IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 28, 2020, accepted February 17, 2020, date of publication February 20, 2020, date of current version March 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975310

Energy-Efficient Computation Offloading in Vehicular Edge Cloud Computing

XIN LI[1,2], **(Student Member, IEEE), YIFAN DANG**[3], **(Student Member, IEEE), MOHAMMAD AAZAM**[4], **(Senior Member, IEEE), XIA PENG**[5], **TEFANG CHEN**[5], **AND CHUNYANG CHEN**[5]

[1]School of Automation, Central South University, Changsha 410083, China
[2]Institute of Transportation Studies, University of California at Berkeley, Berkeley, CA 94806, USA
[3]Department of Computer and Information Science, University of Oregon, Eugene, OR 97403, USA
[4]Computer Science, Carnegie Mellon University, Doha 24866, Qatar
[5]School of Traffic and Transportation Engineering, Central South University, Changsha 410083, China

Corresponding author: Xin Li (lixin9206@gmail.com)

This work was supported by the Doctoral Student Overseas Study Program funded by the China Scholarship Council under Grant 201606370144.

ABSTRACT With the development of electrification, automation, and interconnection of the automobile industry, the demand for vehicular computing has entered an explosive growth era. Massive low time-constrained and computation-intensive vehicular computing operations bring new challenges to vehicles, such as excessive computing power and energy consumption. Computation offloading technology provides a sustainable and low-cost solution to these problems. In this article, we study an adaptive wireless resource allocation strategy of computation offloading service under a three-layered vehicular edge cloud computing framework. We model the computation offloading process at the minimum assignable wireless resource block level, which can better adapt to vehicular computation offloading scenarios and can also rapidly evolve to the 5G network. Subsequently, we propose a method to measure the cost-effectiveness of allocated resources and energy savings, named value density function. Interestingly, with respect to the amount of allocation resource, it can obtain the maximum value density when offloading energy consumption equals to half of local energy consumption. Finally, we propose a low-complexity heuristic resource allocation algorithm based on this novel theoretical discovery. Numerical results corroborate that our designed algorithm can gain above 80% execution time conservation and 62% conservation on energy consumption, and it exhibits fast convergence and superior performance compared to benchmark solutions.

INDEX TERMS Computation augmentation, computation offloading, energy conservation, resource allocation, vehicular edge computing.

I. INTRODUCTION

Along with the rapid research and development of Internet of Things (IoT) in the field of transportation, vehicles are undergoing tremendous changes. Connected and autonomous vehicles (CAVs) technology makes the vehicles converge abundant sorts of sensors, communication systems, and computing units. Large amounts of computing operations are executed on vehicles onboard computers nowadays, such as scene recognition, high-definition localization, and path planning. Besides, demands for vehicular computing will keep exponentially increasing with the development of high-level autonomous vehicles (i.e., Level 4 and Level 5 fully-autonomous driving) [1], [2].

Although vehicles are equipped with more and more resources, there are some unprecedented challenges in their long-term life, such as insufficient computing capability and excessive energy consumption [3], [4]. Some typical scenarios are described as follows:

1) When a large number of time-constrained and computation-intensive applications bring huge energy consumption to vehicles, how can endurance mileage, especially for electric vehicles, be ensured?

2) When the computing capability of formerly factory-installed onboard computers cannot keep up with the growing computing demands, how can consumer markets

The associate editor coordinating the review of this manuscript and approving it for publication was Constantinos Marios Angelopoulos.

cope with the large scale and high cost of hardware upgrade?

3) When the vehicle interconnection technology develops rapidly, how can massive data generated by vehicles at each moment be effectively managed, transferred, and stored?

All of the above problems are ultimately oriented to computing power, energy consumption, and interconnection of vehicles. Computation offloading technology can provide potential solutions to address the preceding problems [5]. Currently, there are three system solutions that can support vehicular computation offloading technology: vehicular cloud computing (VCC), vehicular fog computing (VFC), and vehicular edge computing (VEC).

VCC can obtain sufficient computing and storage capabilities by accessing conventional cloud computing [4]. However, cloud is usually distant, causing unacceptable amounts of delay in the case of vehicular/transport services. Additionally, transferring all data to cloud over public networks is not desirable (due to concerns such as privacy, security, and tremendous instantaneous communication data [6].) In such cases, VCC is not adequate for low-latency computations in vehicular computing.

Fortunately, a trend in cloud computing that relates to bringing cloud-like resources close to the underlying users has recently emerged. New technologies include mobile edge computing (MEC) [7] and fog computing (FC) [8], introduced by the European Telecommunications Standards Institute and Cisco, respectively. In that case, data transmissions and computing operations with low delay constraints can be reached in much lower latency than cloud computing [9], [10]. Considering the particularity of vehicles, in recent years, MEC and FC have been further studied in the vehicular environment as VEC and VFC, respectively. Generally, compute nodes of VEC are deployed in base stations installed with edge devices [11]–[14], but VFC adopts idle resources on nearby vehicles as its compute nodes [15]–[18].

Although VEC is derived from MEC, their differences are subtle but noticeable. We mainly consider the following three aspects: 1) VEC provides more specific services for intelligent transportation systems. Generally, MEC is applicable to IoT devices, whereas VEC is mainly deployed in traffic scenarios, such as vehicle terminal and roadside intelligent infrastructure. 2) In resource allocation management, VEC-based computation offloading has more stringent requirements for delayed management and energy efficiency. 3) In terms of mobility, VEC is more compatible with various types of moving speed than MEC.

In this paper, we expand VEC to a three-layered system framework, i.e., vehicular edge cloud computing (VECC). It has more system resources than traditional VEC to provide abundant vehicular services for CAVs, such as perception, computing, storage, and communication. Notably, since wireless communication paramountly affects the whole system quality of service (QoS) (e.g., system energy efficiency, time efficiency, and amount of serving users), our research work mainly focuses on wireless resource allocation strategy. In terms of resource allocation model, we use the minimum assignable resource block (RB) as the resource allocation granularity, so as to improve the performance and allocation efficiency of resource management. In addition, considering different user mobility, we take the mobility rate as one of the reference variables and consider the impact of spectrum drift on resource efficiency under different moving speeds.

II. RELATED WORK

VEC and computation offloading became a hot topic in recent years. In [6], a vehicular computation offloading method was proposed to realize multi-objective optimization to reduce the execution time of the computing tasks and the energy consumption of the edge device while satisfying the privacy conflicts of the computing tasks. In [19], Gu et al. used matching theory to develop a distributed and context-aware task assignment mechanism for VEC. Two heuristic algorithms were proposed to minimize the system delay. Pu *et al.* [20] proposed a VEC named Chimera to augment network-wide vehicle resources for future large-scale vehicular crowdsensing applications by leveraging cooperative vehicles and VM pool. An online task scheduling algorithm was designed for efficient task workload assignment in edge cloud. Zhao *et al.* [21] studied computing resources allocation scheme for vehicular computation offloading based on VEC and cloud collaboration framework. Its offloading network uses IEEE 802.11p. Dai *et al.* [14] also considered the computation offloading algorithms in VEC with IEEE 802.11p network and tried to maximize the system utility. In [22], Du et al. studied vehicular computation offloading in the TV white space band and proposed an algorithm to tackle a dual-side optimization problem to minimize the cost of VTs and VEC server. Zhang *et al.* [23] proposed a cloud-based VEC. A contract-based offloading and computation resource allocation scheme was proposed to maximize the utility of the MEC service provider. Zhou *et al.* [24] studied the vehicular workload offloading problem and proposed a low-complexity distributed solution to determine the optimal portion of workload to be offloaded based on the dynamic states of energy consumption and latency in local computing, data transmission, workload execution and handover. Resource allocation for a multiuser MEC system based on both time-division multiple access and orthogonal frequency-division multiple access was studied in [25], which optimization objective is to minimize the weighted sum energy consumption for mobile users.

