TASK OFFLOADING IN VEHICULAR MOBILE EDGE COMPUTING

A Matching-Theoretic Framework

Bo Gu and Zhenyu Zhou

©ISTOCKPHOTO.COM/ANDREY SUSLOV

1556-6072/19©2019IEEE

Mobile edge computing (MEC) is an emerging technology that leverages computing, storage, and network resources deployed in the proximity of users to offload terminals from computation-intensive and delay-sensitive tasks. In this article, a vehicular MEC system is studied where edges, such as MEC servers deployed at roadside units (RSUs), and vehicles with excessive computing resources are able to provide offloading opportunities to other vehicles having limited computation capabilities. We first review the latest MEC research. Then, we focus on the task of offloading in an incomplete information environment and model the interactions between tasks and edges as a matching game. Our main objective is to minimize the average delay while taking into account vehicle mobility and energy consumption constraints. We elaborate on two typical application scenarios, namely, interference-free orthogonal multiple-access (OMA) networks and interference nonorthogonal multiple-access (NOMA) networks, and develop two separate heuristic algorithms to solve the delay minimization problem. Simulation results demonstrate the efficiency of the proposed algorithms.

Task Offloading in Vehicular MEC Environments

The explosive growth of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies gives birth to various computation-intensive and delay-sensitive applications, such as augmented-reality and virtual-reality automotive applications, and enables new functionalities, such as self-driving, with the final aim of enriching our life experience. However, those promising applications usually require high computation capacity for data processing, which cannot be provided by local vehicular terminals. Cloud computing allows vehicular users to access a shared pool of configurable computing resources on an on-demand basis, which can significantly augment their computation capacities. It has been predicted that by 2021 approximately 94% of total computation instances will be executed in the cloud and the annual data traffic originated or terminated at data centers will reach 20.6 ZB [1]. The transmission of such a massive volume of data to the cloud will not only pose heavy burdens on backhaul and backbone networks but will also result in unpredictable transmission delays to vehicular users.

Digital Object Identifier 10.1109/MVT.2019.2902637
Date of publication: 26 June 2019

To overcome the limitations of cloud computing, a promising paradigm, MEC, has been proposed to address the latency issue by switching from a conventional centralized architecture to a distributed one. As shown in Figure 1, various existing facilities, such as MEC servers deployed at base stations or RSUs, and vehicles with excess computation capacity can serve as distributed edges. Offloading computation tasks to such edges, which are physically closer to the data sources, can significantly reduce transmission latency and, hence, improve the quality of service (QoS) perceived by end users.

Despite the potential benefits for improving end users' computing experience, task offloading in vehicular MEC environments is highly challenging. The challenge results from the mobility of vehicles, the inherent environmental dynamics associated with the wireless communication medium, the heterogeneous QoS requirements of vehicular users, and the heterogeneous computational capabilities of edges. Furthermore, the reality that both parties (for example, tasks and edges) usually do not have complete information about each other makes the problem even more complex [2].

Many recent research efforts have been dedicated to optimally scheduling computation tasks. In [3], the authors jointly optimized task scheduling and radio-resource allocation to achieve minimum energy cosumption under delay constraints for MEC in 5G heterogeneous networks. In [4], the authors jointly optimized task offloading scheduling and transmit-power allocation to minimize the weighted sum of execution delay and energy consumption by using alternating minimization techniques. The tradeoff between the delay and power consumption is further investigated in [5] and [6]. The authors of [7] and [8] extended task offloading algorithms for vehicular MEC architectures by taking into account the heterogeneous requirements of the computation tasks and vehicle mobility. In [9], the authors proposed a new MEC framework where a centralized software-defined network controller continues collecting information about moving vehicles and determines the path for each V2V task-offloading process. Leveraging the vehicular MEC architectures spurs an array of new services and applications with advanced features, such as real-time heart attack detection services [10] and software-defined content delivery services among connected vehicles [11].

