Resource Allocation based on Graph Neural Networks in Vehicular Communications

Ziyan He*, Liang Wang†, Hao Ye*, Geoffrey Ye Li*, and Biing-Hwang Fred Juang*
*Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA
{heziyan, yehao}@gatech.edu; {liye, juang}@ece.gatech.edu
† School of Computer Science, Shaanxi Normal University, Xi'an, China
wangliang@snnu.edu.cn

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322537

***Abstract*—In this article, we investigate spectrum allocation in vehicle-to-everything (V2X) network. We first express the V2X network into a graph, where each vehicle-to-vehicle (V2V) link is a node in the graph. We apply a graph neural network (GNN) to learn the low-dimensional feature of each node based on the graph information. According to the learned feature, multi-agent reinforcement learning (RL) is used to make spectrum allocation. Deep Q-network is utilized to learn to optimize the sum capacity of the V2X network. Simulation results show that the proposed allocation scheme can achieve near-optimal performance.**

***Index Terms*—Vehicular Communications, Multi-agent RL, GNN, Resource Allocation**

I. Introduction

Recently vehicle-to-everything (V2X) networks have attracted numerous studies with the aspiration to make our experience on wheels safer, greener, and more convenient [1], [2]. In the V2X networks, vehicles, pedestrians and other entities on the road are connected and coordinated to provide a whole new collection of applications, varying from improving road safety to mitigating traffic congestion. The vehicular communication is also supported by the 3rd generation partnership project (3GPP) in the fifth-generation communication system (5G) [3], which will further push the development and deployment of the V2X communication systems. Despite its potential to exert a significant impact on daily human life, there exist some significant challenges in V2X networks, such as lack of quality-of-service (QoS) guarantee and time-varying channels and network configuration [4]. Judicious spectrum allocation is thus needed in the vehicular communication networks to deal with environment dynamics and to guarantee QoS. The resource management problem is often formulated as one of combinatorial optimization, which is generally NP-hard lacking low-complexity and effective universal solutions.

To tackle such issues, some works are focused on obtaining a sub-optimal solution to reduce the complexity. In [5], the reliability requirements of vehicle-to-vehicle (V2V) communications are transformed into optimization constraints, which use only the slowly-varying channel information to make it computable. Similarly, a spectrum and power allocation scheme is proposed in [6] to maximize the ergodic capacity of the vehicle-to-infrastructure (V2I) links, requiring only slowly varying large-scale fading information. In [7], graph partitioning algorithms are exploited to divide highly interfering V2V links into disjoint spectrum-sharing clusters before formulating the spectrum sharing problem, which reduces the computational complexity and network signaling overhead.

In recent years, machine learning, especially reinforcement learning (RL), has shown its power in addressing various engineering problems, including resource allocation in communications [8], [9]. The work in [10] shows that the deep RL framework can address the resource management problems and is comparable and some-times better than heuristic based approaches for a multi-resource cluster scheduling problem. In [11], a deep RL approach is developed to dynamically manage the networking, caching, and computing resources. A distributed spectrum sharing scheme based on multi-agent RL is proposed in [12] to enhance the sum capacity of V2I links and the payload delivery rate of V2V links. In [13], each vehicle is considered as an agent and multiple agents are employed to sequentially make decisions to find available spectrum based on their local observations in V2V broadcast communications. In [14], a base station (BS) is used to aggregate and compress observations of vehicles and the compressed information is then fed back to each agent to help improve the distributed decision performance for spectrum sharing in V2X networks.

Apart from the RL-based optimization methods, several works propose to formulate the optimization problem within a graph model framework and to solve them with graph embedding methods. The critical component is the graph neural network (GNN) [15], which is developed to capture the dependence of graphs via message passing between nodes of graphs. Different from standard neural networks, GNN is used to extract the feature of a node from its neighbors and the graph topology. Recent studies have shown that GNN has attained success in network architectures, optimization techniques, and parallel computation [16]. GNN has been combined with the traditional heuristic algorithms in [17] to solve classic NP-hard optimization problems, such as Minimum Vertex Cover and Maximal Independent Set, which are formulated over weighted graphs. In [18], a device-to-device (D2D) network is expressed as a graph, where D2D pairs are nodes and interference links are edges. A GNN is applied to extract the feature for each node, which is used to

This work was supported in part by the National Science Foundation under Grants 1815637 and 1731017.

make link scheduling by a multi-layer classifier.

