Joint Offloading Decision and Resource Allocation for Vehicular Fog-Edge Computing Networks: A Contract-Stackelberg Approach

Yuwei Li, Bo Yang, Senior Member, IEEE, Hao Wu, Qiaoni Han, Cailian Chen, Member, IEEE, and Xinping Guan, Fellow, IEEE

Abstract

With the popularity of mobile devices and development of computationally intensive applications, researchers are focusing on offloading computation to Mobile Edge Computing (MEC) server due to its high computational efficiency and low communication delay. As the computing resources of an MEC server are limited, vehicles in the urban area who have abundant idle resources should be fully utilized. However, offloading computing tasks to vehicles faces many challenging issues. In this paper, we introduce a vehicular fog-edge computing paradigm and formulate it as a multi-stage Stackelberg game to deal with these issues. Specifically, vehicles are not obligated to share resources, let alone disclose their private information (e.g., stay time and the amount of resources). Therefore, in the first stage, we design a contract-based incentive mechanism to motivate vehicles to contribute their idle resources. Next, due to the complicated interactions among vehicles, road-side unit (RSU), MEC server and mobile device users, it is challenging to coordinate the resources of all parties and design a transaction mechanism to make all entities benefit. In the second and third stages, based on Stackelberg game, we develop pricing strategies that maximize the utilities of all parties. The analytical forms of optimal strategies for each stage are given. Simulation results demonstrate the effectiveness of our proposed incentive mechanism, reveal the trends of energy consumption and offloading decisions of users with various parameters, and present the performance comparison between our framework and existing MEC offloading paradigm in vehicular networks.

Index Terms

Copyright (c) 2022 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Y. Li was with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China and is with Huawei Technologies Co., Ltd. (email: liyuwei8@huawei.com).

B. Yang (Corresponding author), H. Wu, C. Chen, and X. Guan are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China; Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai 200240, China (e-mail: bo.yang@sjtu.edu.cn; wuhao233@sjtu.edu.cn; cailianchen@sjtu.edu.cn; xpguan@sjtu.edu.cn).

Q. Han is with the Department of Automation, Tianjin University, Tianjin 300072, China (e-mail: qghan@tju.edu.cn).

DOI: 10.1109/JIOT.2022.3150955
I. INTRODUCTION

THE development of Internet of Things and wireless communication technologies facilitates the emergence of computationally intensive applications with requirements of low latency and real-time processing, such as computer vision, natural language processing and autonomous driving. However, it is difficult for mobile devices with limited resources to provide required quality of service (QoS) for users [1].

Plenty of attempts have been made to offload computing tasks to the remote cloud server [2]. Although cloud computing can significantly improve computational performance, the long distance transmission may lead to considerable overhead and large latency [3]. By contrast, MEC with low cost servers in the vicinity of users, is a promising solution to computation offloading.

However, the computing resources of MEC server are usually constrained. Offloading computationally intensive tasks to these servers may cause low QoS, especially when the server is bursting with requests. Therefore, it is urgent to expand the resource capacity of MEC servers by exploiting idle resources from existing network entities and then effectively scheduling them for better services. Besides, energy-efficient optimization is also required for large-scale deployment and sustainable development of edge computing [4].

Recently, it is observed that vehicles with abundant idle computing resources can be organized into a vehicular fog computing (VFC) network so as to improve urban communication and computing capabilities [5]. In our daily life, many vehicles stay in the parking lot for a long time, or move slowly on the street. Due to their large number, long-term stay or slow movement, and relatively fixed location, vehicles in the urban area are ideal candidates for static backbone network and edge servers. In addition to the basic functions of computation offloading, such as providing users with computing, storage and application services, VFC also has the characteristics of low deployment cost, proximity to users, dense geographical distribution and so on. In this paper, we propose to utilize urban vehicles’ idle resources for computation offloading.

Nevertheless, there exist some challenging issues to be addressed. Firstly, vehicles are self-interested and have no obligation to share their idle resources. Therefore, a carefully designed incentive mechanism with reasonable rewards is indispensable. Additionally, vehicles are intuitively reluctant to disclose their private information, which creates an information asymmetry between vehicles and the organizer of VFC network like RSU. Secondly, the interactions among vehicles, RSU, the MEC server and mobile device users are complicated and difficult to model, due to their rationality and selfishness, and the tight coupling between resource demands and resource provisions. Thirdly, in order to realize a real-time and scalable computation offloading framework, an effective algorithm for joint offloading decision and resource allocation is preferred.

Motivated by the above issues, we introduce a new computing paradigm, named vehicular fog-edge computing (VFEC). In particular, a multi-stage Stackelberg game with an incentive mechanism is proposed to model the VFEC scenario. In a long-term market, RSU rents computing resources from vehicles. In this process, the information superiority over RSU for vehicles hinders the realization of Pareto optimal configuration. Based on contract theory,
we design incentive mechanism to properly handle this problem. By designing a series of contract items for vehicles to choose from, the contract can reveal the respective types of vehicles, make up for the influence of information asymmetry, and maximize the utilities of two sides of trade. On the other hand, the transaction between RSU and the MEC server, as well as between the MEC server and users, can be considered as a short-term market, which involves various interactions such as publishing prices, offloading decision-making and computing resources purchase. Stackelberg game is suitable for modelling this short-term, complex and dynamic process. Moreover, the solution to equilibrium of Stackelberg game only needs one round of operations, instead of the iterative algorithms commonly used in some auctions and non-cooperative games. This is good news for the time-varying computation offloading service market.

The main contributions of this paper are listed below.

- We propose a vehicular fog-edge computing paradigm to utilize urban vehicles for computation offloading, and develop a multi-stage Stackelberg game with an incentive mechanism to model the interactions among RSU, MEC server and mobile device users.
- We design a contract-based incentive mechanism for RSU to manage the idle computing resources of nearby vehicles. In order to overcome the information asymmetry between vehicles and RSU, the contract items are designed carefully not only to maximize the utility of RSU, but also to satisfy individual rationality and incentive compatibility of vehicles.
- The analytical forms of optimal strategies for each stage of the transaction are given. Numerical evaluation is conducted, which demonstrates the effectiveness of our proposed algorithms, reveal the trends of energy consumption and offloading decisions of users with various parameters.

The rest of this paper is organized as follows. Related works are listed in Section II. Section III describes the system model. Section IV introduces the multi-stage Stackelberg game with a contract-based incentive mechanism. The optimal algorithms are presented elaborately in Section V. The simulation results are shown and interpreted in Section VI. Finally, we conclude this paper in Section VII.

II. RELATED WORKS

MEC has recently attracted widespread attention from academia and industry. MEC servers are located on the edge of wireless network with more computing and storage resources than mobile terminals. Many existing works focus on MEC offloading in wireless networks, some of which are dedicated to minimize energy consumption [4, 6-9], reduce latency [10, 11], or optimize a weighted objective of energy consumption and delay [12, 13]. In addition, most of the above works optimize resource allocation [8] or joint computation offloading and resource allocation [4, 6, 12, 13] for MEC offloading in heterogeneous cellular networks. Due to the roll out of 5G mobile networks and Internet of Things, numerous and diverse users, servers and applications coexist in MEC systems. Therefore, server deployment and resource allocation become quite complicated in such system. Rodrigues et al. presented the utilization of Machine Learning (ML) to tackle these challenges in [14] and [15].