Our work is different from these prior approaches, because we concentrate on task offloading in vehicular MEC architectures with incomplete information (for example, when wireless-channel states and vehicle mobility patterns

MANY RECENT RESEARCH EFFORTS HAVE BEEN DEDICATED TO OPTIMALLY SCHEDULING COMPUTATION TASKS.

are not completely available), where tasks and edges must make decisions based on locally collected information. We formulate the interaction between tasks and edges as a matching game. The main contributions of this article are summarized as follows:

- First, we study the task assignment problem in vehicular MEC architectures. The proposed task offloading mechanism is context aware and optimality driven and can be performed in a distributed manner.
- Second, we elaborate on two typical application scenarios, interference-free OMA networks and interference NOMA networks, and develop two heuristic matching algorithms to solve the delay minimization problem. The proposed algorithms result in stable solutions; that is, neither the vehicles nor the edges have any incentives to deviate from the matching results.
- Finally, simulation results confirm that, by dispatching tasks to edges carefully, the proposed algorithms can significantly reduce the average delay while satisfying energy consumption constrains.

Matching Theory: Fundamentals and Classification

Matching theory is a useful tool for studying the formation of the mutually beneficial relationship between two sets of agents. Each agent has its own preference and ranks agents in the opposite set using a utility function that depends on some measurable parameters. For example, in the college admission problem, students rank colleges according to their academic reputations, while colleges rank students according to their entrance examination scores and so forth. Matching decisions are interactively made by the agents themselves based on locally collected information. Therefore, matching theory-based protocols generally does not require any centralized coordinator and can support good scalability.

Matchings can be classified into three categories according to the quota/capacity of each agent.

- *One-to-one matching*: An agent in both sets can be matched to, at most, one agent in the opposite set. One classical example is the stable-marriage problem [12], which aims to optimally pair up men and women.
- *One-to-many matching*: An agent in one set can be matched to multiple agents in the opposite set, while an agent in the opposite set can be matched to, at most, one agent. Examples include the college admission problem, assigning teachers to high schools, and so on.
- *Many-to-many matching*: An agent in both sets can be matched to, at most, the quota/capacity of agents in the opposite set. Many-to-many matching markets exist in various forms, such as matching advertisers to browsers.

In matching theory, *externalities* refers to the interdependencies between agent preferences. According to the existence of externalities, matching games can be further classified into two categories.

- *Canonical matching*: The preference of each agent depends only on the agent(s) to which it is about to match. Both the stable-marriage problem and the college admission problem fall into this category.
- *Matching with externalities*: The preference of each agent is affected by the dynamic formation of other associations. Examples include dormitory assignments at universities, where the desirability of a specific room may vary among peers in the same or nearby rooms.

Readers can refer to [12] for more details of matching theory.

Task Assignment in MEC: Objectives and Constraints

Aiming to achieve a practical and distributed solution, we realize that the task assignment problem in vehicular MEC architectures can be formulated as a one-to-many matching game. In particular, computation tasks and edges are the two disjointed sets of agents to be matched. For the sake of readability, MEC servers and vehicles with excess computation capacities that offer offloading opportunities to other, resource-limited vehicle nodes (VNs) are collectively denoted as *edge nodes* (*ENs*). ENs can divide their computing resources into several virtual resource units (VRUs), such that tasks can be executed in parallel on those ENs. The number of VRUs at an EN is

