Edge Intelligence for Multi-Dimensional Resource Management in Aerial-Assisted Vehicular Networks

Haixia Peng, Huaqing Wu, and Xuemin (Sherman) Shen

Abstract

A new architecture with drone-assisted multi-access edge computing (MEC) is proposed for vehicular networks to support computation-intensive and delay-sensitive applications and services. Artificial intelligence (AI)-based resource management schemes are developed such that terrestrial and aerial spectrum, computing, and storage resources can be cooperatively allocated for guaranteeing the quality of service requirements from different applications. A case study on the joint management of the spectrum and computing resources is presented to demonstrate the effectiveness of AI-based resource management schemes.

Introduction

Benefiting from advanced wireless communication technologies, onboard computing, and artificial intelligence (AI), vehicles are gradually becoming connected and autonomous [1]. With connected and autonomous vehicle (CAV) technologies, vehicles are allowed to communicate and be driven with less human intervention. In addition to navigation, more computation-intensive and delay-sensitive intelligent transportation services are emerging, such as smart sensing for automated driving, in-car entertainment, and collision warning, to achieve better driving safety and onboard experience. However, vehicles usually have limited onboard computing and storage resources to support the increasing computation needs and satisfy the stringent service delivery requirements [2]. Moreover, services delivered from remote servers (e.g., high-definition [HD] map and video streaming) consume long data transmission delay, and some of the onboard computing tasks offloaded to the remote server also incur long computation response delay. Therefore, the allocation of spectrum and computing resources is essential but challenging in satisfying stringent latency service requirements [3].

To address the aforementioned challenge, combining multi-access edge computing (MEC) with CAV technologies to realize a MEC-enabled vehicular network (MVNET) architecture has drawn increasing attention from both industry and academia [2]. By connecting MEC servers to ground base stations (BSs), additional computing resources are provided for on-demand task processing. Benefiting from the physical proximity between vehicles and MEC servers, computing tasks can be efficiently offloaded with reduced response delay and low task offloading cost. However, a MEC-mounted BS usually has preset computing and storage resources, which may lead to imbalanced supply-to-demand match due to spatio-temporal variations in vehicular communication and computing demands. Taking the advantage of maneuverability and flexibility from aerial devices (e.g., drones[1]), extending the MEC technology to enable aerial computing can effectively mitigate this issue in the MVNET [5–7].

Despite the advantages of real-time aerial computing, aerial-assisted MEC complicates the network architecture. The heterogeneous and dynamic network environment necessitates efficient resource management to improve resource utilization while guaranteeing the quality of service (QoS) requirements for diversified vehicular applications. Moreover, different applications may demand multiple types of resources; for example, spectrum and computing resources are demanded by computing task offloading, while spectrum and storage resources are demanded by content delivery and content caching. How to design a comprehensive resource management framework to adapt to various applications is critical to vehicular networks but challenging due to the highly dynamic network environment [8]. Artificial intelligence (AI) technologies, especially machine learning, are being employed in many technological fields. As a data-driven technology, AI can be applied to CAV networks to realize intelligent service delivery by leveraging a vast amount of data collected from vehicles or road infrastructures (e.g., toll booths and traffic lights). Moreover, with the ability to learn to approximate various functional relations in a complex and dynamic network environment, AI can also be applied to solving resource management problems in real time [9].

In this article, we comprehensively investigate how to use edge intelligence for multi-dimensional resource management to enhance overall resource utilization while guaranteeing vehicular applications' QoS requirements. Specifically, we propose a drone-assisted MVNET architecture, which combines the MEC and drone technologies to enable aerial computing in vehicular networks. This drone-assisted MVNET architecture can effectively accommodate the highly dynamic spectrum, computing, and storage resource demands, and sup-

[1] The carry payloads of a drone are growing, and the energy system is becoming more and more powerful, which makes it possible to enable MEC-mounted drones [4].

Haixia Peng is with California State University Long Beach; Huaqing Wu and Xuemin (Sherman) Shen are with the University of Waterloo.

Digital Object Identifier: 10.1109/MWC.101.2100056

1536-1284/21/$25.00 © 2021 IEEE

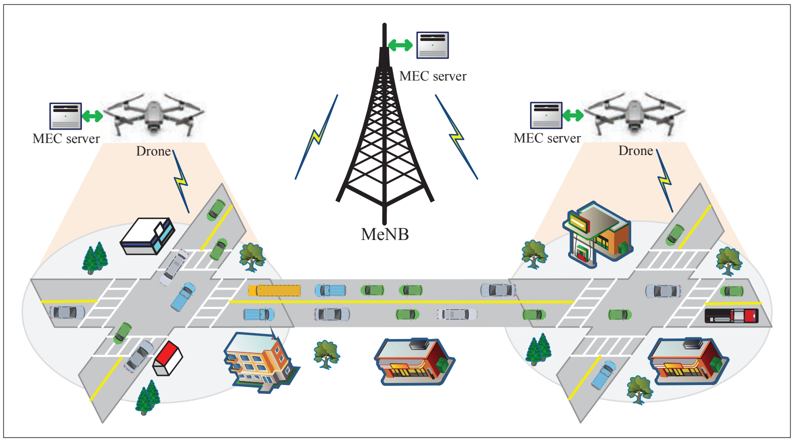

FIGURE 1. An illustration of the drone-assisted MVNET.

