

Received April 23, 2020, accepted May 13, 2020, date of publication May 19, 2020, date of current version June 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995797

Intelligent Resource Allocation for Utility Optimization in RSU-Empowered Vehicular Network

CHAOGANG TANG^{ID 1}, (Member, IEEE), CHUNSHENG ZHU^{ID 2,3}, (Member, IEEE),

XIANGLIN WEI^{ID 4}, HUAMING WU^{ID 5}, (Member, IEEE),

QING LI^{ID 6}, (Senior Member, IEEE), AND JOEL J. P. C. RODRIGUES^{ID 7,8}, (Fellow, IEEE)

¹School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

²SUSTech Institute of Future Networks, Southern University of Science and Technology, Shenzhen 518055, China

³PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518055, China

⁴The 63rd Research Institute, National University of Defense Technology, Changsha 410073, China

⁵Center for Applied Mathematics, Tianjin University, Tianjin 300072, China

⁶Department of Computing, The Hong Kong Polytechnic University, Hong Kong

⁷Federal University of Piauí, Teresina 64049-550, Brazil

⁸Instituto de Telecomunicações, 1049-001 Lisbon, Portugal

Corresponding author: Chunsheng Zhu (chunsheng.tom.zhu@gmail.com)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2019XKQYMS46, in part by the PCL Future Greater-Bay Area Network Facilities for Large-Scale Experiments and Applications Project under Grant LZC0019, in part by the FCT/MCTES through national funds, in part by the EU Funds under Project UIDB/EEA/50008/2020, and in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 309335/2017-5.

ABSTRACT Intelligent transportation system (ITS) has attracted extensive attention in both academia and industry for its potential benefits. For example, ITS is dedicated to convenient, economical and environmentally friendly service provisioning for the drivers and passengers in vehicles via advanced technologies including artificial intelligence (AI), knowledge mining, depth fusion, etc. Besides, several newly emerging computing paradigms revolved around ITS such as vehicular cloud and vehicular fog computing are proposed to fully exploit idle computing and communication resources within connected vehicles. As the number of vehicular applications is explosively increasing, it has posed great challenges to the limited capabilities of vehicle loaded computer systems and communication facility. Accordingly, more intelligent resource allocation strategies are needed for computationally intensive and time sensitive vehicular applications. In this paper we propose a road side unit (RSU) empowered vehicular network that consists of three hierarchical layers—vehicular cloud, RSU-enabled cloudlet, and central cloud, respectively. RSU is enhanced with edge servers such that it can intelligently respond to the resource requests in a real time fashion. To this end, an approximate but efficient resource allocation strategy is proposed that can intelligently optimize the utility value from the perspective of RSU-enabled cloudlet. Extensive experiments are carried out to evaluate the performance of the strategy. The results reveal that the proposed algorithm DbHA shows great advantages over other approaches such as the genetic algorithm (GA) and particle swarm optimization (PSO) in both respects (i.e., performance and response latency).

INDEX TERMS Intelligent transportation system, intelligent, vehicular, RSU, service provisioning.

I. INTRODUCTION

Intelligent transportation system (ITS) is a fully functioning ecosystem which makes the most of the Internet of Things, cloud computing, and mobile internet. ITS further integrates transportation science, artificial intelligence,

The associate editor coordinating the review of this manuscript and approving it for publication was Edith C.-H. Ngai^{ID}.

knowledge mining, depth fusion, with an aim to promote safe, efficient, convenient, economical and environmentally friendly transport operations and services for the drivers and passengers in vehicles. ITS has attracted more and more attention in both academia and industry in recent years. In ITS, on one hand, large masses of data and information [1] can be gathered from mounted facilities such as SIM cards, GPS, sensors, and cameras; on the other hand, with the help

of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication technologies, data and information can be shared and disseminated for better service provisioning. As a consequence, a variety of vehicular applications have been developed for maintaining the stability, profitability and effectiveness of ITS ecosystem [2]. Further to this, newly emerging computing paradigms such as vehicular cloud (VC) [3], [4] and vehicular fog computing (VFC) [5] have ushered significant academic and industry efforts to fully exploit idle computing and communication resources within connected vehicles.

For instance, an enticing application scenario is envisioned in either VC or VFC where tasks and applications from outside can be offloaded and executed at vehicles. These vehicles serve as computing nodes for their own benefits. This computing paradigm caters for the applications and tasks featured by strict delay requirements, since the response latency can be reduced drastically. However, due to limited computing and storing capabilities of individual vehicles, some applications which may require much more resources than individual vehicles can provide necessitate the execution and completion of them outside the vehicles, hopefully to mitigate the pressure on resource consumption of vehicles. Currently, cloud center and cloudlet are two promising solutions for application outsourcing. Cloud center usually has unlimited and low-price resources at the expense of relatively long response latency, whereas the cloudlet deployed at the edge of networks has short response latency at the cost of higher price and limited resources in comparison with the cloud center. Considering the mobility of vehicles in vehicular ad hoc networks (VANETs), it is a challenging task to deploy a cloudlet dedicated for application outsourcing.

However, combining VFC with V2I technologies, the road side unit (RSU) deployed along the road can be easily enhanced with more powerful computing and radio wireless resources to perform the functions of the cloudlet. Indeed, RSUs have already been empowered with more roles in VFC and mobile edge computing (MEC) [6], [7]. For example, RSU can intelligently improve the efficiency of wireless radio resource allocation when the spectrum resources are scheduled. The issue of resource allocation in both VC and VFC has been extensively investigated in the existing literature such as [5], [8], [9], [16]. We notice that most of the current literature focuses on the application execution efficiency such as response latency optimization and success rate improvement. Nevertheless, few of them have considered task scheduling and resource allocation from the perspective of resource providers (RPs) (e.g., cloudlet and RSU) in terms of benefits earned by contributing computing resources. An assumption is usually made that the resources can be utilized unconditionally, e.g., without paying RPs. However, in reality, most of the resources are provided in a pay-as-you-go pattern, which plays an important role in stimulating RPs to contribute the resources dutifully. In this paper, we investigate the resource allocation from the viewpoint of RSUs with a purpose of

maximizing the utility values of them, while considering the real-time requirements of vehicular applications.