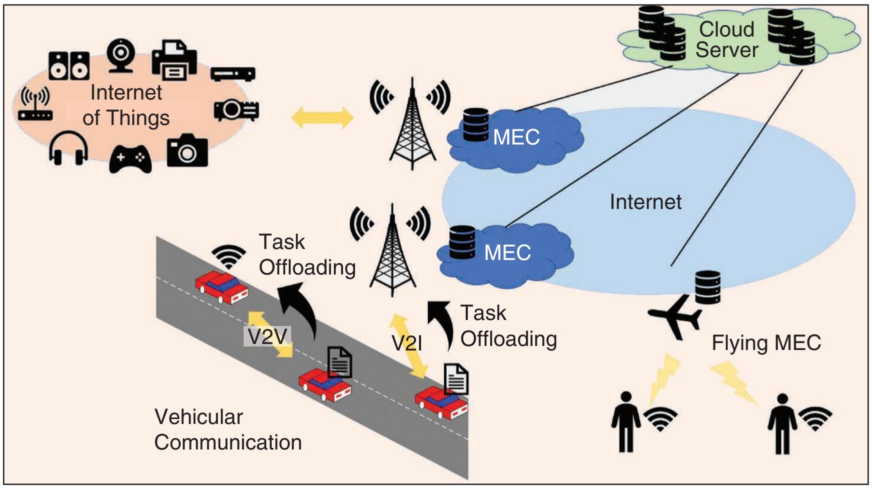

FIGURE 1 Vehicular MEC architectures.

referred to as the *quota* of the EN. The parameters used to describe each computation task are shown in Table 1. In the remainder of this section, we discuss the delay, energy consumption, and mobility constraints that need to be considered in task offloading.

Delay

The overall latency for executing a task of VN i on EN j generally consists of three components: 1) transmission delay, 2) queuing delay, and 3) computation delay. First, the *transmission delay* refers to the time it takes to transmit the input data from VN i to EN j and the outcome of computing from EN j to VN i via uplink and downlink wireless communications, respectively. Transmission delay depends significantly on the wireless-channel states and the achievable data rate between the EN and VN. Second, *queuing delay* is the time that a task waits in a queue until it can be executed. Finally, *computation delay* refers to the length of time that it takes for task execution.

Because task offloading incurs extra transmission latency, the key question in such an environment is whether to offload the task to ENs or execute it locally on terminals. Intuitively, for computation-intensive tasks, computation delay is responsible for a large portion of the overall delay. Hence, computation-intensive tasks tend to be offloaded to an EN with high computation capability to meet their latency constraints, while light tasks with a large quantity of input data may be executed locally to save the transmission-delay cost incurred by task offloading.

Energy Consumption

Similarly, the total energy consumption for executing a task of VN i on EN j has two parts: 1) transmission energy consumption and 2) computation energy consumption. On the one hand, the transmission energy consumption of VN i is directly proportional to the transmission power of VN i and the size of the input data, while it is inversely proportional to the achievable transmission data rate. On the other hand, the computation energy consumption of EN j varies with the number of CPU cycles required and the CPU frequency of EN j.

Mobility Constraints

Link breakage is another problem facing task offloading in vehicular MEC architectures. The *link expiration time* (*LET*) is defined as the duration of the connectivity between the VN and EN. To avoid link breakage during input data transmission, the LET and the time needed for input data transmission must be well predicted. In [13], the authors propose a movement-prediction-based algorithm to anticipate whether a link is likely to expire during data transmission. Considering vehicle mobility, if the accomplishment of the computation task takes a long time, ENs may fail to transmit the computing outcome back to the VNs. In this case, the outcome should be transmitted from the EN that has accomplished the task to an EN that covers the VN destination.

Optimization Problem

The fundamental problem under consideration is then cast as a network-wide average delay minimization problem for task assignment, which is subject to the following set of constraints:

1) Each task is assigned to, at most, one EN.
2) Each EN accepts, at most, its quota of tasks.
3) The energy consumption of each VN and EN should not be higher than its available energy.
4) Each task should be accomplished within its designated delay tolerance, and the input data should be transmitted within the LET.

The optimization problem considered here is in the standard form of a binary linear programming problem with a bundle of constraints, which is proved to be NP hard [14]. In the next section, we investigate two types of wireless networks over which data are transmitted between VNs and ENs: interference-free OMA networks and interference NOMA networks. Then, we propose two separate heuristic matching algorithms for solving the average delay minimization problem. The proposed algorithms achieve a good compromise between computation complexity and delay optimization.