In order to fully take advantage of both RL and GNN, we develop a distributed GNN-augmented RL spectrum-sharing scheme for the V2X network. In our proposed approach, the V2V network is expressed as a graph. Local observations of the V2V pair and the channel gains of the interference links are regarded as the information of nodes and edges, respectively. We apply GNN to learn the low-dimensional feature of each node corresponding to a V2V pair based on the graph information. To exploit RL for resource allocation, each V2V link can be treated as an agent with the extracted feature as its state. Multi-agent RL is applied to learn to optimize the sum capacity of the V2V and V2I links. With GNNs, the updating of the messages is along the edges of the graph. Different from the algorithm in [14], each vehicle aggregates network information by communicating with its neighbors nearby without the help of the BS. Therefore, the proposed approach is fully distributed.

The rest of the article is arranged as following. In Section II, we present the system model. Then, the GNN-augmented RL resource allocation scheme is devised in Section III. Simulation results are presented in Section IV. Finally, we draw conclusions in Section V.

II. System Model

Consider a V2X network with N cellular users (CUEs) and K pairs of V2V users (VUEs) shown in Fig. 1(a). Each cellular user communicates with the BS, which forms a V2I link to support high data rate services, such as video streaming. The V2V link is formed by neighboring vehicles and enabled by D2D communication. Each of the V2I links is assigned an orthogonal spectrum band. Thus, the size of the channel set is the same as the number of the V2I links. We assume all devices are equipped with a single antenna. The sets of CUEs and VUEs are represented by $\mathcal{N} = \{1, 2, ..., N\}$ and $\mathcal{K} = \{1, 2, ..., K\}$, respectively. All V2V links share the spectrum with V2I links to enhance the spectrum utility. Hence, the channel set for all links can also be denoted by $\mathcal{N}$.

As shown in Fig. 1(a), the channel gain of the n-th V2I link is denoted as g_n and the interference channel gain from the transmitter of the k-th V2V pair to the BS over the n-th channel is represented as $h^n_{k,B}$. Hence the capacity of the n-th V2I link can be obtained as

$$R^C_n = B \log_2 \left(1 + \frac{P^C_n |g_n|^2}{\sum_{k=1}^{K} \rho^n_k P^V_k |h^n_{k,B}|^2 + \sigma^2} \right) \quad (1)$$

where B is the bandwidth, σ^2 refers to the power of the Gaussian noise, and P^C_n and P^V_k denote the transmission powers of the n-th CUE and k-th VUE, respectively. In (1), we introduce a binary indicator ρ^n_k, with $\rho^n_k = 1$ if the k-th V2V pair is activated on the n-th channel and $\rho^n_k = 0$ otherwise. Similarly, the rate of the k-th V2V link over the n-th channel can be expressed as

$$R^V_k[n] = B \log_2 \left(1 + \frac{\rho^n_k P^V_k |h^n_{kk}|^2}{\sum_{l \neq k}^{K} \rho^n_l P^V_l |h^n_{lk}|^2 + P^C_n |g^n_k|^2 + \sigma^2} \right) \quad (2)$$

where g^n_k represents the interference channel gain of the n-th CUE to the k-th V2V pair on the n-th channel and h^n_{lk} refers to the channel gain from the l-th V2V pair transmitter to the k-th V2V pair receiver over the n-th channel. From (1) and (2), the set of spectrum allocation indicators $\{\rho^n_k | k \in \mathcal{K}, n \in \mathcal{N}\}$ is critical to the maximization of the capacity of the V2X network.