Considering the practicality and broader coverage of cellular communication technology, we adopt cellular networks as the main communication technology in VECC. For the cellular network resource allocation solutions, Peng *et al.* [26] studied downlink spectrum resource management for VEC, which jointly considered spectrum slicing in base stations, resource allocation among vehicles, and transmission power controlling among WiFi access points. Mao *et al.* [27] proposed a low-complexity online algorithm that jointly decides the offloading decision, the CPU-cycle frequencies for mobile execution, and the transmission power for computation offloading. The objective was minimizing long-term average execution cost in MEC. Ning *et al.* [28] put forward a hybrid VEC for real-time traffic management in 5G networks. They formulated a joint task distribution, sub-channel assignment, and power allocation problem, with the objective of maximizing the sum offloading rate.

Different from existing resource allocation solutions in MEC or VEC, we take the particularity of vehicles into consideration. We study a stricter low-latency and superior performance requirements resource management at the RB level, in addition to mobility. The RB is the minimum assignable wireless resource in 4G LTE and 5G networks. Therefore, the LTE research model in this paper can be quickly evolved into the 5G network. By means of the value density function proposed in this paper, we find that there is a special mathematical relationship between the allocation of RB and the value density. That is, when the energy consumption of computation offloading reaches half of the local computing, the largest value density can be obtained. Through this innovation, we design a low-complexity value density heuristic algorithm, which can achieve fast convergence and superior performance for vehicular computation offloading compared to benchmark solutions.

The main contributions of this work can be summarized as follows.

- We propose a novel three-layer VECC framework that can support real-time computation augmentation, energy conservation, and interconnection for massive time-constrained tasks of CAVs at the edge of the network. VECC has more vehicle-related system resources than traditional VEC, which can provide abundant vehicular services for CAVs and intelligent transportation systems.
- Based on the VECC framework, we first model the vehicular computation offloading process at the minimum assignable resource block level. This model supports stricter resource management requirements for vehicles and more efficient energy consumption management. In addition, this model is able to adapt 5G network with few modifications.
- We propose a value density function to measure the cost-effectiveness of allocated resources and energy consumption. Moreover, we propose a high-precise but low-complexity resource allocation algorithm based on the maximum available value density, namely, Maximum Value Density based Heuristic Allocation (MVDHA). Extensive simulations demonstrate the convergence, fairness, and effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. Section III introduces a three-layered VECC system framework. Section IV describes the vehicular computation offloading model in VECC. Section V presents the formulated resource allocation problem for multi-users computation offloading and methodology. Section VI designs a maximum value density empowered heuristic algorithm. Section VII exhibits extensive simulation results to demonstrate the performance of the resource allocation scheme. Finally, Section VIII concludes this paper.

III. VECC SYSTEM FRAMEWORK

As illustrated in Fig. 1, VECC system architecture consists of three main layers: perception layer, middleware layer, and application layer.

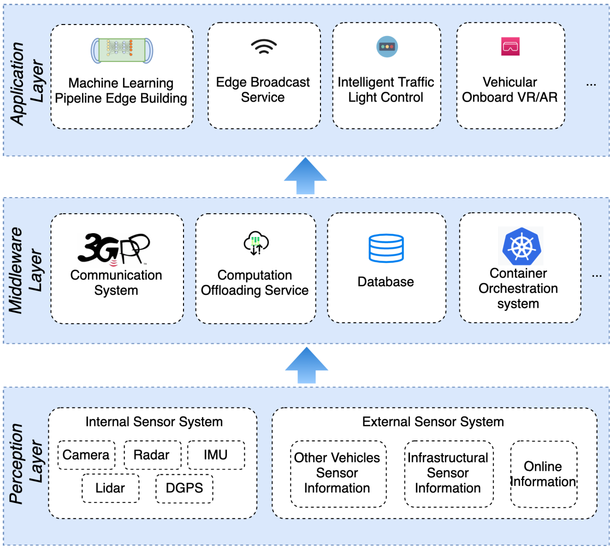


FIGURE 1. VECC System framework.

The bottom perception layer includes an internal sensor system and an external sensor system. The current autonomous vehicle technology mainly focuses on the internal sensor perception, which includes cameras, millimeter wave radar, Lidar, etc. The external sensor perception is a significant component of the future connected autonomous vehicle solution, which will provide extended sensor information from nearby vehicles, infrastructural sensors, and online information. Such a cooperative perception with both internal sensor system and external sensor system can augment perception and situation awareness capability, which can contribute to a better autonomous vehicle in terms of decision-making and planning [29].

The middleware layer includes communication system, computation offloading service system, database, container orchestration system, etc. It mainly supports communication, computation offloading, data collection, processing, and storage.

The upper application layer can be developed based on VECC private libraries or third-party public sources. Some typical applications include: 1) machine learning pipeline edge building, that can provide rapid model parameters training or pipeline architecture building at the edge of network; 2) edge broadcast service, that can broadcast various kinds of information, which includes but not limited to road network congestion information and infrastructure sensor information, in a neighborhood area; 3) intelligent traffic light control, that can achieve higher connected and intelligent traffic light management in not only in connected corridor level but also in the smart city; 4) real-time vehicular onboard VR/AR, that can provide immersive human-vehicle interaction services. For more examples, various kinds of applications mentioned in [4] can be implemented, as well. Furthermore, some of the VFCC applications can also be implemented in the VECC, such as the driver behavior detection system based on fog computing proposed by Aazam *et al.* in [30].

IV. SYSTEM MODEL

In this section, we introduce the RB-based model of the computation offloading service (COS) under the VECC framework.

Without loss of generality, the following two assumptions are made for the model: (1) We assume that the users request services within the VECC service area. (2) Since the offloading tasks we considered are with low-latency constraint (usually from several to hundreds of milliseconds), it will not arise handover or intense signal-to-noise ratio (SNR) fluctuation during offloading by default.

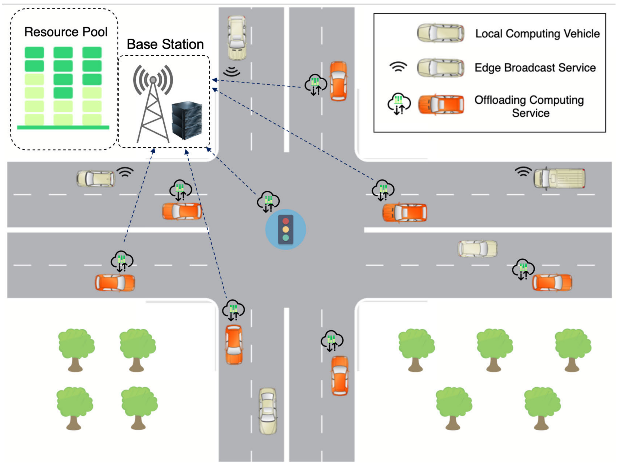


FIGURE 2. VECC service scenario.

Fig. 2 illustrates a representative VECC service scenario. In the upper left corner, a base station equipped with an edge cloud server is providing services for nearby users, including edge broadcast and computation offloading. Edge broadcast service can serve information sharing of Basic Safety Message (BSM) in V2X technology, and offloading computing services can support offloading computing tasks (such as sensor fusion, predictive planning, on-board VR, and AR) of vehicles and roadside infrastructures to the vehicular edge cloud.

This paper focuses on the uplink wireless resource allocation management of VECC for multi-user service requests. A user set is noted as $\mathcal{N} \triangleq \{1, \ldots, i, \ldots, N\}$. The computing task of user i can be described as $\Gamma_i \triangleq \{\delta_i, \beta_i, \gamma_i\}$.

TABLE 1. Main notations.