The major contributions of this paper are summarized as follows.

- We propose an intermediate layer called RSU-enabled cloudlet between vehicular cloud and cloud center. RSU serves as an intelligent agent to perform vehicular applications and tasks with an aim to mitigate the overheads of vehicles. Moreover, compared to task execution in cloud center, the response delay can be drastically reduced. This merit makes it especially suitable for vehicular applications with strict delay requirements.
- An approximate but efficient resource allocation strategy is proposed that can intelligently respond to the resource requests in a real time fashion and optimize the utility value from the perspective of RSU-enabled cloudlet.
- Extensive experiments have been carried out to evaluate the resource allocation strategy. The simulation results show great potentials and prospects.

The rest of this paper is organized as follows. Related works on resource allocation in the contexts of ITS and vehicular cloud are reviewed in Section II. Three hierarchical layers in the RSU-enabled vehicular network are proposed and the problem of resource allocation is modeled and formulated in Section III. An algorithm is put forward to optimize the utility function and find approximate optimal solution in Section IV. A series of experiments are conducted in Section V, followed by the conclusion in Section VI.

II. RELATED WORKS

With the rapid development of information and communication technologies (ICTs), hundreds and even thousands of sensors are deployed within vehicles for event monitoring and data acquisition. Vehicle loaded computers and communication facility ensure that the gathered data can be processed and disseminated quickly. Against this background, intelligent transportation system incorporates such technologies and endeavours to provide better services for passengers in transportation systems [10]. Besides, the number of vehicular applications is also explosively increasing using mobile technologies. These vehicular applications sometimes require massive computing and communication resources, e.g., VR game [11]. Currently, an enormous number of works on resource allocation in ITS mainly fall into two categories – computing resources optimization and communication resources optimization, respectively.

A. COMPUTING RESOURCE RELATED OPTIMIZATION

Due to the limited computing capabilities of vehicles, the way that vehicular applications are outsourced is becoming more and more prevalent. Delay-tolerant applications and tasks can be offloaded to cloud center and executed. However, with the increasing number of delay-sensitive vehicular applications,

this way may not be applicable due to the long response delay caused by the backbone network congestion and delay. Accordingly, to reduce the response latency, mobile edge computing and vehicular fog computing have attracted extensive attention in ITS [7], [12]–[14], [14].

For example, complicated vehicular applications such as augmented reality (AR) have burdened vehicles in ITS in terms of computing resources. Xie *et al.* in [7] introduce the mobile edge computing to the traditional vehicular networks and attempt to enhance the existing vehicular networks. Specifically, they propose an architecture of collaborative vehicular edge computing network (CVECN) and present some technical challenges to enlighten researchers in this field. Considering a smart city scenario where the number of vehicular services is rapidly increasing, to jointly optimize caching, computing resources and networking, Si *et al.* in [12] present a vehicular network architecture which combines the software-defined networking paradigm. In this architecture both the evolved node Bs and on-board units (OBUs) can serve as edge nodes to provision caching and computing resources. The experimental results show that the approach outperforms the existing schemes.

The vehicular edge computing networks (VECNs) that combines the edge computing and vehicular networks has been considered to be a satisfactory solution to deal with the respond latency issue faced by diverse delay-sensitive vehicular applications. However, most of works focusing on VECNs do not consider the load balancing of computing resources. Accordingly, Zhang *et al.* in [13] strive to improve VECNs by a fiber-wireless technology. An algorithm is proposed to minimize the response latency when tasks are offloaded. Experimental results reveal that the approach can improve the computation resource utilization to a great extent.

Similarly, Wang *et al.* in [14] also apply the mobile edge computing to the vehicular networks, enabling vehicular applications to run by computation offloading while meeting the strict delay requirement. Specifically, they try to jointly optimize the computation and communication resources by a low-complexity algorithm. To minimize the transmission energy, the computation offloading problem is modeled as a graph cut problem in [15]. Some operators such as graph compressing and combing are conducted based on label propagation theory.

B. COMMUNICATION RESOURCE RELATED OPTIMIZATION

Wireless communication resources are needed in VANETs, e.g., for data delivery and information dissemination. To improve resource utilization, great efforts are made to optimize wireless channels [16]–[18] and improve the throughput of V2V and V2I wireless links [19], [20].

To realize efficient management of computing and communication resources in cloud-enabled vehicular networks, a coalition game model based approach is proposed in [17]. From the perspective of service providers, they strive to optimize the utility values of computing and bandwidth resources

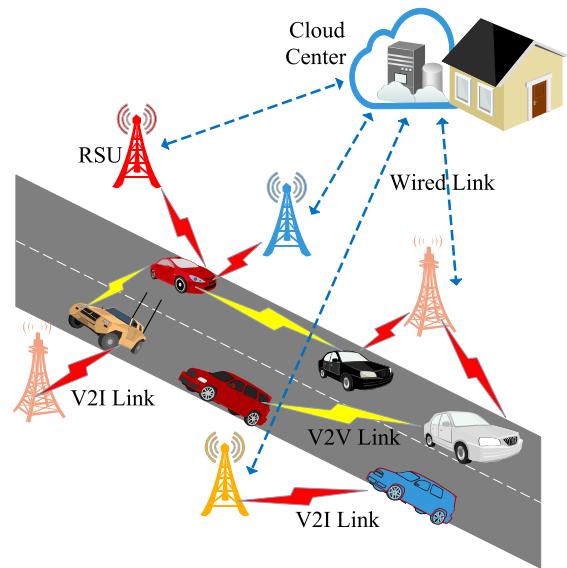


FIGURE 1. RSU-empowered vehicular network environment.

at the same time and the numeric results show the advantages of their approach. To shorten the response delay and transmission cost when tasks are offloaded, a cloud-based mobile edge offloading framework is put forward in the vehicular networks [18]. They investigate the efficiency of existing task offloading strategies and further present a novel approach that ensures tasks can be offloaded adaptively, e.g., either direct uploading or indirect using other entities acting as relay nodes.