port diversified vehicular applications. Also, both terrestrial and aerial MEC can provide resources to support the execution of AI-based algorithms. With the proposed architecture, we comprehensively study how to effectively manage the multi-dimensional resources with AI technologies in terms of network-level spectrum slicing among BSs, spectrum/computing resource management, and joint resource management. To demonstrate the performance of applying AI technologies in resource management, a case study is conducted to evaluate the performance of a joint spectrum and computing resource management scheme. Open research issues are discussed, followed by concluding remarks made in the final section.

A Drone-Assisted MVNET Architecture

In this section, we first summarize the CAV applications in vehicular networks and then present an MVNET architecture with aerial computing.

MEC for Vehicular Networks

For the emerging CAV applications and services, the following new requirements are introduced.

Ultra-Low Latency: In addition to safety-related applications, some advanced use cases in vehicular networks also require ultra-low data transmission latency [10]. For example, to support vehicle platooning[2] for cooperative driving, the maximum end-to-end delay for information exchange among vehicles within the same platoon is 25 ms.

Ubiquitous Connectivity: When driving in urban, suburban, or rural areas, vehicles demand ubiquitous connectivity to support a large range of intelligent transportation system (ITS) applications.

Computation-Intensive: In addition to low latency and ubiquitous connectivity, computing resources are also required by some ITS applications to process generated tasks such as image processing tasks for driving assistance and video analysis tasks for collision warning [11]. Moreover, some new AI-powered services, for example, automated driving and infotainment services with natural language processing, generally have intensive computing requirements.

The MEC technology, bringing the computing and storing capabilities physically close to the end users, is expected to fulfill these requirements and has been widely applied to support vehicular applications [2]. Compared to cloud computing, vehicles do not have to upload/download the related data to/from the core network in an MVNET. Thus, the backhaul link transmission burden can be relieved, and a short response delay for task computing or content delivery can be achieved to satisfy the ultra-low-latency requirements. Moreover, some application programming interfaces (APIs), such as Group Specification (GS) MEC 012 API for radio network information service [2], are defined for the MEC technology to stimulate application innovations and enable flexible implementation of new services.

Drone-Assisted MVNET Architecture

In a terrestrial MVNET, MEC servers are placed at ground BSs to provide computing/storing capabilities for moving vehicles with a short response delay. However, the amount of resources carried by each MEC-mounted BS is usually fixed, which makes it challenging for the network to accommodate the dynamic numbers of offloaded computing tasks and content delivery tasks. Through mounting MEC servers on aerial devices, computing and content requests from moving vehicles can be cooperatively supported and the overall spectrum efficiency can be improved by aerial-to-ground communications on short-distance line-of-sight channels.

As illustrated in Fig. 1, we consider a drone-assisted MVNET where a macro eNodeB (MeNB) and several drones are equipped with MEC servers. The group of drones, denoted by $\mathcal{U}$, fly within the coverage of the MeNB. Both manually operated traditional vehicles and autonomous vehicles with limited onboard computing/storage resources are moving on the road and randomly generate tasks with different delay requirements to support their specific applications. The tasks generated by each vehicle are either communication tasks for data dissemination or computing tasks that need to be offloaded for further processing. In the drone-assisted MVNET, the computing resource provider can be the MeNB, the drones, or the autonomous vehicles[3] with available onboard computing/storage resources. Once a computing task is generated, the vehicle first sends a task offloading request, including the size, CPU cycles, and/or delay requirement of the task, to the MEC-mounted MeNB, a drone, and/or the neighboring autonomous vehicles. According to all received requests, the computing resource provider returns a spectrum/computing resource allocation decision to the vehicle, and then performs task processing once it receives the vehicle's computing task.

Learning-Based Resource Management

In the drone-assisted MVNET, the MEC-mounted MeNBs and drones cooperatively provide communication coverage and on-demand computing resources to different vehicular applications. However, with high vehicle mobility and time-varying resource demands generated by heterogeneous applications, it is difficult to conduct efficient and real-time resource allocation decisions via conventional optimization methods. With recent advances in AI technologies, machine-learning-based methods (e.g., model-free deep reinforcement learning) provide an effective way to solve the resource management problems. The learning architecture for resource management in the drone-assisted MVNET is depicted in Fig. 2. The

[2] Vehicle platooning is a technology that enables vehicles to dynamically form a platoon or a convoy to travel together.