Thanks to MEC’s high computational efficiency and low communication delay, some works have deployed MEC server in vehicular networks, and aim to better support computationally intensive services with requirements of low
latency and real-time processing [16]. In [17], Liu et al. provided an overview of Vehicular Edge Computing (VEC). A mobility-aware computation offloading design for MEC-based vehicular networks was studied in [18]. Due to the limited computing resources of single MEC server and high requirements for timely task processing of a large amount of computations in the emerging mobile applications, some works focus on MEC cooperation or grouping. For instance, a cloud-MEC collaborative computation offloading scheme in vehicular networks was presented in [19]. Considering the cooperative utilization of computing resources of MEC/cloud servers, Dai et al. solved a distributed task assignment problem [20]. Based on game theory, a noncooperative game-based strategy selection algorithm was presented in [21] to realize MEC grouping for task offloading, and a multi-user noncooperative computation offloading game was formulated in [22] to adjust the offloading probability of each vehicle. Besides, some works utilize neighbouring vehicles with idle computing resources to provide offloading opportunities to other vehicles having limited computing capabilities [23–25]. Among them, in [23], Gu et al. addressed the task offloading between MEC servers deployed at RSUs and vehicles with excessive computing resources. By treating vehicles as edge computing infrastructure, Qiao et al. introduced a vehicular edge multi-access network and constructed a cooperative and distributed computing architecture [24]. In [25], both an autonomous vehicular edge (AVE) which shares neighbouring vehicles’ available resources and a hybrid vehicular edge cloud (HVC) which shares accessible resources of RSUs and cloud were introduced. Hou et al. in [26] proposed an edge-computing-enabled software-defined Internet of vehicles (EC-SDIoV) architecture to efficiently orchestrate the heterogeneous edge computing nodes, and provide reliable and low-latency computation offloading, where partial offloading, reliable task allocation and the reprocessing mechanism are jointly considered. In order to deal with the dynamic vehicular environment, Shi et al. in [27] proposed a priority-aware task offloading scheme in the context of vehicular fog computing, where a soft actor-critic method is developed to select the service vehicles and publish dynamic prices. The algorithm in [27] achieved more robust and sample-efficient performance in task completion ratio and offloading delay. More and deeper issues of MEC are surveyed in [28] toward future vehicular networks.

Recently, parked and slow moving vehicles have attracted much attention to improve the performance of vehicular networks. Malandrino et al. investigated the possibility of exploiting parked vehicles to extend the RSU service coverage [29]. In [30], a game theoretic framework of content delivery which utilized parked vehicles was proposed. Sun et al. considered the parked vehicles as relay nodes in [31]. Observing that parked or slow moving vehicles have rich and under-utilized resources for task execution, some works studied utilizing vehicles to develop new computing paradigms [3, 32]. Aiming to minimize the average response time for events reported by vehicles, Wang et al. put forward a feasible solution which enables offloading by moving and parked vehicles for real-time traffic management [33]. With adopting parked vehicles for computation offloading, an energy-efficient parked vehicular computing (PVC) paradigm was developed in [34].

Different from existing works, this paper focuses on some challenging issues. First, vehicles have no obligation and willingness to share their idle computing resources. Second, there is information asymmetry between vehicles and the organizer of VFC network, which makes it difficult to distinguish the qualification of vehicles. Therefore, it is necessary to design an incentive mechanism to reveal the types of vehicles, and encourage them to participate in the computation offloading with a certain reward. Third, the offloading service involves multiple transaction
TABLE I
KEY NOTATIONS AND DEFINITIONS

| Notation | Definition |
|----------|------------|
| \( c_i \) | the computation needed by user \( i \)’s task |
| \( d_i \) | the size of input data of user \( i \)’s task |
| \( t_i^{\text{MAX}} \) | the maximum tolerable latency of user \( i \)’s task |
| \( f^{l}_i \) | the local computing resources of user \( i \) |
| \( f^{e}_{\text{MAX}} \) | the computing resources of MEC server |
| \( f^{v}_{\text{MAX}} \) | the idle computing resources of each vehicle |
| \( U_i, U_{\text{MEC}}, U_{\text{RSU}}, U_{\text{Vm}} \) | the utility of user \( i \) / the MEC server / RSU / a type-\( m \) vehicle |
| \( p \) | the price of MEC server’s computing resources |
| \( e \) | the price of computing resources collected by RSU |
| \( f^{l}_{i} \) | the quantity of resources that user \( i \) purchases from the MEC server |
| \( f^{l}_{\text{thresh}} \) | a threshold of resources for judging whether user \( i \) chooses to offload its task or not |
| \( f_e \) | the computing resources of MEC server used for task offloading |
| \( f_{\text{RSU}} \) | the computing resources that the MEC server purchases from RSU |
| \( e, e_v \) | the unit monetary cost of server’s / vehicle’s energy consumption |
| \( k \) | the effective switched capacitance, which depends on the chip architecture |
| \( \theta_m, m \in \{1, 2, \ldots, M\} \) | the type of a vehicle |
| \( l_{\text{RSU}} \) | the number of type-\( m \) vehicles near RSU |
| \( f_{\text{RSU}} \) | the computing resources that RSU rents from a type-\( m \) vehicle |
| \( p_{\text{RSU}} \) | the rent for leased resources of a type-\( m \) vehicle per unit of time |

processes with multiple participants. How to coordinate the resources of all parties in an integrated way so that all entities can benefit is also a major challenge. In this paper, we introduce a vehicular fog-edge computing paradigm and formulate it as a multi-stage Stackelberg game to deal with these issues.

III. SYSTEM MODEL

A. Overview

We consider a small district in the urban area, where an MEC server, an RSU, some vehicles and some mobile device users are located. We assume that the connections between all involved entities are single hop.

There are \( N \) mobile device users in this region, such as smart phones, vehicles, wearable devices and so on. We denote the ID set of these users as \( \mathcal{N} = \{1, 2, \ldots, N\} \). At some point, each user has a computationally intensive task to be completed, which is characterized by a tuple \( (c_i, d_i, t_i^{\text{MAX}}), \forall i \in \mathcal{N} \). \( c_i \) represents the computation (in CPU cycles) needed by user \( i \)’s task. \( d_i \) is the size of input data of user \( i \)’s task. \( t_i^{\text{MAX}} \) is the maximum tolerable latency of user \( i \). \( f^{l}_{i} \) denotes the local computing resources of user \( i \) (in CPU cycles/s).

There exists an MEC server, whose computing resources, denoted as \( f^{e}_{\text{MAX}} \), are limited. A user may offload its task to the MEC server or execute it locally, depending on the price of resources and how fast its task will be completed.

The resources of MEC server are not always able to meet the needs of all users, which means high latency, expensive prices and low QoS. We may consider making full use of vehicular idle resources to improve users’ experience, where the resources are collected and scheduled by RSU.

All vehicles in this region, whether parking or moving slowly, can establish stable wireless connection with RSU via vehicle-to-infrastructure (V2I) communications. Each vehicle is equipped with an on-board-unit (OBU), which is not a simple device tracking the vehicle location and measuring its speed, but a mobile device with storage, communication and computing capabilities [35]. We assume that the total amount of idle resources owned by each vehicle, \( f^{v}_{\text{MAX}} \), is the same. After signing the contract, OBUs within the communication range of RSU can be organized into a fog cluster so as to enhance service capabilities.
In order to perform computation offloading efficiently after the MEC server has prepared computing resources and users have made their offloading decisions, the HVC architecture in [25] may be used, which designs a workflow that includes steps such as job caching, job scheduling, data transmission, job execution and result data transmission. Additionally, to facilitate the entire transaction process and take care of the interests in all parties to a fair and impartial manner, a third-party central controller is set up in the system, which is subordinate to a computation offloading operator. Its responsibilities include collecting global information, running the proposed algorithms, scheduling users’ tasks and transmitting control messages, etc. As mentioned earlier, this paper focuses on resource scheduling and users’ offloading decisions. The details of these implementations are beyond the scope of this paper and will be the direction of future work.