In general, the V2V links mainly carry essential safety information, while the V2I links support less important entertainment services [12]. In order to ensure the QoS of the V2X network, the transmission of V2V links should be given the priority and supported with high reliability. As a result, the spectrum allocation problem can be formulated as the following optimization problem

$$\max_{\boldsymbol{\rho}} R = \sum_{k=1}^{K} \sum_{n=1}^{N} R^V_k[n] + \omega_C \sum_{n=1}^{N} R^C_n, \quad (3)$$

subject to

$$\rho^n_k \in \{0, 1\}, \forall k \in \mathcal{K}, n \in \mathcal{N}; \quad \sum_{n=1}^{N} \rho^n_k = 1 \quad (4)$$

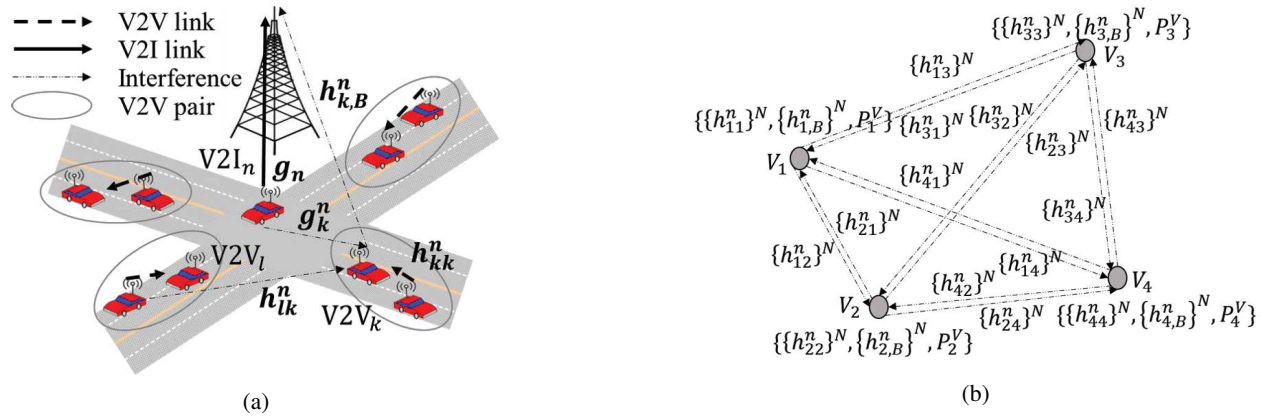
where $\boldsymbol{\rho}$ denotes $\{\rho^n_k | \forall k \in \mathcal{K}, n \in \mathcal{N}\}$ and ω_C is the weight for the sum rate for the V2I link according to the priority. The second constraint in (4) is due to the assumption that each V2V link occupies only one channel.

III. Resource Allocation based on GNN

In this section, we first formulate the V2X network as a graph model and develop a GNN-based technique to extract features relevant to resource optimization for V2V pairs. Then we discuss how to use the GNN-extracted features to address the optimization problem (3) based on RL.

A. GNN Method Design

In order to apply GNNs, the vehicular communication network is first expressed as a graph. Inspired by [18], each of the V2V pairs is regarded as a node while interference links between V2V pairs as edges. The node observation contains the VUEs channel gain and its corresponding transmit power while the edge weights are represented by the interference channel gain. For the network shown in Fig. 1(a), the directed graph representation is given by Fig. 1(b), where $\{h^n_{lk}\}^N = (h^1_{lk}, h^2_{lk}, ..., h^N_{lk})$ and $\{h^n_{k,B}\}^N = (h^1_{k,B}, h^2_{k,B}, ..., h^N_{k,B})$. Thus, the observation of node v and the weight of the link from node u to node v can be written as $\boldsymbol{x}_v = \{\{h^n_{vv}\}^N, \{h^n_{v,B}\}^N, P^V_v\}$ and $\boldsymbol{e}_{uv} = \{h^n_{uv}\}^N$, respectively. Note that only VUEs are included in the graph representation. Hence, all the observations about CUEs, such as the channel gain from the CUE and its transmission power, are not part of the node observation.



(a) (b)

Fig. 1: (a) The structure of vehicular networks. (b) The graph representation of the vehicular network shown in (a).

