

# Deep Learning for MANET Routing

KIRIL DANILCHENKO<sup>1</sup>, RINA AZOULAY<sup>2</sup>, SHULAMIT RECHES<sup>2</sup>,  
AND YORAM HADDAD<sup>2</sup> (Senior Member, IEEE)

<sup>1</sup>Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

<sup>2</sup>Department of Computer Science, Jerusalem College of Technology, Jerusalem 91160, Israel

CORRESPONDING AUTHOR: K. DANILCHENKO (kdanilch@uwaterloo.ca)

This work was supported in part by the Israel Innovations Authority under WIN: The Israeli Consortium for Wireless Intelligent Networks.

**ABSTRACT** Power control and scheduling are among the most well-known resource allocation challenges in wireless networks, and are often solved as optimization problems with constraints. However, solving these optimization challenges by using optimal algorithms often incurs a significant time complexity, which creates considerable discrepancies between the theoretical results and real-time processing required. In this study, we propose a novel machine learning-based perspective to address this issue. We propose a scheduling and power control deep neural network *SPCDNet* method and its modification *SPCDNet<sup>R</sup>*. *SPCDNet* solves the scheduling problem for point-to-point transmission requests while *SPCDNet<sup>R</sup>* solves the more complex problem, where the input transmission list is composed of ordered routes which should be satisfied. Both *SPCDNet* and *SPCDNet<sup>R</sup>* are trained in a supervised manner and show near-optimal performance on the test set. Our results demonstrate that *SPCDNet* and *SPCDNet<sup>R</sup>* can serve as a computationally inexpensive solution (regarding time complexity), compared with state-of-the-art schemes, while showing to be near-optimal approximation solutions to the time scheduling and power control challenges. Moreover, we found that both *SPCDNet* and *SPCDNet<sup>R</sup>* reach efficient solutions for large problem instances, even though they were trained on small problems.

**INDEX TERMS** Ad-hoc networks, 5G, TDMA, machine learning, radio resource management, multi-hop communication.

## I. INTRODUCTION

ADDRESSING resource allocation problems in multi-slot and multi-hop wireless communication systems under different transmitter requirements is one of the most challenging and fundamental tasks in wireless networking. Owing to the broadcast nature of the wireless medium, the transmission power of a transmitter not only “delivers” messages to the receiver, but also inadvertently creates interference for other receivers. Hence, the transmission power needs to be controlled carefully to manage the interference and enhance the overall performance of the system. The problem of transmission power control allocation (in various variations) is NP-hard [12] and has been extensively studied in recent years [13], [14], [17], [20], [41], [42].

In this study, we consider the problem of power allocation and transmission resource scheduling for clusters within the framework of Mobile Ad Hoc Networks (a.k.a. MANET). We assume that one of the cluster members acts as a cluster head and manages the allocation of resources of the

entire cluster in a given time frame. Our goal, in solving the aforementioned challenges, is to maximize the throughput in a multi-time, multi-hop wireless system, given the list of transmission requests. We consider two types of link scheduling problems, namely, the per-link scheduling problem and the per-route scheduling problem. In the per-link scheduling problem, a list of transmission requests is provided, where each request represents a direct point-to-point transmission. On the other hand, in the per-route scheduling problem, the input transmission request list consists of transmission route orderings. Clearly, the per-link scheduling problem is a particular case of the per-route scheduling problem, and both are NP-hard problems [12]. State-of-the-art solutions often involve either exhaustive searches to find the optimal scheduling, or alternatively, heuristic methods to find sub-optimal solutions. However, running time issues hinder the practicality of the optimal search methods, while the heuristic methods may result in inefficient scheduling solutions that are far from the optimal allocation schedules.

We address these problems from a different perspective. We leverage the recent advances in deep learning (DL) to propose a new deep neural network (DNN) architecture to achieve better performance with a shorter running time. In particular, the proposed approach establishes a connection between the throughput maximization problem in multi-slot and multi-hop systems under fairness constraints while minimizing a loss function when training a DNN, and relies on efficient network training and the ensembling of the mechanism to achieve near-optimal power control. Our results demonstrate the attractiveness of the DL method to rapidly solve optimization problems in communication networks while reaching near-optimal solutions.

The contributions of this study can be summarized as follows:

- 1) We present two variations of the scheduling and power control problem for device-to-device (D2D) wireless networks: the “throughput maximization under QoS constraints” (TM-QoS) problem and the “throughput maximization under routing constraints” (TM-RC). TM-QoS was presented in our conference paper [6], and deals with the problem of scheduling and power control given a set of requests. TM-RC is a new problem we tackle in the current study, where the given transmission requests contain the required path to traverse each request. The solution should meet the routing constraints as well.
- 2) We propose a transmission power control strategy for device-to-device (D2D) communication using a DNN for the two variants of the problem. The first DNN [6] solves the TM-QoS problem, while the second DNN solves the TM-RC problem.
- 3) We compare by simulation the results of our proposed solutions with well-known solutions. The simulation results confirm that both proposed DNNs achieve a very good approximation solution to the power allocation and scheduling problems, with a shorter computation time than those of other well-known solutions.
- 4) We demonstrate the ability of the proposed solutions to scale up to different sizes of problems, and, in particular, the ability of machine learning models to reach efficient solutions for large problem instances, even though they were trained on small-sized problems (this later issue of scaling to larger problems than the ones it was trained on is usually considered as challenging).

The remainder of this paper is organized as follows. Section II presents recent related studies, and Section III describes the theoretical model used for the power allocation challenge. Section IV provides the details of the *SPCDNet* and *SPCDNet<sup>R</sup>* DNNs, and the comparison schemes. The details of the simulation process, as well as the simulation results, are described in Section V, while the conclusions and suggestions for future research directions are provided in Section VI.

## II. RELATED WORK

In this study, we propose the use of DNNs for transmission power control and scheduling decisions in cellular communication systems, given the transmission requests. Power control challenges in wireless networks have been widely discussed in recent years [13], [14], [17], [20], [41], [42]. Both centralized and decentralized methods have been proposed for uplink and downlink transmissions, and various machine learning based solutions have been proposed for several variations of this challenge [9], [11], [26], [28], [30], [40].

Several recent studies on communication resource management suggest using DL to reach optimal or near-optimal solutions to control decisions made in communication networks. Usually, a multi-layer neural network is trained, where the network inputs are the network state (in a given representation), and the output, which should be trained, is the resource allocation decision. The DNN can be trained either using a supervised training scheme, given a training set that includes resource allocation solutions, calculated from any optimization method for each input example, or using an unsupervised scheme, by calculating the value of the neural network output and optimizing this value by changing the neural network weights. In the remainder of this section, we provide relevant studies that use DL methods for power control and scheduling problems. Then, we survey relevant DL based solutions that handle the joint routing and scheduling problem in wireless communication methods.

### A. DEEP LEARNING METHODS FOR POWER CONTROL

In general, the common variations of resource allocation problems in communication networks are known to be NP-hard [26]. Thus, over the years, various sub-optimal algorithms have been proposed to deal with resource allocation challenges [11], [28], [30], [31], [40]. Some recent studies have suggested using DL models to derive efficient solutions for scheduling and control decisions in MANET. In this section, we discuss several studies on resource management and highlight the uniqueness of our approach with respect to state-of-the-art studies.

Sun et al. [31] proposed the use of a DL scheme for real-time resource management in interference-limited wireless networks. Their theoretical results indicate that it is possible to train a well-defined optimization algorithm using finite DNNs. To validate their claims, they constructed a DNN for power control problems and trained it to approximate the behavior of the heuristic WMMSE algorithm [30]. Note that Sun et al. considered only a single time period to maximize the weighted system throughput, whereas our study focused on power control over time (TDMA).

Cui et al. [9] proposed a DL approach to schedule interfering links in a dense wireless network with full frequency reuse. They proposed a neural network architecture that takes the geographic spatial convolutions of interfering neighboring nodes. They proposed two methodologies for neural

network training: a supervised learning process, in which the network is trained using a sub-optimal algorithm based on the fractional programming approach, and an unsupervised training process, in which the transmission sum rate is maximized. Similarly, Ahmed et al. [1] developed a supervised learning DL-based resource allocation model with the goal of maximizing the total network throughput. Training data were obtained by solving a non-convex optimization problem using a genetic algorithm. Zappone et al. [43] demonstrated how DL can enable online power allocation to maximize energy efficiency in wireless interference networks. Their problem model consisted of multiple base stations serving multiple users, and a DNN was used to determine the power allocation vector for users to maximize the global energy efficiency of the network. In their study, the DNN was trained using a polynomial sub-optimal algorithm based on fractional programming and sequential optimization. Qian et al. [29] exploited the DNN algorithm to solve the power allocation problem in distributed antenna systems. They trained the DNN based on the traditional iterative algorithm. Alghorani et al. [2] proposed a machine learning-based power allocation scheme using Monte Carlo simulations to improve the link reliability in inter-vehicular communications (IVC). In contrast to the above studies, our proposed training process was performed based on the optimal power allocation, calculated by an linear programming-based optimal solver.

