IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 31, 2019, accepted February 16, 2019, date of publication February 21, 2019, date of current version March 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900530

Efficient Mobility-Aware Task Offloading for Vehicular Edge Computing Networks

CHAO YANG, YI LIU, XIN CHEN, WEIFENG ZHONG, AND SHENGLI XIE, (Fellow, IEEE)
Guangdong Key Laboratory of IoT Information Technology, School of Automation, Guangdong University of Technology, Guangzhou 510006, China

Corresponding author: Xin Chen (xinchen@gdut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61603099, Grant 61773126, Grant 61727810, and Grant 61701125, in part by the Natural Science Foundation of Guangdong under Grant 2018A0303130080, and in part by the Pearl River S&T Nova Program of Guangzhou under Grant 201806010176.

ABSTRACT Vehicular networks are facing the challenges to support ubiquitous connections and high quality of service for numerous vehicles. To address these issues, mobile edge computing (MEC) is explored as a promising technology in vehicular networks by employing computing resources at the edge of vehicular wireless access networks. In this paper, we study the efficient task offloading schemes in vehicular edge computing networks. The vehicles perform the offloading time selection, communication, and computing resource allocations optimally, the mobility of vehicles and the maximum latency of tasks are considered. To minimize the system costs, including the costs of the required communication and computing resources, we first analyze the offloading schemes in the independent MEC servers scenario. The offloading tasks are processed by the MEC servers deployed at the access point (AP) independently. A mobility-aware task offloading scheme is proposed. Then, in the cooperative MEC servers scenario, the MEC servers can further offload the collected overloading tasks to the adjacent servers at the next AP on the vehicles' moving direction. A location-based offloading scheme is proposed. In both scenarios, the tradeoffs between the task completed latency and the required communication and computation resources are mainly considered. Numerical results show that our proposed schemes can reduce the system costs efficiently, while the latency constraints are satisfied.

INDEX TERMS Vehicular network, edge computing, resource allocation, offloading, mobility.

I. INTRODUCTION

With the rapid development of internet of things (IoT) technologies, the vehicular networks have become an indispensable part of the intelligent transportation systems. Including the normal applications (e.g., advertisements, path planning and navigation), the vehicular networks support numerous complex applications for both the vehicles and passengers, such as: automatic driving, intelligent auxiliary driving for vehicles and augmented reality (AR), online interactive gaming and other rich media applications for passengers [1]–[3]. These applications require intensive communication and computation resources. It is a big challenge to ensure these high complexity services, especially in the vehicular networks. Along with the rapid increasing of the traffic density on the road, the gap between the communication/calculation service requirements and the limited capacities of vehicles becomes a serious problem.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhui Yuan.

To address this issue, cloud-based vehicular networks had been envisioned as a potential solution [4]–[6]. The computation tasks are offloaded to the remote cloud servers. However, the long distance between the vehicles and remote central servers and the fluctuant wireless channels lead to considerable latency, which causes the task offloading efficiency is affected seriously. To cope with the time-sensitive and complex task requirements, mobile edge computing (MEC) and other 5G network technologies (e.g., dynamic spectrum access (DSA)) become the promising solutions, to provide both the available communication and computation resources [7]–[10]. With the DSA, the emerging vehicular networks can lease the available spectrum from the existing cellular networks via the base stations (BS) or access points (AP) around the road [11], [12]. The MEC servers with computing and storage resources are deployed at the edge of vehicular networks (i.e, at the roadside AP) [3], [13], the system model of vehicular edge computing networks (VECN) is shown as Fig. 1. The vehicles on road can access the MEC servers by the vehicular-to-infrastructure (V2I)

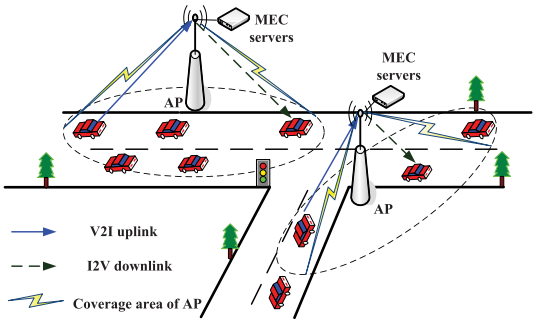


FIGURE 1. System model of vehicular edge computing networks.

communication links. Although the vehicles have a certain local computing resources, they cannot execute the cumbersome and heterogeneous computation tasks by itself. Moreover, not only the computation requests of vehicles, but also the requests from passengers need to be satisfied [1], [3]. The task completed maximum latency thresholds are different under different applications. For example, the maximum latency thresholds of video caching/sharing in AR application (e.g., multi-seconds) for passengers is larger than the intelligent vehicle driving applications (i.e., milliseconds) for vehicles. VECN can help the vehicles to address these challenges.

As shown in Fig. 1, the computation task offloading process in VECN includes three parts: 1) *Offloading*: Same as the data offloading scheme in vehicular Ad Hoc networks [14], the vehicles send the computation tasks to the MEC servers via 5G DSA strategy. Note that the various wireless channel status, the available spectrum resources and the data sizes of tasks affect the offloading latency. 2) *Calculation*: After receiving the offloading tasks, the MEC servers at AP execute the tasks via the computation resource allocation strategies. Meanwhile, when huge amounts of tasks arrive, the MEC servers can further send part of tasks to other MEC servers at neighboring APs. 3) *Computation results feedback*: The MEC servers send the computation results back to the vehicles. Although the task offloading schemes in both MEC networks and VECN had been studied, the computation, communication and the storage resources are analyzed and optimized [8], [15]–[18]. It is still the open problems about how to design an efficient task offloading scheme and how to manage the resources in VECN to maximize the system performances. Different from the previous works of the task offloading and resource allocation in VECN [8], [17], [18], we consider the mobility of vehicles mainly. When the vehicles move closer to the AP, the transmission distance reduces, the channel data transmission rate of V2I communication will be increased. Thus, the vehicles should decide whether/when/how to send the computation tasks, and how much communication and computation resources are needed, based on the initial locations, moving speeds of vehicles and the heterogeneous latency thresholds of tasks.

In this paper, we focus on the design of task offloading schemes in VECN, the tradeoffs between the system costs and the task completed latency are mainly considered. In detail, the system costs include the computation and communication resource costs. In the former, along with the development of microgrids and wireless charging technologies, the roadside APs can be powered by the renewable energy for the convenience consideration [19]. Energy consumption becomes an important concern for the roadside APs. It is reasonable that the MEC servers in APs gain revenues from providing computation resources for the offloading tasks [17]. In the latter, when the traffic density of vehicles increases, the existing spectrum resources may become scarcer. The vehicles can obtain the available spectrum resources via spectrum leasing with the cellular network, to support the task offloading.

In our work, we analyze the efficient task offloading schemes in two scenarios: independent and cooperative MEC servers. In the first scenario, a vehicle passes through the coverage area of a AP with high speed, the computation tasks are processed by the MEC servers deployed at the AP independently. The channel transmission data rate increases when the vehicle is moving close to the AP. The vehicle can send more tasks to AP with low communication cost, via selecting an optimal task offloading time. We propose a mobility-aware task offloading scheme. In the optimization problem of our proposed scheme, the offloading decisions, offloading time and the computation resource allocation in MEC servers are jointly optimized. In the second scenario, we consider a common case, in which many vehicles send the computation tasks to MEC servers. When more tasks with heterogeneous requirements are offloaded, the MEC servers should decide whether perform calculation locally, or further send part/all of the received tasks to other MEC servers deployed at the next AP on the vehicles' moving direction. We propose a location-based task offloading scheme. The initial location and moving speed are two critical factors to lead the vehicles select different offloading schemes: local computing, offloading to the MEC servers or further offloading to the next AP. In fact, the vehicles are divided into different groups, and the resource allocation of vehicles in different groups are formulated as convex optimization problems, respectively. In summary, the main contributions of this paper are as follows:

- We propose the framework of task offloading schemes in VECN, the impacts for the task offloading caused by the moving of vehicles are considered mainly.
- We propose a mobility-aware task offloading scheme in the independent MEC servers scenario, the task offloading decisions, offloading time and the computation resource allocation are jointly optimized.
- We propose a location-based task offloading scheme in the cooperative MEC servers scenario, both the task offloading decisions, offloading time and the cooperation between MEC servers deployed at two adjacent APs are considered.

The rest of this paper is organized as follows. We review the related work in Section II. In Section III and Section V, we describe the system model and formulate the problem

in the independent MEC servers scenario. In Section VI, we analyze the system performances in the cooperative MEC servers scenario. Simulations are conducted in Section VII. Finally, we draw conclusions of our work in Section VIII.