He *et al.* in [19] propose a resource allocation strategy for the uplink vehicular communications networks and aim at optimize V2V throughput while meeting the minimum V2I link throughput requirements. A sub optimal algorithm is adopted to schedule the power among vehicles. To cope with the V2V offloading issues faced by the vehicular network, a software-defined network is introduced to the mobile edge computing architecture in [20]. The corresponding algorithm is put forward to detect whether a V2V path exists with aim to shift the communication traffic in cellular network to V2V links in VANETs.

III. SYSTEM MODEL

A. RSU-EMPOWERED VEHICULAR NETWORK

Fig. 1 shows an RSU-enabled vehicular network, where RSU has been enhanced with powerful computing and radio wireless resources to perform the functions of the cloudlet. Specifically, the network combines the advantages of VFC and VANETs and has three hierarchical layers, i.e., vehicular cloud (VC), RSU-enabled cloudlet (RC), and central cloud (CC), respectively.

1) VEHICULAR CLOUD

Vehicular cloud is actually a local cloud established based on VANETs. VC can guarantee a continuous connection to RC and CC. With the aid of neighboring vehicles, VC extends

the coverage range of RSU such that the success rate of data delivery or task offloading between vehicles and RSUs can be improved. More importantly, vehicular nodes can serve the requests from mobile terminals individually or cooperatively. The cooperative way pools together the individually idle computing and communication resources to deal with more sophisticated requests.

It is worthwhile mentioning that the number of vehicular applications has been explosively growing with the rapid development of ITS in recent years. The demands for computing, communication and storing resources in these in-car applications become more complicated and frequent, which renders that the limited capabilities of on-board computers within vehicles can hardly meet this kind of increasing demands, let alone idle resources sharing and contributing. As a consequence, VC turns to RC or CC for help, e.g., in the form of application outsourcing or task offloading.

2) RSU-ENABLED CLOUDLET

RSU-enabled cloudlet consists of neighboring RSUs using wired connections. As an intermediary layer between VC and CC, RSU-enabled cloudlet actually performs multiply important functionalities. First, comparatively speaking, RC has much more computing and communication resources than VC but relatively less resources than CC. In the traditional sensor-cloud systems, tasks and applications are outsourced to CC and executed. However, the access to CC causes long response delay due to the backbone network congestion and delay. Thus, it cannot satisfy those applications with both computationally intensive and time sensitive characteristics. RC brings the computing and communication resources to the edge of the networks, enabling these complex applications running while meeting these critical requirements. The outsourced applications can be executed in RC and the response latency can be drastically reduced.

Second, RC can serve as a gateway to CC from VC. On one hand, RSU in RC can communicate with vehicles in VC using the V2I communication technology; on the other hand, RSUs connect to the cloud center with wired links, which bridges the connections between VC and CC.

Third, further to the gateway role, RC can provision caching services when content-oriented applications are frequently served. For instance, RC can proactively cache those contents which are frequently requested by vehicular applications from the Internet. Thus, the popular contents can be retrieved in RC and directly delivered to the vehicular applications. As a result, the response latency can be reduced to a great extent. To improve the hit rate, artificial intelligence technologies such as deep learning and reinforcement learning can be applied to the caching schemes.

Last but not least, both the quality of service (QoS) and quality of experience (QoE) can be guaranteed in the sense that outsourcing services are provisioned in a pay-as-you-go model similar to CC. Better service quality means better benefits, which stimulates RC to improve the service provisioning.

TABLE 1. Notations.

Symbol	Descriptions
\mathcal{N}	The set of vehicles
\mathcal{A}	The set of vehicular vehicles
\mathcal{B}	The maximal communication resources that can be leased out in R
\mathcal{C}	The maximal computing resources that can be leased out in R
c_i	The computing resources required by application a_i
b_i	The communication resources required by application a_i
Δt_i	The estimated dwell time of vehicle i within the communication range of R
r_i	The fading rate of its wireless channel selected by a_i
P_i	The transmission power of i 's wireless communication facility
σ^2	The noise power
I_i	The interference caused by other applications using the same channel as a_i
w	The bandwidth of wireless channel
ρ	The processing speed of VM
p_c	The unit cost of computing resources
p_b	The unit cost of communication resources

3) CLOUD CENTER

The cloud center is a powerful infrastructure equipped with sufficient resources in the Internet. CC provides long-term rental services including IaaS, Paas, and SaaS at a low price. Cloud center is especially suitable for those applications and tasks characterized by complex computations and massive data storage demands. These kinds of applications are increasingly gaining popularity in connected vehicles such as VR game, automatic driving, and auto navigation. However, subject to the limited capabilities of vehicle loaded computer systems in VC, cloud center will hence become more and more important in the proposed vehicular networks.

B. PROBLEM FORMULATION

There is a set of $\mathcal{N} = \{1, 2, \dots, n\}$ vehicles and one RSU R in the RSU-empowered vehicular network. R can provision short-term rental computing and communication services to the in-car applications at a reasonable price. The set of applications is denoted by $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ with each element a_i representing the application running on the corresponding vehicle i . The applications considered in this model are of computationally intensive and time-sensitive characteristics. Therefore, these applications need to be outsourced to RC or CC for execution. Considering the high mobility of vehicles, each vehicular application is denoted by a triple $a_i = (c_i, b_{min,i}, \Delta t_i)$, where c_i represents the required computing resources, $b_{min,i}$ the estimated minimal communication resources, Δt_i the estimated dwell time of vehicle i within the coverage range of R , respectively. Assume that the total amount of computing resources and communication resources provided by R is \mathcal{B} and \mathcal{C} , respectively. In specific, if a_i from vehicle i is outsourced, it will consume at least $b_{min,i}$ communication resources and c_i computing resources within the specified time Δt_i . For the sake of convenient and clear expression, we list some key notations in Table 1 to be used through the rest of the paper.