Symbols	Description
N	Number of Computation Offloading Service requests
B	Wireless network bandwidth
δ_i	Uploading task size
β_i	Computation units required
γ_i	Time latency [a]
a_i	Number of resource block allocated to user *i*
SNR_i	Signal to noise ratio of user *i*
$\langle r(a_i)\rangle$	Average uplink data rate of user
λ	Subcarrier per resource block
ϖ	Subcarrier spacing
τ_i^v	Throughput percentage of user *i* at speed v
P_i	Average transmission power of user *i*
f_i^l	Vehicular local computing ability for specific task
f_i^o	Edge cloud computing ability
σ_i	Energy consumption per computing unit
e_i^l, t_i^l	Local energy consumption, and local time consumption
e_i^o, t_i^o	Offloading energy consumption, and offloading time consumption

[a] In this article, time latency constraint is equal to the local computing time consumption, which can ensure the computation augmentation of computation offloading.

The δ_i, β_i and γ_i denote the size of computing offloading data (e.g., the program codes or sensors data sets), the total required computing units (e.g., CPU Cycles in CPU computation or GFlops in GPU computation) and the maximum latency allowed to accomplish computing, respectively. The main notations in the model are shown in Table 1.

A. LOCAL COMPUTING MODEL

When the task of user i is executed locally (i.e., vehicular onboard computer), the energy consumption e_i^l and local execution time t_i^l can be obtained by (1) and (2), respectively.

$$e_i^l = \sigma_i \beta_i \tag{1}$$

$$t_i^l = \frac{\beta_i}{f_i^l} \tag{2}$$

The σ_i in (1) denotes the energy consumption per computing unit. The f_i^l in (2) denotes the vehicle onboard computing ability of vehicular user i.

B. EDGE COMPUTING MODEL

In this model, we investigate the resource allocation scheme in the physical resource block (RB) level. RB is the smallest unit for resource allocation in LTE and 5G network [31]. Each resource block consists of λ subcarriers, and in each subcarrier, it occupies ϖ kHz bandwidth.

We can further calculate the average uplink data rate $\langle r(a_i)\rangle$ of vehicular user i (in the unit of kb/s) based on the Shannon-Hartley capacity as follows:

$$\langle r(a_i)\rangle = \lambda \varpi \tau_i^v a_i \log_2(1 + SNR_i) \tag{3}$$

In (3), τ_i^v denotes the average throughput percentage of vehicular user i at speed v. a_i denotes the number of RBs allocated to user i. SNR_i denotes predicted average signal-to-noise ratio of vehicular user i during computation offloading process, which can be obtained with related works [32], [33].

Considering the capacity of the downlink is much higher than the uplink in cellular network, the computation results are also usually smaller than the computation input data (e.g., program code, and sensing datasets) [34]–[36]. Therefore, we only consider the uplink process in our model. Such method is widely utilized in related research works.

When task i is executed remotely, the overhead consists of offloading energy consumption e_i^o and offloading execution time consumption t_i^o. e_i^o is only composed of the data transmission energy consumption generated in communications. Furthermore, t_i^o is composed of two parts, $t_i^{o,t}$ is generated in data transmission, while the other part $t_i^{o,e}$ is in the remote execution at edge server. The offloading energy consumption e_i^o, and total offloading time consumption t_i^o can be obtained through (4)-(7).

$$e_i^o = \frac{\delta_i P_i}{\langle r(a_i)\rangle} \tag{4}$$

$$t_i^{o,t} = \frac{\delta_i}{\langle r(a_i)\rangle} \tag{5}$$

$$t_i^{o,e} = \frac{\beta_i}{f_i^o} \tag{6}$$

$$t_i^o = \frac{\delta_i}{\langle r(a_i)\rangle} + \frac{\beta_i}{f_i^o} \tag{7}$$

P_i in (4) denotes the average transmission power during computation offloading. f_i^o in (6) denotes the computation capability that the edge cloud allocated to user i. In this paper, we assume that the computation capability in the edge cloud is sufficient, and each offloading user can be assigned f_i^o computing resource.

V. PROBLEM FORMULATION AND METHODOLOGY

In this section, we consider energy-efficient and computation augmentation for COS. The optimization objective is to minimize the overall system energy consumption by complying with the Beneficial Decision policy, which is defined as follows:

Definition 1: The **beneficial decision** represents that an offloading task can obtain not only lower energy consumption than local computing but also shorter accomplishment time than time latency constraint.

It is worth noting that in this paper, we set time latency tolerance to be the same as local execution time by default. Therefore, computation augmentation can be implemented while offloading computing tasks.

A. ENERGY COST MINIMIZATION PROBLEM

We formulate an initial optimization function as follows:

$$(P1): \quad \min \quad (\sum_{i=1}^{N} e_i) \tag{8a}$$

$$s.t. \sum_{i=1}^{N} a_i \leq M \tag{8b}$$

$$e_i^l \geq e_i^o \tag{8c}$$

$$\gamma_i \geq t_i^o \tag{8d}$$

$$\forall i \in \mathcal{N}, \forall a_i \in \mathbb{Z}^+, \forall e_i \in \{e_i^l, e_i^o\} \tag{8e}$$

In (P1), the first constraint (8b) guarantees that the sum of allocated resource blocks cannot exceed the total resource amount M, which is preset by the network bandwidth parameter. Constraints (8c) and (8d) express compliance with Beneficial Decision criteria, i.e., the reduction of energy consumption and execution time, respectively.

A variety of elements affect the consumption of energy and time (e.g., computing ability, time latency, wireless resource, and computer energy efficiency). Therefore, how can the resource management achieve an overall system optimal resource allocation scheme for all service requests at each decision-making slot? We then introduce two key metrics: energy equilibrium point and time equilibrium point.

B. MINIMUM REQUIRED RESOURCE

To precisely describe the decisions of all the users, we use vectors to represent the decisions of the users in the rest of this paper.

Lemma 1: There is an energy equilibrium point ${a_i}^* := \frac{\delta_i P_i}{\lambda \varpi \tau_i^v \sigma_i \beta_i \log_2(1+SNR_i)}$. When the allocation resource amount is larger than this point, user i can achieve energy benefit by using the COS (i.e., $e_i^o \leq e_i^l$).

Proof: If we expand the inequality (9c) in (P1), we will have:

$$e_i^l \geq e_i^o \implies \sigma_i \beta_i \geq \frac{\delta_i P_i}{\langle r(a_i)\rangle} \implies \langle r(a_i)\rangle \geq \frac{\delta_i P_i}{\sigma_i \beta_i}$$
$$\implies a_i \geq \frac{\delta_i P_i}{\lambda \varpi \tau_i^v \sigma_i \beta_i \log_2(1 + SNR_i)}$$

■

Lemma 2: There is a time equilibrium point ${a_i}^\Delta := \frac{\delta_i}{\lambda \varpi \tau_i^v (\gamma_i - \frac{\beta_i}{f_i^o}) \log_2(1+SNR_i)}$. When the allocation resource amount is larger than this point, user i can achieve computation augmentation by using COS (i.e., $\gamma_i \geq t_i^o$).

Proof: If we expand the inequality (9d) in (P1), we will have:

$$\gamma_i \geq t_i^o \implies \gamma_i \geq \frac{\delta_i}{\langle r(a_i)\rangle} + \frac{\beta_i}{f_i^o} \implies \langle r(a_i)\rangle \geq \frac{\delta_i}{\gamma_i - \frac{\beta_i}{f_i^o}}$$
$$\implies a_i \geq \frac{\delta_i}{\lambda \varpi \tau_i^v (\gamma_i - \frac{\beta_i}{f_i^o}) \log_2(1 + SNR_i)}$$

■

Theorem 1: There is a beneficial decision vector $A^\omega \triangleq \{{a_1}^\omega, \ldots, {a_i}^\omega, \ldots {a_n}^\omega\}$, which can enable each user in the user set $\mathcal{N}$ achieve beneficial decisions.

