept. of Electrical and Systems Eng., University of Pennsylvania, Philadelphia, PA.
{sfernandez, rominag, flarroca}@fing.edu.uy, mark.eisen@ieee.org, aribeiro@seas.upenn.edu

***Abstract*—Effective radio resource management is crucial for optimizing both current and future wireless communication networks. Significant research has focused on identifying optimal resource allocation policies, which is a challenging mathematical problem. Consequently, learning-based algorithms, in particular those based on Graph Neural Networks (GNNs), have emerged as a practical and effective solution. However, most studies have relied on synthetic data for testing, which necessarily offers a simplified version of the complex propagation phenomena present in real-life wireless systems. In this paper we address this gap by evaluating these algorithms on a real-life dataset. Our experiments demonstrate that these solutions are indeed viable for real-world applications. Furthermore, since data for training will necessarily stem from the past, we verify that non-stationarities of the real-life networks do not negatively impact the trained algorithm's performance (i.e. *time transferability*). Finally, given that deployed networks are typically designed to follow certain pre-established patterns, we analyze and confirm that these algorithms not only perform well but they can also be transferred between networks while maintaining strong performance (i.e. *spatial transferability*). These results confirm the viability of GNNs as practical tools for resource allocation in real wireless networks.**

***Index Terms*—resource allocation, graph neural networks, transferability**

I. Introduction

Over recent years, wireless systems have seen a dramatic rise in usage. As devices become more ubiquitous and data transmission volumes expand, communication technologies are continuously advancing. The deployment of 5G networks and ongoing research into future 6G networks promise to significantly enhance services and capabilities for users [1], [2]. However, with technological advancements come greater challenges in developing innovative and efficient network management solutions.

Effective resource allocation, crucial for ensuring optimal communication among users, is thus increasingly relevant, even more so as network demands grow. The mathematical formulations required to address these challenges are often complex and computationally costly, leading to the adoption of various heuristic approaches. Notably, machine learning techniques have demonstrated promising results in parameterizing solutions to these problems [3]–[5]. Given that communication networks are naturally represented as graphs, Graph Neural Networks (GNNs) [6] have emerged as a fitting architecture for these algorithms. While studies like [7] and [8] have shown efficient and innovative GNN-based solutions for the resource allocation problem, it is important to note that these algorithms were evaluated on simulations using synthetic data. However, there are important and challenging patterns of the propagation characteristics of a real wireless channel that are very difficult to grasp in simulation. These include correlations between channel gains (or even non-stationarity), non-classical noise distributions due to interference, a difficult to model dependency of attenuation with distance, or simply miscalibrations that generate non-symmetric channels [9]–[11].

It remains unclear whether these characteristics will pose significant obstacles for learning-based algorithms or if they will outperform other heuristics in these challenging scenarios. A potentially more daunting issue is the question of *time transferability*, as training data will inevitably come from a previous time period compared to when the algorithm is in operation. The previously mentioned non-stationarities could represent a significant challenge in this context. Additionally, a network operator managing multiple networks would benefit from training on one network and then applying the learned policy to other networks (i.e., *spatial transferability*).

In this work, we empirically demonstrate that GNN-based resource allocation algorithms are indeed applicable to real-world scenarios by evaluating them over a real-life dataset. Most importantly, we affirmately answer two critical questions regarding the transferability of these solutions. First, can a learning system trained on data from one period of time effectively provide well-performing policies for a future time period? Second, can a model learn relevant features of a specific network and be applied to other networks maintaining a strong performance? While transferability between any two graphs does not necessarily hold (see for instance [12]), the consistent design patterns observed accross networks suggest that certain structural elements can enable successful transferability of trained models.

The rest of the article is organized as follows. Section II presents the problem we worked with. In Sec. III, we describe the setup for the experiments, including details about the dataset and necessary preprocessing. This section also outlines the implemented algorithm and presents the obtained results. Finally, Sec. IV presents some concluding remarks and future work.