After expressing the network as a graph, we use GNN to extract node features. The concept of GNN has been first introduced in [15], which aims at learning a feature embedding $\boldsymbol{\mu}_v \in \mathbb{R}^{n_s}$ of each node, containing the information of adjacent nodes and edges, where n_s is the dimension of the feature embedding. The i-th iteration of $\boldsymbol{\mu}_v$ is described as

$$\boldsymbol{\mu}_v^i = f(\boldsymbol{x}_v, \boldsymbol{x}_{ne}[v], \{\boldsymbol{e}_{uv}\}_{u\in N(v)}, \boldsymbol{\mu}_v^{i-1}, \{\boldsymbol{\mu}_u^{i-1}\}_{u\in N(v)}), \tag{5}$$

where $\boldsymbol{x}_{ne}[v]$ denotes the observation of node v's incoming neighbors, $N(v)$ represents the set of the node v's incoming neighbors, i.e., nodes that have an link to node v, and $f(\cdot)$ is the updating function we need to design.

Motivated by the popular GNN framework in [19], the updating function for the proposed GNN is derived by

$$\boldsymbol{\mu}_v^i = \sigma\left(\boldsymbol{W}_v^i \left[\boldsymbol{x}_v || \boldsymbol{\mu}_v^{i-1} || \sum_{u\in N(v)} \boldsymbol{e}_{uv} || \sum_{u\in N(v)} \boldsymbol{\mu}_u^{i-1}\right]\right), \tag{6}$$

where $\cdot||\cdot$ denotes the vector concatenation and $\boldsymbol{W}_v^i$ is the trainable weight of the node v for the i-th iteration. The rectified linear unit (ReLU) is adopted as the activation function of GNN, $\sigma(x) = \max(0, x)$.

In general, $\boldsymbol{\mu}_v^0 = \mathbf{0}$ at the beginning of feature embedding. To reduce the network signaling overhead, only feature embeddings, $\{\boldsymbol{\mu}_v^i | v \in V\}$, are exchanged among the adjacent nodes at the i-th iteration, where $V = \mathcal{K}$ is the set of the nodes corresponding to V2V pairs. After I iterations, extracted features for nodes are attained as $\{\boldsymbol{\mu}_v^I\}_{v\in V}$.

B. Distributed Deep Q-Network Design

After extracting the features of V2V pairs from GNN, the Q-network is developed to select the spectrum for each V2V pair, which is treated as an agent in the RL framework. For the k-th agent at the time step t, the state is defined as $\boldsymbol{s}_k^{(t)} = \left\{\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_k^{I,(t)}\right\}$, and the action is given by $\boldsymbol{a}_k^{(t)} = \boldsymbol{\rho}_k^{(t)}$, where $\boldsymbol{\rho}_k^{(t)} = \{\rho_k^{1,(t)}, \rho_k^{2,(t)}, ..., \rho_k^{N,(t)}\}$. The reward is designed as R in (3) for all agents globally, i.e., $r_k^{(t+1)} = R^{(t+1)}, \forall k \in \mathcal{K}$. The superscript (t) represents the information as obtained at the time step t. The whole structure of the GNN-RL scheme is shown in Fig. 2.

Algorithm 1 Training Process for the Proposed Framework

Input: GNN structure, Q-network structure for each V2V, and the environment simulator

Output: GNN and allocation policy π_k represented by Q-network $\mathcal{Q}_k$ with parameter $\boldsymbol{\theta}_k$, for all $k \in \mathcal{K}$

Initialisation : Initialize GNN and all Q-network models

for each training episode **do**
  Start simulator and generate vehicles and links.
  for time-step $t = 0, ..., T-1$ **do**
    Observe the graph information denoted by $\boldsymbol{O}^{(t)}$ including node observations $\boldsymbol{x}_k^{(t)}, \forall k \in \mathcal{K}$ and edge weights $\boldsymbol{e}_{lk}^{(t)}, \forall l, k \in \mathcal{K}$
    Each V2V utilizes the proposed GNN in (6), and after I iterations, extracts its feature $\boldsymbol{\mu}_k^{I,(t)}$
    Each V2V takes $\boldsymbol{s}_k^{(t)} = \left\{\boldsymbol{x}_k^{(t)}, \boldsymbol{\mu}_k^{I,(t)}\right\}$ and selects action $\boldsymbol{a}_k^{(t)}$ based on state $\boldsymbol{s}_k^{(t)}$, according to the policy π_k derived from $\mathcal{Q}_k$, e.g., ϵ-greedy policy [20]
    All V2V receive reward $R^{(t+1)}$
    Channels update and new graph information of the next time step $\boldsymbol{O}^{(t+1)}$ is obtained
    Store $\{\boldsymbol{O}^{(t)}, \boldsymbol{a}^{(t)}, R^{(t+1)}, \boldsymbol{O}^{(t+1)}\}$ in memory $\mathcal{M}$
    Sample mini-batch $\mathcal{D}$ from $\mathcal{M}$ uniformly
    for each V2V agent k **do**
      Use $\mathcal{D}$ to train GNN with parameters $\boldsymbol{W}$ and the k-th Q-network with parameters $\boldsymbol{\theta}_k$ jointly by minimizing the mean square error (MSE) (7) between estimation return and the Q-value
      Update the k-th target Q-network: $\boldsymbol{\theta}_k^- \leftarrow \boldsymbol{\theta}_k$ every N_{target} time steps
    end for
    Update target GNN: $\boldsymbol{W}^- \leftarrow \boldsymbol{W}$ every N_{target} time steps
  end for
end for