Proof: From Lemma 1 and Lemma 2, we can get the energy equilibrium point and time equilibrium point, respectively. In such a case, the element ${a_i}^\omega$ at the decision vector A^ω can be obtained as follows:

$${a_i}^\omega = \max(\lceil a_i^* \rceil, \lceil a_i^\Delta \rceil) \tag{9}$$

The symbol $\lceil\bullet\rceil$ denotes rounding up to the next integer function. When the controller allocates at least $\lceil a_i^* \rceil$ RBs to user i, user i will benefit from VECC by costing less (or at least the same) energy consumption than the local computing. Simultaneously, when the controller allocates at least $\lceil a_i^{\Delta} \rceil$ RBs for user i, user i will benefit from VECC by costing less time consumption than its task time latency. Therefore, if we choose to allocate the larger one between $\lceil a_i^* \rceil$ and $\lceil a_i^{\Delta} \rceil$, user i will definitely benefit from COS of VECC on both energy and time conservation. ■

C. TRANSFORMATION OF OPTIMIZATION PROBLEM

In the VECC model, the decisions made at each decision-making slot can be mathematically expressed as two decision variables: 1) the decision-making between local and offloading; 2) the amount of allocation resource for each offloading task.

Therefore, we define two new variable vectors: $\Phi \triangleq \{\varphi_1, \ldots, \varphi_i, \ldots, \varphi_n\}$, which denotes the decision (i.e., local or offloading) set of all the vehicular users; $A \triangleq \{a_1, \ldots, a_i, \ldots, a_n\}$, which represent the amount of resource for each vehicular users. Specifically, $\forall \varphi_i \in \{0, 1\}$, and $\forall a_i \in \mathbb{Z}^+$

Then the (P1) can be literally transformed into (P2).

$$(P2): \min_{\{A,\Phi\}} \left(\sum_{i=1}^{N} \left(\varphi_i e_i^o + (1-\varphi_i) e_i^l \right) \right) \tag{10a}$$

$$s.t. a_i \geq a_i^{\omega} \tag{10b}$$

$$\sum_{i=1}^{N} (\varphi_i a_i) \leq M \tag{10c}$$

$$a_i^{\omega} \in A^{\omega}, a_i \in A, \varphi_i \in \Phi \tag{10d}$$

$$\forall i \in \mathcal{N}, \forall a_i \in \mathbb{Z}^+, \forall \varphi_i \in \{0, 1\} \tag{10e}$$

If we expand (10a), we will get (P3), as shown in follows:

$$(P3): \sum_{i=1}^{N} \left(e_i^l \right) - \max_{\{A,\Phi\}} \left(\sum_{i=1}^{N} \left(\varphi_i \left(e_i^l - e_i^o \right) \right) \right) \tag{11a}$$

$$s.t. same\ as\ (P2) \tag{11b}$$

Because $\sum_{i=1}^{N} (e_i^l)$ is a constant, so only the rear part of (P3) is needed to be considered. The (P3) can be further transformed into (P4).

$$(P4): \max_{\{A,\Phi\}} \left(\sum_{i=1}^{N} \left(\varphi_i \left(e_i^l - e_i^o \right) \right) \right) \tag{12a}$$

$$s.t. same\ as\ (P2) \tag{12b}$$

In the next subsection, we will prove that the literal transformation of (P1), i.e., (P4), is NP-hard.

D. NP-HARD PROOF OF THE RESOURCE ALLOCATION OPTIMIZATION PROBLEM

Theorem 2: Solving the optimization problem (P4) is an NP-hard problem.

Proof: If we simplify one of the elements in (P4) by defaulting $A = A^{\omega}$. (P4) can be reduced to the optimization problem (P5), as follows:

$$(P5): \max_{\{\Phi\}} \left(\sum_{i=1}^{N} \left(\varphi_i \left(e_i^l - e_i^o \right) \right) \right) \tag{13a}$$

$$s.t. \sum_{i=1}^{N} (\varphi_i a_i) \leq M \tag{13b}$$

$$a_i \in A^{\omega}, \varphi_i \in \Phi, \quad \forall i \in \mathcal{N}, \forall \varphi_i \in \{0, 1\} \tag{13c}$$

It is not hard to find out that the (P5) is a transformation of a well-known NP-Complete problem — Knapsack Problem (KP), which tries to maximize the knapsack value by putting different items into the knapsack with a constraint of knapsack capacity. In (P5), we can assume the $(e_i^l - e_i^o)$ as the value of item i, and the φ_i denotes the deciding factor of whether put the item i into the package. The M is the capacity of the knapsack. In such cases, if we denote the optimization problem (P4) as P_A, the optimization problem (P5) as P_B, and the Knapsack Problem as P_C, we will have that $P_B = P_C \prec_p P_A$, which literally means the P_B is the same as the knapsack problem P_C, and P_C is polynomial-time reducible to optimization problem P_A, so the optimization problem P_A is an NP-hard problem [37]. ■

Definition 2: The **value density** is a ratio of the amount of energy conservation to the amount of allocated resource, which can be formulated as follows:

$$\rho_i = \begin{cases} \dfrac{e_i^l - e_i^o}{a_i}, & \text{for}(e_i^l - e_i^o) \geq 0 \quad (14a) \\ 0, & \text{for}(e_i^l - e_i^o) < 0 \quad (14b) \end{cases}$$

Inspired by the greedy heuristic algorithm in KP, the value density can be considered as the same meaning as the cost-performance ratio in KP, which equals to the item value divided by the item weight. Notably, we need to let the $(e_i^l - e_i^o)$ be greater than 0 to make the value density meaningful.

E. THE HIGHEST AVAILABLE VALUE DENSITY

In this article, we adopt a heuristic greedy method to allocate the highest available value density to users. We then involve two new decision vectors: 1) $A^{\hbar} \triangleq \{a_1{}^{\hbar}, \ldots, a_i{}^{\hbar}, \ldots a_n{}^{\hbar}\}$, which represents the highest value density vector; 2) $A^{\varsigma} \triangleq \{a_1{}^{\varsigma}, \ldots, a_i{}^{\varsigma}, \ldots a_n{}^{\varsigma}\}$, which represents the highest available value density vector.

Theorem 3: The user will obtain the highest value density when the offloading energy consumption is equal to half of local energy consumption.

Proof: The statement can be found at the appendix. ■

According to the appendix, the final differential formula of (15a)-(15b) can be formulated as follows.

$$\frac{\partial \rho_i}{\partial a_i} = \begin{cases} \dfrac{2e_i^o - e_i^l}{a_i{}^2}, & \text{for}(e_i^l - e_i^o) \geq 0 \quad (15a) \\ 0, & \text{for}(e_i^l - e_i^o) < 0 \quad (15b) \end{cases}$$