Liang et al. [24] used unsupervised DL to solve the non convex optimization problem of maximizing the sum rate of a fading multi-user interference channel. They proposed a network ensemble with multiple deep networks that were trained independently. Matthiesen et al. [27] developed a DL system for energy-efficient power control in wireless networks. They used an optimal reduced-complexity branch-and-bound procedure to find the globally optimal power policy, and then used the solution set as a training set for a DNN. Similar to Matthiesen et al., in our study, we train the DNN by the optimal power control policy, where in our study, the optimal solution was reached by an optimal solver, based on linear programming.

A transmission power control framework based on a convolutional neural network (CNN) was proposed in [21] to maximize either spectral efficiency (SE) or energy efficiency (EE). The full channel gain information was normalized and taken as the input of the CNN, while the output was the power allocation vector. They also proposed a form of deep power control (DPC) that can be performed in a distributed manner with local channel state information, allowing the signaling overhead to be greatly reduced.

Danilchenko et al. [18] presented the problem of minimizing the transmission in MANET based on multi-hop time-slotted time-division multiple access (TDMA) under routing delay minimization with heterogeneous traffic flows. They considered the challenge of minimizing the overall weighted end-to-end packet delay when the weights are determined according to the priorities of the requests. A delay minimization network that uses DL was introduced, and

simulations demonstrated that the DNN outperformed other state-of-art methods.

Other studies have suggested using reinforcement learning (RL) and deep reinforcement learning (DRL) for resource allocation problems. Ghadimi et al. [15] proposed an RL framework for power control and rate adaptation in the downlink of a radio access network, providing an efficient solution that approaches optimality based on the limited information available in practical systems.

Amiri et al. [4] suggested applying cooperative Q-learning for the power allocation of the dense network, to maximize the capacity of the network while providing quality of service (QoS) and fairness to users. Van Chien et al. [34] used DL to handle the summed spectral efficiency optimization problem in multi-cell massive MIMO systems with varying numbers of active users. Zhang et al. [39] proposed a DRL framework for channel and power allocation in a communication system in which UAVs were used as base stations. In their framework, a UAV base station can allocate both channels and transmission power for the uplink transmission of Internet of Things (IoT) nodes.

Li et al. [22] considered a cognitive radio system that consisted of a primary user and a secondary user. The primary user is assumed to update its transmitted power based on a predefined power-control policy. The secondary user does not have any knowledge about the primary user's transmission power, or its power control strategy, and a set of sensor nodes are spatially deployed to collect the received signal strength information at different locations in the wireless environment. Furthermore, the authors developed a DRL-based method wherein the secondary users can intelligently adjust their transmission power such that after a few rounds of interaction with the primary user, both users can transmit their own data successfully with the required QoS.

Luo et al. [25] solved the downlink max-min power control problem in cell-free massive MIMO systems, using deep deterministic policy gradient algorithm with DNN. They applied this method both for the max-sum and max-product power control problems, achieving better performance than the conventional deep learning algorithm.

In summary, recent studies have used DL to train optimal resource allocation algorithms to efficiently solve the challenges discussed above in online situations. Based on these previous studies, we propose a novel method of using DL to solve the power allocation and request scheduling problems. The uniqueness of our study lies in the fact that our training set is constructed based on optimal solutions for power control and scheduling problems. Therefore, the solutions derived by our DNN have an efficiency close to that of an optimal solution.

## B. JOINT ROUTING AND TDMA LINK SCHEDULING

In the following section, we describe some studies related to the joint optimization challenge of both routing and power control in wireless networks. Initially, we will discuss

heuristic solutions, followed by an exploration of machine learning-based approaches.

Li et al. [23] considered the issue of joint routing and link scheduling in software-defined full-duplex wireless networks, where an exclusive software-defined networking (SDN) controller node was involved. They formulated the optimization problem and presented a minimum-cost routing algorithm to solve it.

Wang et al. [37] focused on the problem of max-throughput (or max-fairness) routing and interference-aware link scheduling for a wireless network. They assumed that different terminals could have different transmission ranges and interference ranges. They formalized the interference-aware joint routing and TDMA link scheduling problem as a linear programming challenge and developed centralized and distributed approximating algorithms, where both achieve flow routing throughput (or fairness) that is at least a constant fraction of the optimum.

Sun et al. [33] proposed an adaptive scheduling and routing scheme to guarantee dynamic end-to-end delay requirements while minimizing the energy consumption in wireless sensor networks (WSNs). In the case where the end-to-end delay requirement of a region changes, they proposed an adaptive adjustment algorithm to locally adjust the wake-up schedule or routing table with the minimum energy cost while satisfying the new delay requirement.

Augusto et al. [5] proposed an algorithm called REUSE, which combines routing and link scheduling and aims to increase the throughput capacity in wireless mesh networks. The proposed mechanism uses a routing metric that favors spatial reuse and a scheduling algorithm that increases the number of simultaneous activated links. However, results obtained by REUSE are still far from the optimal results obtained through linear programming.

Some recent studies have suggested applying DRL to efficiently handle joint routing and scheduling problems. Wang et al. [35], [36] introduced and evaluated a cross-layer protocol that jointly optimizes the power control, rate adaptation and routing strategy in MANETs. The protocol uses a Q-learning method with a diffusion-approximation-based delay estimation model to monitor the environment, and a coordination mechanism was used to achieve a stable learning process.

Cui et al. [10] used an RL approach for simultaneous routing and spectrum access in MANET based on the geographic locations of the nodes. A single agent, trained according to the physical layer, makes routing and spectrum access decisions as it moves along the frontier nodes of each flow. The agent is trained according to the physical-layer characteristics of the environment using a reward function based on the Monte Carlo estimation of the future bottleneck SINR.

Recent research has employed Graph Neural Networks (GNNs) as an innovative solution to overcome challenges in wireless networks. Zhao et al. [44] addressed the issue of link scheduling in these networks. They conceptualized the problem as a maximum weight independent set issue,

and put efficient approximations for the problem, based on a GNN and guided tree search. The training of this network was carried out using a customized reinforcement learning approach. Their numerical experiments demonstrated the superior performance of their proposed method in both single and multi-channel scheduling. Moreover, the method's applicability was tested across different graph types and weight distributions, showing promising results. Wang et al. [38] formulated a general constrained resource allocation problem, and developed a GNN method to solve the general problem. In their study, the constrained optimization problem was converted to a Lagrangian function with dual variables, where the dual optimization problem involves maximizing and minimizing the Lagrangian function with, and the optimal filter tensor of the dual problem is found as the saddle point of the Lagrangian function with the dual variables.

Following previous studies, we suggest using DL to solve the power allocation and request scheduling problems. The uniqueness of our study lies in the fact that our training set consists of the optimal solutions for the power control and scheduling problem, found by using a linear programming-based solver; as a result, the solutions reached by the deep neural network have an efficiency close to the efficiency of the optimal solution.

### III. SYSTEM MODEL

In this section, we begin by presenting the first challenge, namely, the management of a multi-hop time-slotted TDMA mobile ad-hoc cluster with scheduling requirements. We assume that each cluster has a unique "leader" referred to as the cluster head (CH), which is the node responsible for allocating the communication resources within its cluster. The CH receives all the requirements from its cluster members as well as the location of the cluster members. Then, using this information, it assigns the resources within its cluster. We formulate the problems as an optimization graph problem, where each cluster member is a vertex, and each possible connection is considered an edge.

Consider a scenario with set  $N$  of nodes and set  $E$  of edges in the cluster. Set  $\mathcal{K}$  represents the pairs of transmitters and receivers in the cluster (i.e., active links in the cluster), where  $1 \leq |\mathcal{K}| \leq 2|E|$ . We use  $(n_i, n_j)$  to denote the link between transmitters  $n_i$  and  $n_j$ , and  $p_{ij}$  to denote the transmission power on link  $(n_i, n_j)$ . Moreover,  $h_{ij}^t$  represents the channel response between receiver  $n_j$  and transmitter  $n_i$ . Additionally,  $\sigma^2$  denotes the background noise power (thermal noise). The indicator variable  $x_{ij}^t$  for link  $(n_i, n_j)$  in time slot  $t$  is equal to one if node  $n_i$  is scheduled (by the cluster head) to transmit within this time period; otherwise, it is 0. In this work, we assumed a single channel with a bandwidth of  $w$  Hz. Because there is only one channel, we must assume full frequency reuse. In addition, the size of the TDMA frame is  $\mathcal{L}$  slots.