Matching-Based Task Offloading

Let $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$ denote the set of tasks and the set of ENs to be matched together. For simplicity, the task of VN i is represented by $\tau_i (\in \boldsymbol{\tau})$, and EN j is represented by $\epsilon_j (\in \boldsymbol{\epsilon})$. Before matching execution, each agent ranks the agents in the other set in order of preference. Aiming to optimize the average delay, the preference of each task is configured based on the utility function that captures the computation delay, while the preference of each EN is set up based on the utility function that captures the transmission delay.

***TABLE 1** The parameters used to describe a computation task and an EN.*

Computation Task	*EN*
1) Size of input data for computation	1) Number of VRUs
2) Number of required CPU cycles	2) CPU frequency of each VRU
3) Delay tolerance, which is a measure of task-owner time sensitivity and defined as the time from when a computation request is made until the task should be completed	3) Energy-consumption level of each VRU

OMA-Based Networks: Canonical Matching

OMA techniques have been widely used in 4G mobile communication systems. As shown in Figure 2(a), we assume that spectrum resources along the time or frequency dimensions are divided into orthogonal resource blocks (RBs) and that each RB is solely occupied by one task-EN pair for input data transmission. With interference-free OMA-based networks, the achievable data rate, transmission delay, and corresponding preference of each task are independent of other tasks' choices. In other words, each task does not care with whom the other tasks will match. This is called *canonical matching*.

We introduce a one-to-many matching algorithm to solve the delay minimization problem. The proposed matching algorithm is briefly introduced as follows. Each task makes a proposal to its most preferred EN. Once this is done, ENs that have received proposals reject all but their quota of tasks, which they most prefer. Every unmatched task then makes a proposal to its most preferred EN that has not yet rejected it. We continue the previously mentioned iterations until all tasks are dispatched or the constraints shown above cannot be satisfied by any ENs.

Theorem 1

When the proposed algorithm terminates, we have a stable matching between the tasks and ENs.

Proof

The proof can be found in [15] and is omitted here due to space limitations.

NOMA-Based Networks: Matching With Externalities

NOMA is regarded as a radio-access candidate for 5G wireless networks. Unlike OMA, NOMA allows multiple users to share the same RB simultaneously. As a consequence, NOMA can achieve higher spectrum efficiency than OMA while creating interuser interference over the same RB. As shown in Figure 2(b), when more VNs in proximity simultaneously use the same RB for input data transmission, the interuser interference level increases, and the achievable data rate decreases. In such NOMA-based networks, the transmission delay and preference order of each task are strongly affected by the dynamic formation of other task–EN associations. If such externalities are not well managed, tasks must continue changing their preference order in response to changes in other task–EN associations, and a stable result could never be expected. To address the effect of externalities, we propose a swap-matching-based algorithm to obtain a practical and stable solution.

- *Definition 1*: Swap matching. Given a matching $\boldsymbol{\eta}$ and two task–EN pairs $(\tau_i, \epsilon_m), (\tau_j, \epsilon_n) \in \boldsymbol{\eta}$, a swap matching is defined as $\boldsymbol{\eta}_{ij}^{mn} = \{\boldsymbol{\eta} \backslash (\tau_i, \epsilon_m), (\tau_j, \epsilon_n)\} \cup \{(\tau_j, \epsilon_m), (\tau_i, \epsilon_n)\}$.
- *Definition 2*: Two-side exchange stability. A matching $\boldsymbol{\eta}$ is considered two-side exchange stable if there is no agent that has an incentive to swap from its current association.

Given the definitions, a matching $\boldsymbol{\eta}$ with a pair $(\tau_i, \epsilon_m) \in \boldsymbol{\eta}$ is considered to be two-side-exchange stable if no pair $(\tau_j, \epsilon_n) \in \boldsymbol{\eta}$ exists, for which task i prefers EN n over EN m, or any EN m that prefers task j over i. Such a two-side-exchange stability is achieved by guaranteeing that swaps occur if and only if the swaps are beneficial for all of the agents involved. The proposed swap-matching algorithm consists of an initialization stage, a tentative matching stage, and an iterative swap-matching stage.

Initialization Stage

In the initialization stage, the following applies.