Vehicle speed affects drastically the wireless channel quality, and great variation on wireless channels can incur serious fading effects [17]. Furthermore, the fading rate is linearly proportional to the vehicle speed when the speed is higher than a given threshold [21]. The value of the threshold actually depends upon the capabilities of the vehicle loaded communication facility. Therefore, given a threshold v_i^* and the current driving speed v_i for vehicle i , the fading rate of its wireless channel r_i can be given as:

$$r_i = \begin{cases} \delta \cdot v_i & v_i > v_i^* \\ 0 & v_i \leq v_i^* \end{cases} \quad (1)$$

where δ is a constant coefficient that can be predefined to adjust the fading rate. During the dwell time Δt_i , considering the fading effects of the wireless channel, the actual demand for communication resources can be calculated as:

$$b_i = b_{min,i} + \int_{t_{i,0}}^{t_{i,0}+\Delta t_i} r_i(v_i) dt \quad (2)$$

where $t_{i,0}$ denotes the time vehicle i starts to outsource a_i to R . Assume that a_i is offloaded to RC via a selected wireless channel. Then the signal-to-interference-plus-noise-ratio (SINR) at R can be defined as:

$$\gamma_i = \frac{P_i}{\sigma^2 + I_i} \quad (3)$$

where P_i denotes the transmission power of i 's wireless communication facility, σ^2 is the noise power and I_i is the interference arisen from other applications utilizing the same channel as a_i . Then the transmission rate based on Shannon theorem is given as:

$$r_{trans,i} = w \log_2(1 + \gamma_i) \quad (4)$$

where w is the bandwidth of wireless channel. Then the time taken to outsource the a_i from i to R is:

$$t_{trans,i} = \frac{b_i}{r_{trans,i}} \quad (5)$$

When R receives the offloaded task, it will check the current resources. If the remaining resources can meet the requirements of the request, it will immediately create a temporary virtual machine (VM) and allocate the required resources to respond to the request. Accordingly, the execution time of a_i can be given as:

$$t_{exe,i} = \frac{c_i}{\rho} \quad (6)$$

where ρ is the processing speed of VM. Thus, the total time for application outsourcing is

$$t_{total,i} = t_{trans,i} + t_{exe,i} \quad (7)$$

Due to the high mobility of vehicles, vehicle i may not be able to connect to R directly in Δt_i , so data delivery or result return should need other vehicles in the VANET to act as relay nodes for information forwarding, which greatly increase the probability of data loss issue. Thus, we assume

that the response latency from R should be within the dwell time Δt_i . Besides, we neglect the time expended on the result return, for the reason that compared to the size of offloaded data, the size of the result is usually negligible.

1) UTILITY FUNCTION

As mentioned earlier, we investigate the issue of resource allocation from the perspective of resource providers (i.e., R), with an aim to maximize its own benefits. Therefore, the utility function of R should take into account this aspect. Thus, when responding to the resource request from a_i , the utility function of R can be expressed as:

$$\mathcal{U}_i(a_i) = \alpha_i \log(1 + w_b \cdot b_i)(1 + w_c \cdot c_i) - p_c c_i - p_b b_i \quad (8)$$

where $w_c (0 \leq w_c \leq 1)$ and $w_b (0 \leq w_b \leq 1)$ denote the preferences towards computing and communication resources, respectively, and $w_c + w_b = 1$. Note that p_c and p_b denote the unit cost of computing and communication resources, respectively, and α_i denoting the satisfaction level on a_i can be defined as [17]

$$\alpha_i = \frac{\lambda}{1 + e^{-\beta(\vartheta - \mathcal{B} - \mathcal{C})}} \quad (9)$$

where λ and β are constants set by experience, $\vartheta = \max(b_j) + \max(c_k) (1 \leq j, k \leq n)$, that is the sum of the maximal computing and communication resources respectively from all the current requests. Obviously, for the RSU, the more the resources required, the higher their satisfaction level.

2) OBJECTIVE FUNCTION

Since the applications considered in this model are of computationally intensive and time-sensitive characteristics, two options are provided for their outsourcing and execution, namely, they can be served either in the RC or CC. Let ϕ_t be a binary variable to denote where a_i is offloaded. $\phi_t = 1$ if a_i is served at R and $\phi_t = 0$ if a_i is served at the cloud center. Given the indicator vector $\phi = (\phi_1, \phi_2, \dots, \phi_n)$, the overall value of utility function over \mathcal{A} can be expressed as

$$\mathcal{U}_{total}(\mathcal{A}) = \sum_{i=1}^n \phi_i \cdot \mathcal{U}_i(a_i) \quad (10)$$

Considering the constraints, our optimization objective \mathcal{P} can be expressed as:

$$\mathcal{P} : \text{Maximize } \mathcal{U}_{total}(\mathcal{A}) \quad (11)$$

s.t.:

$$t_{total,i} \leq \Delta t_i \quad \forall i \in \mathcal{N} \quad (12)$$

$$\sum_{i=1}^n \phi_i \cdot c_i \leq \mathcal{C} \quad (13)$$

$$\sum_{i=1}^n \phi_i \cdot b_i \leq \mathcal{B} \quad (14)$$

$$\phi_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \quad (15)$$

where inequation (12) ensures that the response latency of a_i should not exceed the dwell time of vehicle i . The inequations (13) and (14) guarantee that the total computing and communication resources to be leased out cannot exceed the respective resources which R can provide (i.e., \mathcal{B} and \mathcal{C}).

IV. SOLUTION TO UTILITY MAXIMIZATION

A. PRELIMINARIES

The problem \mathcal{P} can be modeled as a special knapsack problem and thus is of a NP-hard problem with the regards to the time taken to obtain an exact solution. To reduce time complexity, some heuristic algorithms such as genetic algorithm (GA) and particle swarm optimization (PSO) are often applied to seeking an approximate solution. However, considering the dynamics of vehicles in VANET, neither GA nor PSO is appropriate for the vehicular applications characterized by strict delay requirements. Accordingly, in this paper, we investigate the feature of utility function and propose a distance based heuristic algorithm to obtain an approximate solution.