The rate  $r_{ij}$  in link  $(n_i, n_j)$  at a specific time slot  $t$  is defined by the Shannon-Hartley theorem, presented in Equation 1.

$$r_{ij}^t = w \log \left( 1 + \frac{|h_{ij}|^2 p_{ij}^t x_{ij}^t}{\sum_{kj \in E \setminus ij} |h_{kj}|^2 p_{kj}^t x_{kj}^t + \sigma^2} \right), \quad (1)$$

We consider  $p_{ij}^t = \{p : 0 \leq p \leq P_{max}\}, \forall (n_i, n_j) \in \mathcal{K}$ , where  $P_{max}$  is the maximum power that transmitters can use. Note that  $p_{ij}^t \geq 0$ . Thus, a user may choose not to transmit at all in time slot  $t$ .

Then, in the second variation of the time scheduling and power control problem, we consider the fact that messages are transmitted by routes. Let  $\Theta$  denote the set of all the routes. Route  $\vartheta^i \in \Theta$  defines a path  $\vartheta^i = \{n_s^1, \dots, n_d^k\}$  of size  $k - 1$  hops, where each pair  $(n_a^j, n_b^{j+1})$  represents one transmission link in the route  $\vartheta^i$ .

It is important to note that each transmission link in the route should be assigned in a specific order to preserve the route integrity. The set  $\mathcal{K}_\Theta$  is the union of all transmitter-receiver pairs in  $\Theta$ , and  $\mathcal{K}_{\vartheta^i}$  defines all transmitter-receiver pairs in  $\vartheta^i$ .

#### A. THROUGHPUT MAXIMIZATION UNDER QoS CONSTRAINTS (TM-QoS)

In the first variation of this problem, we assume that a list of requests is provided and the goal is to maximize the total throughput while ensuring that the receiver rate of each link meets a minimum rate requirement. This minimum rate acts as the quality-of-service requirement of the optimization problem. The inputs of this problem include the position of each device, the requirements of each transmitter, and the size of the frame measured in slots.

Each link rate  $r_{ij}$  should follow the QoS constraint and maintain the minimum required rate  $R_{ij}$  of the receiver on the link  $(n_i, n_j)$ , i.e.,  $r_{ij} \geq R_{ij}, \forall (n_i, n_j) \in \mathcal{K}$ . More specifically, the scheduling problem is formulated as:

$$\max \sum_{t=1}^{\mathcal{L}} \sum_{(n_i, n_j) \in \mathcal{K}} w \log \left( 1 + \frac{|h_{ij}|^2 p_{ij}^t x_{ij}^t}{\sum_{kj \in \mathcal{K} \setminus (n_i, n_j)} |h_{kj}|^2 p_{kj}^t x_{kj}^t + \sigma^2} \right) \quad (2)$$

$$\text{s.t. } x_{ij}^t = \{0, 1\}, \forall (n_i, n_j) \in E$$

$$\sum_{j=1}^N x_{ij}^t \leq 1, \forall (n_i, n_j) \in E$$

$$\sum_{t=1}^{\mathcal{L}} r_{ij}^t \geq R_{ij}$$

$$0 \leq p_{ij}^t \leq P_{max} \quad (3)$$

In other words, the goal is to optimally choose  $x_{ij}^t$  and  $p_{ij}^t$  across time slots  $0 \leq t < \mathcal{L}$ . The aim is to maximize the total transmission throughput, defined as the sum of all transmission rates. Each transmission rate,  $r_{ij}$ , is defined according to the Shannon-Hartley theorem, as defined in Equation 1.

#### B. THROUGHPUT MAXIMIZATION UNDER ROUTING CONSTRAINTS (TM-RC)

The more complex variation of the scheduling problem has two objectives. First, it aims to maximize the throughput while ensuring all receivers meet the minimum rate requirement. Second, it maintains the order of transmission in the flow according to the given pre-calculated routes.

Let us denote the operation that returns the time slot when node  $n_a^l$ , in position  $l$  from route  $\vartheta^j$  transmits a message to node  $n_c^{l+1}$  as  $f_s(\vartheta^j, n_a^l)$ . Throughput maximization under routing constraints can be formally expressed as

$$\begin{aligned} & \max \sum_{t=1}^{\mathcal{L}} \sum_{(n_i, n_j) \in \mathcal{K}_\Theta} w \log \left( 1 + \frac{|h_{ij}|^2 p_{ij}^t x_{ij}^t}{\sum_{kj \in \mathcal{K}_\Theta \setminus (n_i, n_j)} |h_{kj}|^2 p_{kj}^t x_{kj}^t + \sigma^2} \right) \\ & \text{s.t. } x_{ij}^t = \{0, 1\}, \forall (n_i, n_j) \in E \\ & \quad \sum_{j=1}^N x_{ij}^t \leq 1, \forall (n_i, n_j) \in E \\ & \quad \sum_{t=1}^{\mathcal{L}} r_{ij}^t \geq R_{ij} \\ & \quad 0 \leq p_{ij}^t \leq P_{max} \\ & \quad \forall \vartheta^j \in \Theta, \forall 1 \leq l < |\vartheta^j| - 1; f_s(\vartheta^j, n_a^{l+1}) < f_s(\vartheta^j, n_b^{l+1}) \\ & \quad \forall \vartheta^j \in \Theta, \forall 1 \leq l < |\vartheta^j| - 1; x_{(n^l, n^{l+1}) \in \vartheta^j}^t = 1 \\ & \quad \forall \vartheta^j \in \Theta, \forall 1 \leq l < |\vartheta^j| - 1; p_{(n^l, n^{l+1}) \in \vartheta^j}^t > 0 \end{aligned} \quad (4)$$

In fact, Equation 4 extends Equation 2, while including an additional constraint related to the transmission schedule within a route. This constraint ensures that each link in route  $\vartheta^j$  is allocated before the subsequent link.

#### IV. THE DNN STRUCTURE

After having formally defined the two problem variants namely throughput maximization under QoS constraints (TM-QoS) and throughput maximization under routing constraints (TM-RC), we present, in the following, the DNNs developed to address these variants. First, we describe *SPCDNet*, a DNN tailored to solve the TM-QoS variant. This includes details on the DNN design and a supervised learning-based training mechanism. Subsequently, we introduce *SPCDNet<sup>R</sup>*, a DNN crafted for the TM-RC variant.

In the domain of DL, our main aim is to identify near-optimal solutions for a broad spectrum of scenarios. A fundamental characteristic in machine learning is that solutions optimized through training data should demonstrate effective generalization when applied to unseen data. For this purpose, we utilize the gradient descent algorithm to minimize the loss function, which serves as an indicator of the divergence between the optimal solution and the prediction generated by the neural network.

Nevertheless, achieving flawless generalization to unseen data is not always possible, often due to issues like overfitting. To tackle this issue, our research implements strategic

$$Req = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad Fil = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

FIGURE 1. Filter matrix calculation.

measures designed to alleviate overfitting and enhance the capacity of generalization of our model. While theoretical results do offer worst-case guarantees for the convergence of the gradient descent based methods [3], [7], [16], it is important to acknowledge that practical outcomes can be influenced by a variety of factors, including the choice of optimization algorithm, the initial parameter setup, and the architecture of the model. In our research, we took great care to prevent our model from overfitting the training data and to ensure its strong generalization to new data.

### A. SPCDNet NETWORK STRUCTURE

In this section, we outline the structure of the DNN designed to address the problem of throughput maximization under QoS constraints. Our proposed solution employs a fully connected neural network featuring two input layers, five fully connected hidden layers ( $HL = 5$ ), and a single output layer.

The network's first input includes the distance matrix ( $Dis$ ) and the matrix of transmission requests ( $Req$ ), concatenated together. It is important to note that  $Req$  is a matrix representation of the set  $\mathcal{K}$ .