[3] We assume that an autonomous vehicle only offers computing support to its neighboring vehicles.

resource management policy can be learned by neural networks (NNs). With states and actions being input and output, we can train the NNs for obtaining the optimal resource management policy to maximize the accumulated reward. In what follows, we investigate how to apply reinforcement learning methods to address the resource management problems in the drone-assisted MVNET, as summarized in Table 1.

Spectrum Resource Management

We summarize the process of spectrum management from different time granularities.

Long-Term Spectrum Slicing: This is slicing the aggregated spectrum resources among different BSs. Time is partitioned into $\widehat{W}$ consecutive slicing windows, where each slicing window is composed of $\widehat{t}$ time slots.[4] The spectrum slicing is controlled by the radio resource management (RRM) unit at the MeNB and is executed at the beginning of each slicing window. Define the communication workload of a BS as the number of communication tasks generated by vehicles within its coverage. Based on the average communication workloads of the MeNB and drones in each slicing window, the MeNB makes spectrum slicing and communication workload division decisions for the MeNB and drones. Available spectrum resources, S_{max}, are sliced with ratios α_m and α_d and are allocated to the MeNB and drones, respectively, where the drones share the spectrum resources, $S_{max}\alpha_d$, to increase the multiplexing gain. Furthermore, the communication workload under the coverage area of drone i is divided into two parts with the partitioning ratios, $\beta_{i,m}$ and $\beta_{i,u}$, which are assigned to the MeNB and drone i, respectively.

Short-Term Spectrum Allocation: This is operated at the beginning of each time slot within a slicing window. Once the spectrum slicing is done in a target slicing window, the amounts of spectrum resources available to the MeNB and the drones are fixed at $\alpha_m S_{max}$ and $\alpha_d S_{max}$, respectively. At a time slot within the target slicing window, vehicles send their communication requests to accessible MEC servers. According to the received communication requests, the MeNB and drone i then cooperatively determine the vehicle-MeNB and vehicle-drone association patterns for vehicles under the coverage of drones and allocate proper amounts of spectrum resources to the associated vehicles.

In what follows, we take the long-term spectrum slicing for uplink transmission as an example to show how to utilize the learning-based methods for spectrum management (the short-term spectrum allocation is studied below).

For time slot t within a target slicing window, vehicles generate communication tasks with probability p^s, size $\sigma(\tau)$, and delay requirement $\tau^s(t)$. Let the MeNB act as an agent. The environment state at a target slicing window can be designed according to traffic flow, task generation frequency, and the average spectrum efficiencies from a vehicle to the MeNB and from a vehicle to drone i. Actions taken under a given environment state should include the spectrum slicing ratios and communication workload partitions. To guarantee the delay requirements for different communication tasks while maximizing the overall spectrum utilization, a higher reward is obtained when more tasks are completed within τ^s, and a low reward is achieved otherwise.

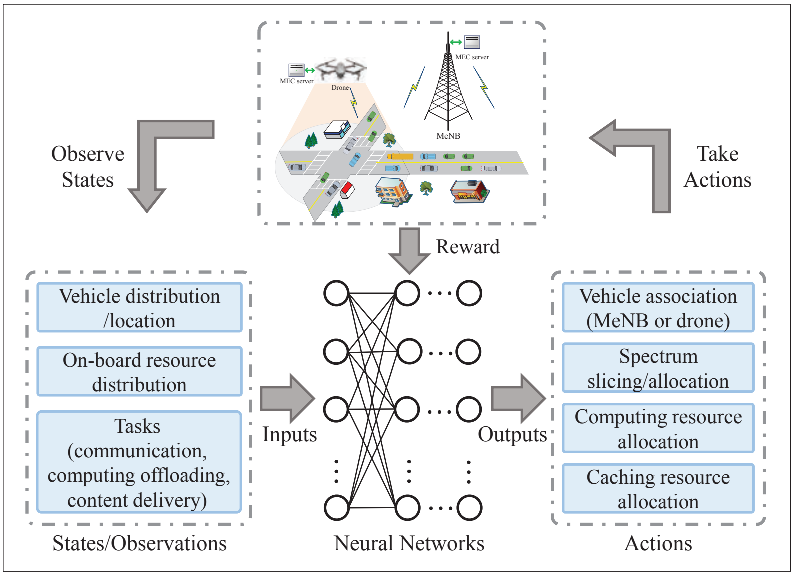

FIGURE 2. Learning architecture overview.

As both spectrum slicing ratios and workload partitions are continuous variables, the spectrum slicing policy can be efficiently learned by adopting policy-gradient algorithms, such as actor-critic, deep deterministic policy gradient (DDPG), and proximal policy optimization, for solving the long-term spectrum slicing problem. With the learned spectrum slicing policy, the agent can make an optimal spectrum slicing and workload division decision for each given environment state to achieve high spectrum utilization in the long run.