Let's focus on the utility and further assume that c_i and b_i are continuous variables. We investigate the feature of the utility using Hessian matrix. First, we calculate the partial derivatives by differentiating $\mathcal{U}_i(c_i, b_i)$ w.r.t. c_i and b_i , respectively. The first partial derivatives are given as:

$$\begin{aligned}\frac{\partial \mathcal{U}_i(a_i)}{\partial c_i} &= \frac{\alpha_i \cdot w_c}{(1 + w_c \cdot c_i) \cdot \ln 2} - p_c \\ \frac{\partial \mathcal{U}_i(a_i)}{\partial b_i} &= \frac{\alpha_i \cdot w_b}{(1 + w_b \cdot b_i) \cdot \ln 2} - p_b\end{aligned}$$

The second partial derivatives are given as:

$$\begin{aligned}A &= \frac{\partial^2 \mathcal{U}_i(a_i)}{\partial c_i^2} = \frac{-\alpha_i \cdot w_c^2}{(1 + w_c \cdot c_i)^2 \cdot \ln 2} \\ B &= \frac{\partial^2 \mathcal{U}_i(a_i)}{\partial b_i \partial c_i} = 0 \\ C &= \frac{\partial^2 \mathcal{U}_i(a_i)}{\partial b_i^2} = \frac{-\alpha_i \cdot w_b^2}{(1 + w_b \cdot b_i)^2 \cdot \ln 2}\end{aligned}$$

Thus, the Hessian matrix can be constructed as

$$\mathbf{H}_{\mathcal{U}} = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$

Since $A < 0$ and $|\mathbf{H}_{\mathcal{U}}| = AC - B^2 > 0$, $\mathbf{H}_{\mathcal{U}}$ is a negative definite matrix and thus the maximal value of \mathcal{U} exists. Furthermore, by solving the equation set as follows

$$\begin{cases} \frac{\partial \mathcal{U}_i(a_i)}{\partial c_i} = 0 \\ \frac{\partial \mathcal{U}_i(a_i)}{\partial b_i} = 0 \end{cases}$$

we can obtain the maximal value as follows

$$(c^*, b^*) = \left(\frac{w_c \alpha_i - p_c \ln 2}{w_c p_c \ln 2}, \frac{w_b \alpha_i - p_b \ln 2}{w_b p_b \ln 2} \right) \quad (16)$$

where c^* and b^* denote the optimal values of computing and communication resources respectively when the maximal

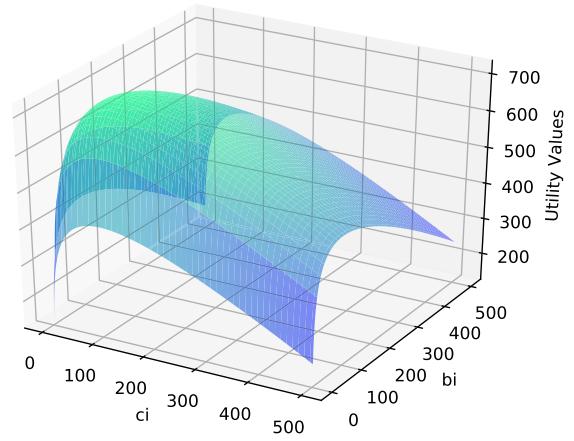


FIGURE 2. A generic distribution of $\mathcal{U}(c_i, b_i)$ with $\alpha_i = 100$, $w_b = w_c = 1$, $p_c = p_b = 1$.

value of utility function is achieved. Furthermore, the maximal value of utility function can be obtained as

$$\mathcal{U}^*(c_i, b_i) = \alpha_i \log \frac{w_b w_c a_i^2}{p_c p_b \ln^2 2} - p_c c_i - p_b b_i$$

Specifically, a generic distribution of $\mathcal{U}(c_i, b_i)$ is sketched in Fig. 2, where the x-coordinate represents the continuous variable c_i and the y-coordinate represents b_i . From Fig. 2 we can easily observe that (c^*, b^*) guarantees that only optimal value of $\mathcal{U}(c_i, b_i)$ can be obtained when $c_i > 0$ and $b_i > 0$. According to the limit definition, we have

$$\lim_{\substack{c_i \rightarrow c^* \\ b_i \rightarrow b^*}} \mathcal{U}(c_i, b_i) = \mathcal{U}^*(c^*, b^*)$$

which makes intuitive sense in that the closer to (c^*, b^*) w.r.t. the Euclidean distance, the closer to \mathcal{U}^* . This observation hence inspires us to propose a distance based heuristic to guide our solution searching when $c_i > 0$ and $b_i > 0$ are discretized.

B. ALGORITHM DESIGN

Based on the description above, we in this section propose a heuristic algorithm to solve problem \mathcal{P} . At the beginning, R periodically broadcasts the beacon information to the vehicles under the coverage of its communication range. The information includes the resources available to lease out and the corresponding prices (e.g., \mathcal{B} , \mathcal{C} , p_c and p_b). After receiving the beacon information, vehicles in vicinity that have vehicular applications to outsource will respond to R by sending the reply information. The reply information states their resource requirement and status information in the specified field. With a purpose of maximizing the total utility value while satisfying the resource requirements, R strives to make a real-time decision on which requests to accept and which to refuse. Specifically, the proposed algorithm performs its function for the decision making at this stage. The refused requests in R are assumed to be forwarded to CC directly. Since we focus

Algorithm 1 Distance Based Heuristic Algorithm for Utility Optimization (DbHA)

Input: $\mathcal{N}, \mathcal{A}, \mathcal{B}, \mathcal{C}, \alpha, p_c, p_b, w_c, w_b, P_i, \sigma^2, I_i$
Output: $\mathcal{U}_{total}(\mathcal{A})$