II. Problem Formulation

A. Radio Resource Allocation

Consider a wireless system consisting of m transmitters and n receivers. Each transmitter $i \in \{1,\ldots,m\}$ is linked to a specific receiver $r(i) \in \{1,\ldots,n\}$. Multiple transmitters can be associated to the same receiver. Time is divided into slots, with connections between nodes in each slot characterized by fading channel coefficients. Let $h_{ij}(t)$ denote the channel between transmitter i and receiver $r(j)$ at time slot t. We arrange all channel coefficients in the matrix $\mathbf{H}(t)$, with entries $[\mathbf{H}(t)]_{ij} = h_{ij}(t)$. In order to represent the node states we introduce variables $x_i(t)$ representing the random state of communication between transmitter i and receiver $r(i)$ at time slot t. The states could represent for example the processing capabilities of each agent or the amount of data ready for transmission. These state variables come together in the vector $\mathbf{x}(t)$ with entries $[\mathbf{x}(t)]_i = x_i(t)$.

The radio resource allocation problem we consider here is to choose the power p_i of each transmitter (i.e. a vector $\mathbf{p}(t)$ has to be chosen for each t), so that a certain network-wise performance indicator is maximized. In this choice, there are naturally certain constraints, typically in the form of a power budget constraint. One way to address the problem would be to use the set of random fading states $\mathbf{H}(t)$ and the set of random node states $\mathbf{x}(t)$ to determine an instantaneous allocation policy $\mathbf{p}(t) = \mathbf{p}(\mathbf{H}(t),\mathbf{x}(t))$. This policy is obtained by maximizing the performance indicator function while respecting the constraints. Once the policy $\mathbf{p}(t)$ is chosen, the system pair $\mathbf{H}(t),\mathbf{x}(t)$ produces an instantaneous vector reward $\mathbf{r}(t)$.

However, this strategy implies a high computational cost and potentially poses an infeasible computational burden because it requires solving a constrained optimization problem for each instance of the channel and node states. Moreover, in fast fading scenarios, the instantaneous performance of the system tends to vary very quickly, and users experience instead its long-term average across time slots. Let us consider the sum over all transmitters of their Shannon capacity as the network indicator, and assume that they may either transmit with power p_0 or not at all. Then, taking the above into consideration, one can formulate the following optimization problem with average power constraints:

$$
\begin{aligned}
R^* = \max_{\mathbf{p}(\mathbf{H},\mathbf{x})} \quad & \sum_{i=1}^{m} r_i, \qquad\qquad (1)\\
\text{s.t.} \quad & r_i = \mathbb{E}\left[\log\left(1+\frac{|h_{ii}|^2 p_i(\mathbf{H},\mathbf{x})}{\sigma^2+\sum_{j\neq i}|h_{ji}|^2 p_j(\mathbf{H},\mathbf{x})}\right)\right],\\
& \mathbb{E}\left[\mathbf{1}^T\mathbf{p}(\mathbf{H},\mathbf{x})\right] \le P_{\max}, \quad \mathbf{p}(\mathbf{H},\mathbf{x}) \in \{0,p_0\}^m.
\end{aligned}
$$

In (1), we seek to find a binary power allocation policy that maximizes the sum of rate capacities subject to two average value constraints. The first is a reward value constraint, where we use the capacity experienced by each transmitter over an AWGN channel with noise power σ^2 and interference among users. The second is a power constraint, where we restrict the average power consumed by all transmitters in the network. Several other forms of the radio resource allocation problem exist, and it is rather straightforward to modify (1) to consider them (see for instance [7], [13]). However, the formulation in (1) is a very popular and well studied problem, which is the reason behind our choice.

B. Learning to Solve the Radio Resource Allocation Problem

Generally, there is no exact solution for problem (1). One possible heuristic is to use data-driven learning techniques to parameterize and learn the resource allocation policy. However, both the performance indicator function as well as the constraints are based on expected values, posing a significant challenge for traditional learning algorithms. Expected or mean rewards are the subject of Reinforcement Learning [14]. However, learning constrained policies is a challenging task, which we will tackle through a dual representation of the problem.