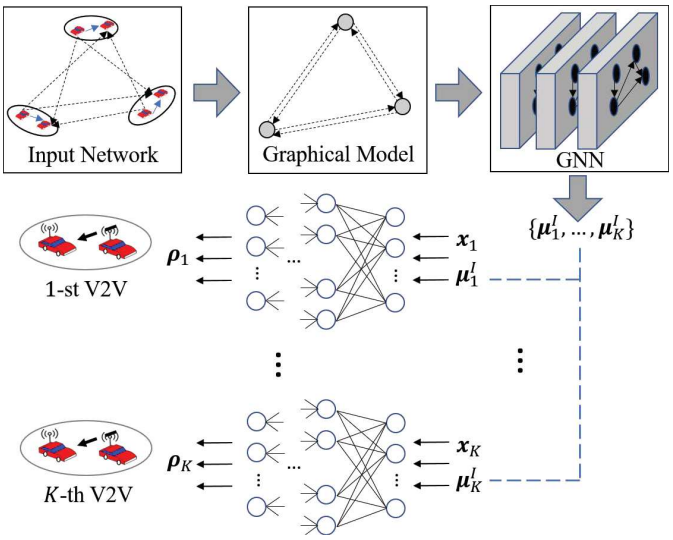


Fig. 2: The structure of GNN-RL.

The details of the training process for the proposed GNN-augmented RL framework is demonstrated in Algorithm 1. Here, $\boldsymbol{a}^{(t)} = \{\boldsymbol{a}_1^{(t)}, ..., \boldsymbol{a}_K^{(t)}\}$ and the MSE for the k-th agent is defined as

$$\sum_{\mathcal{D}} [R^{(t+1)} + \gamma \max_{\boldsymbol{a}'} \mathcal{Q}_k(\boldsymbol{s}_k^{(t+1)}(\boldsymbol{W}^-, \boldsymbol{O}^{(t+1)}), \boldsymbol{a}'; \boldsymbol{\theta}_k^-) - \mathcal{Q}_k(\boldsymbol{s}_k^{(t)}(\boldsymbol{W}, \boldsymbol{O}^{(t)}), \boldsymbol{a}_k^{(t)}; \boldsymbol{\theta}_k)]^2, \quad (7)$$

where γ is the discount parameter. $\boldsymbol{s}_k^{(t)}(\boldsymbol{W}, \boldsymbol{O}^{(t)})$ implies the state is derived by the GNN with parameters $\boldsymbol{W} = \{\boldsymbol{W}_k^i | \forall k \in \mathcal{K}, i = 1, ..., I\}$, which infers that parameters $\boldsymbol{W}$ and $\boldsymbol{\theta}_k$ can be trained simultaneously by minimizing (7).