The second input is a binary matrix of dimension  $\mathcal{L} \times |N|^2$ , where each column represents all possible (valid) assignments in a specific time slot. We denote this matrix as  $Fil$ . This matrix acts as a binary filter: a nonzero entry indicates that the corresponding node is a transmitter, as depicted in Fig. 1. This design ensures that the DNN does not allocate power to nodes that are not transmitters. To construct  $Fil$ , we first flatten the  $Req$  matrix into a  $1 \times |N|^2$  vector, then concatenate  $\mathcal{L}$  instances of this vector to produce a  $\mathcal{L} \times |N|^2$  matrix.

Fig. 1 depicts an example of the calculation of  $Fil$ . The left side of Fig. 1 shows a matrix  $Req$  of a cluster with 3 nodes, and the right side shows the filtered matrix.

The first hidden layer reshapes the input matrix into a one-dimensional vector of length  $2(\mathcal{L} - 3)|N|^2$ . The second hidden layer then takes this vector and reshapes it again, this time into a one-dimensional vector of length  $2(\mathcal{L} - 2)|N|^2$ . The next two hidden layers continue this reshaping process until the output is a one-dimensional vector of length  $2\mathcal{L}|N|^2$ . The final hidden layer reshapes this vector back into a matrix of dimensions  $2|N|^2 \times \mathcal{L}$ . Each column of the output matrix represents a time slot, with each value within the column signifying the power allocation for a specific link.

The output of the network represents the power allocation for each transmitter in each time slot. We used the sigmoid function as the activation function for the hidden layers. Specifically, we employed the standard sigmoid function:

$$Sig(x) = \frac{1}{1 + \exp(-x)} \quad (5)$$

Moreover, to enforce the power constraint in Equation 3, we implemented a specialized activation function from Equation 6.

$$y(x) = \min(ReLU(x), P_{max}). \quad (6)$$

A detailed explanation of the  $SPCDNet$  architecture is presented in Fig. 2. The output layer, sized  $2|N|^2 \times \mathcal{L}$ , determines the transmission power of each link  $ij$  for each time slot  $0 \leq t < \mathcal{L}$ . The computation for this layer differs from that of the previous layers, as described in Equation 3."

### B. SPCDNet<sup>R</sup> NETWORK STRUCTURE

In this section, we describe the proposed  $SPCDNet^R$ , which is a DNN designed to solve the throughput maximization under routing constraints ( $TM-RC$ ). We provide the architecture details of the DNN, while the supervised training mechanism for both  $SPCDNet$  and  $SPCDNet^R$  is presented in detail in Section IV.

Note that  $TM-RC$  is an extension of  $TM-QoSC$ , with the additional requirement of following the scheduling ordering constraints due to the given request routes. As a result, the network structure of  $SPCDNet^R$  is very similar to the network structure of  $SPCDNet$ . The only difference between  $SPCDNet$  and  $SPCDNet^R$  is in the first input layer, while all remaining layers are the same. In  $SPCDNet^R$  The first input of the network includes the distance matrix concatenated with the matrix of requirements, and  $|\Theta|$  matrices, to represent the transmission requests routes, where each matrix  $j$  includes the order of route  $\vartheta^j$ . A detailed explanation of the  $SPCDNet^R$  architecture is presented in Fig. 3.

## V. EXPERIMENTAL SETUP AND SIMULATION RESULTS

To evaluate the performance of the deep neural network, we conducted a set of experiments with simulated transmission graphs and requirements. In this section, we describe the experimental setup and the simulation results. In particular, we give the reference schemes used for comparison with the DNNs' performance. Then, we describe the data generation process, the division to train and test sets, training details, and the test process. Finally, we provide our numerical results.

### A. REFERENCE SCHEMES

In order to conduct numerical simulations to verify the effectiveness of the proposed  $SPCDNet$  and  $SPCDNet^R$ , we compared them with state-of-the-art power control methods. Thus, we implemented the following schemes to handle the power control problem:

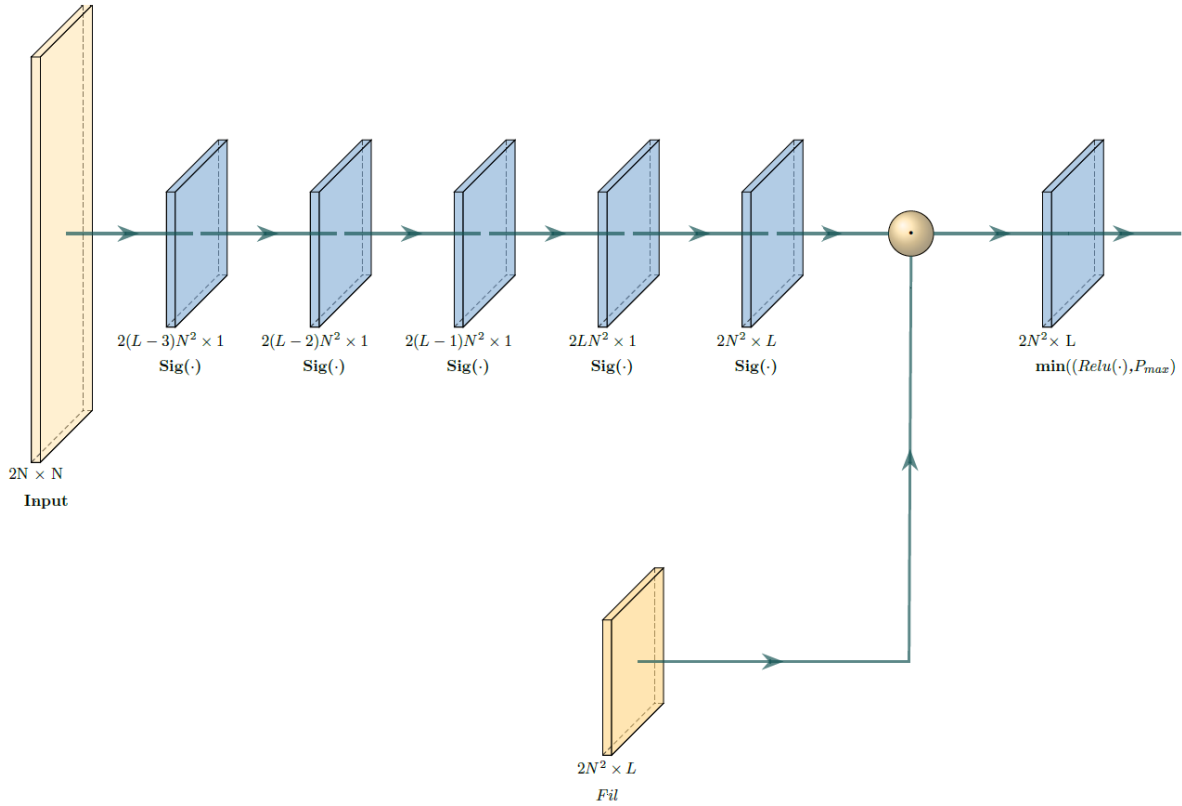


FIGURE 2. DNN architecture for SPCDNet.

#### 1) WEIGHTED MMSE (WMMSE) [30]

We use the well-known WMMSE algorithm as a benchmark. The original WMMSE was designed for problems where variables are beamformer vectors with complex entries. In this paper we adopt the simplified version of the algorithm from [32] where this algorithm proves to work in real domain.

#### 2) ROUND-ROBIN POWER CONTROL (RR)

The RR algorithm proposed in [8] is implemented for comparison. The basic idea of RR is to randomly initialize the power of each transmitter and update the power of one transmitter while keeping others fixed. The algorithm stops when the following condition is satisfied

$$\left| \frac{R(P^{(t)}) - R(P^{(t-1)})}{R(P^{(t-1)})} \right| \leq 10^{-4} \quad (7)$$

where  $R(P^{(t)})$  and  $R(P^{(t-1)})$  denote the throughput in the current and last iterations, respectively.

#### 3) EQUAL POWER SCHEME

In the equal power allocation policy, we allocate a power of  $p = P_{max}/|\mathcal{K}|$  for all transmitters.