II. RELATED WORK

The MEC servers can bring flexible computation and storage resources to the edge of wireless access network, to cope with the delay-sensitive, computational intensive and high reliability applications. MEC has attracted increasing attention from both academic and industrial areas [7]. The existing studies always focus on multi-resources management schemes, the optimal design of task offloading polices in both 5G and IoT networks [9], [10], to minimize the task processing energy consumption or latency [7], [16], [20]–[31]. In [23], based on the time-division multiple access (TDMA) and orthogonal frequency-division multiple access (OFDMA) strategies, energy efficient resource allocation schemes are studied for a multi-user system. In [7]and [21], energy efficient task offloading schemes are analyzed, in which the tradeoffs between energy consumptions and task completed latencies are considered. In [24], an offloading framework with one device and multiple APs is proposed, the flexible CPU frequency of AP and the task offloading are optimized. The wireless channels' status affect the efficiency of computation task offloading. In [31], the multi-user computation task offloading problem is analyzed, while the interferences among channels are considered. Furthermore, in [20], the computation task offloading scheme and interference management are jointly optimized. According to the multi-access characteristics of file transmission in 5G networks, the energy-efficient offloading schemes are proposed to help the devices make optimal decisions [25], [26]. Multi-access task offloading schemes in 5G ultra-dense networks are analyzed in [9] and [10]. Although the MEC servers endow computational resources, the computing capabilities are limited due to the installation and operation maintenance cost. In order to prevent the quality of service (QoS) degradation when the traffic load is huge, the MEC servers can further relay the excessive workloads to the remote cloud [27], [32], or other around servers [28], [29]. The tradeoffs between the computation delay and communication delay are considered. However, all of the above researches are based on a condition that the devices are fixed or moving with low speeds. In VECN, the characteristics of vehicular networks cause that the system performances will be degraded when the existing task offloading schemes in 5G networks are utilized directly.

In the VECN, MEC becomes an efficient solution to support the automatic driving services, a two-level architecture including the vehicles and roadside BSs is proposed in [3]. To overcome the limited computational capacity of MEC server, a collaborative MEC framework for vehicular networks is analyzed in [33]. In [34], an autonomous vehicular edge framework is proposed, the vehicles on road are teamed up via the help of GPS information, an efficient job caching scheme is designed to support the task offloading. In [17] and [8], two cloud-based vehicular edge computing offloading frameworks are proposed. The contract theory and Stackelberg game theoretic approaches are used respectively, to maximize the benefits of both the MEC servers and vehicles. Task offloading scheme in VECN is the special case of the data offloading in vehicular networks [14]. In [18], two efficient collaborative task offloading schemes are proposed in the fiber-wireless (FiWi) enhanced VECN. In [35], a credit-based clustering scheme is proposed, the cluster head and the covered vehicles share the interested geo data. Similarly, a vehicular fog computing framework is proposed in [36], in which the near-located vehicle resources are utilized cooperatively, to improve the communication and computation capacities of the operation vehicles on road. However, none of above cited papers consider the fast moving characteristics of vehicles. In fact, the channel transmission data rate and efficient contact durations in V2I communication are affected by the data transmission distance. In this paper, our proposed task offloading schemes aim at the VECN system benefits of minimizing the task completed system costs, while the task completed latency, the mobility and locations of vehicles are mainly considered.

III. SYSTEM MODEL

In this section, we present the framework of VECN firstly, specially, the spectrum leasing scheme among the APs and the covered vehicles is introduced. Then, we will give the detail of the analytical model, including the communication process, computation model and system costs.

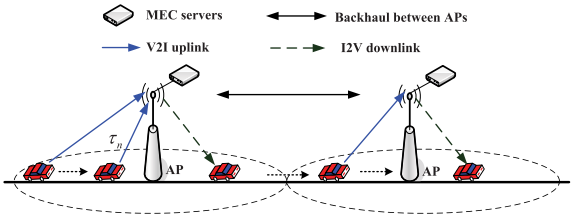


FIGURE 2. The vehicular edge computing network framework.

A. FRAMEWORK OF VECN

Fig. 2 shows the framework of VECN. With finite communication, computation and storage resources, the MEC servers are deployed at the roadside APs. The APs are connected with a wired communication line, and the channel transmission rates are fixed. With the powerful antenna and the adequate power supply from onboard energy, the vehicles can perform data transmission and computation simultaneously. When the vehicles enter the coverage area of a AP, upon the computation requests arrive, the vehicles should decide whether/when/how to offload computation tasks to the MEC servers. Heterogeneity of tasks and the tradeoffs between system costs (i.e., costs caused by the leased spectrum and computation resources) and the task completed latencies (i.e., data offloading and processing latencies) are taken into account by the proposed optimal task offloading decisions. The locations of vehicles affect the channel transmission data rate between the vehicles and AP. When the vehicles move

closer to the AP with high speed, the data rate becomes larger as the data transmission distance decreases. The task offloading time should be optimized. When overwhelming workloads arrive, the MEC servers can further send part of them to other servers deployed at the adjacent AP on the vehicles' moving direction. Moreover, when the APs send back the task calculation results to vehicles, the distances between the APs and vehicles should be estimated.

The V2I communication network performs spectrum leasing with the cellular network. We set that the roadside APs perform as the network operator, to lease both the available spectrum resource blocks in cellular network and computation resources in MEC servers to vehicles, in exchange for remuneration [17]. Through the leased spectrum resource blocks, the vehicles can send excessive computation tasks to the nearest MEC servers. In detail, the roadside APs broadcast the available spectrum resource blocks periodically. Then, the vehicles determine the optimal task offloading decisions cooperatively, based on the data size, required CPU cycles and other constraints of tasks in each vehicle. The vehicles should decide how wide a spectrum band is needed and for how long. When multiple vehicles offload the tasks simultaneously, the vehicles perform data transmission in different spectrum bands via OFDMA technology, same as the IEEE standard 802.11p/D3.0 for vehicular networks [37], [38], no mutual interference among different spectrum bands exists.

B. ANALYTICAL MODEL

We consider a VECN system with N vehicles and two adjacent APs, the coverage areas of each AP are the same. The VECN is a saturation network, each vehicle has multiple tasks to be executed all the time. The computation task i of vehicle n is described as a tuple $\{s_{n,i}, c_{n,i}\}, n \in \{1, 2, \cdots, N\}$, $i \in \{1, 2, \cdots, I\}$, where $s_{n,i}$ denotes the data size, $c_{n,i}$ denotes the required CPU cycles to process the task. Due to the distinguished applications and the predefined priorities, a total task completed maximum latency threshold T_n^{max} exists for vehicle n. Assume the initial location of vehicle is l_n^0, when the computation request arrives. If the task i of vehicle n is executed locally, the completion time $T_{loc}^{n,i}$ is given as

$$T_{loc}^{n,i} = c_{n,i}/c_L, \tag{1}$$

where c_L denotes the local computing ability of vehicle n, we set that the local computing abilities of vehicles are the same.

In fact, in order to obtain better system performances, the vehicle n may offload part of computation tasks to the MEC servers via the V2I communication. For simplicity, the MEC servers at AP have non-overlapping steps, that they should begin to calculate the computation tasks sequentially after collecting. Hence, the vehicles should select an optimal offloading time τ_n, when they are moving close to the AP. As the distance between the vehicle and AP becomes shorter, the vehicle can send more tasks to the MEC servers with lower communication cost and transmission latency. At the offloading time τ_n, the channel transmission rate is expressed as

$$R_n = B_n \log_2 \left\{1 + \frac{P_{tx} h_{ng} (d_{ng})^{-r}}{\bar{w}_0}\right\}, \tag{2}$$

where B_n is the bandwidth of the leased transmission channel between vehicle n and AP, P_{tx} is the transmission power[1], h_{ng} denotes the channel gain, $\bar{w}_0$ is the power level of white noise and d_{ng} denotes the distance between the vehicle n and AP, which is associated with the initial location of vehicle l_n^0, the moving speed v_n and the offloading time τ_n, is shown as

$$d_{ng} = \begin{cases} \sqrt{l_g^2 + \left(D_g/2 - l_n^0 - v_n\tau_n\right)^2}, & if\ l_n^0 \le D_g/2, \\ \sqrt{l_g^2 + \left(l_n^0 - D_g/2 + v_n\tau_n\right)^2}, & if\ l_n^0 > D_g/2, \end{cases}$$

where D_g is the coverage area and l_g is the height of the AP. When vehicles are moving close to the AP, $l_n^0 \le D_g/2$, the distance between the vehicles and AP reduces with τ_n, otherwise, it increases with τ_n.