IV. Simulation Results

The simulation scenario is set up for the urban case in Annex A of [3]. The simulation area size is 1,299 m $\times$ 750 m, where the detailed parameters and the corresponding channel models are the same as Table I and II in [14], respectively. The weight of the sum rate for the V2I link is set as $\omega_C = 0.1$. We assume the dimension of the feature n_s is the same for all $\boldsymbol{\mu}_k^i$, $\forall k \in \mathcal{K}$ and $i = 0, 1, ..., I$. Hence, the trainable weights $\boldsymbol{W}_k^i$ of the GNN have the same dimension $n_s \times n_g$, where n_g is the length of the concatenate graph information $\left[\boldsymbol{x}_v || \boldsymbol{\mu}_v^{i-1} || \sum_{u \in N(v)} \boldsymbol{e}_{uv} || \sum_{u \in N(v)} \boldsymbol{\mu}_u^{i-1}\right]$. The total iteration of the GNN is set to be $I = 3$. The Q-network for each V2V pair has three fully connected hidden layers with 80, 40, and 20 neurons, respectively. The ReLU function is chosen as the default activation function of the Q-network. Besides, the activation function of the last iteration of GNN and the output layers in Q-networks are set to be a linear function. Adam optimizer [21] is adopted to update the trainable parameters with a learning rate of 0.001 during training. The number of training and testing episodes is 10,000 and 2,000, respectively. Each episode consists $T = 1,000$ time steps. We adopt ϵ-greedy policy [20] to balance the exploration and exploitation and the exploration rate, ϵ, linearly decreases from 1 to 0.01 over the beginning 8,000 episodes of training. The discount factor, γ, is chosen as 0.05. The target networks update their parameters every 500 steps. The size of the mini-batch is set as 512.

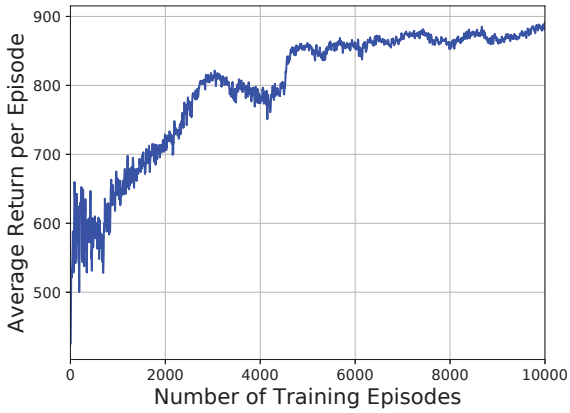


Fig. 3: Average return per episode in training stage

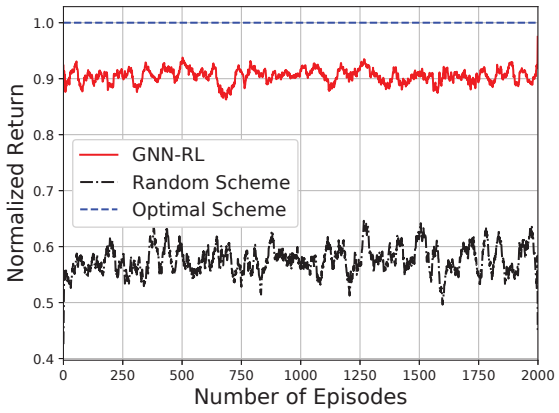


Fig. 4: The return comparison among GNN-RL scheme, random scheme and optimal scheme with $N = K = 4$

Fig. 3 shows the average return against the number of training episodes with $N = K = 4$ and $n_s = 20$. The average return increases dramatically at the beginning and gradually converges with the increasing training episodes, which demonstrates the number of training episodes, $10,000$, is enough to guarantee the convergence of the proposed scheme. Scheme training converges slowly, but it is not a significant problem since the training stage is totally offline.

Fig. 4 compares return R defined in (3) among three resource allocation schemes with $N = K = 4$ and $n_s = 20$. The optimal scheme applies a brute-force full search to select the optimal channel for each V2V pair. In the random scheme, V2V pairs choose the spectrum uniformly in the channel set $\mathcal{N}$. Here, we normalize the return with the maximum return obtained by the optimal scheme and smooth the return

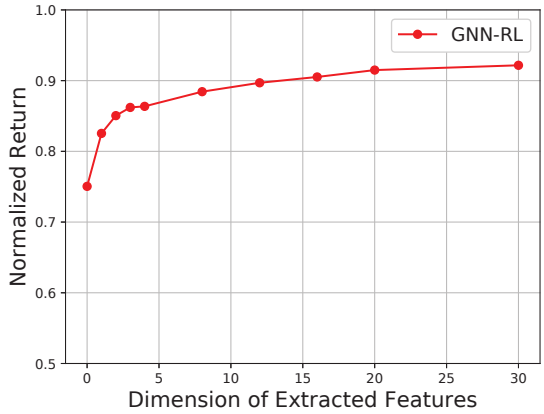


Fig. 5: Average normalized return against the dimension of the extracted features n_s