Computing Resource Management

Different levels of computing capabilities are fixed at edge servers and autonomous vehicles. Thus, different from spectrum resources, managing the overall computing resources among vehicles is always achieved via adjusting the vehicle association patterns for offloading computing tasks to different resource providers. With a given spectrum management strategy, the computing resource management problem can be transformed into a distributed computing task offloading decision making problem with an acceptable communication overhead to the network.

To be consistent with practical applications, we assume each vehicle generates a computing task with probability p^c at a given time slot. Different computing tasks generated by vehicles have random task sizes, numbers of CPU cycles required for task processing, and maximum tolerable delays. In each time slot, the MeNB, drones and autonomous vehicles update their current available computing resources to vehicles within their service areas. According to the received resource information, each vehicle makes an offloading association decision[5] and offloads its computing task to the associated provider. When receiving a computing task offloaded from a vehicle, the provider allocates the same number of CPU cycles as required by the vehicle to process the task if it has enough available computing resources, and discards the vehicle's task otherwise.

As vehicles compete for the providers' computing resources with each other, we can model the distributed task offloading decision making prob-

[4] With a larger $\widehat{t}$, the spectrum slicing cost would be reduced while resulting in low overall spectrum utilization. Thus, dynamically adjusting the slicing window size to balance this trade-off is critical.

[5] Allowing each vehicle to make its own offloading association decision would achieve relatively low computational complexity for resource management. We can also let each resource provider centrally allocate its computing resources among vehicles.

	Spectrum resource management	Computing resource management (given spectrum management strategy)	Joint spectrum and computing resource management
Objective	Maximize spectrum resource utilization	Maximize computing resource utilization	Maximize spectrum/computing resource utilization
Constraint	Communication delay requirement	Computing delay requirement, task migration cost	Total time consumption requirement
States	1. Traffic flow 2. Task generation frequency 3. Vehicle-MEC spectrum efficiency	1. Vehicle location 2. Task offloading requests 3. Available computing resources	1. Vehicle location 2. Task offloading requests 3. Vehicle-provider spectrum efficiency 4. Available spectrum and computing resources
Actions	1. Spectrum slicing ratios 2. Communication workload division	Vehicle-provider association	1. Vehicle-provider association 2. Computing resource allocation 3. Spectrum resource allocation
Rewards	Spectrum resource utilization	Computing resource utilization	Spectrum and computing resource utilization
Decision making manner	Centralized	Distributed	Centralized/distributed
Decision making frequency	Every slicing window (long-term)	Every time slot (short-term)	Every time slot (short-term)

TABLE 1. The application of AI technologies to resource management.

lem as a competitive multi-agent learning problem [12]. Let vehicles with generated computing tasks act as learning agents. The local observation of a learning agent can be built according to its received resource information from the accessible providers, namely, the amounts of available computing resources carried by the MeNB and the drone as well as neighboring autonomous vehicles. Under a local observation, each agent makes a discrete action, including the offloading association pattern between the vehicle and the provider. To balance the task migration cost and task processing delay, an agent can define its reward by combining these two parts. For the task offloaded to a provider, a positive reward should be returned when the task is completed within the tolerated delay while involving a penalty according to the task migration cost. Since each agent has discrete actions, learning algorithms that are suitable for dealing with discrete actions, such as deep Q network (DQN), can be employed to learn each agent's task offloading decision making policy. However, due to the resource competition relation among different vehicles, each agent may face a non-stationary environment when other agents are adjusting their policies. To address this issue, the fingerprint method can be adopted, which allows each agent to augment its observation space with the estimated policies of other agents.

Joint Resource Management

The computing task offloading problem involves both spectrum and computing resource management. In general, there are five steps to complete a computing task offloaded from a vehicle to a resource provider:

Step 1: The vehicle sends a task offloading request to the provider.

Step 2: The provider allocates proper amounts of spectrum and computing resources for transmitting and processing the vehicle's task.

Step 3: The vehicle offloads its task to the provider over the allocated spectrum resources.

Step 4: The provider processes the received task with the allocated computing resources.

Step 5: The provider returns the task processing result to the corresponding vehicle.

Considering the small sizes of task offloading request and task processing result, transmission time on steps 1 and 5 are usually ignored. Thus, the total time consumption on the entire computing task offloading process mainly depends on how efficiently and optimally the spectrum and computing resources are allocated during steps 2–4.

The objective of the joint spectrum and computing resource management is to maximize the number of offloaded tasks with satisfied delay requirements. Thus, the vehicle-provider association patterns, including vehicle to MeNB, vehicle to drone *i*, and vehicle to neighboring autonomous vehicle *v* association patterns, are to be optimized. The proportions of the spectrum and computing resources allocated to each vehicle from the associated resource provider are also to be optimized. When formulating the joint resource management problem, we use the step function to evaluate whether an offloaded task's delay requirement is met or not, where the function value is equal to 1 when the delay requirement is satisfied. Furthermore, the problem has vehicle association constraints that each vehicle can offload its task to only one provider, and the spectrum and computing resource constraints on the providers.