- Tasks broadcast their information to the ENs in their proximity, indicating the input data size, the required CPU cycles, and the tolerable delay.
- ENs that have received the request will update their preference lists by estimating the achievable data rate and corresponding transmission latency, based on the received interference signal power, and then reply with their computation capabilities.
- Tasks set up their preference list (with the EN that achieves the minimum computation delay on the top).

Tentative Matching Stage

For the tentative matching stage, the following applies.

- Each task makes a proposal to its most preferred EN.

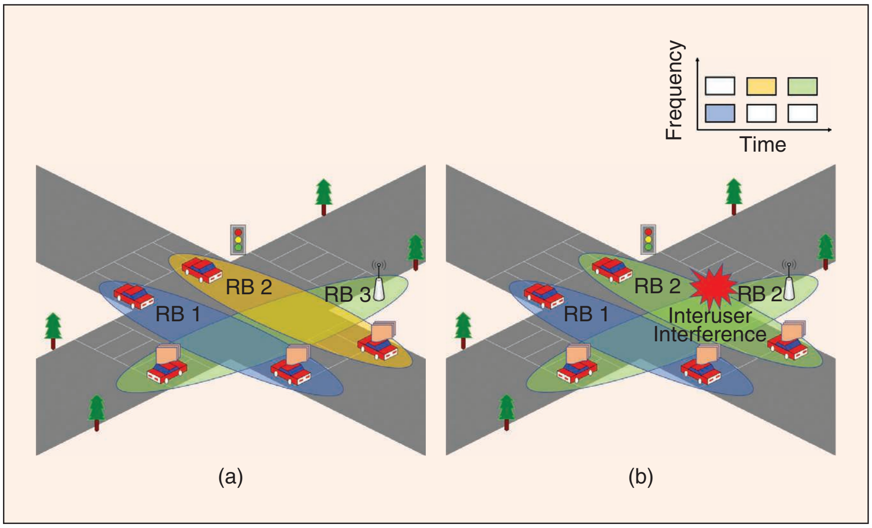

FIGURE 2 Task assignment: (a) canonical matching in OMA-based networks and (b) matching with externalities in NOMA-based networks.

- ENs that have received proposals will tentatively accept their quota of tasks on a first come, first served basis.

Swap-Matching Stage

The following is applicable to the swap-matching stage.

- For each EN, all matched tasks that use the same RB for input data transmission recalculate the transmission delay by considering the interuser interference and update their preference list accordingly.
- If the most preferred EN in the updated preference list is not the current partner and the swap is beneficial to all of the agents involved, the swap operation will be approved. Otherwise, the swap operation will be rejected.

The swap proceeds iteratively and terminates when no task has an incentive to swap from its current association.

Case Study

Simulations are conducted to evaluate the performance of the proposed algorithms. We simulate a 10-km road with two lanes: fast and slow. The width of each lane is 3 m. One hundred MEC servers are evenly deployed along the side of the fast lane. Each MEC server owns two VRUs, each of which has a 4-GHz CPU. At the same time, 100 vehicles are evenly placed in the two lanes. The average speed of vehicles in the fast lane is 50 km/h, and that of vehicles in the slow lane is 40 km/h. The CPU frequency of each vehicle is randomly selected within the range of 0.5–1 GHz. For simplicity, we assume that each vehicle generates zero or one task. The vehicles generating no tasks can act as edges to offer offloading opportunities to other vehicles. Furthermore, the input data size and number of required CPU cycles for each task are randomly selected within the range of 0–1 MB and $(0, 1 \times 10^9)$, respectively. The delay tolerance of each task is set to 2 s. We set the transmission power to 46 dBm, noise power spectral density to –110 dBm/Hz, and system bandwidth to 2 MHz. The total bandwidth is divided into six orthogonal RBs. The channel power gain at each receiver is set to $-40\, d^4$ db, where d is the distance between the transceiver and the receiver. The length of each slot is 10 s.