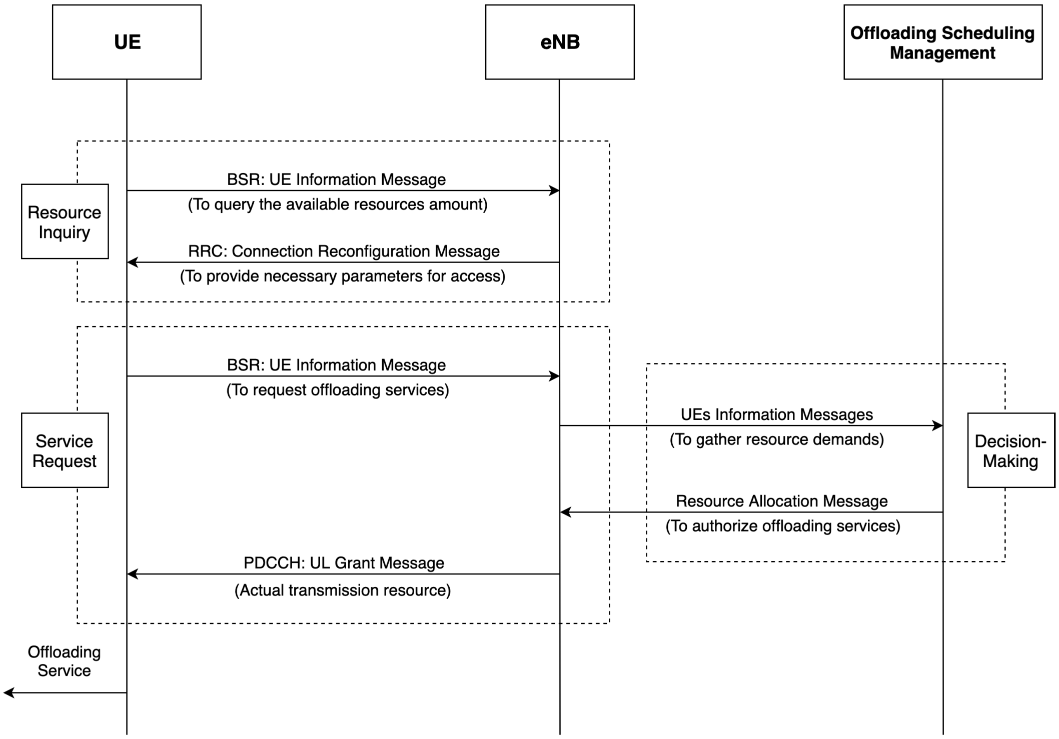

UE
eNB
Offloading Scheduling Management
Resource Inquiry
BSR: UE Information Message
(To query the available resources amount)
RRC: Connection Reconfiguration Message
(To provide necessary parameters for access)
Service Request
BSR: UE Information Message
(To request offloading services)
UEs Information Messages
(To gather resource demands)
Decision-Making
Resource Allocation Message
(To authorize offloading services)
PDCCH: UL Grant Message
(Actual transmission resource)
Offloading Service

FIGURE 3. The resource allocation processing flow of VECC computation offloading.

Corollary 1: The highest value density point ${a_i}^{\hbar} := [\frac{2\delta_i P_i}{\lambda\varpi\tau_i^{v}\varphi\beta_i\log_2(1+SNR_i)}]$.
Where $[\bullet]$ denotes the rounding function which obtains the nearest integer. *Proof:* According to the Theorem 3, we can expand the highest value density condition as follows:

$$2e_i^o - e_i^l = 0 \implies \frac{2\delta_i P_i}{\langle r(a_i)\rangle} = \sigma_i\beta_i \implies \langle r(a_i)\rangle = \frac{2\delta_i P_i}{\sigma_i\beta_i}$$
$$\implies a_i = \frac{2\delta_i P_i}{\lambda\varpi\varphi\beta_i\log_2(1+SNR_i)}$$

Because the resource block can only be allocated in integer, the rounding function can ensure the obtained point is the closest integer to the highest value density point. ■

Theorem 4: The highest available value density point ${a_i}^{\varsigma} := \max({a_i}^{\omega}, {a_i}^{\hbar})$.

Proof: Considering the constraint in the minimum required resource amount, the maximum available value density decision element can be obtained by choosing the larger one between ${a_i}^{\omega}$ and ${a_i}^{\hbar}$. ■

VI. ALGORITHM DESIGN

In this section, we introduce the resource allocation process and MVDHA algorithm.

The resource allocation algorithm complies with the existing 3GPP protocol standards, and uses Buffer Status Report (BSR), Radio Resource Control (RRC) and UpLink (UL) Grant to configure transmission parameters and message formats.

As shown in Fig. 3, resource allocation processing in VECC mainly includes three parts: Resource Inquiry, Service Request, and Decision-Making.

First, users (UEs) send BSR messages to the eNB to query the remaining resources, and the eNB performs message feedback through RRC messages. When the resource margin meets the minimum requirements of the UE, the UE will enter the Resource Inquiry phase. Then the qualified users send another BSR information for the service request. After that, the eNB sends multiple user requests received in the same time slot t to offloading scheduling management (OSM), where the offloading decision and resource allocation are performed. The UEs who are finally approved to offload will receive the UpLink (UL) grant through the Physical Uplink Control Channel (PUCCH), and then obtain offloading services.

As shown in algorithm 1, the UE first sends a resource query request to the eNodeB. When the remaining resources of the eNodeB are greater than the required resources (i.e., $R(t) \geq {a_i}^{\omega}$), the UE initiates a computation offload request to eNodeB. If approved, then starts the offloading service.

Algorithm 2 runs at the eNodeB side to make decisions and allocate resources. Initially, radio resource management (RRM) of eNodeB initializes the remaining resources $R(t)$ at time slot t. Then enter the decision-making and resource allocation stage. OSM preprocess and sort the request messages received by the slot t. When the allocable resources are greater than the minimum requirement of users,

Algorithm 1 Resource Allocation Algorithm_User Side

```
Initialization:
initialize each user individual decision profile.
End initialization.

send a BSR to eNodeB to query the available resource amount
if $R(t) \geq a_i^{\omega}$ then
    send a BSR to eNodeB to request offloading services.
    if receive a UL Grant then
        $a_i \leftarrow a_i^{receive}$ and $\varphi_i \leftarrow 1$.
    else
        $a_i \leftarrow 0$ and $\varphi_i \leftarrow 0$.
    end if
end if
```

decision-making and resource allocation are performed. Otherwise, the eNodeB can not provide the offloading service due to insufficient resources. Generally, the MVDHA algorithm allocates resources to users in descending order of value density. The amount of allocated resources is rounding to the maximum value density point. When resources are limited, the minimum required resource is allocated instead. The computation complexities of the MVDHA algorithm for the UE and eNodeB side are $O(1)$ and $O(n^2)$, respectively.

Where $a_i^{receive}$ in Algorithm 1 denotes the amount of resource allocated to user i.

VII. SIMULATION RESULTS

In this section, we conduct extensive simulations to verify the theoretical results derived in Section V and evaluate the performance of the MVDHA algorithm.

In simulations, radio channels follow Rayleigh fading. The Physical Uplink Shared Channel (PUSCH) throughput τ_i^v is measured by MATLAB *LTE ToolBox*^{TM} under conformance test conditions as defined in TS36.104. The rest of the main parameters are as follows. Subcarriers in each resource block $\lambda = 12$. Subcarrier spacing $\varpi = 15$ kHz (in 5G network, it supports scalable subcarrier spacing, e.g. $\{15, 30, 60, 120, 240\}$ kHz) [38], [39]. Number of vehicular users $N = \{20, 30, 40, 50\}$. Wireless network bandwidth $B = \{10, 15, 20\}$ MHz. Transmission power of vehicular user $P_i = 100$ mWatt. Vehicular local computing ability f_i^l is randomly assigned in the range of [500, 1000] Megacycles or Gigaflop per second. Edge cloud computing ability is 200000 Megacycles or Gigaflop per second. Uploading task size, computation units, and energy consumption per computing unit are uniformly distributed in the range of [100, 1000] kB, [400, 12000] Megacycles or Gigaflop, and $[2.0 * 10^{-10}, 2.0 * 10^{-6}]$ Watt per computing unit, respectively. The vehicles maintain a constant movement speed at either 5 MPH or 70 MPH. The average signal to noise ratio of vehicular users during offloading is in the range of [-2dB, 2dB].