By solving the joint resource management problem, the optimal vehicle-provider association patterns and the optimal fractions of spectrum/computing resources allocated to each vehicle are obtained. Due to the high network dynamic, coupled relation among the management of spectrum and computing resources, the integer decision variables (i.e., vehicle-provider association pattern variables), and the non-convex objective function, solving the problem using conventional optimization methods is computationally complex. Thus, we next show how to solve the problem by adopting learning-based methods to obtain an efficient joint resource management scheme. Considering different network scales and supportable wireless communication overhead, centralized and distributed methods are employed here.

Centralized Method: For scenarios with a relatively small number of vehicles within the coverage of the MeNB, we let the MeNB act as an agent and build a single-agent learning framework. The environment state at a given time slot can be designed according to the received information about locations, computing tasks, and available computing resources from vehicles and drones. Actions made by the agent should include the vehicle-provider association patterns and the fractions of spectrum/computing resources allocated to each vehicle from all providers. During the learning stage, the agent's policy is adjusted to obtain the optimal action that maximizes the accumulated reward. To avoid allocating more resources to parts of the tasks for achieving a higher reward and guarantee fairness among vehicles, we integrate the logarithm function to define each vehicle's reward. Namely, the vehicle's reward is defined as $\log_2(T'/T^{de})$, where T^{de} and T' denote the tolerable delay and the total time consumed for the vehicle's task offloading and processing, respectively. Then the system reward can be designed as the average reward over all vehicles within the coverage of the MeNB. Considering the discrete action elements and the continuous action elements, the learning algorithms that are appropriate for parametrized actions (a combination of discrete and continuous action elements), such as parametrized DQN (P-DQN) and multi-pass DQN (MP-DQN), should be employed here. The discrete action elements can also be loosened into real numbers for using policy-gradient algorithms.

Distributed Method: To avoid spectrum and time cost on wirelessly updating drones' and autonomous vehicles' resource information to the MeNB, we can allow each provider to manage its own spectrum/computing resources. Therefore, the joint resource management problem can be decomposed into multiple subproblems, each of which is solved by one provider. Let the providers act as learning agents and receive computing task offloading requests from vehicles under their service areas. The observation of the MeNB agent is the same as the state in the centralized method. For drone i and autonomous vehicle v, their observations only include the information of vehicles' locations and generated tasks under their service area. Actions made by a provider agent, including the MeNB agent, drone i agent, and autonomous vehicle v agent, are designed according to vehicle-provider association patterns and the fractions of spectrum/computing resources allocated by the provider to the vehicles. As vehicles under the service areas of a drone and an autonomous vehicle are also covered by the MeNB, non-stationarity would occur when one agent's observation no longer reflects the actual environment dynamics due to the other agents' policy changing at the same time. Therefore, instead of building separate learning modules for each agent, we integrate the learning algorithms, such as P-DQN, MP-DQN, and policy-gradient algorithms, with a multi-agent learning architecture. By adopting a cooperative multi-agent learning algorithm [9], each agent's learning module is trained simultaneously to achieve a maximum long-term common reward. In the implementation stage, each agent can make real-time resource management and task offloading decisions based on the trained module.

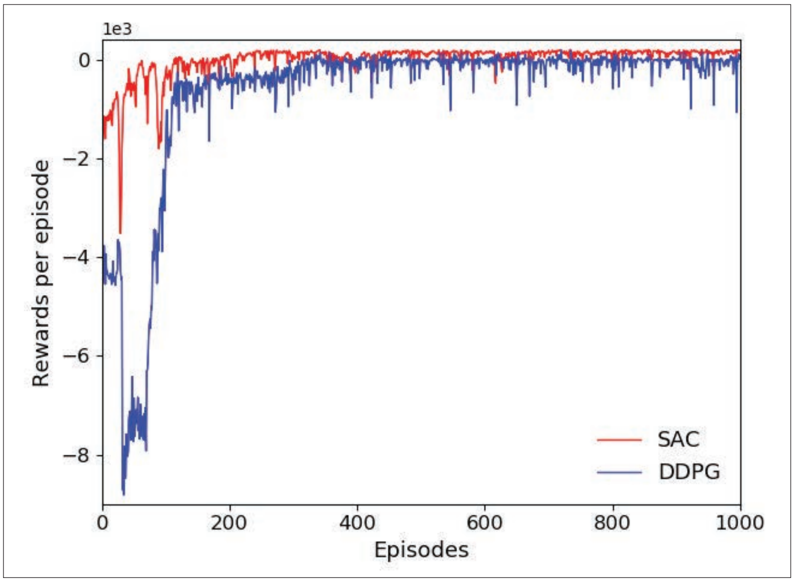

FIGURE 3. Convergence performance of the learning algorithms.