B. Three-stage Stackelberg game

In order to ensure that vehicles, RSU, the MEC server and users are willing to participate in computation offloading, we design a multi-stage Stackelberg game for this vehicular fog-edge computing paradigm. In this game, a contract-based incentive mechanism is developed for RSU to recruit the idle computing resources of vehicles, while the interactions between RSU and the MEC server, the MEC server and users are modeled as a multi-stage Stackelberg game.

Specifically, in the first stage, a contract-based trading mechanism is designed to employ vehicles for users’ task execution. The RSU acts as an employer who offers different contract items to vehicles, while each vehicle is an employee who selects a certain type of contract which suits it best. At the same time, RSU also acts as the leader of RSU-MEC Stackelberg game and announces the price of computing resources to the MEC server.

In the second stage, the MEC server is both the follower of RSU-MEC Stackelberg game and the leader of MEC-user Stackelberg game, determining the quantity of computing resources purchased from RSU and broadcasting the price of resources to all mobile device users.

In the third stage, all users are the followers of MEC-user Stackelberg game. Each user decides whether to offload its task and the quantity of computing resources purchased from the MEC server.
Note that since parked or slow moving vehicles are considered, the leasing of idle vehicular computing resources is a long-term market. In contrast, the deadline for a user’s task is shorter, so the RSU-MEC Stackelberg game and the MEC-user Stackelberg game are played slot by slot.

C. Utility functions

a) mobile device users: The utility function of user $i$ is defined as

$$U_i(f_i) = \frac{C}{t_i^{\text{max}}} \ln \left( \frac{f_i}{f_i + \delta} \right) - pf_i.$$  

(1)

In fact, users tend to purchase more computing resources to minimize delay. However, with the continuous growth of obtained resources, the shortened completion time is reduced, and the benefits of unit computing resource are gradually declining as well. That is, diminishing marginal effect appears. Therefore, the benefits of a user can be characterized by a concave function of the purchased computing resources over its local resources. So we may adopt a logarithmic function. $C > 0$ is a constant, and $\tau_i \triangleq C/t_i^{\text{max}}$ represents user $i$’s sensitivity to latency. $\delta \geq 1$, is to make sure that the logarithmic function has a positive value. $p$ denotes the price of MEC server’s computing resources, and $f_i$ is the quantity of resources that user $i$ purchases from the MEC server. The utility of a mobile device user is defined as the benefits of offloading service minus the expenditure of resources.

If user $i$ decides to execute its task locally, the completion time is

$$t_i^l = \frac{c_i}{f_i^l}.$$  

(2)

Otherwise, user $i$ offloads its task to the MEC server. The completion time is composed of upload latency and execution delay. Then the completion latency for offloading user $i$’s task to the MEC server is

$$t_i^e = \frac{d_i}{r_i} + \frac{c_i}{f_i^e},$$  

(3)

where $r_i$ represents the upload transmission rate from user $i$ to the MEC server. When $t_i^e < t_i^l$, user $i$ may prefer to offload its computationally intensive task to the MEC server. Let $f_i^{\text{thresh}} = \frac{c_i}{\tau_i - \frac{d_i}{r_i}}$. If

$$f_i > f_i^{\text{thresh}},$$  

(4)

then user $i$ will choose to offload its task.

b) the MEC server: The utility function of the MEC server is defined as

$$U_{\text{MEC}}(p, f_e, f_{\text{RSU}}) = \sum_{i=1}^{N} pf_i - e k f_e^2 - c f_{\text{RSU}},$$  

(5)

where $f_e$ is the computing resources of MEC server used for task offloading, and $e$ denotes the unit cost of server’s energy consumption. $k f_e^2$ represents the energy of MEC server consumed by computation. Here, the energy

1Usually the results of processing are much smaller than the input data, so here we ignore the download transmission delay [10].

2In fact, a user’s task may be offloaded to a certain vehicle. Then the upload latency in (3) should be replaced by the upload delay from the user to the vehicle. However, we can’t figure out in advance whether and which vehicle a task will be delivered to. Additionally, considering that users, the MEC server and vehicles are all located in the same small district, we use the upload delay from the user to the MEC server to replace the specific transmission delay.
consumption model of computation is referenced from [10]. $k$ is the effective switched capacitance, which depends on the chip architecture. In order to simplify expressions, let $k_v = e_v k$. $e$ denotes the price of resources collected by RSU, and $f_{RSU}$ is the computing resources that the MEC server purchases from RSU. The utility of MEC server is the revenues from users minus the energy cost and the expenditure of purchasing computing resources from the RSU.

c) the RSU: Before the contract is signed, we assume that RSU has obtained the current stay time of each vehicle within its communication range, which is denoted as $\theta_v'$. The type of a vehicle $v$ is defined as the length of time it continues to park or stay, $\theta_v$. Of course, RSU prefers the vehicle who has a longer stay. Due to the existence of information asymmetry between vehicles and RSU, the latter does not know the exact types of former. However, through statistics on the historical data, RSU can obtain the probability cumulative function of vehicle stay time $F(t)$ about the timing in a day, vehicle position and other factors. Then we have:

$$ P(\theta \geq \theta_v + \theta_v' | \theta \geq \theta_v') = \frac{1 - F(\theta_v + \theta_v')}{1 - F(\theta_v')} $$

(6)

In order to facilitate the subsequent analysis, we discretize the type of vehicles into $M$ items: $\theta_1, \theta_2, \ldots, \theta_M$ with $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_M$. Then it can be seen that RSU can calculate the probability distribution of the vehicles' continued stay time at the current moment, $f(\theta_m)$ with $\sum_{m=1}^{M} f(\theta_m) = 1$. Therefore, the number of type-$m$ vehicles near the RSU is $l_m = f(\theta_m)L$, where $L$ is the total number of vehicles.

The utility function of RSU is

$$ U_{RSU}(f_m, p_m, e) = e f_{RSU} - \sum_{m=1}^{M} l_m \theta_m p_m, $$

(7)

where $p_m$ represents the rent for leased resources of a type-$m$ vehicle per unit of time. The utility of RSU is defined as payment from the MEC server minus the cost of renting computing resources from vehicles.

d) vehicles: The utility function of a type-$m$ vehicle is defined as

$$ U_{V_m}(f_m, p_m) = \theta_m p_m - e_v k f_m^2, $$

(8)

where $f_m$ denotes the quantity of computing resources offered by a type-$m$ vehicle, and $e_v$ is the unit monetary cost of vehicle’s energy consumption. The utility of a vehicle is the rewards from RSU minus the cost of energy consumption on computing.

IV. Problem Formulation

A. Stage I

To motivate the vehicles to participate the transaction and select the contract item which fits their types best, the following IR and IC conditions should be satisfied [36].