#### 4) RANDOM POWER SCHEME

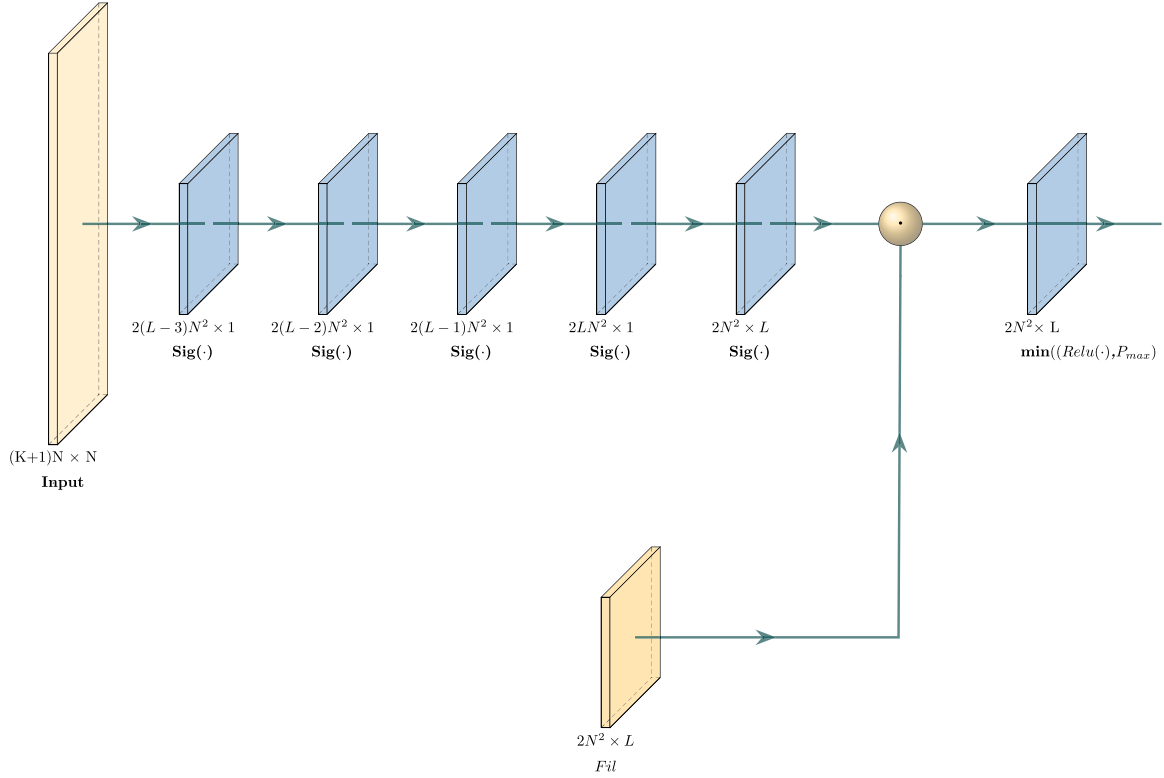
In the random power allocation policy, we allocate a power of  $p \sim U[1, P_{max}]$  for all transmitters.

#### B. DATA GENERATION

To train *SPCDNet* and *SPCDNet<sup>R</sup>* to be able to find efficient solutions for instances of problems *TM-QoSC* and *TM-RC* respectively, we generated training and test data, where each data item consists of a problem instance and an optimal solution. The data were generated as follows. We created two versions of the DNN, as explained above in Sections IV-A and IV-B, respectively. The DNNs were trained using optimal solutions that were calculated by the optimal solver based on linear programming optimization methods. The optimal solver was implemented in the Wolfram Language, where we implement the optimal solver for each problem *TM-QoSC* and *TM-RC* defined in Section III.

First, we explain the process of creating one instance of data for *TM-QoSC*. We randomly distributed the devices on a square area, where the area was  $500m \times 500m$ . Next, uniformly and randomly, we chose the number of devices  $|N| \sim U[4, 10]$ . Then, in a uniform and random manner, we chose the positions of each device in the square area. In the same manner, we chose the set  $\mathcal{K}$  of transmission requests for each instance of *TM-QoSC*.

Recall from Section IV-A that the output of the solver is a matrix of dimension  $\mathcal{L} \times |N|^2$ , where each column represents the time slot, and each value represents the power allocation for a specific link. We denote this matrix as  $M_o$ . Then, we repeated the above process multiple times to generate the dataset. The size of the entire dataset was approximately


 FIGURE 3. DNN architecture for  $SPCDNet^R$ .

500000 instances. We randomly split the dataset into two sets: the training set and the test set, where the size of each set was 80% and 20% of the entire dataset, respectively. We use  $T_{train}$  and  $T_{test}$  to denote the training and test sets, respectively. Therefore, the optimal solver solves Equation 2 for a specific device location and set  $\mathcal{K}$  of requests.

Next, we explain the process of creating one instance of the data for  $TM$ -RC. Similar to  $TM$ -QoSC we located  $|N|$  users in the same way, where we set  $|N| = 15$ . From these users, we randomly chose  $|K|$  pairs of sources and destinations for the routes. Then, for each pair, we chose nodes within the root between the source and destination, using Dijkstra's shortest path algorithm. Then, the set  $\Theta$  is the set of all generated routes, and  $\mathcal{K}_\Theta$  is the set of all transmitter-receiver pairs that appear in any route  $\vartheta \in \Theta$ .

In the last step, we used the optimal solver to solve the problem in this specific scenario. The solver receives, as input, the set  $N$  of users, the set  $\Theta$  of the routes among the sources and destinations, and the positions of each device. As previously mentioned, the output of the solver is also a matrix  $M_o$ , which represents whether, for each time slot  $0 \leq t < \mathcal{L}$  and each pair of transmitter  $n_i$  and receiver  $n_j$ , the link  $(n_i, n_j)$  is active in time slot  $t$ . Therefore, the optimal solvers can solve Equation 4 for a specific node's location and route requests. For this instance of the problem, we created approximately 200000 instances and divide them randomly into two sets: the training set and test set in a ratio of 80% and 20%, respectively.

### C. TRAINING PROCESS

We used the entire training dataset  $T_{train}$ , which includes 80% of the optimally solved instances, to train the neural network weights to be able to calculate, for each instance in the training set, the power control and scheduling solution, which is as close as possible to the solved optimal solution. For the loss function, we used the mean squared error between  $M_o$  and the network's output. We used an Adam optimizer [19] as the step rule for optimization. In addition, we studied the impact of batch size and the learning rate of  $SPCDNet$  evaluated on the validation set and the total training time.

In the experiment described in Fig. 4, we tested different batch sizes and analyzed their influence on the MSE of the test set, for varying number of epochs, with a learning rate of 0.001. Based on the results shown in Fig. 4, we used a batch size of 128 in the remainder of our experiments.

In our next experiment, as presented in Fig. 5, we gradually decreased the learning rate when the validation error did not decrease. In the case of  $SPCDNet^R$  we used the same hyper-parameters as for  $SPCDNet$  and the same optimizer.

#### 1) TESTING PROCESS

In the testing stage, we utilized dataset  $T_{test}$ , passed each instance through the trained  $SPCDNet$  (or  $SPCDNet^R$ ), and collected the result in a matrix. Then, we computed the resulting throughput between the optimal solution and the solution based on the power allocation generated by  $SPCDNet$  (or  $SPCDNet^R$ ).



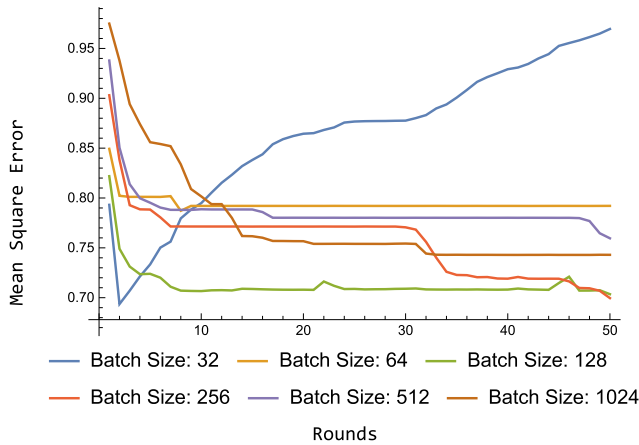


FIGURE 4. Batch size selection for SPCDNet.

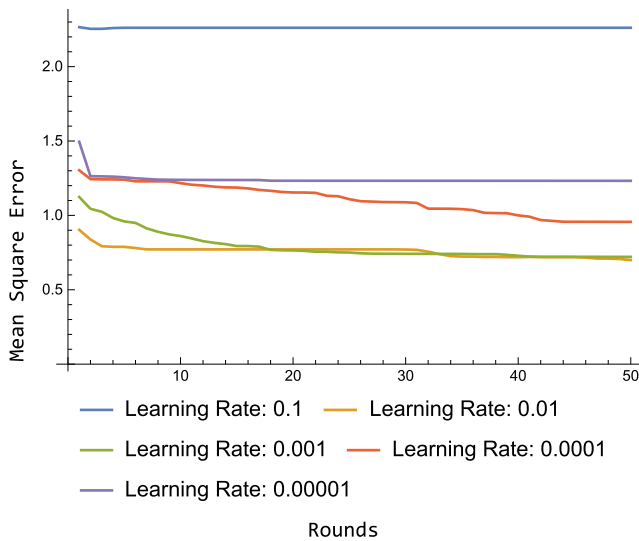


FIGURE 5. Learning rate selection for SPCDNet.