Case Study

In this section, we conduct a case study to demonstrate the effectiveness of employing learning-based methods to solve the joint spectrum and computing resource management problem in a drone-assisted MVNET. Two drones are flying with a constant speed of 10 m/s and an altitude of 40 m above a two-lane bidirectional road segment to cooperatively provide computing capabilities for vehicles within the communication coverage of an MeNB. The amounts of spectrum resources available to vehicle-to-MeNB and vehicle-to-drone communications are 6 MHz and 0.5 MHz, respectively. The communication range of the MeNB is 600 m, and that of a drone is 100 m. We use VISSIM to generate a real vehicle moving trace. At each time slot, a vehicle generates a computing task, where the task size is in [0.5, 1] kb, the required number of CPU cycles for task processing is in [50, 100] cycles/s, and the maximum delay tolerated by the task is in [10, 50] ms.

For the joint spectrum and computing resource optimization problem, we relax the integer decision variables (i.e., vehicle-MeNB and vehicle-drone association pattern variables) into real numbers in [0,1]. Based on the centralized method discussed previously, we utilize two policy-gradient algorithms, DDPG and soft actor-critic (SAC), to learn the joint resource management policy. Specifically, the DDPG and SAC algorithms are appropriate for continuous actions and can achieve good convergence performance [13]. However, the sizes of states and actions for the learning architecture both depend on the time-varying vehicle density, which is contradictory to the fixed sizes of input and output of an NN. Thus, to handle this issue, we set a relatively large size for both state and action, and let an element of the state or action be 0 when there is no vehicle corresponding to that element.

Figure 3 shows the convergence performance of the SAC and DDPG algorithms in solving the

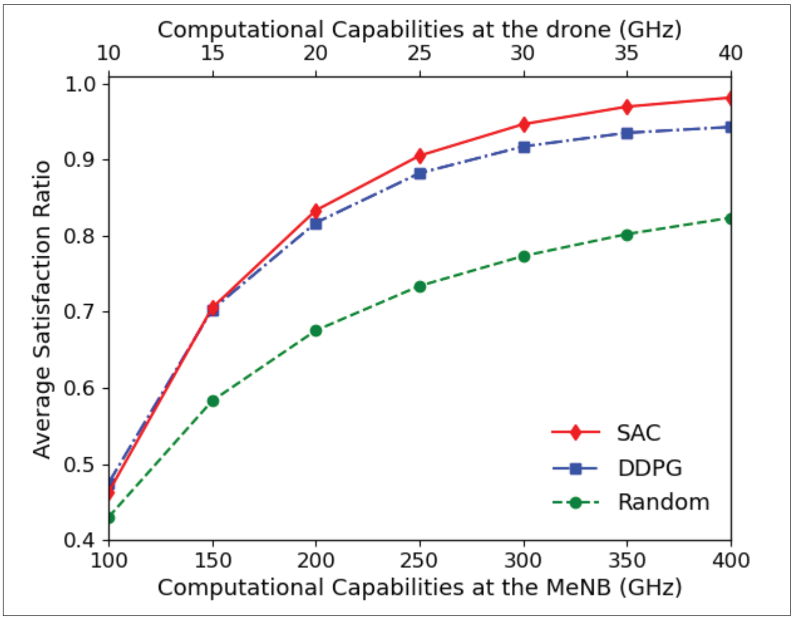

FIGURE 4. Performance of the joint resource management schemes.

joint spectrum/computing resource management problem. The x-axis represents the number of simulation episodes, where each episode contains 1000 time slots. The y-axis shows the total rewards achieved by the agent in each episode. From the figure, we can see that under the same simulation setting,[6] the SAC algorithm converges after around 200 episodes, which is 100 episodes faster than the DDPG algorithm. Moreover, the SAC algorithm can achieve higher total rewards in each episode with less fluctuation. This is because compared to the DDPG algorithm, the SAC algorithm can further maximize the entropy of the policy while optimizing it for higher cumulative rewards.

The definition of reward indicates that a positive reward can be achieved when the delay requirement of an offloaded task is satisfied by the spectrum and computing resources allocated by the MEC server. During the training stage, with a higher reward, the SAC algorithm can better manage the overall spectrum/computing resources to satisfy the offloaded tasks' requirements than DDPG. To further demonstrate the performance of the SAC- and DDPG-based joint resource management schemes, we use the average delay satisfaction ratio, defined as the average proportion of offloaded tasks that are completed within its tolerated delay over 10,000 time slots, as the evaluation criterion in the test stage. Figure 4 shows the delay satisfaction ratios achieved by the SAC-based, DDPG-based, and random resource management schemes under scenarios with different amounts of available computing resources. Considering the limited battery power and relatively small coverage area of drones, we assume the computing capability of each drone is one-tenth that of the MeNB. We can see that with the increase of amounts of available computing resources, more computing resources can be allocated to each offloaded task, resulting in higher average delay satisfaction ratios for the three schemes. Compared to the random resource management scheme, learning-based schemes can achieve better performance. Moreover, under the same scenario, higher delay satisfaction ratios are achieved by the SAC-based resource management scheme than the DDPG-based one.