**Definition 1: [Individual Rationality (IR)]** Since the vehicles are rational, IR condition committed a nonnegative utility to a vehicle if it accepts the contract item designed for its type. The IR conditions can be formulated as

$$ \theta_m p_m - k_v f_m^2 \geq 0, \forall m \in \{1, 2, \ldots, M\}. $$

(9)

Similar to the MEC server, let $k_v = e_v k$ to simplify the derivation process.
Definition 2: Incentive Compatibility (IC)] IC conditions guarantee that a type-\( m \) vehicle will select the contract \((f_m, p_m)\), rather than any other contract items \((f_j, p_j)\), \( \forall j \in \{1, 2, \cdots, M\}\setminus\{m\} \). The IC conditions can be written as

\[
\theta_m p_m - k_v f_m^2 \geq \theta_m p_j - k_v f_j^2, \forall m, j \in \{1, 2, \cdots, M\}, m \neq j.
\] (10)

Meanwhile, the RSU also plays the leader of RSU-MEC Stackelberg game and announces the price of computing resources \( c \) to the MEC server. We formulate the utility maximization problem of RSU as

\[
\max_{f_m, p_m, c} U_{RSU}(f_m, p_m, c)
\]

s.t. \( C1: c \geq 0 \)

\( C2: 0 \leq f_m \leq f_m^{\text{max}}, \forall m \in \{1, 2, \cdots, M\} \)

\( C3: \sum_{m=1}^{M} l_m f_m \geq f_{RSU} \)

\( C4: (9) \)

\( C5: (10) \)

\( C3 \) ensures that the MEC server does not purchase more resources than the resources RSU renting from vehicles.

B. Stage II

In this transaction, the MEC server plays two roles. It is the follower of RSU-MEC Stackelberg game as well as the leader of MEC-user Stackelberg game. The MEC server determines computing resources \( f_{RSU} \) purchased from the RSU and computing resources \( f_e \) used locally based on the price of unit resources \( c \) of RSU. At the same time, it broadcasts the price of unit resources \( p \) to all mobile device users. We can describe the utility maximization problem of MEC server as

\[
\max_{p, f_e, f_{RSU}} U_{MEC}(p, f_e, f_{RSU})
\]

s.t. \( C6: p \geq 0 \)

\( C7: 0 \leq f_e \leq f_e^{\text{max}} \)

\( C8: f_{RSU} \geq 0 \)

\( C9: f_e + f_{RSU} = \sum_{i=1}^{N} f_i \)

\[
(12)
\]

C. Stage III

The mobile device user \( i \), as a follower of MEC-user Stackelberg game, determines the quantity of computing resources \( f_i \) purchased from the MEC server based on the price \( p \) to maximize its own utility. We can formulate the optimization problem of user \( i \) as

\[
\max_{f_i} U_i(f_i)
\]

s.t. \( C10: (4) \)

March 1, 2022 DRAFT
V. OPTIMAL ALGORITHM

In this section, we use the backward induction method to analyze this multi-stage Stackelberg game and the contract-based incentive mechanism.

A. Solution of Stage III

Since the utility function of user \(i\) is a concave function, the zero point of its first derivative is the optimal solution. That is,

\[
f_i^* = \begin{cases} \frac{\tau_i}{p} - f_i^1 \delta, & p < \frac{\tau_i}{f_i^1 \delta + f_i^\text{thresh}} \\ 0, & p \geq \frac{\tau_i}{f_i^1 \delta + f_i^\text{thresh}} \end{cases}
\] (14)

We can see that if the price of computing resources is too high, \(i.e., p \geq \frac{\tau_i}{f_i^1 \delta + f_i^\text{thresh}}\), user \(i\) is unwilling to offload its task.

B. Solution of Stage II

Given the quantity of computing resources (14) purchased by users, we can substitute it into the utility function of MEC server. Note that \(f_i^*\) is a piecewise function, we introduce the following indicator variable for user \(i\)

\[
\chi_i = \begin{cases} 1, & p < \frac{\tau_i}{f_i^1 \delta + f_i^\text{thresh}} \\ 0, & p \geq \frac{\tau_i}{f_i^1 \delta + f_i^\text{thresh}} \end{cases}
\] (15)

Then (12) can be rewritten as

\[
\max_{p, f_e, f_{RSU}, \chi} \sum_{i=1}^{N} \chi_i \left( \tau_i - f_i^1 \delta p \right) - k_e f_e^2 - c f_{RSU}
\]

s.t. C6–C9

C11: \(\chi_i \in \{0, 1\}, \forall i \in \mathcal{N}\)

where \(\chi = [\chi_1, \chi_2, \cdots, \chi_N]^T\). Since \(\chi\) is a binary vector, and \(p, f_e, f_{RSU}\) are continuous variables, the optimization problem (16) is a mixed integer nonlinear programming problem. Given the indicator vector \(\chi\), it is easy to verify that this problem is concave.

We assume that the users are sorted in the following order:

\[
\frac{\tau_1}{f_1^1 \delta + f_1^\text{thresh}} \geq \frac{\tau_2}{f_2^1 \delta + f_2^\text{thresh}} \geq \cdots \geq \frac{\tau_N}{f_N^1 \delta + f_N^\text{thresh}}.
\] (17)

According to (15), we can see that whether a user chooses to offload its task depends on the price of unit computing resources of MEC server, \(i.e., p\). Further, in our transaction framework, \(p\) depends on the price of unit computing resources of RSU, \(i.e., c\). Thus, we consider a special case of (16), in which we assume that \(c\) is small enough.
such that all users choose to participate in the offloading. That is, the indicators for all users are equal to 1. Under this assumption, (16) can be rewritten as

$$\max_{p, f_e, f_{RSU}} \sum_{i=1}^{N} (\tau_i - f_i^l \delta p) - k_e f_e^2 - c f_{RSU}$$

s.t. C6–C8

(18)

Proposition 1: The optimal solution of (18) is

$$p^* = \sqrt{\frac{c \sum_{i=1}^{N} \tau_i}{\delta \sum_{i=1}^{N} f_i^l}}$$

(19)

$$f_e^* = \min \left\{ \frac{c}{2k_e}, f_e^{\text{max}} \right\}$$

(20)

when $2k_e f_e^{\text{max}} \leq T_N$, $f_{RSU}^*$ is shown as

$$f_{RSU}^* = \begin{cases} \sqrt{\frac{\tau \Delta}{c} - \frac{c}{2k_e}}, & c \leq 2k_e f_e^{\text{max}} \\ \sqrt{\frac{\tau \Delta}{c} - \frac{f_e^{\text{max}}}{c}}, & 2k_e f_e^{\text{max}} < c < T_N \end{cases}$$

(21)

when $2k_e f_e^{\text{max}} > T_N$,

$$f_{RSU}^* = \sqrt{\frac{\tau \Delta}{c} - \frac{\Delta}{2k_e c}} + \left( \frac{\Delta}{2k_e c} - \frac{c}{2k_e} \right)$$

(22)

where $\tau = \sum_{i=1}^{N} \tau_i$, $\Delta = \sum_{i=1}^{N} f_i^l \delta$, $T_N = \frac{\tau N \sqrt{\delta \sum_{i=1}^{N} f_i}}{(f_{RSU}^* + f_e^{\text{max}})(\sum_{i=1}^{N} \tau_i)}$. The proof of this proposition is shown in part A of Appendix.

Proposition 2: The price of MEC server’s computing resources given by (19) and the quantity of computing resources purchased by the MEC server from RSU given by (21) and (22) are optimal solution of (16) if and only if $c < T_N$. The proof of Proposition 2 is shown in part B of Appendix.

Then the optimal solution of (16) is given by the following theorem.