- 1 Initialize a heap h ;
- 2 $Sum_b = 0$;
- 3 $Sum_c = 0$;
- 4 $\mathcal{U} = 0$;
- 5 Calculate (c^*, b^*) based on Eq. (16);
- 6 **for** $i = 1$ to n **do**
- 7 Compute b_i using Eq. (2);
- 8 $d_i = \sqrt{(c_i - c^*)^2 + (b_i - b^*)^2}$;
- 9 Push (i, d_i) into h ;
- 10 **end**
- 11 **while** h not empty **do**
- 12 $h.pop(idx, d_{idx})$;
- 13 **if** $Sum_b + b_{idx} \leq \mathcal{B}$ and $Sum_c + c_{idx} \leq \mathcal{C}$ **then**
- 14 Calculate $t_{total, idx}$ based on Eq. (7);
- 15 **if** $t_{total, idx} \leq \Delta t_{idx}$ **then**
- 16 $Sum_b = Sum_b + b_{idx}$;
- 17 $Sum_c = Sum_c + c_{idx}$;
- 18 Calculate utility \mathcal{U}_{idx} based on Eq. (8);
- 19 $\mathcal{U} = \mathcal{U} + \mathcal{U}_{idx}$;
- 20 **end**
- 21 **end**
- 22 **end**
- 23 Return \mathcal{U} ;

on the utility optimization in RSU cloudlet, the application outsourcing and execution in CC are beyond the scope of our discussion in this paper. The algorithm (DbHA) is shown in Algorithm 1 with the procedure to be detailed as follows.

R can immediately obtain the best value pair (c^*, b^*) based on Eq. (16) after all the resources requests are received. For each request, the algorithm iteratively calculates the communication resources they truly need based on Eq. (2). Thus, the Euclidean distance d_i between pair $(c_i, b_i)(1 \leq i \leq n)$ and (c^*, b^*) can be obtained by $d_i = \sqrt{(c_i - c^*)^2 + (b_i - b^*)^2}$. To efficiently get the minimal distance every time, a Min-Heap h is used to store the distance of each vehicular application. Then, DbHA traverses h until it is empty. Each time a value is retrieved from h , the algorithm checks whether the corresponding application violates the constraints listed in inequations (12), (13) and (14). If all the constraint conditions hold, then R will accept this request, send confirmation message, allocate the corresponding resources, and update the related information such as the available resources and the total utility value (lines 16-19). In the end, the total utility value \mathcal{U} is returned.

DbHA takes $O(\log n)$ to initialize h and $O(n \log n)$ to traverse h so as to accomplish the utility optimization. Here, n denotes the number of vehicular applications. The total time complexity of DbHA is $O(n \log n)$, which is much smaller

than the evolutionary algorithms such as GA and PSO. The low time complexity makes DbHA specially suitable for the decision making with the strict delay requirement in the stage of request confirmation.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. STRATEGY COMPARISON

In this section, we investigate the efficiency and effectiveness of our approach compared to other three strategies – GA, PSO, and Greedy strategies.

GA. GA searches the feasible solutions over the potential solution space iteratively. The efficiency of GA is affected by multiple parameters involved in selection, crossover, mutation operators. We empirically set these parameters when comparing it with DbHA. To be more specific, we set the size of the population to 100, the crossover probability to 0.2, the mutation probability to 0.02, the number of iterations to 300, and the chromosome length to the number of vehicular applications.

PSO. As an evolutionary algorithm, PSO also strives to search the feasible solution over the population based solution space, which is featured by simple deployment and rapid convergence. Further to this, the number of parameters is also fewer than GA, which brings about shorter response delay to some extent. However, PSO is originally proposed to solve the continuous non-linear optimization problems in the continuous solution space. Thus, the original version is not suited for the binary problem optimization in this paper. To meet the requirement, we tailor PSO by using the Sigmoid function and $sig(x) = 1/(1 + e^{-x})$ where x represents the element in the velocity vector. A random variable $\rho(0 \leq \rho \leq 1)$ is used for assigning binary values to x . For instance,

$$x_{ij}^{t+1} = \begin{cases} 0 & \rho \leq sig(v_{ij}^{t+1}) \\ 1 & \rho > sig(v_{ij}^{t+1}) \end{cases}$$

where v_{ij}^{t+1} denotes the j th velocity component of particle i in the $(t + 1)$ th iteration and x_{ij} denotes the j th location component of particle i in the $(t + 1)$ th iteration. Besides, the parameters involved in PSO are set as follows. The inertia weight factor is set to 0.6. Both the cognitive and social learning coefficients are set to 2. The number of particles is 5 and the number of iterations is 300.

Greedy strategy. DbHA is actually a heuristically greedy algorithm, which preferentially responds to the resource requests with the shortest distance to (c^*, b^*) so far. Considering the feature of the utility function, we propose another heuristic rule to compare to DbHA. Intuitively, the larger the product of computing resources and communication resources (i.e., $c_i * b_i$), the larger the value of the first part of \mathcal{U} . Accordingly, we design another simple greedy rule to guide the resource allocation, i.e., the resource request with the largest value of $c_i * b_i$ is always responded first.

B. RESULT ANALYSIS

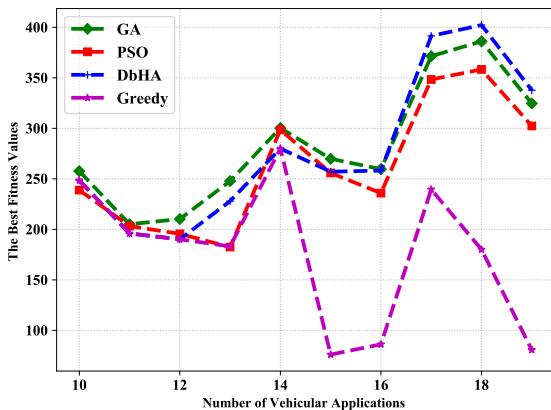
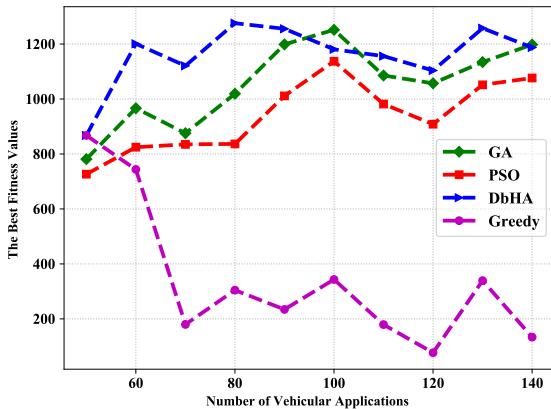
In the first set of experiments, we compare the efficiency of our approach with other three approaches above with regards

TABLE 2. Running time with small number of resource requests (seconds).