To this end, consider the Lagrangian of the constrained problem $\max_{\mu\ge 0}\min_{\mathbf{r}} -\sum_{i=1}^{m} r_i - \mu\left(P_{\max} - \mathbf{1}^T\mathbf{p}\right)$, where r_i and $\mathbf{p}$ are now ergodic averages of the Shannon capacity and transmitted power (e.g. for an observed period of T timeslots, $p_i = \sum_{t=1}^{T} p_i(t)/T$). Following [7], we will take a random policy provided by a parametric function $\mathbf{\Phi}(\mathbf{H},\mathbf{x};\boldsymbol{\alpha})$. That is to say, for each time-step t, $\mathbf{\Phi}(\mathbf{H}(t),\mathbf{x}(t);\boldsymbol{\alpha})$ will provide the probability for each node to either transmit at power p_0 or not transmit in that particular time-step. This function will be learned through a modified version the REINFORCE algorithm, a classic policy gradient method [14], [15]. For each iteration, we consider a batch of graphs and sample the power allocation policy. The rewards for each time slot are computed, as well as the accumulated rewards. These are then used in order to update the dual variable (through a simple gradient ascent) as well as the parameters $\boldsymbol{\alpha}$ (through REINFORCE). This procedure is summarized in Algorithm 1.

Algorithm 1 REINFORCE algorithm with constraints

Initialize $\mu_0, \boldsymbol{\alpha}_0$
for iteration j in 1, ..., num_iterations **do**
  Run iteration:
    Sample $\mathbf{H}_t$ and $\mathbf{p}_t \sim \mathbf{\Phi}(\mathbf{H}_t;\mathbf{x};\boldsymbol{\alpha}_j)$
    Compute rewards R_t for times t in $\{1,...,T\}$,
    $R_t = \sum_{i=1}^{m} r_i^t - \mu_j \cdot u(\mathbf{r}^t)$
  Compute accumulated rewards $\tilde{R}_t = \sum_{l=t}^{T} R_t$
  $Loss(\boldsymbol{\alpha}_j) = -\sum_{t=1}^{T}\tilde{R}_t \log(\Phi(\mathbf{H}_t;\mathbf{x};\boldsymbol{\alpha}_j))$
  $\boldsymbol{\alpha}_{j+1} \leftarrow \text{Adam}(Loss(\boldsymbol{\alpha}_j))$
  $\mu_{j+1} \leftarrow \mu_j - \epsilon \cdot u(r_j)$

Where u is the constraint function associated with the available power in the network. The considered family of functions $\mathbf{\Phi}(\mathbf{H},\mathbf{x};\alpha)$ from where we will choose the resource allocation policy are GNNs. The decision is supported by their attractive properties such as transferability (i.e. they can generalize patterns across different graph structures), and stability to deformations of the underlying graph (i.e. small changes in the graph result in small changes of the output) [16].

In order to properly use this architecture we must reinterpret the following objects. First, the node state vector $\mathbf{x} \in \mathbb{R}^m$ will be considered a signal supported on nodes $i = 1, ..., m$. Second, the channel matrix $\mathbf{H} \in \mathbb{R}^{m \times m}$ will be used as an adjacency matrix representation of each graph. In a nutshell, GNNs consists of a concatenation of several layers consisting of a graph convolutional filters supported on the graph matrix $\mathbf{H}$ followed by a pointwise non-linearity σ. That is to say, a single layer is simply $\mathbf{y} = \sigma\left(\sum_{k=0}^{K-1} \mathbf{H}^k \mathbf{x} \alpha_k\right)$. In this case, $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_{K-1})$ is the set of K filter coefficients.

III. Experiments

In order to evaluate the performance of the proposed solution, a series of experiments were conducted. A real dataset was analyzed and preprocessed for the work. We first introduce the characteristics of said dataset. We then describe the implementation of Algorithm 1 and the settings employed for the tests, as well as presenting the obtained results.