[6] When training the DDPG and SAC algorithms, we set the learning rates for the actor and critic to be 0.02 and 0.0002, respectively. The reward discount factor used in the Q-value functions is 0.9. The smoothing coefficient used to softly update the parameters of the target networks is 0.01. The replay buffer size and batch size are 10,000 and 32, respectively.

Open Research Issues

To effectively manage the multi-dimensional resources in drone-assisted MVNETs, there are open research issues that require further investigation. Designing more flexible resource management schemes to dynamically accommodate the diverse applications still needs effort. Moreover, investigating comprehensive and data-privacy-preserving resource management schemes that integrate drone trajectory planning and satellite technology to further improve resource utilization is still in its infancy.

Multi-Dimensional Resource Management

In addition to providing computing resources for computing-intensive applications, MEC servers can also benefit many data-sensitive applications, such as video streaming and online gaming. Caching some popular content at the edge can help to mitigate the traffic congestion in the core network and improve content delivery efficiency. Optimally placing the content at the MEC servers to maximize the storage resource utilization while allocating a proper amount of spectrum for delivering the content to vehicles is one of the critical research problems in the MVNETs. On the other hand, with more vehicular applications emerging, diverse tasks such as computing task offloading and content delivery usually require multi-dimensional resource support. Designing multi-dimensional resource management schemes to effectively manage the uplink/downlink communication, computing, and storage resources is still urgent.

Considering the heterogeneous QoS requirements of different applications and the highly dynamic vehicular network environment, AI technologies can be utilized to effectively manage the multi-dimensional resources in drone-assisted MVNETs. For scenarios where diversified vehicular services have different priorities, the procedure of multi-dimensional resource management can be summarized in two stages: multi-dimensional resource planning among services (also referred to as service slicing) and multi-dimensional resource scheduling among vehicles. We can slice the multi-dimensional resources among different services based on priority level and the statistical resources demanded by each service via either optimization or AI methods. To reduce the complexity of the learning algorithm and increase the convergence performance, the multi-dimensional resource scheduling problem can be decomposed into the computing task offloading subproblem and content placement/delivery subproblem, and then solved by combining learning algorithms with the hierarchical learning architecture [14].

Trajectory Planning

In our designed joint resource management scheme, we assume each drone flies with a fixed trajectory. In general, the trajectory of each drone should be dynamically adjusted according to the traffic flow, resource demand distribution, and drone power consumption to further improve the overall resource utilization and energy efficiency in drone-assisted MVNETs. For example, by optimally adjusting the flying direction and speed in each time slot, we can increase the service coverage and endurance of a drone. However, the drones' trajectory planning significantly increases the network environment dynamic and challenges the convergence of the learning algorithms. How to design a learning architecture and build the learning model to cooperatively manage the overall resources and plan the trajectory for drones still needs effort. One of the potential methods is to decompose trajectory planning from resource management and build separate learning models for each of them. To cooperatively learn the two models, the drone's trajectory obtained from the trajectory planning model should be regarded as a part of the environment state of the resource management learning model.

Satellite Integration

Integrating satellite networks with drone-assisted MVNETs (i.e., enabling satellite-air-ground MVNETs) can provide seamless signal coverage, high-capacity communication, and more flexible Internet access to support the increasing number of applications and services [15]. For example, satellite antennas can be embedded into the roof of a vehicle to allow the vehicle to receive satellite signals even without signal coverage of the MeNB or drones. Moreover, embedding satellite antennas into small BSs to enable satellite-based small cells can provide vehicles with a high-capacity backhaul to access the cloud computing server. However, due to the mobility of vehicles, drones, and satellites, challenges arise in cooperatively managing the overall resources, including the satellite/terrestrial spectrum resources and the computing/storage resources carried by the cloud server, edge server, and autonomous vehicles. How to achieve fast, adaptive, and efficient resource management is an urgent issue for satellite-air-ground MVNETs.

Data-Privacy Preserving

Data privacy preservation is always an important concern for vehicles. Vehicles have to share their data with the cloud or MEC servers to support some AI-powered applications and services. For example, to enable automated driving, connected vehicles need to share their sensing data with the server to learn the optimal automated driving strategy. How to preserve the privacy of vehicles' data while allowing them to benefit from these applications is very important. One of the potential technologies to preserve data privacy in AI-powered applications is federated learning, also known as collaborative learning. With the federated learning technique, we can enable decentralized learning, namely, building the learning model at the server while training it in vehicles without uploading raw data to the server.

Conclusion

In this article, we have proposed a drone-assisted MVNET architecture to facilitate responsive and adaptive multi-dimensional resource management. With the developed AI-based resource management schemes, computation-intensive and delay-sensitive applications can be supported with satisfied QoS requirements in the highly dynamic vehicular network scenarios. For our future works, we will integrate satellite communication technologies to enable a satellite-air-ground MVNET for global coverage and diversified applications support.