Theorem 1: The optimal solution of (16) is

$$p^* = \begin{cases} q_N, & c < T_N \\ q_{N-1}, & T_N \leq c < T_{N-1} \\ \vdots & \vdots \\ q_1, & T_2 \leq c < T_1 \end{cases}$$

(23)

(i): $2k_e f_e^{\text{max}} \geq T_1$,

$$f_{RSU}^* = \begin{cases} \left[ L_N - \frac{f_{RSU}^*}{2k_e} \right]^+, & c < T_N \\ \left[ L_{N-1} - \frac{f_{RSU}^*}{2k_e} \right]^+, & T_N \leq c < T_{N-1} \\ \vdots & \vdots \\ \left[ L_1 - \frac{f_{RSU}^*}{2k_e} \right]^+, & T_2 \leq c < T_1 \end{cases}$$

(24)
(ii): $2k_e f_e^{\text{max}} \leq T_N$

$$
f^*_\text{RSU} = \begin{cases} 
\left[ L_N - \frac{c}{2k_e} \right]^+, & c \leq 2k_e f_e^{\text{max}} \\
\left[ L_N - f_e^{\text{max}} \right]^+, & 2k_e f_e^{\text{max}} < c < T_N \\
\left[ L_{N-1} - f_e^{\text{max}} \right]^+, & T_N \leq c < T_{N-1} \\
\vdots & \vdots \\
\left[ L_1 - f_e^{\text{max}} \right]^+, & T_2 \leq c < T_1
\end{cases}
$$

(iii): $T_N < 2k_e f_e^{\text{max}} < T_1$ and $T_o \leq 2k_e f_e^{\text{max}} < T_{o-1}, \forall o \in \{2, 3, \cdots, N\}$

$$
f^*_\text{RSU} = \begin{cases} 
\left[ L_N - \frac{c}{2k_e} \right]^+, & c < T_N \\
\vdots & \vdots \\
\left[ L_o - \frac{c}{2k_e} \right]^+, & T_o \leq c \leq 2k_e f_e^{\text{max}} \\
\left[ L_o - f_e^{\text{max}} \right]^+, & 2k_e f_e^{\text{max}} < c < T_{o-1} \\
\vdots & \vdots \\
\left[ L_1 - f_e^{\text{max}} \right]^+, & T_2 \leq c < T_1
\end{cases}
$$

where $q_k = \sqrt{\frac{c \sum_{i=1}^{k} \tau_i}{\delta \sum_{i=1}^{k} f_i}}, T_k = \frac{\tau_k \sqrt{\delta \sum_{i=1}^{k} f_i}}{(f_{k, \text{opt}} + f_{k, \text{break}}) (\sum_{i=1}^{k} \tau_i)}, L_k = \sqrt{\left( \frac{\sum_{i=1}^{k} \tau_i}{c} \right) \frac{\delta \sum_{i=1}^{k} f_i}{\tau_k}} - \frac{\delta \sum_{i=1}^{k} f_i}{\tau_k}, \forall k \in \{1, 2, \cdots, N\}$.

**Proof:** If $c < T_N$, the optimal solution of (16) is obtained by Proposition 2. For $c$ in other intervals, the solution can be obtained similarly as Proposition 2, and is thus omitted.

The algorithm for optimal solution of Stage II is shown as below:

**Algorithm 1** the optimal solution of Stage II

**Step 1:** Initialize $k = N$;

**Step 2:** Sort all users according to (17);

**Step 3:** Compute $T_k = \frac{\tau_k \sqrt{\delta \sum_{i=1}^{k} f_i}}{(f_{k, \text{opt}} + f_{k, \text{break}})(\sum_{i=1}^{k} \tau_i)}$;

**Step 4:** Compare $T_k$ with $c$. If $c \geq T_k$, remove user $k$ from the game, set $k = k - 1$, and go to step 3; otherwise, go to step 5;

**Step 5:** According to Theorem 1, find the optimal solution $p^*$ and $f^*_\text{RSU}$.

C. Solution of Stage I

It can be seen that the expression of $f^*_\text{RSU}$ can be divided into three cases depending on the relationship between the values of $2k_e f_e^{\text{max}}$ and $T_k, \forall k \in \{1, 2, \cdots, N\}$. Here, we only give the solving process of Stage I in the case of $2k_e f_e^{\text{max}} \geq T_1$. The other two cases can be solved in similar ways. To solve the utility maximization problem
of RSU (11), we substitute (24) into (11). Since the expression of $f_{RSU}^*$ is piecewise, the problem is decomposed into $N$ subproblems. Assuming that $T_k \leq c < T_{k-1}$, (11) can be rewritten as:

$$\min_{f_m, p_m, c} \sum_{m=1}^{M} l_m \theta_m p_m - c \left( L_k - \frac{c}{2k_c} \right)$$

s.t. C2, C4, C5

$$\text{CT}: T_k \leq c < T_{k-1}$$

$$\text{C3}: L_k - \frac{c}{2k_c} \leq \sum_{m=1}^{M} l_m f_m$$

$$\text{C11}: L_k - \frac{c}{2k_c} \geq 0$$

The optimization problem (27) is difficult to solve due to $M$ IR constraints and $M(M-1)$ IC constraints. Below we propose several lemmas to simplify these constraints.

**Lemma 1:** For any feasible contract, if $\theta_i > \theta_j$, then we have $p_i > p_j$, $\forall i, j \in \{1, 2, \cdots, M\}$.

**Lemma 2:** For any feasible contract, $p_i > p_j$ if and only if $f_i > f_j$, $\forall i, j \in \{1, 2, \cdots, M\}$.

The above two lemmas have been proved in [36]. Then, we can reduce IR and IC constraints by the following lemmas.

**Lemma 3:** If the IR constraint of type-1 vehicles is satisfied, then all other IR constraints of type-$m$, $m \in \{2, 3, \cdots, M\}$ vehicles are also satisfied. That is,

$$\theta_m p_m - k_v f_m^2 \geq \theta_1 p_1 - k_v f_1^2 \geq 0.$$  

**Lemma 4:** The IC conditions can be reduced to the local downward incentive compatibility (LDIC) conditions and the local upward incentive compatibility (LUIC):

$$\theta_m p_m - k_v f_m^2 \geq \theta_m p_{m-1} - k_v f_{m-1}^2, \forall m \in \{2, 3, \cdots, M\},$$  

$$\theta_m p_m - k_v f_m^2 \geq \theta_m p_{m+1} - k_v f_{m+1}^2, \forall m \in \{1, 2, \cdots, M-1\}.$$  

**Lemma 5:** Under the optimal contract, all the LDIC constraints are active, and the IR constraint of type-1 vehicles is active as well. That is,

$$\theta_m p_m - k_v f_m^2 = \theta_m p_{m-1} - k_v f_{m-1}^2, \forall m \in \{2, 3, \cdots, M\},$$

$$\theta_1 p_1 - k_v f_1^2 = 0.$$  

**Lemma 6:** If all the LDIC constraints are active, then all the LUIC constraints are satisfied.

See the Appendix for proofs of above four lemmas.
Then, the IR and IC conditions are reduced to active type-1 IR constraint and active LDIC constraints. Note that C11 is not concave, we transform C11 to a concave function of $c$. Thus, (27) can be rewritten as

$$
\min_{f_m, p_m, c} \sum_{m=1}^{M} l_m \theta_m p_m - c \left( L_k - \frac{c}{2k_e} \right)
$$

s.t. C1, C2, C3

$$
\mathbf{C4} : \theta_1 p_1 - k_v f_1^2 = 0
$$

$$
\mathbf{C5} : \theta_m p_m - k_v f_m^2 = \theta_m p_{m-1} - k_v f_{m-1}^2, \ m \geq 2
$$

$$
\mathbf{C11} : c \left( L_k - \frac{c}{2k_e} \right) \geq 0
$$

$$
\mathbf{C12} : 0 \leq p_1 \leq p_2 \leq \cdots \leq p_M
$$

(33)

Obviously, the optimization problem (33) is convex and can be solved by Lagrangian multiplier method. The Lagrangian function is defined as

$$
\mathcal{L} = \sum_{m=1}^{M} l_m \theta_m p_m - c \left( L_k - \frac{c}{2k_e} \right) + \gamma (\theta_1 p_1 - k_v f_1^2) + \beta \left( -cL_k + \frac{c^2}{2k_e} \right) + \sum_{m=2}^{M} \mu_m \left( \theta_m (p_{m-1} - p_m) - k_v (f_{m-1}^2 - f_m^2) \right) + \eta \left( L_k - \frac{c}{2k_e} - \sum_{m=1}^{M} l_m f_m \right),
$$

(34)

where $\gamma, \beta, \mu = \{\mu_m\}, \eta$ are Lagrangian multipliers for constraints C4, C11, C5 and C3. C12 is used to verify feasibility of the obtained optimal $p_m, \forall m \in \{1, 2, \cdots, M\}$. Then, we can get the optimal solution presented in Theorem 2.