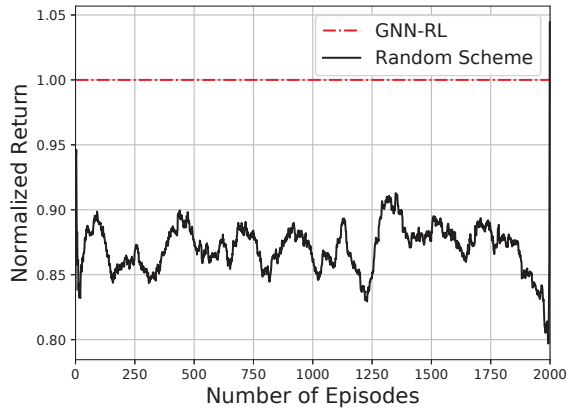


Fig. 6: The return comparison between GNN-RL scheme and random scheme with $N = K = 10$

over adjacent episodes for clarity in demonstration. From the figure, the average performance of GNN-RL framework is approximate 90%, while the random scheme can only achieve below 60% of the optimal.

Fig. 5 depicts the average normalized return of the GNN-RL framework against n_s with $N = K = 4$. In particular, $n_s = 0$ implies that each V2V pair makes the decision only based on its local observation and GNN is not working. The result in Fig. 5 indicates that GNN can aggregate more information for decision making and significantly enhance allocation performance.

In Fig. 6, we set $N = K = 10$, $n_s = 20$ and compare the return between the GNN-RL scheme and he random scheme. Note that the optimal scheme is deemed computationally infeasible in this case. The return is normalized with the return of the GNN-RL scheme. In the large V2X network with $N = K = 10$, the random scheme can achieve around 83% of the GNN-RL framework.

V. Conclusion

We present a GNN-augmented RL spectrum allocation scheme for vehicular networks. GNN extracts the feature of each V2V pair according to the network topology and neighbors' observations. Based on the extracted features and local observations, V2V pairs can make decisions distributively with Q-networks. Simulation results show that the proposed GNN-RL scheme can achieve near-optimal performance.

References

[1] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: A physical layer perspective," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10647–10659, Dec. 2017.

[2] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, Feb. 2018.

[3] "Technical Specification Group Radio Access Network: Study on LTE-based V2X Services; (Release 14)," *3GPP, TR 36.885 V14.0.0*, Jun. 2016.

[4] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "Lte for vehicular networking: a survey," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 148–157, May. 2013.

[5] W. Sun, E. G. Ström, F. Brännström, K. C. Sou, and Y. Sui, "Radio resource management for D2D-based v2v communication," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6636–6650, Aug. 2015.

[6] L. Liang, G. Y. Li, and W. Xu, "Resource allocation for D2D-enabled vehicular communications," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 3186–3197, Jul. 2017.

[7] L. Liang, S. Xie, G. Y. Li, Z. Ding, and X. Yu, "Graph-based resource sharing in vehicular communication," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4579–4592, Jul. 2018.

[8] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.

[9] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341 – 356, Feb. 2020.

[10] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proc. 15th ACM Workshop Hot Topics Netw.*, Nov. 2016, pp. 50–56.

[11] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[12] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[13] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[14] L. Wang, H. Ye, L. Liang, and G. Y. Li, "Learn to compress CSI and allocate resources in vehicular networks," *IEEE Trans. Commun.*, vol. 68, no. 6, pp. 3640–3653, Jun. 2020.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[16] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[17] Z. Li, Q. Chen, and V. Koltun, "Combinatorial optimization with graph convolutional networks and guided tree search," in *Proc. Adv. NIPS*, 2018, pp. 539–548.

[18] M. Lee, G. Yu, and G. Y. Li, "Graph embedding based wireless link scheduling with few training samples," *arXiv preprint arXiv:1906.02871*, 2019.

[19] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. NIPS*, 2017, pp. 1024–1034.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2018.

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