Since task offloading may produce the potential benefits, the vehicles, who had already leased the available spectrum blocks, can offload part of tasks to the MEC servers. Denote the price of unit leased spectrum resource block is ϕ_1 per $Hzsec$. The communication cost of vehicle n is associated with the bandwidth of the leased spectrum B_n and the occupied time duration t_u, denoted as F_n^1, is formulated as

$$F_n^1 = \phi_1 B_n t_u,$$

where t_u also denotes the uplink transmission time duration, the data size of the offloading task affects it directly. Let $a_{n,i}$ denotes the offloading decision. $a_{n,i} = 1$ denotes that the task i of vehicle n is offloaded to the MEC servers deployed at the AP, $a_{n,i} = 0$ denotes that the task is processed locally. Then, we have

$$t_u = \sum_{i=1}^{I} a_{n,i} s_{n,i}/R_n.$$

Denote the price of the unit leased computation resource as ϕ_2 per $cycle/sec$. When more computation resources are tenant, the computation latency becomes smaller, but the computation cost increases. The computation cost of vehicle n, F_n^2 is formulated as

$$F_n^2 = \phi_2 \sum_{i=1}^{I} \beta_{n,i} c_{th},$$

where c_{th} denotes the computing capacity of the MEC servers, $\beta_{n,i}$ denotes the fraction of computation capacity for task i in vehicle n. The vehicle should determine the task offloading schemes optimally, while the tradeoff between the task completed latency and the system cost (i.e., communication and computation costs) is mainly executed.

IV. TASK OFFLOADING FOR INDEPENDENT MEC SERVERS

In this paper, we focus on the design of task offloading schemes in VECN. We aim to minimize the task processing system costs under the task completed maximum latency and

[1]We set that the transmission powers of vehicles are the same. The energy consumptions of data transmission in vehicles are not considered.

other special constraints in vehicular networks. We analyze this problem in a simplified scenario firstly, a vehicle passes through the coverage area of AP with fast moving speed, and the offloading tasks can be processed by the MEC servers deployed at AP independently. In this scenario, the vehicle should determine the leased spectrum and computation resources based on the data size, the computation request and the latency of tasks.

A. PROBLEM FORMULATION

When the vehicle n moves into the coverage area of AP, it should select an optimal offloading time τ_n to transmit part of the computation tasks to the MEC servers. Actually, the offloading time selection affects the system communication cost directly. Within a given offloading task, when the vehicles move closing to AP, the channel transmission data rate between the vehicles and AP increases, and the communication cost will be decreased. We propose the task processing for vehicle n in the independent MEC servers scenario, as Fig. 3.

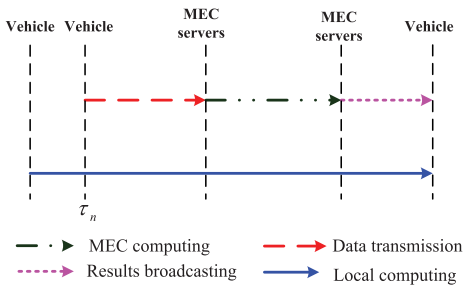


FIGURE 3. Task processing for vehicle *n* in independent MEC servers scenario.

As shown in Fig. 3, when the task computation requests arrive, the vehicle n schedules the tasks based on the predefined priorities firstly. The vehicle calculates part of them $\{(1-a_{n,i})s_{n,i}\}$ locally. Then, at the offloading time τ_n, other tasks $\{a_{n,i}s_{n,i}\}$ are offloaded to the MEC servers. After the data uplink transmission phase, the MEC servers perform task processing. Finally, the AP sends back computation results to the vehicle n via the I2V downlink channels. For the data size of computation results is smaller than the one of the offloading data, the AP can broadcast the results in its coverage area, the results feedback time is fixed as t_B. The task completed latency $t_{n,i}$ is shown as

$$t_{n,i} = (1-a_{n,i})\frac{c_{n,i}}{c_L} + a_{n,i}\left(\frac{s_{n,i}}{R_n} + \frac{c_{n,i}}{\beta_{n,i}c_{th}} + t_B\right). \quad (3)$$

The total task completed latency of all tasks in vehicle n, t_n is given by

$$t_n = \max\left\{\sum_{i=1}^{I}(1-a_{n,i})\frac{c_{n,i}}{c_L}, \tau_n + \sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n} + \frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B\right\}. \quad (4)$$

Then, the task processing system cost of vehicle n, F_n, includes the communication and computation costs, as

$$F_n = F_n^1 + F_n^2.$$

In the single vehicle case, the objective of the proposed optimization problem is to minimize the task completed system cost of vehicle n, while the task completed maximum latency is satisfied. The task offloading decision $a_{n,i}$, the bandwidth B_n of the leased spectrum resources, the fraction $\beta_{n,i}$ of the leased computation resources, and the offloading time τ_n are jointly optimized. The optimization problem is formulated as **P1**.

$$\min_{\{a_{n,i},B_n,\tau_n,\beta_{n,i}\}} F_n,$$
$$\text{subject to } t_n \le T_n^{max}, \quad (5)$$
$$a_{n,i} \in \{0,1\}, \quad 0 \le \beta_{n,i} \le \beta_{max}, \ \forall i, \quad (6)$$
$$B_n \le B_{max}, \quad (7)$$

where B_{max} is the maximum bandwidth of the leased spectrum resources, β_{max} is the maximum fraction of the available computation resources.

It is a challenge to solve the problem **P**1 which has integer constraint $a_{n,i} \in \{0,1\}$ and division mathematical operation of the variables. **P**1 is a mixed integer non-linear programming problem (MINLP), and it is usually NP-hard. To make the problem tractable, we analyze the objective function firstly.

$$F_n = \phi_1 B_n \sum_{i=1}^{I} a_{n,i}s_{n,i}/R_n + \phi_2 \sum_{i=1}^{I}\beta_{n,i}c_{th},$$
$$= \phi_1 \frac{\sum_{i=1}^{I} a_{n,i}s_{n,i}}{\log_2\left\{1 + \frac{P_{tx}h_{ng}(d_{ng})^{-r}}{\bar{w}_0}\right\}} + \phi_2 \sum_{i=1}^{I}\beta_{n,i}c_{th}, \quad (8)$$

we find that the value of B_n do not affect the objective function result. Then, based on the constraints Eqs. (5)(7), we set that $B_n = B_{max}$. In the independent MEC servers scenario, the vehicle can access the leased available spectrum with the maximum bandwidth.

Lemma 1: When the variables $a_{n,i}$ are relaxed to $0 \le a_{n,i} \le 1, \forall i$, the problem **P**1 is a convex problem.

Proof: See Appendix.

Note that the problem **P**1 is transformed as a convex problem with multiple variables, the existing typical convex optimal algorithms (e.g., dual decomposition algorithm [39]) can be used to solve it directly. However, the solving overhead is huge, especially when the number of tasks I becomes excessive. We can find other simple solving algorithms, via the formulations of objective function and the constraints in the problem.

B. MOBILITY-AWARE TASK OFFLOADING SCHEME

According to the formulations of task completed system cost F_n and latency t_n, we find that the initial location l_n^0,

the moving speed v_n of vehicle and the maximum latency threshold T_n^{max} affect the performance of the proposed optimization problem directly.

When the computation task requests arrive, if the vehicle n is moving close to the AP, $l_n^0 \leq D_g/2$, the vehicle can select an optimal data offloading time τ_n to reduce the communication cost. In detail, under the leased spectrum resource B_{max}, when the data size of offloading $\sum_{i=1}^{I} \alpha_{n,i} s_{n,i}$ is determined, the distance between the vehicle n and AP decreases, the data uplink transmission duration and the communication cost are reduced. For the vehicle can perform data transmission and calculation simultaneously, as shown in Fig. 3 and Eq. (4), it is an efficient way that the time duration of local computing in vehicle is the same as the one of task offloading in MEC servers. Thus, we have

$$\sum_{i=1}^{I}(1-a_{n,i})\frac{c_{n,i}}{c_L} = \tau_n + \sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n} + \frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B \leq T_n^{max}. \tag{9}$$

For the objective of problem **P**1 is to minimize the task completed system cost, which is associated with the data size of offloading tasks directly. Thus, the vehicle n should perform local computing as much as possible. Set a predefined threshold ε_n, following the descending order of the required CPU cycles of tasks $c_{n,i}$, we can find a set of tasks to make

$$\left(T_n^{max} - \sum_{i=1}^{I}(1-a_{n,i})\frac{c_{n,i}}{c_L}\right) \leq \varepsilon_n. \tag{10}$$

Then, other tasks are offloaded to the MEC servers, task offloading decisions $\{a_{n,i}\}$ are obtained. The offloading time τ_n is expressed as the function with $\beta_{n,i}$.

$$\tau_n = \sum_{i=1}^{I}\frac{(1-a_{n,i})c_{n,i}}{c_L} - \left(\sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n} + \frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B\right). \tag{11}$$

Then, **P**1 becomes an optimal problem with variables $\{\beta_{n,i}\}$, as sub-problem **P**1-1.

$$\underset{\{\beta_{n,i}\}}{max.}\ F_n,$$

subject to Eq. (11) and $0 \leq \beta_{n,i} \leq \beta_{max}, \forall i$.

The sub-problem **P**1-1 is a typical convex optimal problem, some convex optimization algorithms [39] can be used to solve it directly.