Algorithm 2 Resource Allocation Period_EnodeB Side

```
Initialization:
if receive queries about available resource then
    send $R(t)$ to request users.
    if receive request about offloading services then
        add the requests to resource allocation buffer and sort the vectors of $A^{\omega}$ and $A^{\varsigma}$ from larger to smaller as $\overline{A^{\omega}}$ and $\overline{A^{\varsigma}}$, respectively.
    end if
end if
End initialization.

while $\min(\overline{A^{\omega}}) \leq R(t)$ do
    if $\min(\overline{A^{\varsigma}}) \leq R(t)$ then
        for i in $\overline{A^{\varsigma}}$ do
            if $a_i{}^{\varsigma} \leq R(t)$ then
                send UL Grant to user $i$, $a_i^{receive} \leftarrow a_i{}^{\varsigma}$, $R(t+1) \leftarrow R(t) - a_i{}^{\varsigma}$, and $t \leftarrow t+1$. Remove $a_i{}^{\varsigma}$ and $a_i{}^{\omega}$ from $\overline{A^{\varsigma}}$ and $\overline{A^{\omega}}$, respectively.
            end if
        end for
    else
        for i in $\overline{A^{\omega}}$ do
            if $a_i{}^{\omega} \leq R(t)$ then
                send UL Grant to user $i$, $a_i^{receive} \leftarrow a_i{}^{\omega}$, $R(t+1) \leftarrow R(t) - a_i{}^{\omega}$, and $t \leftarrow t+1$. Remove $a_i{}^{\varsigma}$ and $a_i{}^{\omega}$ from $\overline{A^{\varsigma}}$ and $\overline{A^{\omega}}$, respectively.
            end if
        end for
    end if
end while
```

A. THEORETICAL RESULTS VERIFICATION

In this subsection, we verify the correctness of the highest available value density theory. As described in Theorem 3 of Section V, the highest value density will be obtained when the offloading energy consumption is equal to half of local energy consumption.

As shown in Fig. 4, we randomly select eight computation task data sets to demonstrate the relationship between value density and allocation resource. It can be seen that the individual value density curve firstly arises with the increase of allocated resource block, and then gradually decreases. This verifies that the value density function is concave. Some bottom curves do not follow this trend because the number of allocated resource blocks does not meet the preset computation offloading constraints, such as the offloading energy or the time consumption threshold. Therefore, we hide this part of the curve.

In addition, we select User No. 1 and User No. 2 in Fig. 4 as two representative users for further analysis. As shown in Fig. 5, the highest value density point of User 1 is at 1 RB, and the energy consumption reduces to 51.96% of local energy consumption. The energy consumption of User 2

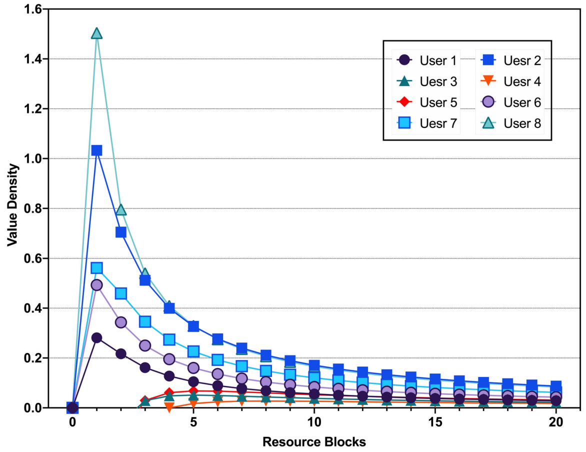


FIGURE 4. Value density change process with allocation resource.

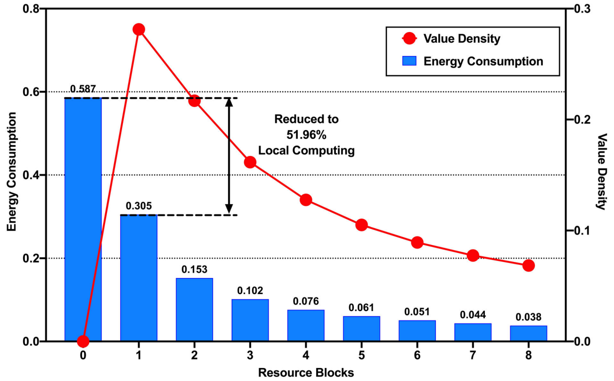


FIGURE 5. The individual energy consumption change process vs. allocated resource blocks of user No. 1.

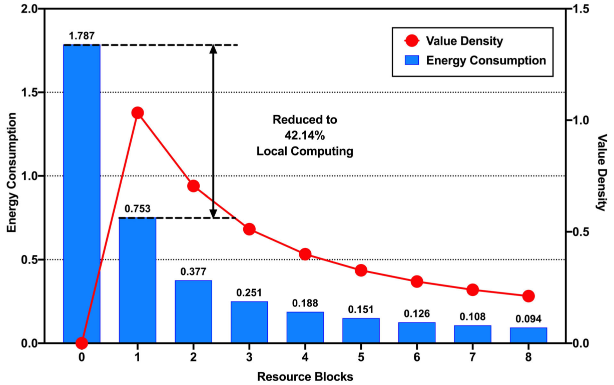


FIGURE 6. The individual energy consumption change process vs. allocated resource blocks of user No. 2.

reduces to 42.1% of local energy consumption, as illustrated in Fig. 6. Considering that the RB can only be allocated in integer, their energy-saving conclusions are both very close to 50%. Therefore, it satisfies the description of Theorem 3.

B. PERFORMANCE EVALUATION WITH DIFFERENT NETWORK PARAMETERS

In this subsection, we verify the performance of the VECC and the MVDHA algorithm by comparing it with local computing and optimal solution.

As shown in Fig. 7 and 8, we can see that the blue column with the highest value represents local computing, and the red and green columns represent the results of the MVDHA and the optimal solution obtained by using the brute force algorithm (BFA). The similar results of these two algorithms show the superior performance of the MVDHA algorithm. Fig. 7 shows that the VECC solution can save more than 62% energy consumption. Fig. 8 shows that VECC performs more than 80% time saving compared to local computing. They reflect the effectiveness of VECC and the superior performance of MVDHA in energy-saving and computation augmentation.

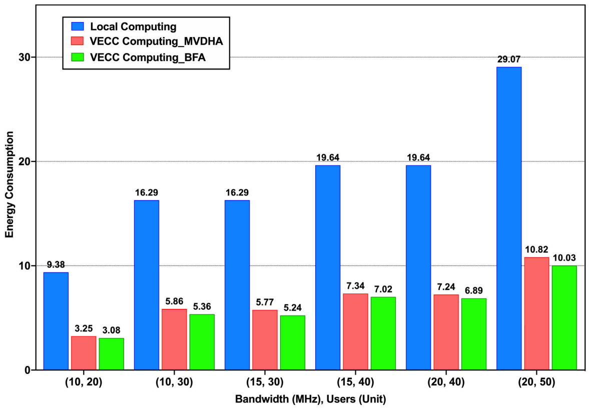


FIGURE 7. VECC and Local computing energy consumption under different simulation parameters.

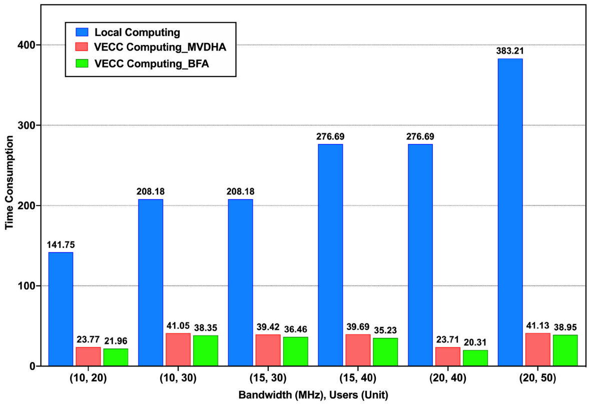


FIGURE 8. VECC and Local computing time consumption under different simulation parameters.

C. PERFORMANCE EVALUATION AT DIFFERENT MOVING SPEEDS

To illustrate the VECC performance for vehicular users at different mobility, we conducted 100 tests with $N = 30$ Users

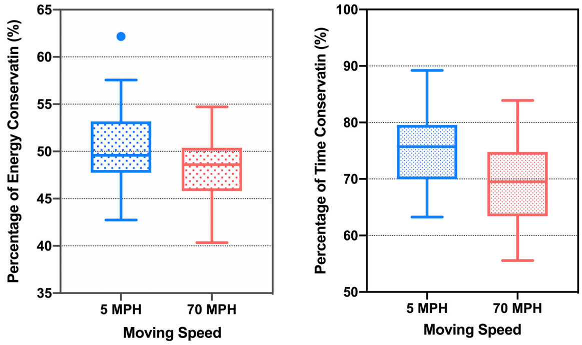


FIGURE 9. **Percentage of energy conservation and time conservation at different moving speed.**

and $B = 15$ MHz in different mobility, respectively. The statistical results show in Fig. 9.