No. Strategies	10	11	12	13	14	15	16	17	18	19
GA	3.0201	2.8829	3.2363	3.4471	3.6241	3.8337	4.1544	4.2931	4.1295	4.7469
PSO	0.0809	0.0749	0.0863	0.0883	0.0967	0.1148	0.1023	0.1103	0.1087	0.1222
DbHA	0	0.0009	0.0010	0	0	0	0	0	0	0
Greedy	0.0010	0	0	0.0009	0	0	0.0010	0.0009	0	0

TABLE 3. Running time with large number of resource requests (seconds).

No. Strategies	50	60	70	80	90	100	110	120	130	140
GA	5.4732	6.3842	7.3776	8.5194	9.6751	10.4312	11.3179	12.6253	13.4005	14.8435
PSO	0.1914	0.1902	0.2330	0.2487	0.3025	0.3161	0.3716	0.3660	0.4257	0.4773
DbHA	0.0010	0.0015	0.0011	0.0008	0.0009	0.0008	0.0009	0.0013	0.0019	0.0019
Greedy	0.0009	0.0004	0.0020	0.0011	0.0010	0.0009	0.0009	0.0016	0.0009	0.0020

**FIGURE 3.** The performance comparison with small number of vehicular application requests.**FIGURE 4.** The performance comparison with large number of vehicular application requests.

to the best fitness values. The experimental results are shown in Fig. 3 and Fig. 4, where the x-coordinate represents the number of resource requests and the y-coordinate represents the best fitness values retrieved by these approaches. The difference between the experimental results shown in Fig. 3 and Fig. 4 lies in the scale of resource requests. For example, in Fig. 3 the number of vehicular applications ranges from 10 to 19, while in Fig. 4 the number of vehicular applications

ranges from 50 to 140 with a step of 10. When the number of resource requests is small as shown in Fig. 3, e.g., ranging from 10 to 14, we can find that the performance of DbHA, GA and PSO is almost the same, but these three approaches are much better than the greedy approach. On another hand, we can observe that DbHA performs the best and the Greedy approach performs the worst when the number of resource request is large. As for the evolutionary algorithms GA and PSO, the performance of them are both inferior to that of DbHA yet better than the Greedy approach.

What is interesting is that the fitness values are not always increasing with the increasing number of resource requests. For instance, in Fig. 4, the best values of the fitness function with the number of requests equal to 120 are smaller than the case with the number of requests equal to 110. However, two factors may cause this consequence. First, the data in the experiment is generated randomly such as the vehicular applications in each iteration, thus the vehicular applications are totally independent in different iterations. Second, considering the constraint conditions including the resource consumption and delay latency requirement, it is understandable and acceptable that the best fitness values are not always increasing as the number of vehicular applications increases.

To sum up, the performance of DbHA is not obviously better than GA or PSO when the number of requests is small. However, with the number of requests increasing, the advantage of DbHA becomes more and more apparent in terms of best fitness values retrieving.

Apart from this merit, DbHA is proposed to cater for the time sensitive characteristic of vehicular applications in RSU-empowered vehicular network. To investigate the time on decision making, the second set of experiments have been conducted in the next. Similar to the first set of experiments, the second one also takes into account two different scales of resource requests and the corresponding results are shown in Tables 2 and 3, respectively. From the results, we can easily make the following observations: (1) Regardless of the scales of resource requests, DbHA, Greedy approach and PSO can achieve almost the real-time response while GA responds to the requests and make decision on resource allocation in seconds. (2) GA is the worst among these approaches in

terms of response time and when the number of requests is large (e.g., ranging from 100 to 140), the response latency is more than ten seconds. GA actually evolves towards the best solution based on multiple operations on the population. Thus, it takes time to update the population frequently. In a word, this kind of response latency is unacceptable in the RSU-empowered vehicular networks. The vehicles may go out of the communication range of RSU with a high probability and the data delivery and result return tend to fail. The situation may be worsened when considering the unstable and intermittent wireless connections caused by interference and barrier obstruction. (3) The response latency of PSO can be acceptable. In contrast to GA, PSO has much shorter running time for the reason that both the deployment and the implementation are simple enough. However, the performance of PSO on utility value optimization may not be that effective in comparison with DbHA; (4) The Greedy approach may perform slightly better than DbHA w.r.t. the response latency. However, the performance on utility value optimization is the worst.

To sum up, GA is not appropriate for the resource allocation featured by strict response latency requirement. DbHA proposed in this paper shows great advantages over other three approaches in both respects (i.e., performance and response latency).

VI. CONCLUSION

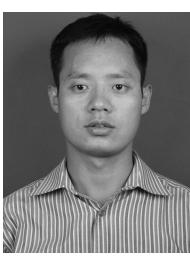
The number of vehicular applications is explosively increasing with the development of ITS. The computing and communication resources required for running these vehicular applications have posed great challenges for the limited capabilities of vehicle loaded computer systems and communication facility in ITS. We enhance RSU with powerful computing and wireless radio resources such that it can act as an intelligent agent to perform the functions of the cloudlet. For example, when the vehicular applications are outsourced to RSU for execution, RSU can decide both computing and communication resources allocation among these applications intelligently. To improve the efficiency of resource allocation, we have proposed a Euclidean distance based algorithm to optimize the utility values from the perspective of RSU cloudlet. The results have revealed that our approach outperforms other resource allocation strategies in terms of both performance and the response latency.

To realize efficient resource allocation in RSU-empowered vehicular network, more efficient and effective algorithms are needed, especially considering the high mobility of vehicles in VANETs. For the future work, we are planning to design efficient algorithms to cater for the flexible feature of vehicles in RSU-empowered vehicular network and further support vehicular applications with strict delay requirements.

REFERENCES

- [1] W. Zhang, Z. Zhang, and H.-C. Chao, "Cooperative fog computing for dealing with big data in the Internet of vehicles: Architecture and hierarchical resource management," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 60–67, Dec. 2017.