When the computation task requests arrive, if the vehicle n is leaving the coverage area of AP, $l_n^0 > D_g/2$, the distance between the vehicle and MEC servers is increasing. When the tasks are offloaded latter, the vehicles need to pay for more data transmission costs. Thus, the offloading tasks should be transmitted to MEC servers at the offloading time $\tau_n = 0$. The data rate R_n^0 is

$$R_n^0 = B_n \log_2\left\{1 + \frac{P_{tx}h_{ng}(d_{ng}(\tau_n=0))^{-r}}{\bar{w}_0}\right\},$$

where

$$d_{ng}(\tau_n = 0) = \sqrt{l_g^2 + \left(l_n^0 - D_g/2\right)^2}.$$

Similarly, with the expression of Eq. (9), we obtain the task offloading decisions $\{a_{n,i}\}$, **P**1 becomes an optimal problem with variables $\{\beta_{n,i}\}$, as the sub-problem **P**1-2.

$$\underset{\{\beta_{n,i}\}}{min.}\ F_n, \tag{12}$$

subject to the obtained optimal $\{a_{n,i}\}$, $B_n = B_{max}$, $\tau_n = 0$ and $0 \leq \beta_{n,i} \leq \beta_{n,max}, \forall i$.

The main difference between the sub-problems **P**1-1 and **P**1-2 is that in problem **P**1-2, the offloading time is $\tau_n = 0$. The same convex optimal algorithms can be used to solve these two problems directly. Above all, based on the solution of problem **P**1, the task offloading scheme in independent MEC servers scenario named as "Mobility-Aware Task Offloading (MATO) scheme", is shown as Algorithm 1.

Algorithm 1 MATO Scheme for the Vehicle n

Require: For vehicle n, the computation task i arrives, $i \in \{1, 2, \cdots, I\}$, $\{c_{n,i}, s_{n,i}\}$, v_n, c_L, c_{th}, T_n^{max}.

1: **if** $\sum_{i=1}^{I} c_{n,i}/c_L \leq T_n^{max}$ **then**
2: The vehicle n selects the local computing;
3: **else**
4: The vehicle n sends part of tasks $\{a_{n,i}s_{n,i}\}$ to MEC servers at the offloading time τ_n;
5: **if** $l_n^0 \leq D_g/2$ **then**
6: $B_n = B_{n,max}$, $\{a_{n,i}\}$ is obtained via Eq. (10), $\tau_n = \sum_{i=1}^{I}\frac{(1-a_{n,i})c_{n,i}}{c_L} - \left(\sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n} + \frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B\right)$;
7: $\{\beta_{n,i}\}$ is obtained via convex optimization algorithm for sub-problem **P**1-1;
8: **else**
9: $\tau_n = 0$, $B_n = B_{max}$;
10: $\{\beta_{n,i}\}$ is obtained via convex optimization algorithm for sub-problem **P**1-2;
11: **end if**
12: **end if**

Ensure: The suitable mobility-aware task offloading scheme.

Compared with the original optimization problem **P**1, the number of independent variables in the problems of the proposed MATO scheme are reduced from $\{a_{n,i}, B_n, \tau_n, \beta_{n,i}\}$ to $\{\beta_{n,i}\}$. The solution computational complexity is reduced directly. For the complexities are different when we use different algorithms to solve these problems, in this paper, we select the typical Newton-Raphson methods [39] to solve the convex problems **P**1-1 and **P**1-2. Set $N_M^n = \sum_{i=1}^{I} a_{n,i}$, the computational complexity is shown to be $\mathcal{O}(N_M^n)$.

V. TASK OFFLOADING FOR COOPERATIVE MEC SERVERS

In this section, we analyze the design of task offloading schemes in VECN in a common scenario: cooperative MEC servers scenario. When multiple vehicles enter the coverage area of a AP, the task computation requests increase. Many vehicles may offload computation tasks to the MEC servers. Both the communication and computation resource allocations are executed for the heterogeneous applications of tasks. However, the computing capacity of MEC servers is limited. To reduce the task completed latency, one of the recommended solution is that the excessive workloads are outsourced, to further be offloaded to other MEC servers deployed at the adjacent AP on the vehicles' moving direction. The cooperation between the MEC servers deployed at these two adjacent APs should be considered.

A. PROBLEM FORMULATION

For vehicle n, $n \in \{1, 2, \cdots, N\}$, when the computation requests arrive and the local computing capacity cannot meet the latency constraints, the vehicle can offload part of tasks $\{a_{n,i}s_{n,i}\}$ to the MEC servers. In detail, for the MEC servers at AP, when multiple tasks arrive or the data size of tasks is huge, it can further send part of tasks to the MEC servers deployed at the adjacent AP. The vehicles perform data transmission via the OFDMA. The interferences between channels are not considered. The computing capacities of MEC servers at these two APs are the same as c_{th}. Part of the computation resources $\beta_{max}c_{th}$ are allocated to the calculation requests from the vehicles in the coverage area of AP. And other resources $(1-\beta_{max})c_{th}$ are prepared for the outsourced computing tasks from the adjacent AP. Same as the analysis in single vehicle case, $a_{n,i} \in \{0, 1\}$. We denote $b_{n,i} \in \{0, 1\}$ as the further offloading decision for task i in MEC servers. Specially, $b_{n,i} = 1$ denotes the MEC servers decide to further offload task i to the adjacent servers, while $b_{n,i} = 0$ otherwise. In order to express the task offloading scheme in the cooperative MEC servers scenario clearly, we propose the model of task processing for vehicle n, as Fig. 4.

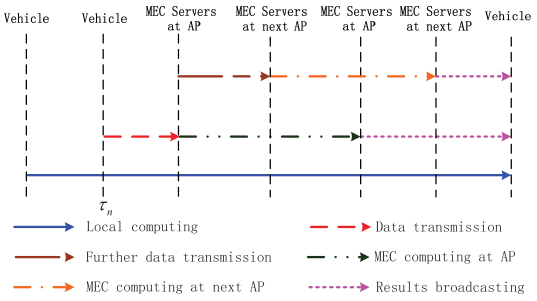


FIGURE 4. Task processing for vehicle n in cooperative MEC servers scenario.

As shown in Fig. 4, after the data uplink transmission between the vehicles and MEC servers, the MEC servers decide whether to calculate the received tasks locally or further to send part of them out. Specially, both the vehicles and MEC servers can perform data transmission and calculation simultaneously. The calculation results will be sent back to the vehicles. All of the tasks should be processed before the maximum latency threshold T_n^{max}. The task completed latency of task i in vehicle n is shown as

$$t_{n,i} = (1-a_{n,i})\frac{c_{n,i}}{c_L} + a_{n,i}(1-b_{n,i})\left(\frac{s_{n,i}}{R_n} + \frac{c_{n,i}}{\beta_{n,i}c_{th}} + t_B\right) + a_{n,i}b_{n,i}\left(\frac{s_{n,i}}{R_n} + \frac{s_{n,i}}{R_f} + \frac{c_{n,i}}{\beta'_{n,i}c_{th}} + t_B\right),$$

where R_f is the backhaul channel transmission data rate between APs. $\beta'_{n,i}$ denotes the computation resource allocation in the adjacent MEC servers. The total task completed latency of tasks in vehicle n is given by

$$t_n = \max\left\{\sum_{i=1}^{I}\frac{(1-a_{n,i})c_{n,i}}{c_L}, \tau_n + \sum_{i=1}^{I}a_{n,i}(1-b_{n,i})\left(\frac{s_{n,i}}{R_n} + \frac{c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B, \tau_n + \sum_{i=1}^{I}a_{n,i}b_{n,i}\left(\frac{s_{n,i}}{R_n} + \frac{s_{n,i}}{R_f} + \frac{c_{n,i}}{\beta'_{n,i}c_{th}}\right) + t_B\right\}. \quad (13)$$

The task completed system cost is shown as

$$F = \sum_{n=1}^{N}\left\{\phi_1 B_n \sum_{i=1}^{I}\frac{a_{n,i}s_{n,i}}{c_L} + \phi_f B_f \sum_{i=1}^{I}\frac{a_{n,i}b_{n,i}s_{n,i}}{R_f} + \phi_2\sum_{i=1}^{I}\left(\beta_{n,i}c_{th} + \beta'_{n,i}c_{th}\right)\right\}, \quad (14)$$

where ϕ_f and B_f are the unit communication cost and bandwidth of the backhaul channel, respectively. We set that the values of ϕ_f and B_f are fixed. Simplify, in order to reduce the computation latency, we set that the MEC servers deployed at the next AP provide all of the prepared computation resources for the outsourced computation requests, as

$$\sum_{n=1}^{N}\sum_{i=1}^{I}\beta'_{n,i}c_{th} = (1-\beta_{max})c_{th}. \quad (15)$$

In the cooperative MEC servers scenario, we aim to minimize the total task processing system cost. The offloading decisions $a_{n,i}$ and $b_{n,i}$, the uplink spectrum resource allocation B_n, computation resource allocation $\beta_{n,i}$ and the offloading time τ_n are jointly optimized. The optimization problem is formulated as **P2**.

$$\min_{\{a_{n,i},b_{n,i},\beta_{n,i},B_n,\tau_n\}} \quad F,$$

$$\text{subject to} \quad t_n \le T_n^{max}, \quad \forall n, \quad (16)$$

$$\sum_{n=1}^{N} B_n \le B_{max}, \quad (17)$$

$$0 \le \sum_{n=1}^{N}\beta_{n,i} \le \beta_{max}, \quad (18)$$

$$a_{n,i} \in \{0,1\}, \quad b_{n,i} \in \{0,1\}, \quad \forall n, i. \quad (19)$$

Due to the fact that $a_{n,i}$ and $b_{n,i}$ are binary variables, and in the formulations of objective function F as Eq. (14) and constraint t_n as Eq. (13), there exist product relationships between variables $a_{n,i}$ and $b_{n,i}$, the problem **P**2 is a complex non-convex MINLP. It is difficult to solve it directly, Thus, a sub-optimal and low-complexity solution for problem **P**2 is proposed in the next subsection.