#### D. NUMERICAL RESULTS FOR SPCDNet

To verify the effectiveness of the proposed methods and compare them with the optimal solutions and other schemes, we conducted numerical simulations for various environment parameters while checking several machine learning meta-parameters. The DNN implementation, and training and testing phases were implemented in Wolfram 12.0 environment. The training phase was performed on an Nvidia GPU GeForce GTX 1080Ti, and the test phase was performed on a desktop computer with an Intel CPU Core i7-8700K @ 3.70 GHz. The detailed simulations allowed us to study the approximation ratio of the proposed SPCDNet, expressed as the ratio of the proposed throughput to the optimal throughput. Specifically, we examined the the approximation ratio obtained by SPCDNet for different numbers of nodes and different numbers of active links.

Fig. 6 presents the average approximation ratio of the proposed algorithm versus the optimal solution for the case where the number of nodes  $|N|$  is uniformly distributed  $|N| =$

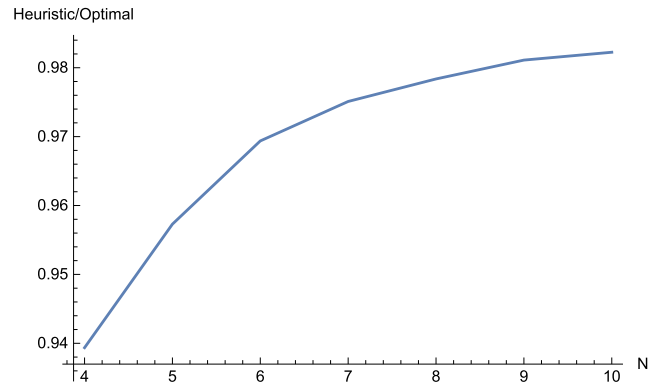


FIGURE 6. Performance ratio of SPCDNet on test data.

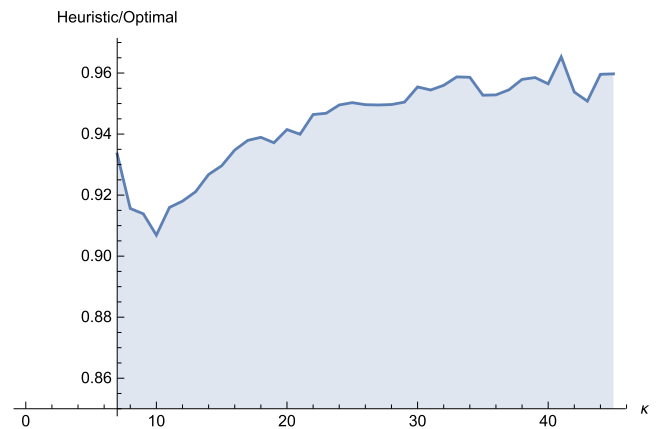
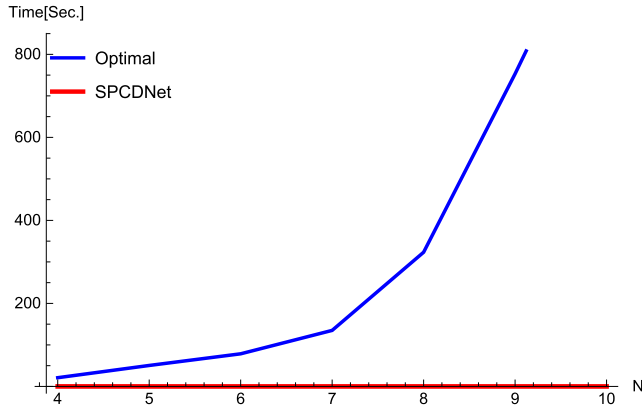
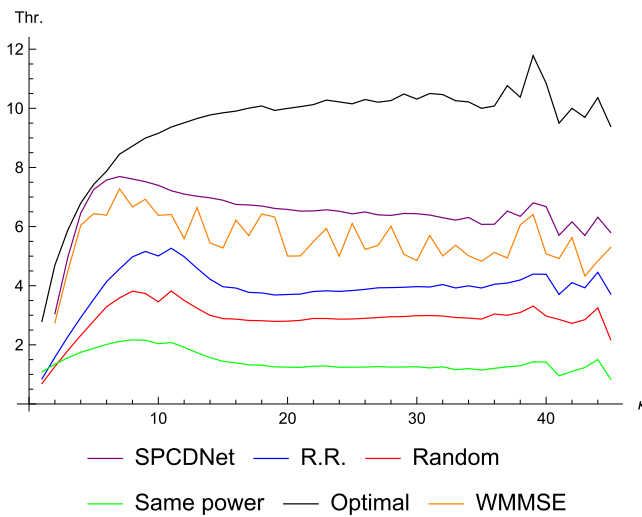


FIGURE 7. Performance ratio with varying numbers of links.

$U \sim [4, 10]$ . This figure shows that SPCDNet achieves near-optimal solutions when the size of the instances in the test set is the same as in the training set.

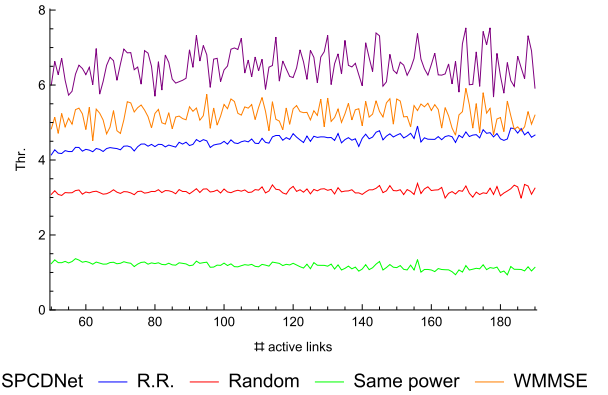
In Fig. 7 we demonstrate the performance of the proposed method as a function of the size of set  $\mathcal{K}$ , when the size of the set  $N$  is uniformly distributed  $|N| = U \sim [4, 10]$  as well. We can see that SPCDNet performed very well with varying numbers of active links. In Fig. 8 we present the runtime of SPCDNet versus the runtime of the optimal solution. SPCDNet's runtime is mainly constant and is indeed several orders of magnitude less than the optimum baseline for networks with a different number of nodes and active links. We would like to emphasize that the runtime presented in the figure corresponds specifically to the test phase, where the DNN model is evaluated after being trained on the provided data. During the training phase, it is crucial to consider the time taken by the optimal solver to solve each instance. This factor plays an essential role in ensuring that the DNN captures relevant features necessary for efficient approximation. By incorporating the runtime of the optimal solver during training, we enable the DNN to learn from the optimal solutions and improve its ability to approximate them accurately. Since it is important to provide a fair comparison between the DNN approach and the optimal solution, we took into account


**FIGURE 8.** Time complexity of *SPCDNet* v.s. optimal solution.

**FIGURE 9.** Performance of *SPCDNet* v.s. comparison schemes.

the time-consuming nature of the data generation phase. Our experiments reveal that the calculation of each instance takes between 10 – 800 seconds. These observations highlight the significant impact of data generation on the overall computational complexity and emphasize the fact that this should be considered when comparing different approaches.

In Fig. 9, we illustrate the performance of *SPCDNet* in comparison with other schemes as a function of the size of set  $\mathcal{K}$ , where the size of  $\mathcal{K}$  varies from 2 to 44. It is important to note that the *SPCDNet* performance demonstrated here is based on test data that have properties similar to the training data. Consequently, the problems encountered in the test data were of the same size as those in the training data set. From the results, we observe that *SPCDNet* performed exceptionally well when facing a varying number of active links.

In Fig. 10, we demonstrate the performance of the *SPCDNet* versus all comparison schemes as a function of the size of set  $\mathcal{K}$ , where the size of  $\mathcal{K}$  varies from 44 to 186. It is important to note that the size of the wireless network in Fig. 10 is larger than the one encountered during the training phase. As a result, the DNN algorithm generates solutions


**FIGURE 10.** Performance when the number of active links is higher than the *SPCDNet* trained on it.

for network configurations which it has not been previously encountered, especially for those with a larger number of users. These results highlight the generalization capabilities of our approach, demonstrating its potential to handle real-world scenarios effectively. Evaluating performance on unseen network instances allows for a more comprehensive assessment of our algorithm's practical applicability.