A. Dataset

The dataset used in these experiments was provided by Plan Ceibal [9]. It contains data from various school buildings across Uruguay, ranging from multi-story buildings, housing hundreds of students to small rural schools with only a few dozen students. The network graphs were constructed with double-band WiFi Access Points (APs) as nodes and the wireless links between them as edges. The initial dataset comprised information on the transmission power, the channels used and the Radio Signal Strength Indicator (RSSI) measurements that each node detects from others within range. Data were collected throughout the school year 2018, with samples taken every hour from March to December.

Preprocessing of the dataset was required for the data to be suitable for the experiments. Most of the datafields were removed, retaining essentially the transmission and reception power, so as to compute the attenuation between nodes. These values were used to construct the adjacency matrix of the graph. In particular, in order to construct the channel matrix $\mathbf{H}$, it was necessary to define for each transmitter i the corresponding pair $r(i)$. The criterion used towards this end was to assigned each transmitter the receiver with the best channel condition. This is illustrated with an example in Fig. 1. In addition to dropping irrelevant fields, school buildings with missing data for several months or with corrupted information were excluded from the analysis. Ultimately, the dataset included data from 1,424 buildings, with network sizes up to 33 APs. Experiments were conducted using network states for both the 2.4 GHz and the 5 GHz frequency bands, employing various scaling factors to account for differences in channel power values.

B. Numerical Results

The architecture consists of a 5-layer GNN, using a filter of order $K = 3$. The dataset is organized in batches of 64 graphs. The same learning rate was used both for updating the dual variable and the weights of the neural network. The step was defined as 5×10^{-4}. The ADAM optimizer was used for updating the GNN weights $\boldsymbol{\alpha}$ [17].

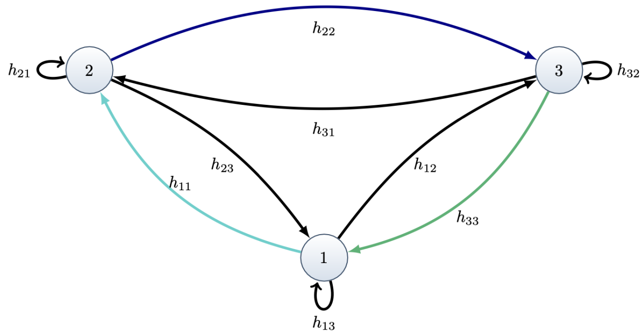

Fig. 1. Example of a small network where we show channel matrix coefficients h_{ij} and transmitter-receiver pairs $r(i)$ using colored edges. For example, h_{11} represents the channel gain between transmitter 1 and its receiver, whereas h_{12} represents the channel gain between transmitter 1 and the receiver of transmitter 2.

The experiments were run using a 20-node graph as reference, so as to evaluate the algorithm on a graph of considerable size within the dataset. The graph from the building with ID 856 was considered for training, using several other school buildings for validation of the transferability properties of our algorithm.[1]

In order to compare and grasp a better idea of the performance of the algorithm, various baseline implementations were conducted. The first, referenced to as *baseline 1*, defines a random amount of transmitting nodes using a probability $\theta = P_{max}/(mp_0)$, Here, P_{max} is the available power, p_0 is the power used for the transmission of a single node and m is the amount of nodes considered. This baseline divides the mean available power between the nodes that are set to transmit. A second baseline, *baseline 2*, divides the available power equally between all network nodes, each assigned P_{max}/m. This implies that all nodes are transmitting throughout the entire experiment. It is reasonable to expect that *baseline 1* will perform better than *baseline 2*, given that not all nodes are transmitting at all times. Finally, we consider a policy variation of the popular WMMSE heuristic [18]. This baseline consists on running the traditional WMMSE algorithm on the graph for each timestep independently. This yields a policy vector $\mathbf{p}_{opt}(t)$ which represents the optimal power combination given the state of the network and does not necessarily need to be a binary policy. Then, in order to conduct a fair comparison and to satisfy the average power constraint, we define node probabilities for transmission using the optimal allocation. Let $\theta_i = (p_{opt\,i} P_{max})/(p_0 \mathbf{1}^T \mathbf{p}_{opt})$ for all nodes in the network. This implies that nodes that received higher power through the traditional WMMSE algorithm will have a higher probability of transmission.