B. LOCATION-BASED TASK OFFLOADING SCHEME

In this section, we present a Location-Based Task Offloading (LBTO) scheme to obtain a sub-optimal solution for problem **P**2. Based on the different initial locations, moving speeds of vehicles, and latency requirements, we divide the vehicles into different groups. The task offloading schemes in each group are analyzed separately. The solution becomes low-complexity. In detail, the LBTO scheme includes three steps, which are shown as follows.

1) VEHICLE GROUPING

The objective of problem **P**2 is to minimize the system costs, under the task completed latency thresholds. Thus, the vehicles should select local computing as much as possible. We divide these vehicles into a group, denoted as $\mathcal{G}_{\mathcal{L}}$, in which the local computing capacity can meet the latency constraint, $a_{n,i} = 0, \forall i$. When $\sum_{i=1}^{I} c_{n,i}/c_L \leq T_n^{max}$, we have $n \in \mathcal{G}_{\mathcal{L}}$.

The second group, denoted as $\mathcal{G}_{\mathcal{F}}$, is defined that the vehicles should send part of tasks to the MEC servers, and these offloaded tasks should further be transmitted to the MEC servers deployed at the next AP on the moving direction of vehicles. For the calculation results will be sent back to vehicles, the vehicles should arrive the coverage area of the next AP before the task processing completed. The initial locations and moving speeds of vehicles affect the arriving positions. When $\sum_{i=1}^{I} c_{n,i}/c_L > T_n^{max}$ and $(D_g - l_n^0)/v_n \leq T_n^{max}$, we have $n \in \mathcal{G}_{\mathcal{F}}$.

Besides these two groups mentioned before, the other vehicles are assigned to the group $\mathcal{G}_{\mathcal{O}}$, $n \in \mathcal{G}_{\mathcal{O}}$. The vehicles belonged to the group $\mathcal{G}_{\mathcal{O}}$, must offload part of their computation tasks to the MEC servers, and the tasks should be processed when the vehicles stay at the coverage area of the AP. Same as the analysis in the single vehicle case, the vehicles in the group are divided into two sub-groups $\mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$, $\mathcal{G}_{\mathcal{O}}^{\mathcal{U}} \cup \mathcal{G}_{\mathcal{O}}^{\mathcal{L}} = \mathcal{G}_{\mathcal{O}}$, based on the initial positions of vehicles. When $l_n^0 \leq D_g/2, n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$. Otherwise, $l_n^0 > D_g/2, n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$.

2) OFFLOADING TIME CALCULATION

The vehicle n can send part of tasks to MEC servers at the offloading time τ_n. For the vehicles in group $\mathcal{G}_{\mathcal{F}}$, $n \in \mathcal{G}_{\mathcal{F}}$, their offloading tasks will further be transmitted to MEC servers deployed at the next AP, $b_{n,i} = 1, \forall i$. The vehicles are leaving the coverage area of the AP, the data transmission distances between the vehicles and AP increase. Therefore, the vehicles should send the data out as soon as possible, the offloading time is $\tau_n = 0$, when $n \in \mathcal{G}_{\mathcal{F}}$.

For the vehicles in group $\mathcal{G}_{\mathcal{O}}, n \in \mathcal{G}_{\mathcal{O}}$, the vehicles can send part of tasks to the MEC servers. When $l_n^0 \leq D_g/2, n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$, the vehicle n is moving close to the AP, the optimal offloading time τ_n is obtained via the solution of problem **P**1-1. Otherwise, $l_n^0 > D_g/2, n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$, the vehicle n is leaving the coverage area of the AP, the offloading time is $\tau_n = 0$.

3) RESOURCE ALLOCATION

In the third step, both the communication and computation resources are allocated, when the vehicles perform data offloading synchronously. All of the vehicles in group $\mathcal{G}_{\mathcal{F}}$ and sub-group $\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$ send the computation tasks to MEC servers when $\tau_n = 0$, resource allocation should be considered for these vehicles.

When $n \in \mathcal{G}_{\mathcal{F}}$, we have $b_{n,i} = 1, \forall i$. Due to the formulations of t_n and F in Eqs. (13)(14), the task completed latency and system cost of vehicle n in group $\mathcal{G}_{\mathcal{F}}$, denoted as t_n^1 and $\hat{F}_n^1$, are shown as

$$t_n^1 = \max\left\{ \sum_{i=1}^{I} \frac{(1-a_{n,i})c_{n,i}}{c_L}, \sum_{i=1}^{I} a_{n,i}\left(\frac{s_{n,i}}{R_n^0} + \frac{s_{n,i}}{R_f} + \frac{c_{n,i}}{\beta'_{n,i}c_{th}}\right) + t_B \right\},$$

$$\hat{F}_n^1 = \phi_1 B_n \sum_{i=1}^{I} \frac{a_{n,i}s_{n,i}}{R_n^0} + \phi_f B_f \sum_{i=1}^{I} \frac{a_{n,i}s_{n,i}}{R_f} + \phi_2 \sum_{i=1}^{I} \beta'_{n,i}c_{th}.$$

When $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$, we have $b_{n,i} = 0$, the task completed latency and system cost, denoted as t_n^2 and $\hat{F}_n^2$, are shown as

$$t_n^2 = \max\left\{ \sum_{i=1}^{I}(1-a_{n,i})\frac{c_{n,i}}{c_L}, \sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n^0} + \frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right) + t_B \right\},$$

$$\hat{F}_n^2 = \phi_1 B_n \sum_{i=1}^{I} \frac{a_{n,i}s_{n,i}}{R_n^0} + \phi_2 \sum_{i=1}^{I} \beta_{n,i}c_{th}.$$

The problem **P**2 is rewritten as a sub-problem **P**2-1, as

$$\min_{\{a_{n,i},B_n,\beta_{n,i}\}} \sum_{n\in\mathcal{G}_{\mathcal{F}}} \hat{F}_n^1 + \sum_{n\in\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}} \hat{F}_n^2,$$

$$\text{subject to } t_n^1 \leq T_n^{max}, \quad n \in \mathcal{G}_{\mathcal{F}}, \tag{20}$$

$$t_n^2 \leq T_n^{max}, \quad n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}, \tag{21}$$

$$\sum_{n\in(\mathcal{G}_{\mathcal{F}}\cup\mathcal{G}_{\mathcal{O}}^{\mathcal{L}})} B_n \leq B_{max}, \tag{22}$$

$$0 \leq \sum_{n\in\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}} \beta_{n,i} \leq \beta_{max}, \tag{23}$$

$$a_{n,i} \in \{0,1\}, \quad \forall i, n \in (\mathcal{G}_{\mathcal{F}} \cup \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}). \tag{24}$$

We set that the MEC servers deployed at the next AP provide the remain computation resources for the arriving tasks, as Eq. (15), we have $\sum_{n\in\mathcal{G}_{\mathcal{F}}}\sum_{i=1}^{I}\beta'_{n,i} = (1-\beta_{max})$. Same as the analysis in problem **P**1-1, the efficient way to minimize the system cost is that the latency of local computing is the same as the one of task offloading. Thus, when $n \in \mathcal{G}_{\mathcal{F}}$, we have

$$\sum_{i=1}^{I}\frac{(1-a_{n,i})c_{n,i}}{c_L} = \sum_{i=1}^{I}a_{n,i}\left(\frac{s_{n,i}}{R_n^0}+\frac{s_{n,i}}{R_f}+\frac{c_{n,i}}{\beta'_{n,i}c_{th}}\right)+t_B \le T_n^{max}. \quad (25)$$

With Eqs. (10) and (25), we can obtain the task offloading decisions $\{a_{n,i}\}$. And the bandwidth B_n of vehicle n in group $\mathcal{G}_{\mathcal{F}}$ is expressed as a equation with $\{a_{n,i}\}$.

When $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$, we have

$$\sum_{i=1}^{I}(1-a_{n,i})\frac{c_{n,i}}{c_L} = \sum_{i=1}^{I}\left(\frac{a_{n,i}s_{n,i}}{R_n^0}+\frac{a_{n,i}c_{n,i}}{\beta_{n,i}c_{th}}\right)+t_B \le T_n^{max}. \quad (26)$$

The task offloading decisions $\{a_{n,i}\}$ can also be obtained via Eqs. (10) and (26). Then, the variables of problem **P**2-1 becomes $\{B_n, \beta_{n,i}\}$, $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$. **P**2-1 becomes a convex optimal problem, and some typical algorithms can be used to solve it.