Recall that the performance of *SPCDNet* is demonstrated on a test dataset, which includes problems of various sizes, including those larger than the ones used for training *SPCDNet*.

In particular, the test data is obtained from graphs with a larger number of nodes and links compared to the training dataset.

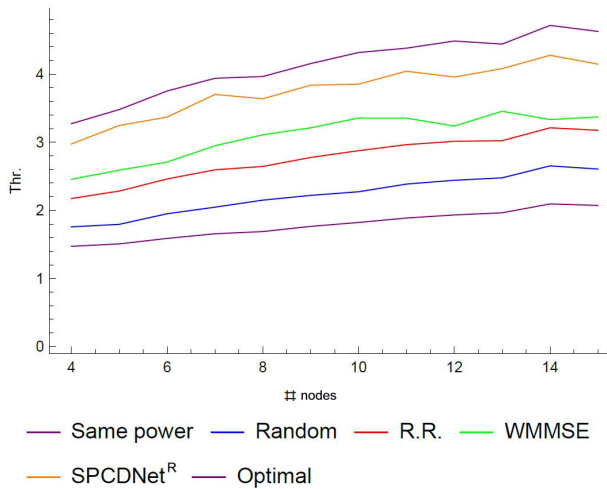
We can see that *SPCDNet* performed very well with varying numbers of active links. Furthermore, we observe the superiority of *SPCDNet* over the comparison schemes (presented in Section V-A) used for handling the power control problem.

In Fig. 13, we present the performance of the *SPCDNet* scheme in comparison to other methods as a function of the set size  $N$ . Here, users are distributed according to a Truncated Normal Distribution process with a mean of 0.5 and a standard deviation of 0.2. As can be seen, *SPCDNet<sup>R</sup>* outperforms the comparison schemes, even when the size of the test graph is larger and users are distributed by a different distribution.

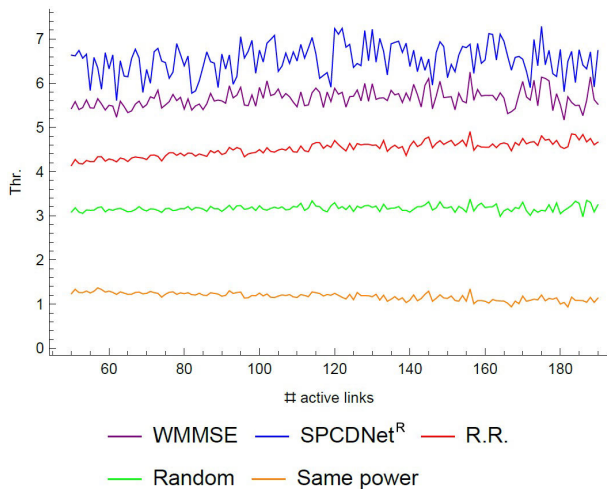
### E. NUMERICAL RESULTS FOR *SPCDNet<sup>R</sup>*

In this section, we describe our results when considering *SPCDNet<sup>R</sup>* for problems involving power control and time schedule, where routing constraints are given in addition to the transmission costs. In Fig. 12, we present the performance of the *SPCDNet<sup>R</sup>* compared to other schemes as a function of the number of active links.

Specifically, we evaluate the performance of *SPCDNet<sup>R</sup>* on a test dataset consisting of graphs with the same number of nodes as the training dataset but with varying numbers of active links. We can observe that *SPCDNet<sup>R</sup>* performs



**FIGURE 11.** Performance of  $SPCDNet^R$ , when the size of the test graph is same as a train graph.

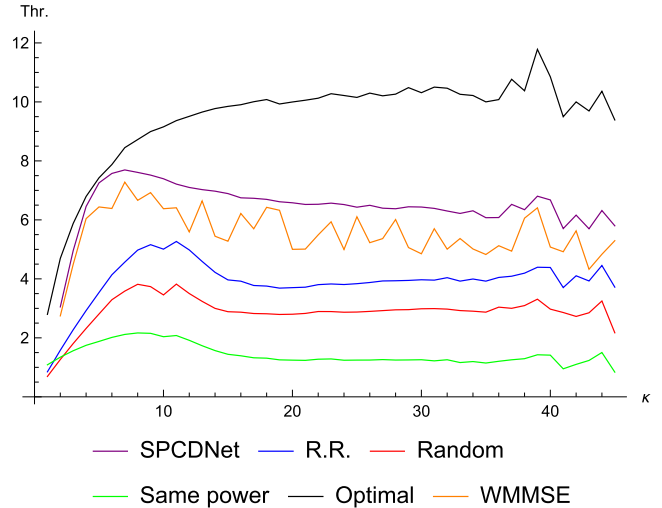


**FIGURE 12.** Performance of  $SPCDNet^R$  when the size of the test graph size varies and may be larger than that of the train graph.

exceptionally well with varying numbers of active links in the graphs across different scenarios.

In Fig. 11, we present the performance comparison of  $SPCDNet^R$  with other schemes as a function of the number of nodes. The figure shows the performance of  $SPCDNet^R$  on the test dataset, where the graphs have a greater number of transmission nodes and more active links compared to the training dataset. As observed,  $SPCDNet^R$  exhibits impressive performance across varying numbers of nodes. The results demonstrate that even for test data with sizes not included in the training,  $SPCDNet^R$  is still able to outperform other schemes and reach near-optimal solutions.

In summary, our experimental results demonstrate that both  $SPCDNet$  and  $SPCDNet^R$  networks, upon completing the training session, are capable of generating near-optimal power control and scheduling solutions. These DNNs effi-



**FIGURE 13.** Performance of  $SPCDNet^R$  when the size of the test graph size varies and may be larger than that of the train graph, and ground user distributed by Poisson distribution.

ciently handle instances of different sizes, requiring minimal computation time during the testing phase.

Our results emphasize the potential of employing DL based method for real-time optimization challenges in communication networks. This approach can be particularly valuable for complex optimization problems, where alternative sub-optimal solutions can deviate significantly from the optimum, and finding exact optimal solutions may be impractical due to the inherent difficulty of solving these complex problems.

## VI. CONCLUSION AND FUTURE WORK

In this study, we considered power allocation and request scheduling in MANET clusters. In particular, we consider the throughput maximization under quality-of-service constraints, in which an ad-hoc cluster of mobile nodes exists, and the cluster head is given a set of transmission requests, given as a transmitter-receiver pairs, that should be allocated respective time slots. The method controls the transmitters' power to maximize the total transmission throughput while satisfying the minimum rate requirement of all receivers. Inspired by recent advances in artificial intelligence, we proposed using deep learning to address the scheduling and power control problem for interference management.

For the first variation of the problem, where a list of requests is provided, we developed  $SPCDNet$  a fully connected multi-layer neural network. This network accepts the distance matrix and requirements as input, and then outputs the transmit power of all transmitters at each time slot. We employed a supervised learning strategy for training  $SPCDNet$  using optimal solutions as the training dataset.

Next, we considered a second variation of the scheduling and power control problem: throughput maximization under routing constraints. The objective now was to maximize the throughput while satisfying the minimum rate requirement of

all receivers, and maintaining the order of transmission in the flow according to the given pre-calculated routes. For this second variation, we developed *SPCDNet<sup>R</sup>* a DNN with a structure similar to *SPCDNet* but with a modification in the first network layer.

We found through simulations that both *SPCDNet* and *SPCDNet<sup>R</sup>* performed exceptionally well with varying numbers of active links across different graphs. We also observed that when presented with test data items of sizes not included in their training, both DNNs were still capable of outperforming other schemes aimed at solving these problem variations.

Our results are encouraging in many respects. The remarkable low time complexity of the DNN and the highly efficient solutions reached by it are impressive. Furthermore, we trained our model on a small instance of the problem and tested it on a larger instance, still achieving excellent results. Thus, the key outcome of our research is that a DNN can serve as a computationally inexpensive component of resource-intensive optimization algorithms in real-time tasks with a very good approximation of these problems, even when trained on small instances of the problem.

There are many interesting challenges that should be addressed in the future. We intend to consider different properties of requests, such as different request sizes and priorities, combining power control with the routing procedure, and handling situations in which interference may be caused by units that are not part of the cluster. In addition, We are also interested in the joint routing and scheduling problem, wherein a multi-layer neural approach should be taken to simultaneously handle both challenges, which depend on each other. A key further step is to create a framework where a trained model can be applied to another related task. Here, transfer learning can be a valuable tool. Lastly, we plan to explore the effect of a clustering structure on the efficiency of resource allocation solutions.