**Theorem 2:** The optimal solution of (33) is given as

$$
\mu_m = \frac{\mu_{m+1} \theta_{m+1}}{\theta_m} + l_m, \forall m \in \{2, 3, \cdots, M-1\}, \mu_M = l_M,
$$

(35)

$$
\eta 1 = \frac{l_1 \eta}{2k_v \left( l_1 + \mu_2 \theta_2 - \theta_1 \right)}.
$$

(36)

$$
f_m = \frac{l_m \eta}{2k_v (\mu_m - \mu_{m+1})}, \forall m \in \{2, 3, \cdots, M-1\},
$$

(37)

$$
f_M = \frac{\eta \theta_M}{2k_v},
$$

(38)

$$
p_1 = \frac{k_v f_1^2}{\theta_1},
$$

(39)

$$
p_m = p_{m-1} + \frac{k_v (f_m^2 - f_{m-1}^2)}{\theta_m}, \forall m \in \{2, 3, \cdots, M\}.
$$

(40)

The proof of Theorem 2 is shown in part G of Appendix.

March 1, 2022 DRAFT
The subgradient method is used to update Lagrangian multipliers $\eta$ and $\beta$.

\[
\eta^{(l+1)} = \left[ \eta^{(l)} + k^{(l)} \left( L_k^{(l)} - \frac{c^{(l)}}{2k_e} - \sum_{m=1}^{M} l_m f_m^{(l)} \right) \right]^+, \\
\beta^{(l+1)} = \left[ \beta^{(l)} + k^{(l)} \left( \frac{c^{(l)}_2}{2k_e} - c^{(l)} L_k^{(l)} \right) \right]^+, \\
\]

(41)

where $k^{(l)} = \frac{q}{\sqrt{l}}$ is the learning rate, $l$ is the number of iterations, and $q > 0$ is a small positive constant.

The optimal solution $f^*_m, p^*_m$ and $c^*$ of Stage I is selected from the optimal solutions of $N$ optimization subproblems, i.e., the one which makes $U_{RSU}$ largest. The other two cases $T_N < 2k_e f_e^{max} < T_1$ and $2k_e f_e^{max} \leq T_N$ can be solved in similar ways. Lagrangian multiplier method for Stage I is shown in Algorithm 2.

**Algorithm 2** Lagrangian multiplier method for the Stage I

1: Initialize Lagrangian multipliers $\eta^{(l)}, \beta^{(l)}$, set $l = 0, \mu_M = l_M$;

2: for $m = M - 1$ to 2 do
3: $\mu_m = \frac{\mu_{m+1} \theta_{m+1}}{\theta_m} + l_m$
4: end for

5: repeat
6: $f_1^{(l)} = \frac{l_1 \eta^{(l)}}{2k_v \left( 1 + \mu_2 \frac{\theta_2 - \theta_1}{\theta_1} \right)}$
7: $f_M^{(l)} = \frac{\mu_v^{(l)}}{2k_v}$
8: for $m = M - 1$ to 2 do
9: $f_m^{(l)} = \frac{l_m \eta^{(l)}}{2k_v (\mu_m - \mu_{m+1})}$
10: end for
11: $p_1^{(l)} = \frac{k_v f_1^{(l)2}}{\theta_1}$
12: for $m = 2$ to $M$ do
13: $p_m^{(l)} = p_m^{(l-1)} + \frac{k_v \left( f_m^{(l)2} - f_m^{(l-1)2} \right)}{p_m^{(l-1)}}$
14: end for
15: $c^{(l)} = \left[ c^{(l-1)} - \frac{\partial L}{\partial c^{(l-1)}} \right] \Delta k - 1$
16: Update $\eta^{(l+1)}, \beta^{(l+1)}$ according to (41);
17: $l = l + 1$
18: until $U_{RSU}$ converges

As already proved in [37], the convergence of Algorithm 2 can be guaranteed by adopting decreasing step sizes $k^{(l)}$ for (41). Since we define $k^{(l)} = \frac{q}{\sqrt{l}}$, the conditions in [37], i.e., $\sum_{l=1}^{\infty} k^{(l)} = \infty$ and $\sum_{l=1}^{\infty} (k^{(l)})^2 < \infty$, are satisfied. Therefore, Algorithm 2 converges to the optimal solution given in Theorem 2.

The algorithm for optimal solution of the multi-stage Stackelberg game and the contract-based incentive mechanism is shown below.
D. The computational complexity analysis

a) Stage I: The comparison of $T_k$ and $2k_e f_e^{\text{max}}$ will take $O(N)$ time. According to Theorem 2 and (\ref{th2}), we can see that the time complexity of one-step iteration of primal variables and Lagrangian multipliers is $O(M)$. We assume that when $\|U_l^{(l+1)} - U_l^{(l)}\|_{\text{RSU}} \leq \varepsilon$, Lagrangian multiplier method will stop. Then it can be seen that the number of iterations $l$ is $O(\ln \varepsilon)$ [38]. Since the optimal solution of Stage I is selected from the optimal solutions of $N$ optimization subproblems, the total time complexity of this Stage is $O(N + NM \ln \varepsilon) = O(NM \ln \varepsilon)$.

b) Stage II: In the worst case, the time complexity of Algorithm 1 depends on Step 3 and Step 4, namely $O(N^2)$.

c) Stage III: According to (14), the time complexity of solving $f^*_i$ is a constant, $O(1)$.

Based on the above analysis, our proposed algorithm is a polynomial-time algorithm, which is efficient.

Algorithm 3 the algorithm for optimal solution of the multi-stage Stackelberg game and the contract-based incentive mechanism

Step 1: Compare $T_k, \forall k \in \{1, 2, \cdots, N\}$ with $2k_e f_e^{\text{max}}$ and use Algorithm 2 or similar methods to obtain the RSU’s optimal solution $f^*_m, p^*_m$ and $c^*, \forall m \in \{1, 2, \cdots, M\}$;

Step 2: Use Algorithm 1 to find the MEC server’s optimal solution $f^*_{\text{RSU}}$ and $p^*$;

Step 3: According to (15), user $i$ determines whether to offload its task to the MEC server. If yes, the computing resources purchased by user $i$ is obtained based on (13).

VI. NUMERICAL RESULTS

A. Parameter Setting

In this section, we use numerical simulation to evaluate the performance of our proposed algorithm. In the simulation, we set $\delta = 1, C = 100$. We assume that there are $M = 10$ types of vehicles in this $200m \times 200m$ district. The probability of vehicles’ types and some other parameters are borrowed from [34]. We set the time interval $t = 20$ min and the value of type $m$ is defined as $\theta_m = (m \cdot t)/60, \forall m \in \{1, 2, \cdots, M\}$. The total computing resources of MEC server is $f_e^{\text{max}} = 300$GHz, while each user and vehicle has 1GHz local computing resources, i.e., $f^*_i = f^*_v = 1$GHz, $\forall i \in \mathcal{N}$. The maximum delay tolerance of user $i$, $t^*_i$ is assigned as a random number in $[0.5, 2]$ ms.