In Fig. 9, the left sub-figure shows that the average percentage of energy conservation is 49.5% at 5 MPH and 48.4% at 70 MPH. Also, the upper range of low-speed energy conservation is higher than at high speed. Furthermore, in Fig. 8 right part, the average percentage of time conservation is 75.7% at 5 MPH and 69.5% at 70 MPH. In general, the algorithm has better performance on both time consumption and energy consumption in the low-speed environment, because the transmission speed is affected by the moving speed. However, it still shows a decent performance in high-mobility situations.

D. PERFORMANCE EVALUATION WITH DIFFERENT ALGORITHMS

In this subsection, we further compare the MVDMA algorithm with two benchmark algorithms to verify the consistent performance of the MVDHA algorithm under fast convergence.

We compare MVDHA with two benchmark algorithms — randomly select algorithm (RSA) and sorted select algorithm (SSA). RSA is a stochastic resource allocation algorithm based on the competitive game theory, which is proposed in [35], and is currently a benchmark algorithm in this research field. The other SSA is an algorithm based on strict sorting to allocate the minimum amount of resources. It has a similar theoretical approach to MVDHA, but lacks user fairness.

In order to avoid the contingency of the algorithm results, we conducted 100 tests and statistically analyze the results as follows.

As shown in Fig. 10, the box and whisker plot shows the statistical results of the energy conservation percentage under these three different resource allocation strategies. We can see that the RSA ranges from 20.4% to 71.8%, with a mean of 53.7%, the SSA ranges from 47.3% to 70.4%, with a mean of 59.9%, and the MVDHA ranges from 53.1% to 72.9%, with a mean of 65.6%. RSA shows the largest value interval, which reveals the lowest consistency of performance among these three algorithms. In contrast, VHDA shows the best performance, not only in terms of energy conservation percentage but also in performance stability.

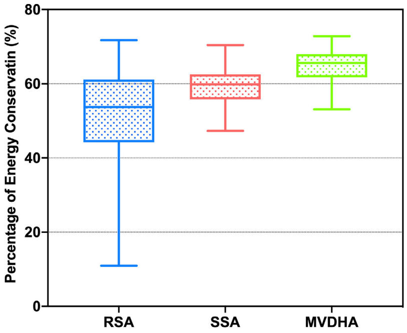


FIGURE 10. **100 statistic results 15MHz + 30 UEs of energy conservation.**

In Fig. 11, we can see that the RSA performs in a range of 17.3% to 85.8%, with a mean of 48.7%. The SSA performs in a range of 75.5% to 96.2%, with a mean of 87.8%. The MVDHA performs in a range of 76.1% to 95.9%, with a mean at 84.8%. Compared with MVDHA, SSA can provide an average of 3% extra time consumption.

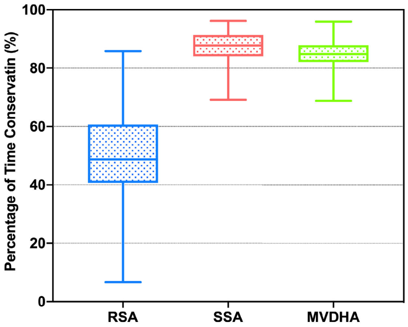
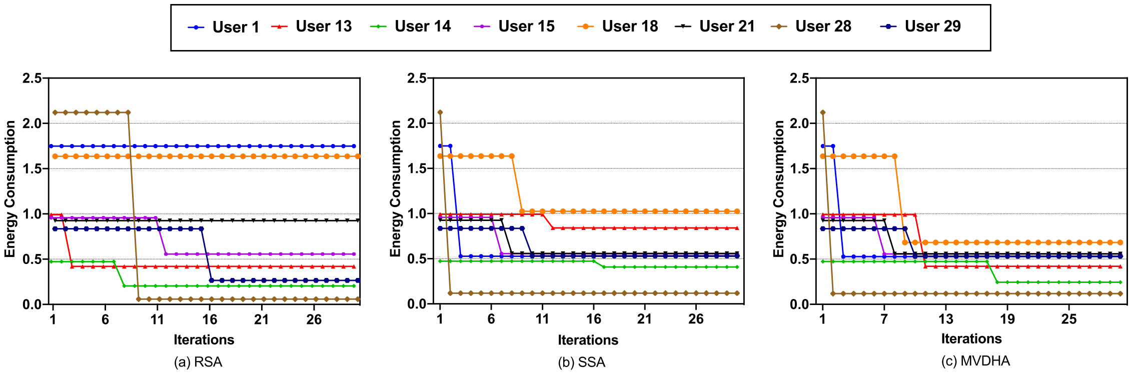


FIGURE 11. **100 statistic results 15MHz + 30 UEs of time conservation.**

We can see that MVDHA has better convergence and stability than the other two benchmark algorithms in energy saving. In terms of time savings, while SSA is better than MVDHA, their difference is small. To demonstrate more details on energy conservation among these three schemes, Fig. 12 vividly illustrates the energy consumption change curve of 8 random selected UEs in 20 MHz bandwidth and $\mathcal{N} = 30$ with RSA, SSA, and MVDHA three schemes, respectively.

As we can see, RSA fails in accurately selecting some users with higher energy consumption, such as user 1 and user 18. However, both SSA and MVDHA chose them as offloading users at an early stage. In addition, SSA does not allocate an appropriate amount of resources to users, but MVDHA does. As shown by users 13 and 14, precise resource allocation can bring greater returns to both individual users and the entire system. Therefore, MVDHA achieves the best performance in terms of energy-saving.



(a) RSA (b) SSA (c) MVDHA

FIGURE 12. The individual energy consumption change process in 20 MHz & 30 users scenario.

VIII. CONCLUSION

In this paper, we presented the VECC framework as a promising solution to energy conservation and computation augmentation for vehicular computing. The VECC could resolve several inevitable computing dilemmas for large-scale implementation of CAVs. At the same time, to achieve a high energy-efficient VECC framework, we studied the wireless resource allocation scheme for vehicular COS at RB level. We proposed the value density function to evaluate the cost-effectiveness of allocated resources and energy savings. Numerical results showed that with the increase of allocation resources to users, the value density function was concave. Especially when the offloading energy consumption was equal to half of the local computation energy consumption, it could obtain the maximum value density. Based on such discovery, we proposed MVDHA algorithm to solve the formulated NP-hard energy optimization problem. In the simulations, it showed fast convergence and superior performance on both energy and time conservation. For future work, we will further study the other modules of VECC and adaptable strategies in resource management. Furthermore, we also aim at integrating the VECC framework with the California Connected Vehicle Testbed to have more field tests research.