- [2] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2663–2668.
- [3] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *Int. J. Pervas. Comput. Commun.*, vol. 7, no. 1, pp. 7–21, Apr. 2011.
- [4] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Netw. Comput. Appl.*, vol. 40, pp. 325–344, Apr. 2014.
- [5] 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.
- [6] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Trans. Veh. Technol.*, early access, Jan. 31, 2020, doi: [10.1109/TVT.2020.2970763](https://doi.org/10.1109/TVT.2020.2970763).
- [7] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Collaborative vehicular edge computing networks: Architecture design and research challenges," *IEEE Access*, vol. 7, pp. 178942–178952, 2019.
- [8] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172–70185, 2018.
- [9] Y. Deng, Z. Chen, X. Yao, S. Hassan, and J. Wu, "Task scheduling for smart city applications based on multi-server mobile edge computing," *IEEE Access*, vol. 7, pp. 14410–14421, 2019.
- [10] Z. Li, R. Y. Fei, and Y. Wang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2018.
- [11] L. Hu, Y. Tian, J. Yang, T. Taleb, L. Xiang, and Y. Hao, "Ready player one: UAV-Clustering-Based multi-task offloading for vehicular VR/AR gaming," *IEEE Netw.*, vol. 33, no. 3, pp. 42–48, May 2019.
- [12] M. Li, P. Si, and Y. Zhang, "Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9073–9086, Oct. 2018.
- [13] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.
- [14] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. S. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62624–62632, 2019.
- [15] L. Dong, M. N. Satpute, J. Shan, B. Liu, Y. Yu, and T. Yan, "Computation offloading for mobile-edge computing with multi-user," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 841–850.
- [16] 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.
- [17] R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing, and Y. Zhang, "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7938–7951, Dec. 2015.
- [18] 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.
- [19] C. He, Q. Chen, C. Pan, X. Li, and F.-C. Zheng, "Resource allocation schemes based on coalition games for vehicular communications," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2340–2343, Dec. 2019.
- [20] C.-M. Huang, M.-S. Chiang, D.-T. Dao, W.-L. 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.
- [21] L. Cimini, "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," *IEEE Trans. Commun.*, vol. 33, no. 7, pp. 665–675, Jul. 1985.



CHAOGANG TANG (Member, IEEE) received the B.S. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, and the Department of Computer Science, City University of Hong Kong, through a joint Ph.D. program, in 2012. He is currently an Assistant Professor with the China University of Mining and Technology. His research interests include mobile cloud computing, fog computing, the Internet of Things, big data, and WSN.



CHUNSHENG ZHU (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Canada. He is an Associate Professor with the SUSTech Institute of Future Networks, Southern University of Science and Technology, China. He is also an Associate Researcher with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, China. He has authored more than 100 publications published by refereed international journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, *ACM Transactions on Embedded Computing Systems*, *ACM Transactions on Cyber-Physical Systems*, and the IEEE WIRELESS COMMUNICATIONS, magazines, such as the *IEEE Communications Magazine* and *IEEE Network Magazine*, and in conferences such as the IEEE INFOCOM, the IEEE IECON, the IEEE SECON, the IEEE DCOSS, the IEEE ICC, and the IEEE GLOBECOM. His research interests mainly include the Internet of Things, wireless sensor networks, cloud computing, big data, and social networks and security.



QING LI (Senior Member, IEEE) was the former Founding Director of the Multimedia Software Engineering Research Centre (MERC) and a Professor at the City University of Hong Kong, where he worked with the Department of Computer Science, from 1998 to 2018. Prior to these, he was also with The Hong Kong University of Science and Technology and Australian National University, Canberra, Australia. He is currently a Chair Professor (data science) and the Head of the Department of Computing, The Hong Kong Polytechnic University. He is currently a Fellow of the IET/IEE, and a member of ACM-SIGMOD and the IEEE Technical Committee on Data Engineering. He is the Chairperson of the Hong Kong Web Society, and has also served/is serving as an Executive Committee (EXCO) Member of the IEEE Hong Kong Computer Chapter and the ACM Hong Kong Chapter. In addition, he serves as a Councilor of the Database Society of the Chinese Computer Federation (CCF), a member of the Big Data Expert Committee of the CCF, and is a Steering Committee Member of DASFAA, ER, ICWL, UMEDIA, and the WISE Society.



XIANLIN WEI received the bachelor's degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2007, and the Ph.D. degree from the University of Science and Technology, Nanjing, in 2012. He is currently a Researcher with the National University of Defense Technology, Changsha, China. His research interests include cloud computing, peer-to-peer networks, network anomaly detection, network measurement, and distributed system design and optimization. He has served as an editorial board member of many international journals and a TPC member of several international conferences. He has also organized a few special issues for many reputable journals.



HUAMING WU (Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, China, in 2009 and 2011, respectively, and the Ph.D. degree (Hons.) in computer science from Freie Universität Berlin, Germany, in 2015. He is currently an Associate Professor with the Center for Applied Mathematics, Tianjin University. His research interests include model-based evaluation, wireless and mobile network systems, mobile cloud computing, and deep learning.



JOEL J. P. C. RODRIGUES (Fellow, IEEE) is a Professor with the Federal University of Piauí, Brazil, a Senior Researcher with the Instituto de Telecomunicações, Portugal, and a Collaborator of the Post-Graduation Program on Teleinformatics Engineering with the Federal University of Ceará (UFC), Brazil. He has authored or coauthored over 850 articles in refereed international journals and conferences, three books, and one ITU-T Recommendation. He holds two patents. He is the Leader of the Next Generation Networks and Applications (NetGNA) Research Group (CNPq), an IEEE Distinguished Lecturer, and a Member Representative of the IEEE Communications Society, IEEE Biometrics Council. He was the Director of conference development of the IEEE ComSoc Board of Governors, a Past-Chair of the IEEE ComSoc TCs on eHealth and Communications Software, and a Steering Committee Member of the IEEE Life Sciences Technical Community. He is the editor-in-chief of an international journal and an editorial board member of several journals.