B. Convergence and Effectiveness

With $e = e_c = a = 60$, Fig. 2 shows the convergence of utility of RSU with different number of users, and demonstrates the convergence of Algorithm 2. Fig. 3 and Fig. 4 show the optimal contracts between the RSU and vehicles. As shown in Fig. 3 with the number of mobile device users rising, the computing resources that the RSU borrows from different types of vehicles also rise. Fig. 3 verifies Lemma 2, showing that the higher type a vehicle is, the more computing resources the RSU purchases. Fig. 4 shows that when the computing resources purchased by RSU increase, the expenses paid by RSU to each type of vehicles also increase. Fig. 4 demonstrates Lemma 1 as well, showing that the higher type a vehicle is, the more payment it gets.
Fig. 2. The convergence of utility of RSU with different number of users.

Fig. 3. The computing resources purchased by RSU with different number of users.

Fig. 4. The payment by RSU with different number of users.
Fig. 5. Utility of the RSU versus the number of users

Fig. 6. Utility of the MEC server versus the number of users

Fig. 5 and Fig. 6 reveal the relationship between the utility of RSU or MEC server and the number of users. Fixed the parameter $a$, it can be seen that with the number of users rising, the utility of RSU and MEC server also rises. However, when the number of users is fixed, with $a$ increasing, the utility of RSU and MEC server decreases.

Fig. 7 shows that, with $a$ increasing, the energy consumption of MEC server decreases. This means that the MEC server tends to use external resources. Fixed $a$, with the number of users rising, the energy consumption of MEC server also rises.

Fig. 8 shows the relationship between the number of users choosing to offload and the energy cost coefficient $a$. We can see that, fixed the number of users, when $a$ increases, since the energy consumption of MEC server decreases, users must pay more for computing resources and tend not to offload.
Fig. 7. Energy consumption of MEC server versus energy cost coefficient $a$

Fig. 8. The number of users choosing to offload versus the total number of users

C. Comparative Study

First, we compare our approach with the existing schemes for vehicular MEC offloading. That is, the pure MEC offloading with dedicated MEC server in [18], the cloud-MEC collaborative computation offloading in [19], where an MEC server and a cloud server coexist, and the autonomous MEC offloading using neighbouring vehicles’ idle computing resources in [24]. In table II, we present the performance comparison between our proposed mechanism and the existing pure MEC offloading, cloud-MEC collaborative offloading and autonomous MEC offloading, which shows the advantages of our proposed mechanism in offloading delay, energy consumption and cost.

Then, we compare our proposed optimal computing resource allocation scheme with random allocation scheme (RAS) and uniform allocation scheme (UAS). Given $N = 100$ users, RAS randomly allocates computing resources of the MEC server or vehicles, while in UAS, we have $f_{RSU}^{av} = \{0.08, 0.1\}$ GHz, $f_{MEC}^{av} = \{2.8, 3\}$ GHz for each user. As shown in Fig. 9, the energy consumption of MEC server in our scheme is lower than that in RAS and
TABLE II

PERFORMANCE COMPARISON BETWEEN OUR MECHANISM VS PURE MEC OFFLOADING [18], CLOUD-MEC COLLABORATIVE OFFLOADING [19], AND AUTONOMOUS MEC OFFLOADING [24]

| Compared Based On | Our Mechanism | Pure MEC Offloading | Cloud-MEC Collaborative Offloading | Autonomous MEC Offloading |
|-------------------|---------------|---------------------|-----------------------------------|---------------------------|
| Delay             | low           | medium              | high                              | low                       |
| Energy Consumption| low           | high                | medium                            | low                       |
| Availability      | almost always | within server’s range| always                            | always but not reliable   |
| Cost              | low           | medium              | high                              | low                       |

Fig. 9. Energy consumption of MEC server in different schemes

VII. CONCLUSION

In this paper, we propose a vehicular fog-edge computing network for computation offloading. In order to portray this scene, we formulate it as a multi-stage Stackelberg game and a contract-based incentive mechanism. We solve this complex three-stage optimization problem by the backward induction method. Then we obtain the optimal price and computing resource demand strategies in each stage. Numerical results demonstrate the effectiveness of our proposed incentive mechanism and trading mechanism, reveal the trends of energy consumption and offloading decisions on various parameters, and show the performance comparison between our work and the other existing MEC offloading framework in vehicular networks. Our future work will include two aspects. First, based on HVC framework, combined with specific vehicle-to-everything applications (e.g., sensor data sharing and processing, use cases of automatic driving or advanced driving), the proposed algorithm will be verified with real objects. Besides, taking the heterogeneity of mobile device users and of vehicles, a finer-grained task assignment and resource scheduling algorithm will be studied.
This work was supported by the National Key Research and Development Program of China (Grant No.2018YFB1702300), and in part by the NSF of China (Grants No. 61731012, 62025305, 61933009 and 92167205).

Appendix

A. Proof of Proposition 1

Proof: Obviously, the objective function of problem (18) is concave. Rewrite (18) as

\[
\min_{p, f_e, f_{RSU}} - \tau + \Delta p + k_e f_e^2 + c f_{RSU}
\]

s.t. \( C6 : p \geq 0 \)

\( C7 : 0 \leq f_e \leq f_{e, \text{max}} \) \hspace{1cm} (42)

\( C8 : f_{RSU} \geq 0 \)

\( \frac{\tau}{p} - \Delta = f_e + f_{RSU} \) \hspace{1cm} (43)

The Lagrangian function is defined as

\[
L = -\tau + \Delta p + k_e f_e^2 + c f_{RSU} + \alpha \left( \frac{\tau}{p} - \Delta - f_e - f_{RSU} \right) + \gamma (f_e - f_{e, \text{max}}) - \zeta p - \xi f_e - \nu f_{RSU}.
\]

According to the Karush-Kuhn-Tucker (KKT) conditions, we have

\[
\frac{\partial L}{\partial p} = \Delta - \frac{\alpha \tau}{p^2} - \zeta = 0, \hspace{1cm} (44)
\]

\[
\frac{\partial L}{\partial f_e} = 2k_e f_e - \alpha + \gamma - \xi = 0, \hspace{1cm} (45)
\]

\[
\frac{\partial L}{\partial f_{RSU}} = c - \alpha - \nu = 0, \hspace{1cm} (46)
\]

\[
\frac{\tau}{p} - \Delta - f_e - f_{RSU} = 0, \hspace{1cm} (47)
\]

\[
\gamma (f_e - f_{e, \text{max}}) = 0, \hspace{1cm} (48)
\]

\[
\zeta p = 0, \quad \xi f_e = 0, \quad \nu f_{RSU} = 0, \hspace{1cm} (49)
\]

\[ p, f_e, f_{RSU}, \alpha, \gamma, \zeta, \xi, \nu \geq 0. \] \hspace{1cm} (50)

Lemma 7: \( \zeta = 0 \)

Proof: Suppose \( \zeta \neq 0 \). Since \( \zeta p = 0 \), it follows that \( p = 0 \). However, if \( p = 0 \), then (47) does not hold. Thus, we have \( \zeta = 0 \). \( \blacksquare \)

Lemma 8: \( \xi = 0 \)

Proof: Suppose \( \xi \neq 0 \). Similarly, we have \( f_e = 0 \). If \( f_e = 0 \), from (48) we know \( \gamma = 0 \). From (45), we have \( \alpha = -\xi \). Substituting \( \alpha < 0 \) into \( \frac{\partial L}{\partial p} \), we can obtain that \( \frac{\partial L}{\partial p} > 0 \), which contradicts (44). Thus, \( \xi = 0 \). \( \blacksquare \)
Lemma 9: $\nu = 0$

Proof: Suppose $\nu \neq 0$. Then we have $f_{RSU} = 0$. It means that the MEC server does not buy any computing resource from RSU. Therefore, the price of RSU’s resources is $c = 0$. When $c = 0$, from (46) we can obtain that $\alpha = -\nu < 0$, which creates contradiction like Lemma 8. Thus, we have $\nu = 0$.