The model from the proposed solution was trained using data from the months of March through August. In Fig. 2, the sum capacity throughout the experiment is recorded

[1]The code for our experiments is available in our repository: https://github.com/sfernandezr/wireless-learning

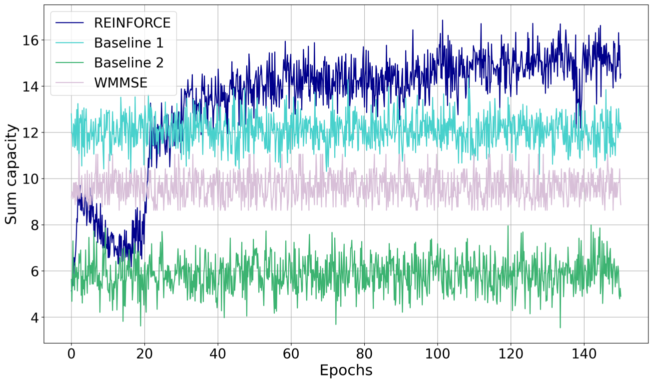

Fig. 2. Training process of the proposed algorithm. Specifically, the evolution of the sum capacity is presented. Baselines are considered for comparison. The REINFORCE algorithm outperforms them all.

and compared to the previously presented baselines. The REINFORCE algorithm outperforms the baselines after only a few iterations, resulting in an increase of around 20%.

In order to test spatial and temporal transferability properties two experiments were conducted. In both of these experiments the model weights were trained using school building 856 and then transferred to various other schools. Specifically, the following schools were considered: building 84 (26 APs), building 814 (24 APs), building 67 (22 APs), building 800 (15 APs), building 838 (15 APs), building 1141 (15 APs), building 201 (10 APs) and building 211 (10 APs). For each of these schools, the model was validated. A first experiment with aim to test spatial transferability was conducted. The model was trained with data during the months of March to December and then validated on this same time period. A second experiment attempted to test not only spatial but also temporal transferability. In this case the model was trained with data during the months of March to August and then validated with data from September to December.

The results of the transferability experiments are shown in Fig.3 and Fig.4. The first corresponding to spatial transferability and the latter corresponding to both spatial and temporal transferability scenarios. The experiments aim to compare the performance of the REINFORCE algorithm to the corresponding baselines. Not only does the proposed algorithm continue to outperform the baselines in nearly all schools, but it also succeeds in performing well in various scenarios. Note that the proposed algorithm outperforms all other baselines except for a single school building (the smallest one, where the WMMSE baseline slightly outperforms the learning algorithm). In general, the REINFORCE algorithm maintains strong performance when applied to both larger and smaller sized networks, in comparison to the original 20-node network.

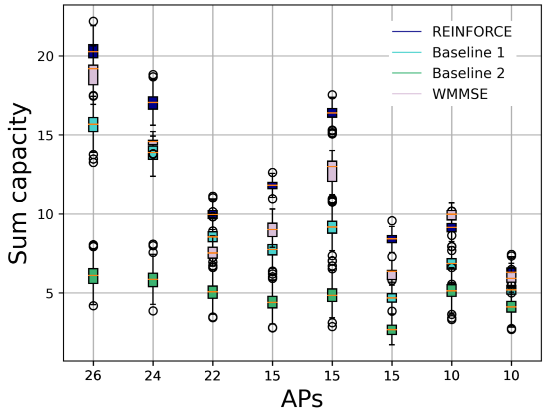

Fig. 3. Resulting box plots of spatial transferability experiments. The model is trained on building with ID 856 (with 20 nodes) from March to December. It is then validated in other school buildings during the same months. The GNN-based resource allocation algorithm maintains a good performance even in networks it has not been trained on.