For the other vehicles in subgroup $\mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$, $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$, the vehicles send part of computation tasks to the MEC servers at different offloading times $\{\tau_n\}$. Thus, the whole communication resources are utilized, $B_n = B_{max}$. The optimal offloading decisions $\{a_{n,i}\}$ and other variables of each vehicle n are obtained via the solution in problem **P**1-1, separately.

Above all, we obtain the task offloading scheme named: "Location-Based Task Offloading (LBTO) scheme", the detail is shown as Algorithm 2.

In the proposed LBTO scheme, the computational complexity primarily come from the step 2 and step 3. In step 2, offloading time calculation, $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$, the vehicles send the tasks at different times, the solution algorithms (i.e., Newton-Raphson methods [39]) used in problem **P**1-1 can also be used. Set $N_M^n = \sum_{i=1}^{I} a_{n,i}$, the computational complexity is $\mathcal{O}(\sum_{n\in\mathcal{G}_{\mathcal{O}}^{\mathcal{U}}} N_M^n)$. In step 3, resource allocation, we select the same Newton-Raphson methods to solve the convex optimization problem **P**2-1. Set $|\mathcal{G}_{\mathcal{F}}|$ and $|\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}|$ as the number of vehicles in groups $\mathcal{G}_{\mathcal{F}}$ and $\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$, the computational complexity is $\mathcal{O}((|\mathcal{G}_{\mathcal{F}}|+|\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}|)\sum_{n\in(\mathcal{G}_{\mathcal{F}}\cup\mathcal{G}_{\mathcal{O}}^{\mathcal{L}})} N_M^n)$. Then, the total computational complexity of the solution algorithms in LBTO scheme is $\mathcal{O}(\sum_{n\in\mathcal{G}_{\mathcal{O}}^{\mathcal{U}}} N_M^n + (|\mathcal{G}_{\mathcal{F}}|+|\mathcal{G}_{\mathcal{O}}^{\mathcal{L}}|)\sum_{n\in(\mathcal{G}_{\mathcal{F}}\cup\mathcal{G}_{\mathcal{O}}^{\mathcal{L}})} N_M^n)$.

VI. NUMERICAL RESULTS

In this section, we conduct numerical simulations to evaluate the performance of the proposed task offloading schemes in both independent and cooperative MEC servers scenarios. We consider that two APs are deployed at the roadside. Similar as the ref. [40], we set the coverage of a AP is $D_g = 600m$, the AP is located at the center of the coverage area, the altitude of AP is $l_g = 20m$. The bandwidth of available channels among the vehicles and AP is $B_{max} = 20MHz$, and the path loss efficient is $r = 2.5$. The V2I channel is modeled as Rayleigh fading channel with average power loss $10^{-3}W$. For the VECN, we consider the computing capacities of vehicles are the same, as $c_L = 2\times10^8$ $cycles/sec$. The computing capacities of MEC servers are the same, as $c_{th} = 8\times10^8$ $cycles/sec$. For the system cost, we set that the cost of unit spectrum resource block is $\phi_1 = 1$ $RMB/(MHzsec)$, the cost of unit computation resources in MEC servers is $\phi_2 = 3$ $RMB/(cycles/sec)$, and the unit communication cost in backhaul channel is $\phi_f = 1$ $RMB/(MHzsec)$, $\beta_{max} = 0.8$, $\bar{w}_0 = 10^{-9}W$. If not specified, we summarize the default simulation parameters in Table 1.

Algorithm 2 LBTO Scheme for Multiple Vehicles

Require: For vehicle n, $n \in \{1,2,\cdots,N\}$, computation tasks, $\{c_{n,i}, s_{n,i}\}$, $i \in \{1,2,\cdots,I\}$, v_n, c_L, c_{th}, T_n^{max}, β_{max}, B_{max}, R_f.

Stage 1: Vehicle grouping
if $\sum_{i=1}^{I} c_{n,i}/c_L \le T_n^{max}$ **then**
  $n \in \mathcal{G}_{\mathcal{L}}$;
else
  if $(D_g - l_n^0)/v_n \le T_n^{max}$ **then**
    $n \in \mathcal{G}_{\mathcal{F}}$;
  else
    if $l_g^0 \le D_g/2$ **then**
      $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$;
    else
      $n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{L}}$;
    end if
  end if
end if
Step 2: Offloading time calculation
$n \in \mathcal{G}_{\mathcal{L}}$, the vehicles select the local computing;
$n \in (\mathcal{G}_{\mathcal{F}} \cup \mathcal{G}_{\mathcal{O}}^{\mathcal{L}})$, $\tau_n = 0$;
$n \in \mathcal{G}_{\mathcal{O}}^{\mathcal{U}}$, $B_n = B_{max}$, τ_n is obtained via the solution of problem **P**1-1, while the resource allocations $\{a_{n,i}, B_n, \beta_{n,i}\}$ of vehicle n are obtained separately.
Step 3: Resource allocation
$n \in (\mathcal{G}_{\mathcal{F}} \cup \mathcal{G}_{\mathcal{O}}^{\mathcal{L}})$, the resource allocations $\{a_{n,i}, B_n, \beta_{n,i}\}$ are obtained via the convex optimization algorithms utilized in the problem **P**2-1.

Ensure: The suitable location-based task offloading scheme.

TABLE 1. Default parameter setup.

Parameter	Definition	Value
R_f	Backhaul channel data rate between APs	$20Mbps$
B_f	Backhaul channel bandwidth between APs	$10MHz$
P_{tx}	Transmission powers of vehicle	$1W$
t_B	Results feedback time	$0.1sec$

The performances of the proposed task offloading schemes (i.e., MATO and LBTO schemes) are compared with the following baseline or simple schemes:

- **Random offloading scheme**: The vehicles offload the computation tasks to the MEC serves or process them locally randomly. $a_{n,i} \in \{0, 1\}$. The generation of values 0 and 1 follows equal probability [30]. Other parameters, such as offloading times $\{\tau_n\}$, resource allocations $\{\beta_{n,i}, B_n\}$ are optimized as usual.
- **Direct offloading scheme**: The vehicles send the computation tasks when they arrive, the offloading time is $\tau_n = 0$. Other parameters $\{a_{n,i}, \beta_{n,i}, B_n\}$ are optimized as usual.
- **TDMA-based offloading scheme**: The vehicles send the computation tasks in different time slots, the offloading times $\{\tau_n\}$ are optimized. For the offloading process in vehicle n, the bandwidth of the V2I channel is $B_n = B_{max}$ [41], [42]. This scheme is suitable for the cooperative MEC servers scenario, multiple vehicles send tasks to the AP.

A. INDEPENDENT MEC SERVERS SCENARIO

In the independent MEC servers scenario, we consider that one vehicle passes through the coverage area of AP with high speed. $v_n = 100km/h$. We consider the number of computation tasks in vehicle n is $I = 20$. The data size of each task $\{s_{n,i}\}$ is selected randomly from $[2, 6]Mbit$, and the corresponding task required CPU cycles $\{c_{n,i}\}$ are randomly distributed in range $[40, 120]Mcycle$. The performance of the proposed MATO scheme will be compared with other schemes. Besides the data sizes and the required CPU cycles of tasks, the task offloading time τ_n, the initial positions of vehicle l_n^0 and the task completed latency threshold T_n^{max} affect the system performance directly. The simulation results are shown as Fig. 5-7.

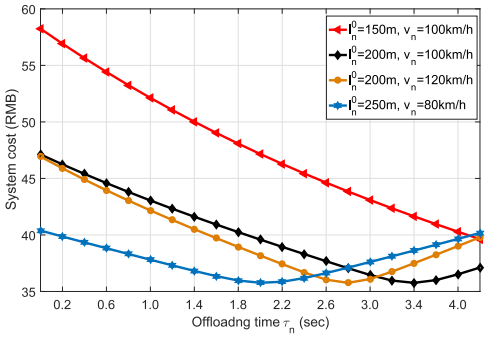


FIGURE 5. The task completed system costs of the proposed MATO scheme.

Fig. 5 shows the task completed system costs of the proposed MATO scheme under different offloading times τ_n, initial locations l_n^0, and moving speeds v_n. The task completed maximum latency threshold is $T_n^{max} = 6\ sec$. From Fig. 5, we can find that when the initial location of vehicle is far from the AP and it is moving close to the AP, $l_n^0 = 150m$, the system cost decreases with the increasing of offloading time. This is because that the vehicle n is moving close to AP, when the value of τ_n increases, the distance between the vehicle and AP is reduced, and the vehicle can send computation tasks to the MEC servers deployed at AP with less communication cost. Moreover, when the calculation tasks arrive, the vehicle n is closing to the AP, as in case $l_n^0 = 250m, v_n = 80km/h$, the system cost will reduce as the increasing of τ_n. When $\tau_n > 2.04\ sec$, the system cost will increase. The reason is that when the vehicle selects a large offloading time, the vehicle may pass the AP. The distance between the vehicle and AP is increasing, more communication costs are needed. It is necessary to select an optimal offloading time, and both the initial locations and moving speeds affect the optimal selection.