References

[1] X. Ge, "Ultra-Reliable Low-Latency Communications in Autonomous Vehicular Networks," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 5, May 2019, pp. 5005–16.

[2] F. Giust *et al.*, "Multi-Access Edge Computing: The Driver Behind the Wheel of 5G-Connected Cars," *IEEE Commun. Stds. Mag.*, vol. 2, no. 3, Sept. 2018, pp. 66–73.

[3] Y. Du *et al.*, "Joint Resources and Workflow Scheduling in UAV-Enabled Wirelessly-Powered MEC for IoT Systems," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 10, Oct. 2019, pp. 10,187–200.

[4] "WSU Researchers Trial Liquid Hydrogen for UAVs"; https://www.h2-view.com/story/wsu-researchers-trial-liquid-hydrogen-for-uavs/, accessed May 8, 2021.

[5] W. Shi *et al.*, "Drone Assisted Vehicular Networks: Architecture, Challenges, and Opportunities," *IEEE Network*, vol. 32, no. 3, May/June 2018, pp. 130–37.

[6] X. Chen *et al.*, "Information Freshness-Aware Task Offloading in Air-Ground Integrated Edge Computing Systems," 2020, arXiv preprint arXiv:2007.10129.

[7] "Building an Ecosystem for Responsible Drone Use and Development on Microsoft Azure"; https://azure.microsoft.com/en-ca/blog/building-an-ecosystem-for-responsible-drone-use-and-development-on-microsoft-azure/, accessed Jan. 1, 2021.

[8] W. Zhuang *et al.*, "SDN/NFV-Empowered Future IoV with Enhanced Communication, Computing, and Caching," *Proc. IEEE*, vol. 108, no. 2, Feb. 2020, pp. 274–91.

[9] H. Peng and X. Shen, "Multi-Agent Reinforcement Learning Based Resource Management in MEC- and UAV-Assisted Vehicular Networks," *IEEE JSAC*, vol. 39, no. 1, Jan. 2021, pp. 131–41.

[10] 3GPP TS 22.186 V15.3.0, "Service Requirements for Enhanced V2X Scenarios" (Release 15), July 2018; https://www.etsi.org/deliver/etsi ts/122100 122199/122186/15.03.00 60/ts 122186v150300p.pdf

[11] P. Dai *et al.*, "Multi-Armed Bandit Learning for Computation-Intensive Services in MEC-Empowered Vehicular Networks," *IEEE Trans. Vehic. Tech.*, vol. 69, no. 7, July 2020, pp. 7821–34.

[12] X. Chen *et al.*, "Computation Offloading in Beyond 5G Networks: A Distributed Learning Framework and Applications," arXiv preprint arXiv:2007.08001, 2020.

[13] T. Haarnoja *et al.*, "Soft Actor-Critic Algorithms and Applications,", 2019; https://arxiv.org/abs/1812.05905.

[14] T. D. Kulkarni *et al.*, "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation," *Adv. Neur. In.*, 2016, pp. 3675–83.

[15] N. Zhang *et al.*, "Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions," *IEEE Commun. Mag.*, vol. 55, no. 7, July 2017, pp. 101–09.

Biographies

Haixia Peng [M'21] received her two Ph.D. degrees in computer science and electrical and computer engineering from Northeastern University, Shenyang, China, in 2017 and the University of Waterloo, Ontario, Canada, in 2021, respectively. She is now an assistant professor with the Department of Computer Engineering and Computer Science, California State University Long Beach. Her current research focuses on satellite-terrestrial vehicular networks, multi-access edge computing, resource management, and reinforcement learning.

Huaqing Wu [M'21] received her Ph.D. degree from the University of Waterloo in 2021. She received her B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, China, in 2014 and 2017, respectively. She is now a postdoctoral research fellow at McMaster University. She was the recipient of the Best Paper Award at IEEE GLOBECOM 2018. Her current research interests include vehicular networks with emphasis on edge caching, wireless resource management, space-air-ground integrated networks, and application of artificial intelligence for wireless networks.

Xuemin (Sherman) Shen [M'97, SM'02, F'09] is currently a university professor with the Department of Electrical and Computer Engineering, University of Waterloo. His research focuses on network resource management, wireless network security, social networks, 5G and beyond, and vehicular networks. He is a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Chinese Academy of Engineering Foreign Fellow. He received the 2021 Canadian Award for Telecommunications Research, the R.A. Fessenden Award in 2019 from IEEE, Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, and the Education Award in 2017 from the IEEE Communications Society.

With the developed AI-based resource management schemes, computation-intensive and delay-sensitive applications can be supported with satisfied QoS requirements in the highly dynamic vehicular network scenarios. For our future works, we will integrate the satellite communication technologies to enable a satellite-air-ground MVNET for global coverage and diversified applications support.