## REFERENCES

- [1] K. I. Ahmed, H. Tabassum, and E. Hossain, "Deep learning for radio resource allocation in multi-cell networks," *IEEE Netw.*, vol. 33, no. 6, pp. 188–195, Nov. 2019.
- [2] Y. Alghorani, A. S. Chekkouri, D. A. Chekired, and S. Pierre, "Improved S-AF and S-DF relaying schemes using machine learning based power allocation over cascaded Rayleigh fading channels," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7508–7520, Dec. 2021.
- [3] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 242–252.
- [4] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak, "A machine learning approach for power allocation in HetNets considering QoS," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.
- [5] C. H. P. Augusto, C. B. Carvalho, M. W. R. da Silva, and J. F. de Rezende, "REUSE: A combined routing and link scheduling mechanism for wireless mesh networks," *Comput. Commun.*, vol. 34, no. 18, pp. 2207–2216, Dec. 2011.
- [6] R. Azoulay, K. Danilchenko, Y. Haddad, and S. Reches, "Transmission power control using deep neural networks in TDMA-based ad-hoc network clusters," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 406–411.
- [7] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [8] C. S. Chen, K. W. Shum, and C. W. Sung, "Round-robin power control for the weighted sum rate maximization of wireless networks over multiple interfering links," *Eur. Trans. Telecommun.*, vol. 22, no. 8, pp. 458–470, Dec. 2011.
- [9] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.
- [10] W. Cui and W. Yu, "Scalable deep reinforcement learning for routing and spectrum access in physical layer," 2020, *arXiv:2012.11783*.
- [11] C. Wu and D. P. Bertsekas, "Distributed power control algorithms for wireless networks," *IEEE Trans. Veh. Technol.*, vol. 50, no. 2, pp. 504–514, Mar. 2001.
- [12] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.
- [13] G. J. Foschini and Z. Miljanic, "A simple distributed autonomous power control algorithm and its convergence," *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 641–646, Nov. 1993.
- [14] Y. Fu, Y. Chen, and C. W. Sung, "Distributed power control for the downlink of multi-cell NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6207–6220, Sep. 2017.
- [15] E. Ghadimi, F. Davide Calabrese, G. Peters, and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.
- [16] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," 2014, *arXiv:1412.6544*.
- [17] D. Goodman and N. Mandayam, "Power control for wireless data," *IEEE Pers. Commun.*, vol. 7, no. 2, pp. 48–54, Apr. 2000.
- [18] K. Danilchenko, R. Azoulay, S. Reches, and Y. Haddad, "Deep learning method for delay minimization in MANET," *ICT Exp.*, vol. 8, no. 1, pp. 7–10, Mar. 2022.
- [19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [20] R. Knopp and P. A. Humblet, "Information capacity and power control in single-cell multiuser communications," in *Proc. IEEE Int. Conf. Commun.*, Jun. 1995, pp. 331–335.
- [21] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, Jun. 2018.
- [22] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen, and H. Li, "Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach," *IEEE Access*, vol. 6, pp. 25463–25473, 2018.
- [23] Z. Li, X. Chen, L. Li, and X. Wang, "Joint routing and scheduling for transmission service in software-defined full-duplex wireless networks," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 422–429, Mar. 2019.
- [24] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760–1776, Mar. 2020.
- [25] L. Luo, J. Zhang, S. Chen, X. Zhang, B. Ai, and D. W. K. Ng, "Downlink power control for cell-free massive MIMO with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6772–6777, Jun. 2022.
- [26] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.
- [27] B. Matthiesen, A. Zappone, E. A. Jorswieck, and M. Debbah, "Deep learning for real-time energy-efficient power control in mobile networks," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.
- [28] Y. Mirsky and Y. Haddad, "A linear downlink power control algorithm for wireless networks," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2013, pp. 1–7.
- [29] G. Qian, Z. Li, C. He, X. Li, and X. Ding, "Power allocation schemes based on deep learning for distributed antenna systems," *IEEE Access*, vol. 8, pp. 31245–31253, 2020.
- [30] 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 Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.
- [31] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[32] 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," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–6.

[33] Y. Sun, C. Chen, and H. Luo, "Adaptive scheduling and routing scheme for delay guarantee in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 8, Aug. 2013, Art. no. 801758.

[34] T. Van Chien, T. Nguyen Canh, E. Björnson, and E. G. Larsson, "Power control in cellular massive MIMO with varying user activity: A deep learning solution," *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 5732–5748, Sep. 2020.

[35] K. Wang, T. Y. Chai, and W.-C. Wong, "Routing, power control and rate adaptation: A Q-learning-based cross-layer design," *Comput. Netw.*, vol. 102, pp. 20–37, Jun. 2016.

[36] K. Wang, W.-C. Wong, and T. Yoong Chai, "An adaptive delay-based power control and routing scheme," in *Proc. 7th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2013, pp. 1–7.

[37] Y. Wang, W. Wang, X.-Y. Li, and W.-Z. Song, "Interference-aware joint routing and TDMA link scheduling for static wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 12, pp. 1709–1726, Dec. 2008.

[38] Z. Wang, M. Eisen, and A. Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 1850–1863, 2022.

[39] Y. Cao, L. Zhang, and Y.-C. Liang, "Deep reinforcement learning for channel and power allocation in UAV-enabled IoT systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[40] Y. Du, K. Wang, K. Yang, and G. Zhang, "Energy-efficient resource allocation in UAV based MEC system for IoT devices," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[41] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

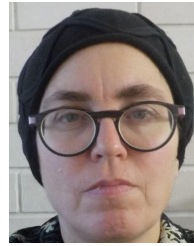
[42] J. Zander, "Distributed cochannel interference control in cellular radio systems," *IEEE Trans. Veh. Technol.*, vol. 41, no. 3, pp. 305–311, Aug. 1992.

[43] A. Zappone, M. Debbah, and Z. Altman, "Online energy-efficient power control in wireless networks by deep neural networks," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2018, pp. 1–5.

[44] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Link scheduling using graph neural networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 6, pp. 3997–4012, Jun. 2023.



**KIRIL DANILCHENKO** received the Ph.D. degree from the Department of Communication Systems Engineering in 2023. Currently, he is a Post-Doctoral Fellow with the University of Waterloo. His research interests include complex network algorithms, modeling, analysis, leveraging optimization methods, and machine learning techniques.



**RINA AZOULAY** received the B.A. degree in economics and computer science and the M.Sc. and Ph.D. degrees in computer science from Bar Ilan University in 2001. She is currently a Senior Lecturer with the Computer Science Department, Jerusalem College of Technology, Israel. Her main research interests include protocols for reaching agreements, the application of machine learning methods, and intelligent tutoring systems.



**SHULAMIT RECHES** received the B.A. degree in mathematics from the Jerusalem College of Engineering and the M.Sc. degree in mathematics and computer science and the Ph.D. degree in computer science from Bar Ilan University in 2011. She is currently a Senior Lecturer with the Mathematics Department, Jerusalem College of Technology, Israel. Her main research interests include artificial intelligent and algebraic combinatorics.



**YORAM HADDAD** (Senior Member, IEEE) received the B.Sc., Engineering Diploma, and M.Sc. degrees (Hons.) in radiocommunications from SUPELEC (leading engineering school in Paris, France) in 2004 and 2005, respectively, the Ph.D. degree (Hons.) in computer science and networks from Telecom ParisTech in 2010, and the Habilitation (H.D.R.) degree from Sorbonne University in 2019. He was a Kreitman Post-Doctoral Fellow with Ben-Gurion University, Israel, between 2011 and 2012. Since 2010, he has been a Faculty Member (actually at the rank of an Associate Professor) with the Jerusalem College of Technology (JCT)-Lev Academic Center, Jerusalem, Israel, and the Head of the FTNet Laboratory. He has published about 50 papers in international conferences (e.g., SODA), book chapters, and journals. He served on the technical program committee for major IEEE conferences. He was the Chair of a working group at European COST action on Autonomous Control for a Reliable Internet of Services (ACROSS). He was a recipient of the Henry and Betty Rosenfelder Outstanding Researcher Award for the year 2013 and of the Computer Science Outstanding Lecturer Award in 2018. His main research interests are in the area of wireless networks and algorithms for networks, wireless software defined networks (SDN) and ML for 5G cellular networks and beyond, and the lightweight security for IoT.