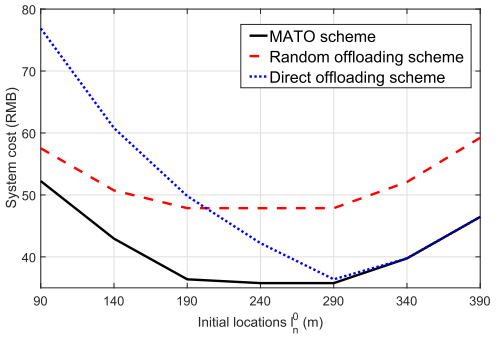


FIGURE 6. The task completed system costs under different initial positions of vehicles l_n^0.

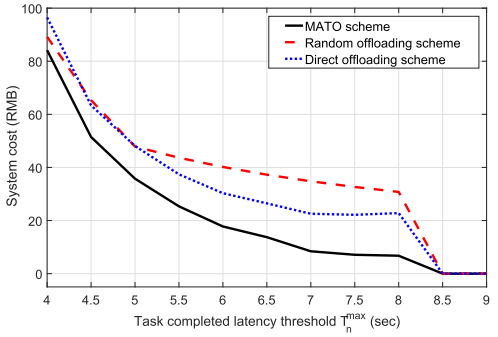


FIGURE 7. The task completed system costs under different latency thresholds T_n^{max}.

Fig. 6 shows the task completed system cost comparisons under different initial locations l_n^0. We can find that: under all of the schemes, when $l_n^0 < 300m$, the system cost decreases as the increasing of the value of l_n^0, otherwise, when $l_n^0 > 300m$, the system cost increases. The reason is that the initial locations affect the distances between the vehicles and AP. When the distance is reducing, the communication cost for the computation tasks reduces. Meanwhile, the performance of the proposed MATO scheme is better than other two schemes, which means that the offloading time optimal selection is beneficial, the tradeoff between the transmission efficiency and communication cost is mainly considered. But, when $l_n^0 > 300m$, the performances of MATO is the same as

Direct offloading scheme. This is because when the vehicle is leaving the coverage area of AP, the vehicle in MATO scheme selects offload the data out at $\tau_n = 0$, same as the selection in Direct offloading scheme.

Fig. 7 shows the task completed system costs under different latency thresholds T_n^{max}. We observe that the higher the value of T_n^{max} is, the lower system cost is. The reason is that we consider that the vehicles calculate tasks locally free. They need to pay for the communication and computation costs when they offload part/all of the tasks to the MEC servers. When the value of T_n^{max} becomes larger, more tasks can be calculated locally. The performance of the proposed MATO scheme is better than others, and when $T_n^{max} > 8.5\ sec$, the system cost becomes zero. This is because when the latency threshold is larger, all of the tasks are processed locally. Task offloading is not needed and the system cost is zero.

In summary, in the independent MEC servers scenario, when the vehicle n enters the coverage area of the AP and performs the task offloading scheme, the performance of our proposed MATO scheme is better than other schemes. The reason is that the mobility of vehicle is considered mainly, the task offloading time and the computation resource allocation are optimized jointly.

B. COOPERATIVE MEC SERVERS SCENARIO

In the cooperative MEC servers scenario, we consider the number of moving vehicles in the coverage area of AP is $N = 10$. When the computation tasks arrive, the initial locations of vehicles are randomly selected from range $[50, 550]m$. The moving speeds of vehicles are randomly selected from range $[80, 100]km/h$. To ensure the traffic safety, we set the distances between two adjacent vehicles are larger than $40m$. For convenience, when the value of the vehicle index n is large, the initial location l_n^0 becomes larger. For the calculation tasks, we set the data sizes $\{s_{n,i}\}$ are selected randomly from $[2, 6]Mbit$, and the corresponding task required CPU cycles $\{c_{n,i}\}$ are randomly distributed in range $[40, 120]Mcycle$, same as the setting in single vehicle case. The performances of the proposed LBTO scheme are compared with Random offloading scheme and Direct offloading scheme.

Fig. 8 shows the system costs of each vehicle. The task completed latency thresholds T_n^{max} of vehicles are distributed in range $[4, 8]sec$ randomly. We can see that: for the vehicles 1, 2 and 4, the system costs of the proposed LBTO scheme are lower than Random offloading scheme and Direct offloading scheme. The reason is that in Random offloading scheme, the vehicle selects the offloading tasks randomly, more tasks may be offloaded. In Direct offloading scheme, the vehicle sends the calculation tasks directly, the channel transmission conditions between the vehicle and AP are not considered. In our proposed LBTO scheme, the vehicle selects an optimal offloading time, both the communication and computation resource allocations are considered. The tradeoff between the task completed latency and the system costs (i.e., the required

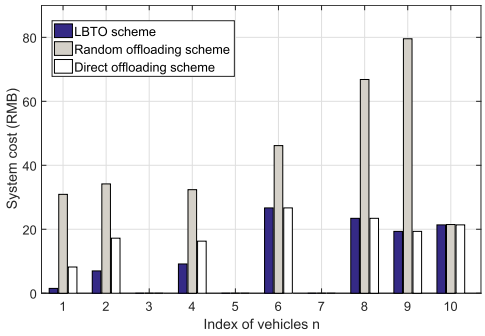


FIGURE 8. The task completed system costs of each vehicle.

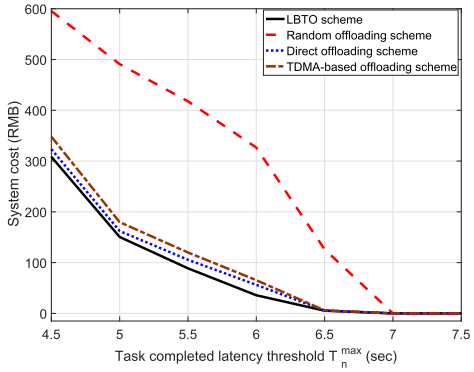


FIGURE 9. The task completed system costs of vehicles under different latency thresholds T_n^{max}.

communication and the computation resources) is considered. For the vehicles 3, 5 and 7, the system costs are zeros. The reason is that when the data sizes of computation tasks are limited, and the task completed latency threshold is large, the vehicles can finish the task processing locally, no other system cost happens. In addition, for the vehicles 6, 8, 9 and 10, the performance of LBTO scheme is the same as the Direct offloading scheme. This is because that these vehicles are leaving the coverage area of the AP, $l_n^0 > 300m$. Based on our analysis, the optimal offloading time is $\tau_n = 0$.

Fig. 9 shows the task completed total system cost comparisons, under different latency thresholds. We set that the task completed latency thresholds T_n^{max} of vehicles are the same. We find that the performance of LBTO scheme is better than other three schemes. As the increasing of T_n^{max}, the gap of total system cost between the LBTO scheme and other schemes becomes smaller. The reason is that when the latency threshold increases, more tasks are processed locally, the amount of the offloading calculation tasks decreases. The performance of the proposed LBTO scheme is better than the TDMA-based offloading scheme. The reason is that in the TDMA-based offloading scheme, the vehicles send their computation tasks in different times τ_n. Only the offloading times τ_n are optimized, based on the data sizes of tasks and the initial locations of vehicles. The vehicles access all of the maximum leased spectrum resources, and the offloading communication costs are large. When $T_n^{max} > 7\ sec$, we find that the system costs become zero. This is because that in

this scenario, the latency is larger enough that all of the tasks can be processed locally. No task offloading is needed. The proposed LBTO scheme can optimize the offloading times and perform both the communication and computation resources allocations optimally. It is more suitable for the task offloading in VECN.

VII. CONCLUSION

In this paper, we investigate the task offloading mechanisms in vehicular edge computing networks, to minimize the system costs (i.e. communication cost and computation cost), while the task completed maximum latency and other constraints are satisfied. Specially, the fast moving characteristic of vehicles is mainly considered. The vehicles should pay for the leased communication and computation resources. We analyze the design of task offloading schemes in both independent and cooperative MEC servers scenarios. When number of tasks arrive, the vehicles can send part of them to the MEC servers. Moreover, when the offloading tasks cannot be processed under the latency requirements, the MEC servers can further offload part of tasks to the next AP on the vehicles' moving direction. Two optimization problems are proposed. Based on the initial locations, the moving speeds of vehicles and the task completed latencies, the vehicles can select local computing, offloading or further offloading optimally, to balance the tradeoff between the task completed latency and system cost. Extensive simulations are proposed to demonstrate the effectiveness of the proposed schemes.