The efficiency of canonical matching algorithms has been well investigated in [15]. Thus, we focus on the task offloading over NOMA-based networks and evaluate the performance of the proposed swap-matching algorithm in terms of average latency and power consumption. We compare the proposed algorithm with two heuristic algorithms: the random-matching algorithm and the exhaustive-searching algorithm.

- *Random matching*: Tasks and ENs are randomly paired as long as all conditions shown earlier are satisfied.
- *Exhaustive searching*: This concerns the algorithm that examines all possible combinations to find the global optimal solution. The result serves as an upper performance benchmark.

TO ADDRESS THE EFFECT OF EXTERNALITIES, WE PROPOSE A SWAP-MATCHING-BASED ALGORITHM TO OBTAIN A PRACTICAL AND STABLE SOLUTION.

Figures 3 and 4 show the average delay and power consumption versus task generation probability. Here, *task generation probability* means the probability that a vehicle will generate a task in each slot. As we can observe, the average delay increases with the task generation probability, and the growth becomes slower as the task generation probability increases. Furthermore, the proposed swap-matching algorithm shows comparable performance compared to the exhaustive-searching algorithm while outperforming the random-matching algorithm in terms of average delay and power consumption. These results are not surprising because, in the proposed method, swap-matching algorithm tasks and ENs rank the agents in the opposite set based on the utility function that captures the computation and transmission delays. Meanwhile, the lower transmission and computation delays imply lower transmission and computation

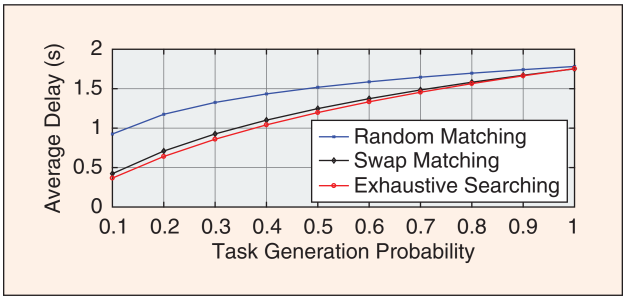

FIGURE 3 The average delay versus task generation probability.

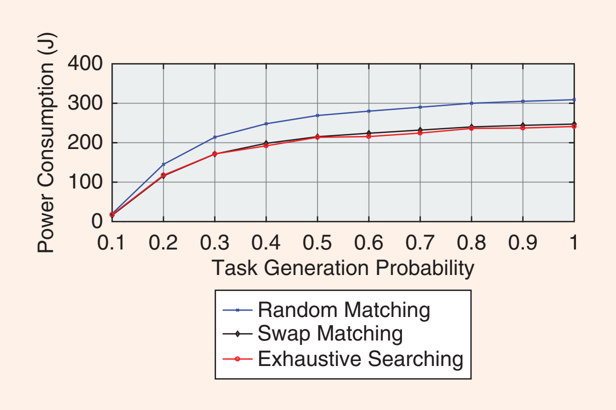

FIGURE 4 Power consumption versus task generation probability.

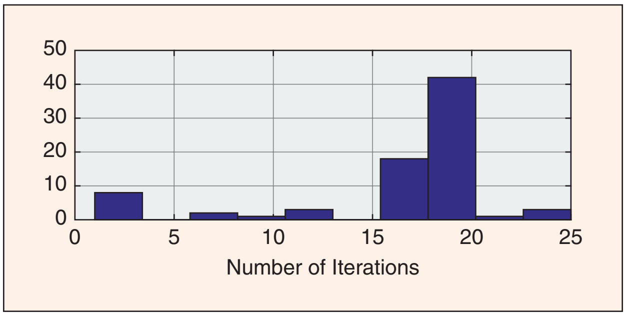

FIGURE 5 A histogram of the convergence speed.

energy consumption. Thus, the matching-based algorithm by its nature guarantees that tasks will be intelligently offloaded to ENs that generate low latency and energy consumption.