APPENDIX

PROOF OF THE LEMMA 3

According to (14a), we will have:

$$\begin{aligned}
\frac{\partial \rho_i}{\partial a_i} &= \frac{\partial}{\partial a_i}(\frac{e_i^l - e_i^o}{a_i}) = \frac{\partial}{\partial a_i}\left(\frac{\sigma_i\beta_i - \frac{\delta_i P_i}{r(a_i)}}{a_i}\right) \\
&= \frac{\partial}{\partial a_i}\left(\frac{\sigma_i\beta_i - \frac{\delta_i P_i}{\lambda\varpi\tau_i^v a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}}{a_i}\right) \\
&= \frac{2\delta_i P_i - \lambda\varpi\tau_i^v\sigma_i\beta_i a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}{\lambda\varpi\tau_i^v \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})a_i^3} \\
&= \frac{\frac{2\delta_i P_i - \lambda\varpi\tau_i^v\sigma_i\beta_i a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}{\lambda\varpi\tau_i^v a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}}{a_i^2}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\frac{2\delta_i P_i}{\lambda\varpi\tau_i^v a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})} - \frac{\lambda\varpi\tau_i^v\sigma_i\beta_i a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}{\lambda\varpi\tau_i^v a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})}}{a_i^2} \\
&= \frac{\frac{2\delta_i P_i}{\lambda\varpi\tau_i^v a_i \log_2(1+\frac{P_i|H_{i,s}|^2}{N_0})} - \sigma_i\beta_i}{a_i^2} \\
&= \frac{2e_i^o - e_i^l}{a_i^2}
\end{aligned}$$

It is easy to find out that $\frac{\partial \rho_i}{\partial a_i}$ will have a zero when $(2e_i^o - e_i^l)$ is equal to 0, which also means the ρ_i will have extremum (or extrema) under the same condition. Furthermore, because the $(2e_i^o - e_i^l)$ is a monotonically-decreasing function with the increasing of allocated resource, the e_i^l is constant. So with the increase of a_i, the $(2e_i^o - e_i^l)$ will firstly be positive, and then be negative. In such case, the extremum can be inferred to be the maximum. Furthermore, the highest value of value density will be obtained when the offloading energy consumption is half of the local energy consumption (i.e., $(2e_i^o - e_i^l) = 0$).

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable and insightful help in improving this research. Meanwhile, the authors also would like to thank Professor Kun Zhou, Dr. Matthew Weber at the University of California, Berkeley, Dr. Hans-Joerg Voegel at Germany BMW Group Research, Professor Weisong Shi at Wayne State University, and Professor Carlo Ratti at Massachusetts Institute of Technology for their helpful insights and suggestions for this research.

REFERENCES

[1] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mech. Syst. Signal Process.*, vol. 100, pp. 482–500, Feb. 2018.

[2] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 62–68, Jan. 2017.

[3] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," *ACM SIGPLAN Notices*, vol. 53, no. 2, pp. 751–766, Mar. 2018.

[4] A. Ashok, P. Steenkiste, and F. Bai, "Vehicular cloud computing through dynamic computation offloading," *Comput. Commun.*, vol. 120, pp. 125–137, May 2018.

[5] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Gener. Comput. Syst.*, vol. 87, pp. 278–289, Oct. 2018.

[6] X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, and S. Wan, "An edge computing-enabled computation offloading method with privacy preservation for Internet of connected vehicles," *Future Gener. Comput. Syst.*, vol. 96, pp. 89–100, Jul. 2019.

[7] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Valbonne, France, White Paper 11, 2015, pp. 1–16.

[8] F. Computing, "Fog computing and the Internet of Things: Extend the cloud to where the things are," Cisco, San Jose, CA, USA, White Paper, 2015.

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[10] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," in *Proc. 55th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*, Sep. 2016, pp. 1122–1127.

[11] X. Li, Y. Dang, and T. Chen, "Vehicular edge cloud computing: Depressurize the intelligent vehicles onboard computational power," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3421–3426.

[12] 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.

[13] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.

[14] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.

[15] A. Ghazizadeh and P. Ghazizadeh, "Design and analysis of a communication protocol for dynamic vehicular clouds in smart cities," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 1–6.

[16] 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.

[17] 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.

[18] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 93–107, Feb. 2018.

[19] B. Gu and Z. Zhou, "Task offloading in vehicular mobile edge computing: A matching-theoretic framework," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 100–106, Sep. 2019.

[20] L. Pu, X. Chen, G. Mao, Q. Xie, and J. Xu, "Chimera: An energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 84–99, Feb. 2019.

[21] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

[22] 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.

[23] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th Int. Workshop Resilient Netw. Des. Modeling (RNDM)*, Sep. 2016, pp. 288–294.

[24] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus ADMM approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 5087–5099, May 2019.

[25] 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.

[26] H. Peng, Q. Ye, and X. Shen, "Spectrum management for multi-access edge computing in autonomous vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[27] Y. Mao, J. Zhang, 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.

[28] Z. Ning, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5G-enabled traffic management systems," *IEEE Trans Ind. Informat.*, vol. 15, no. 5, pp. 3058–3067, May 2019.

[29] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 39–50, Jul. 2015.

[30] M. Aazam and X. Fernando, "Fog assisted driver behavior monitoring for intelligent transportation system," in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–5.

[31] *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) Radio Transmission and Reception*, document TS 36.101 version 8.23.0 Release 8, 3GPP, 2014.

[32] M. Ni, X. Xu, and R. Mathar, "A channel feedback model with robust sinr prediction for lte systems," in *Proc. 7th Eur. Conf. Antennas Propag. (EuCAP)*, 2013, pp. 1866–1870.

[33] R. Falkenberg, B. Sliwa, N. Piatkowski, and C. Wietfeld, "Machine learning based uplink transmission power prediction for LTE and upcoming 5G networks using passive downlink indicators," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–7.

[34] A. Ashok, P. Steenkiste, and F. Bai, "Adaptive cloud offloading for vehicular applications," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2016, pp. 1–8.

[35] 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.

[36] 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.

[37] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *NP-Complete Problems*. New York, NY, USA: McGraw-Hill, 2008, ch. 8.

[38] *LTE Physical Layer Overview*. Keysight Technologies. [Online]. Available: http://rfmw.em.keysight.com/wireless/helpfiles/89600b/webhelp/subsystems/lte/content/lte_overview.htm

[39] *5G Physical Layer Overview*. Keysight Technologies. [Online]. Available: https://www.keysight.com/upload/cmc_upload/All/Understanding_the_5G_NR_Physical_Layer.pdf

XIN LI (Student Member, IEEE) received the B.E. degree in electrical and electronics engineering from Central South University, Changsha, China, in 2014, where he is currently pursuing the Ph.D. degree. He has been a Visiting Scholar with the Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA, since 2016. His collaborator at UC Berkeley is Prof. Wei-Bin Zhang and Prof. Edward Lee. His research interests include vehicular edge computing, artificial intelligent, and intelligent transportation systems. For additional details: www.lixin618.com.

YIFAN DANG (Student Member, IEEE) received the B.E. degree in software engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, and the M.S. degree in computer and information science from the University of Oregon Eugene, OR, USA. His research interests include artificial intelligent, vehicular edge computing, and V2X communication systems.

MOHAMMAD AAZAM (Senior Member, IEEE) received the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2015. In addition to that, he has completed a course on data science with R from Harvard University, USA, in 2017, and a course on Internet of Things (IoT) from King's College London, U.K., in 2016. He worked with Carleton University, Canada, and Ryerson University, Canada as a Postdoctoral Fellow. He is currently working as a Senior Research Scientist with Carnegie Mellon University, Qatar. He has more than 100 publications, including three patents. He is also a Founding Member of the IEEE SIG Intelligent Internet Edge (IIE). For additional details: www.aazamcs.com.

XIA PENG received the B.E. degree in electrical and electronics engineering from the Hunan Institute of Engineering, Xiangtan, China, in 2016. She is currently pursuing the master's degree with Central South University, Changsha, China. Her research interests include vehicular cloud computing and vehicular communication networks.

TEFANG CHEN has been a Full Professor with Central South University, China, since 2004, where he is currently a Professor with the School of Traffic and Transportation Engineering. His main research areas include intelligent railway transportation systems and vehicular network systems.

CHUNYANG CHEN has been a Vice President of Central South University, since 2013. He is currently a Professor with the School of Traffic and Transportation Engineering, Central South University, China. His main research areas include railway transportation systems, intelligent transportation systems, and V2X communication systems. He is a member of Expert Panel for Joint Action Plan of Chinese High-Speed Train Innovation signed by the Ministry of Science and Technology and Ministry of Railways, the leader of 863 Program, a member of Expert Panel for national support project Key Components and Major Equipment Development of Power Electronics, and a member of the General Evaluation Committee of National Engineering Research Center.