APPENDIX

To prove **P**1 is a convex optimization problem under the condition $0 \leq a_{n,i} \leq 1$, we should prove the objective function F_n and the main constraint function t_n are convex with the variables $\{a_{n,i}, \beta_{n,i}, \tau_n, B_n\}$. The objective function is shown as

$$F_n = \phi_1 B_n \sum_{i=1}^{I} a_{n,i} s_{n,i} / R_n + \phi_2 \sum_{i=1}^{I} \beta_{n,i} c_{th}.$$

The bandwidth B_n do not affect the values of F_n. According to the constraint Eq. (4), we have $B_n = B_{max}$. In addition, F_n is linear function with $\beta_{n,i}$. Then, set $F_n^p = \frac{a_{n,i} s_{n,i}}{R_n}$, we need to calculate the Hessian matrix of F_n^p, shown as

$$H(F_n^p) = \begin{bmatrix} \frac{\partial^2 F_n^p}{\partial \tau_n^2} & \frac{\partial^2 F_n^p}{\partial \tau_n \partial a_{n,i}} \\ \frac{\partial^2 F_n^p}{\partial a_{n,i} \partial \tau_n} & \frac{\partial^2 F_n^p}{\partial a_{n,i}^2} \end{bmatrix} \tag{27}$$

We have

$$\frac{\partial^2 F_n^p}{\partial \tau_n^2} = 0,$$

when the vehicle is moving close to the AP, $l_n^0 \leq D_g/2$, we have

$$\frac{\partial^2 F_n^p}{\partial \tau_n \partial a_{n,i}} = \frac{\partial^2 F_n^p}{\partial a_{n,i} \partial \tau_n} = \frac{B}{2 \ln_2 R_n} \frac{-r s_{n,i}}{\bar{w}_o} (v_n) < 0,$$

when $l_n^0 > D_g/2$, we have

$$\frac{\partial^2 F_n^p}{\partial \tau_n \partial a_{n,i}} = \frac{\partial^2 F_n^p}{\partial a_{n,i} \partial \tau_n} = \frac{B}{2 \ln_2 R_n} \frac{-r s_{n,i}}{\bar{w}_o} (-v_n) > 0.$$

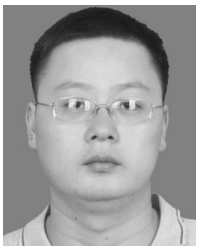
Then, we obtain $H(F_n^p) < 0$. The objective function is a concave function with the variables $\{a_{n,i}, \tau_n, \beta_{n,i}\}$. Same as the analysis, we can find that the constraint function t_n is also concave function. Thus, the problem **P**1 is a convex optimization problem.

The proof is completed.

REFERENCES

[1] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: From a communication perspective," *Mech. Syst. Signal Process.*, vol. 25, no. 6, pp. 2020–2038, Aug. 2011.
[2] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 996–1014, Mar. 2018.
[3] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan./Feb. 2018.
[4] A. Boukerchea and R. E. De Grande, "Vehicular cloud computing: Architectures, applications, and mobility," *Comput. Netw.*, vol. 135, pp. 171–189, Apr. 2017.
[5] M. H. Eiza, Q. Ni, and Q. Shi, "Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7868–7881, Oct. 2016.
[6] N. Cordeschi, D. Amendola, M. Shojafar, P. G. V. Naranjo, and E. Baccarelli, "Memory and memoryless optimal time-window controllers for secondary users in vehicular networks," in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst.*, 2015, pp. 1–7.
[7] Y. Mao, J. Zhang, Z. Chen, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.
[8] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive offloading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.
[9] H. Guo, J. Liu, and J. Zhang, "Computation offloading for multi-access mobile edge computing in ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 14–19, Aug. 2018.
[10] H. Guo, J. Liu, J. Zhang, W. Sun, and N. Kato, "Mobile-edge computation offloading for ultradense IoT networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4977–4988, Dec. 2018.
[11] P. Dong, T. Zheng, S. Yu, H. Zhang, and X. Yan, "Enhancing vehicular communication using 5G-enabled smart collaborative networking," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 72–79, Dec. 2017.
[12] B. M. Masini, A. Bazzi, and E. Natalizio, "Radio access for future 5G vehicular networks," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–7.
[13] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 94–100, Jul. 2017.
[14] H. Zhou, H. Wang, X. Chen, X. Li, and S. Xu, "Data offloading techniques through vehicular ad hoc networks: A survey," *IEEE Access*, vol. 6, pp. 65250–65259, 2018.
[15] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.
[16] C. Liang, Y. He, F. R. Yu, and N. Zhao, "Enhancing QoE-aware wireless edge caching with software-defined wireless networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6912–6925, Oct. 2017.
[17] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[18] H. Guo, J. Zhang, and J. Liu, "FiWi-enhanced vehicular edge computing networks: Collaborative task offloading," *IEEE Veh. Technol. Mag.*, vol. 14, no. 1, pp. 45–53, Mar. 2019.
[19] S. Bu, F. R. Yu, Y. Cai, and X. P. Liu, "When the smart grid meets energy-efficient communications: Green wireless cellular networks powered by the smart grid," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 3014–3024, Aug. 2012.

[20] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7432–7445, Aug. 2017.
[21] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.
[22] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 774–777, Dec. 2017.
[23] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.
[24] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Trans. Commun.*, vol. 65, no. 8, pp. 3571–3584, Aug. 2017.
[25] K. Zhang *et al.*, "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016.
[26] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.
[27] X. Ma, S. Zhang, W. Li, P. Zhang, C. Lin, and X. Shen, "Cost-efficient workload scheduling in cloud assisted mobile edge computing," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–10.
[28] W. Fan, Y. Liu, B. Tang, F. Wu, and Z. Wang, "Computation offloading based on cooperations of mobile edge computing-enabled base stations," *IEEE Access*, vol. 6, pp. 22622–22633, 2018.
[29] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.
[30] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.
[31] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[32] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber–wireless networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4514–4526, May 2018.
[33] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Netw.*, vol. 32, no. 5, pp. 112–117, Sep./Oct. 2018.
[34] J. Feng, Z. Liu, C. Wu, and Y. Ji, "AVE: Autonomous vehicular edge computing framework with ACO-based scheduling," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 10660–10675, Dec. 2017.
[35] C.-M. Huang, Y.-F. Chen, S. Xu, and H. Zhou, "The vehicular social network (VSN)-based sharing of downloaded Geo data using the credit-based clustering scheme," *IEEE Access*, vol. 6, pp. 58254–58271, 2018.
[36] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[37] H. Zhang, Y. Ma, D. Yuan, and H.-H. Chen, "Quality-of-service driven power and sub-carrier allocation policy for vehicular communication networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 197–206, Jan. 2011.
[38] R. A. Osman, X.-H. Peng, and M. A. Omar, "Adaptive cooperative communications for enhancing QoS in vehicular networks," *Phys. Commun.*, to be published.
[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[40] B. Zhang, X. Jia, K. Yang, and R. Xie, "Design of analytical model and algorithm for optimal roadside AP placement in VANETs," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7708–7718, Sep. 2016.
[41] R. Zhang, J. Lee, X. Shen, X. Cheng, L. Yang, and B. Jiao, "A unified TDMA-based scheduling protocol for vehicle-to-infrastructure communications," in *Proc. WCSP*, Oct. 2013, pp. 1–6.
[42] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, "TDMA-based MAC protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2461–2492, 4th Quart., 2015.



CHAO YANG received the Ph.D. degree in signal and information processing from the South China University of Technology, Guangzhou, China, in 2013. From 2014 to 2016, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University, Guangzhou. He is currently with the School of Automation, Guangdong University of Technology. His research interests include VANETs, smart grid, and edge computing.

YI LIU received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2011. After that, he held a Postdoctoral position with the Singapore University of Technology and Design. In 2014, he was with the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, China, where he is currently a Full Professor. His research interests include wireless communication networks, cooperative communications, smart grid, and intelligent edge computing.

XIN CHEN received the Ph.D. degree in bioinformatics from Harbin Medical University, Harbin, China, in 2012. After that, she was a Postdoctoral Fellow with the Faculty of Health Sciences, University of Macau. Since 2016, she has been with the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou, China. Her research interests include computational biology, wireless big data analysis, and convex optimization.

WEIFENG ZHONG received the B.Eng. and M.Eng. degrees from the Guangdong University of Technology, Guangzhou, China, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in control science and engineering. In 2016, he was a Visiting Student with The Hong Kong University of Science and Technology for six months. His research interests include energy management of vehicle-to-grid, power system, and edge computing.

SHENGLI XIE (M'01–SM'02–F'19) received the M.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1992, and the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, China, in 1997. He is currently a Full Professor and the Head of the Institute of Intelligent Information Processing, Guangdong University of Technology, Guangzhou. He has authored or co-authored two books and over 150 scientific papers in journals and conference proceedings. His research interests include wireless networks, automatic control, and blind signal processing. He was a recipient of the Second Prize in China's State Natural Science Award in 2009 for his research on blind source separation and identification.