Then, we evaluate the convergence speed of the proposed swap-matching algorithm. Figure 5 shows the histograms of the number of iterations that the proposed algorithm requires to converge to stable matching. We can observe that the number of iterations is usually approximately 20 and never more than 25. For a problem with 100 vehicles and 100 MEC servers, the speed of convergence is quite acceptable. Thus, we also believe that the proposed algorithm is appropriate for large-scale vehicular MEC platforms.

Conclusions

In this article, we focused on using matching theory for developing a distributed and context-aware task assignment mechanism for vehicular MEC. First, we provided a comprehensive review of the latest MEC research results. Second, we introduced the fundamentals of matching theory, which is regarded as a powerful tool to study the formation of a mutually beneficial relationship between two sets of agents. Third, we formulated the task assignment problem in vehicular MEC architectures as a matching game, with the objective of minimizing the average latency. Then, we investigated two types of wireless networks over which data were transmitted between VNs and ENs. After demonstrating that solving the delay minimization problem is NP hard, we proposed two heuristic algorithms. Finally, simulations were carried out to confirm the efficiency of the proposed algorithms.

Acknowledgments

This work was partially supported by the National Science Foundation of China under grant 61601181, the Fundamental Research Funds for the Central Universities under grant 2017MS001, the Beijing Natural Science Foundation under grant 4174104, and Beijing Outstanding Young Talent under grant 2016000020124G081.

Author Information

Bo Gu (gubo@mail.sysu.edu.cn) is an associate professor in the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include the Internet of Things, edge computing, network economics, and machine learning. He is a Member of the IEEE.

Zhenyu Zhou (zhenyu.zhou@ncepu.edu.cn) is a professor in the School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China. His research interests include green communications, vehicular communications, and smart grid communications. He is a Senior Member of the IEEE, Chinese Institute of Engineers, and China Institute of Communications. Zhou is the corresponding author.

References

[1] Cisco, "Cisco global cloud index: Forecast and methodology, 2016–2021 white paper," Cisco Systems, San Jose, CA, 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html

[2] R. Shrestha, R. Bajracharya, and S. Y. Nam, "Challenges of future VANET and cloud-based approaches," *Wireless Commun. Mobile Comput.*, 2018. doi: 10.1155/2018/5603518.

[3] K. Zhang et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[4] Y. Mao, J. Zhang, and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. 2017 IEEE Wireless Communications Networking Conf. (WCNC)*, San Francisco, CA, 2017, pp. 1–6.

[5] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.

[6] B. Gu, Y. Chen, H. Liao, Z. Zhou, and D. Zhang, "A distributed and context-aware task assignment mechanism for collaborative mobile edge computing," *Sensors*, vol. 18, no. 8, 2018. doi: 10.3390/s18082423.

[7] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, July 2018.

[8] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 48–54, Aug. 2018.

[9] C. Huang, M. Chiang, D. Dao, W. Su, S. Xu, and H. Zhou, "V2v data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture," *IEEE Access*, vol. 6, pp. 17741–17755, 2018.

[10] S. Ali and M. Ghazal, "Real-time heart attack mobile detection service (RHAMDS): An IoT use case for software defined networks," in *Proc. 2017 IEEE 30th Canadian Conf. Electrical and Computer Engineering (CCECE)*, Windsor, Canada, 2017, pp. 1–6.

[11] J. Al-Badarneh, Y. Jararweh, M. Al-Ayyoub, R. Fontes, M. Al-Smadi, and C. Rothenberg, "Cooperative mobile edge computing system for VANET-based software-defined content delivery," *Comput. Electr. Eng.*, vol. 71, no. 1, pp. 388–397, Oct. 2018.

[12] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *Amer. Math. Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[13] H. Menouar, M. Lenardi, and F. Filali, "Movement prediction-based routing (MOPR) concept for position-based routing in vehicular networks," in *Proc. 2007 IEEE 66th Vehicular Technology Conf.*, Baltimore, MD, Sept. 2007, pp. 2101–2105.

[14] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, Eds. Boston, MA: Springer, 1972, pp. 85–103.

[15] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: Fundamentals and applications," *IEEE Commun. Mag.*, vol. 53, no. 5, pp. 52–59, May 2015.

VT