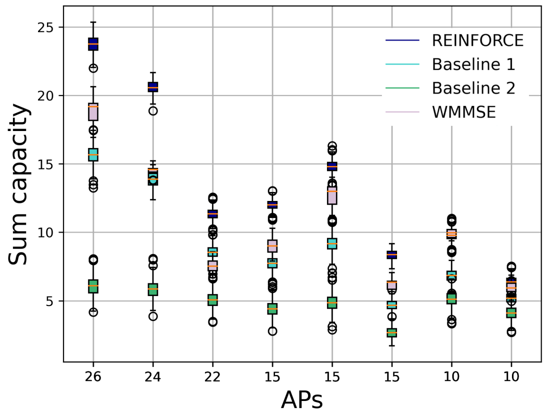

Fig. 4. Resulting box plots of the spatial and time transferability experiments. The model is trained on building with ID 856 (with 20 nodes) from March to August. It is then validated in other school buildings during the months of September through December. The GNN-based resource allocation algorithm maintains a good performance even in networks it has not been trained on.

IV. Conclusions

We considered the problem of learning optimal resource allocation policies in wireless networks. A possible solution for this problem was discussed and implemented, with the intention of testing said solution in a real-world dataset. In particular, a data-driven learning technique based on the REINFORCE algorithm was studied, which uses GNNs as the learning architecture. It was shown that the implemented algorithm results in good performance, specifically in comparison to the defined baselines. Most importantly, both *time* and *spatial* transferability experiments were conducted, and strong performance results were maintained when testing in different scenarios and time periods.

For future work, we plan to consider other more challenging variants of the resource allocation problem and evaluate them on this real-life dataset. For instance, frequency selection policies or user assignment and handover. In this case, we may compare the learned policy with that of the propietary algorithm used by the Wi-Fi controller.

REFERENCES

[1] N. Al-Falahy and O. Y. Alani, "Technologies for 5g networks: Challenges and opportunities," *IT Professional*, vol. 19, no. 1, pp. 12–20, 2017.

[2] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, "On the road to 6g: Visions, requirements, key technologies, and testbeds," *IEEE Communications Surveys Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[3] J. Suárez-Varela, P. Almasan, M. Ferriol-Galmés, K. Rusek, F. Geyer, X. Cheng, X. Shi, S. Xiao, F. Scarselli, A. Cabellos-Aparicio, and P. Barlet-Ros, "Graph neural networks for communication networks: Context, use cases and opportunities," *IEEE Network*, vol. 37, no. 3, pp. 146–153, 2023.

[4] M. Randall, P. Belzarena, F. Larroca, and P. Casas, "Deep reinforcement learning and graph neural networks for efficient resource allocation in 5g networks," in *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, 2022, pp. 1–6.

[5] Z. Wang, M. Eisen, and A. Ribeiro, "Decentralized wireless resource allocation with graph neural networks," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 299–303.

[6] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 128–138, 2020.

[7] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2977–2991, 2020.

[8] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5898–5912, 2022.

[9] G. Capdehourat, F. Larroca, and G. Morales, "A nation-wide wi-fi rssi dataset: Statistical analysis and resulting insights," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 370–378.

[10] F. Larroca and F. Rodríguez, "An overview of wlan performance, some important case-scenarios and their associated models," *Wireless Personal Communications*, vol. 79, pp. 131–184, 2014.

[11] G. Capdehourat, P. Bermolen, M. Fiori, N. Frevenza, F. Larroca, G. Morales, C. Rattaro, and G. Zunino, "Large-scale 802.11 wireless networks data analysis based on graph clustering," *Wireless Personal Communications*, vol. 120, pp. 1791–1819, 2021.

[12] L. Ruiz, L. Chamon, and A. Ribeiro, "Graphon neural networks and the transferability of graph neural networks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1702–1712.

[13] M. Eisen, C. Zhang, L. F. O. Chamon, D. D. Lee, and A. Ribeiro, "Learning optimal resource allocations in wireless systems," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2775–2790, 2019.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[15] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[16] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5680–5695, 2020.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[18] 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," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.