According to Lemma 7, 8 and 9, we can obtain the optimal solution of (42) by the KKT conditions easily. This problem is based on the assumption that all users choose to offload their tasks. Therefore, the price of MEC server’s computing resources must follow this condition:

$$p < \min \left\{ \frac{\tau_1}{f_1^1 + f_1^{thresh}}, \frac{\tau_2}{f_2^1 + f_2^{thresh}}, \ldots, \frac{\tau_N}{f_N^1 + f_N^{thresh}} \right\},$$

(51)

From (17), we can reduce this condition to $p < \frac{\tau_N}{f_N^1 + f_N^{thresh}}$. Then, from (19), we have $\sqrt{\frac{c \sum_{i=1}^{N-1} \tau_i}{\delta \sum_{i=1}^{N-1} f_i^1}} < \frac{\tau_N}{f_N^1 + f_N^{thresh}}$. Thus, it follows that $c < T_N$. Then, the optimal solution of (18) can be solved.

B. Proof of Proposition 2

Proof: The sufficiency part has been proved in the proof of Proposition 1. Now we consider the necessity part. Suppose $T_N \leq c < T_{N-1}$, and the optimal price of MEC server’s computing resources is $p^*$ given by (23). Since $c \geq T_N$, we have $p \geq \frac{\tau_N}{f_N^1 + f_N^{thresh}}$ and $f_N^* = 0$. Then (16) can be rewritten as

$$\max_{p, f_e, f_{RSU}} \sum_{i=1}^{N-1} (\tau_i - f_i^1 \delta p) - k_e f_e^2 - c f_{RSU}$$

s.t. C6–C8

$$\sum_{i=1}^{N-1} \left( \frac{\tau_i}{p} - f_i^1 \delta \right) = f_e + f_{RSU}$$

(52)

This problem has the same structure as (13). Therefore, from the proof of the sufficiency part, we can see that the optimal price of MEC server’s computing resources for this problem is

$$p^* = \sqrt{\frac{c \sum_{i=1}^{N-1} \tau_i}{\delta \sum_{i=1}^{N-1} f_i^1}}$$

(53)

We can also obtain the optimal quantity of resources purchased by the MEC server from RSU.

Obviously, the optimal solution of (52) is different from that of (13). This contradicts with our presumption. Thus, $p^*$ given by (19) is the optimal solution of (13) only if $c < T_N$. The necessity part is thus proved.

C. Proof of Lemma 3

Proof: According to IR conditions, we have

$$\theta_1 p_1 - k_e f_e^2 \geq 0.$$  

(54)

Due to IC conditions, we have

$$\theta_m p_m - k_e f_m^2 \geq \theta_m p_1 - k_e f_1^2, \forall m \in \{2, 3, \ldots, M\}.$$  

(55)
As we know, $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_M$, thus
\[ \theta_m p_m - k_v f_m^2 \geq \theta_1 p_1 - k_v f_1^2 \geq 0, \forall m \in \{2, 3, \cdots, M\}. \] (56)
That is, if the IR condition of type-1 vehicles is satisfied, the other IR conditions also hold.

\begin{enumerate}
\item[D.] \textbf{Proof of Lemma 4}

\textit{Proof:} According to IC conditions, we have
\[ \theta_{m+1} p_{m+1} - k_v f_{m+1}^2 \geq \theta_{m+1} p_m - k_v f_m^2, \] (57)
\[ \theta_m p_m - k_v f_m^2 \geq \theta_{m+1} p_m - k_v f_m^2. \] (58)

According to Lemma 1 and 2, we have
\[ \theta_{m+1} (p_m - p_{m-1}) \geq \theta_m (p_m - p_{m-1}). \] (59)

Combining (58) and (59), we get
\[ \theta_{m+1} p_m - k_v f_m^2 \geq \theta_{m+1} p_{m-1} - k_v f_{m-1}^2. \] (60)

Combining (57) and (60), we get
\[ \theta_{m+1} p_{m+1} - k_v f_{m+1}^2 \geq \theta_{m+1} p_{m-1} - k_v f_{m-1}^2. \] (61)

Recursively like this, we can prove Lemma 4.

\item[E.] \textbf{Proof of Lemma 5}

\textit{Proof:} Assume that an LDIC constraint is not active, \textit{i.e.}, for type-$m$ vehicles,
\[ \theta_m p_m - k_v f_m^2 > \theta_{m-1} p_{m-1} - k_v f_{m-1}^2. \] (62)

At this time, the RSU can gradually reduce the payment $p_m$ until both sides of (62) equal. This measure does not violate the LDIC constraints, but improves the utility of RSU. Therefore, under the optimal contract, all the LDIC constraints must be active.

Similarly, we can prove that the IR constraint of type-1 vehicles is active as well.

\item[F.] \textbf{Proof of Lemma 6}

\textit{Proof:} According to Lemma 1, we have
\[ \theta_m (p_m - p_{m-1}) > \theta_{m-1} (p_m - p_{m-1}). \] (63)

From Lemma 5, we get
\[ k_v (f_m^2 - f_{m-1}^2) = \theta_m (p_m - p_{m-1}). \] (64)

Then,
\[ k_v (f_m^2 - f_{m-1}^2) > \theta_{m-1} (p_m - p_{m-1}) \]
\[ \Rightarrow \theta_{m-1} p_{m-1} - k_v f_{m-1}^2 > \theta_{m-1} p_m - k_v f_m^2, \] (65)
i.e., the LUIC constraints hold.
G. Proof of Theorem 2

Proof: The first-order conditions of Lagrangian function (34) are

\[ \frac{\partial L}{\partial p_m} = \mu_{m+1} \theta_{m+1} - \mu_m \theta_m + l_m \theta_m = 0, \quad \forall m \in \{2, 3, \cdots, M - 1\}, \]  
(66)

\[ \frac{\partial L}{\partial p_1} = l_1 \theta_1 + \gamma \theta_1 + \mu_2 \theta_2 = 0, \]  
(67)

\[ \frac{\partial L}{\partial p_M} = l_M \theta_M - \mu_M \theta_M = 0. \]  
(68)

Combining (66) and (68), we can obtain (35). From (67), we know that \( \gamma = -l_1 - \mu_2 \frac{\theta_2}{\theta_1}. \)

\[ \frac{\partial L}{\partial f_m} = 2k_v (\mu_m - \mu_{m+1}) f_m - l_m \eta = 0, \quad \forall m \in \{2, 3, \cdots, M - 1\}, \]  
(69)

\[ \frac{\partial L}{\partial f_1} = -2k_v (\mu_2 + \gamma) f_1 - l_1 \eta = 0, \]  
(70)

\[ \frac{\partial L}{\partial f_M} = 2k_v \mu_M f_M - l_M \eta = 0. \]  
(71)

From (69), (70) and (71), we can obtain (37), (36) and (38).

According to the reduced IR constraint \( C_4 \), the solution of \( p_1 \) can be written as (39). Then, given the solution \( f_i^\star, \quad \forall i \in \{1, 2, \cdots, M\} \) and the reduced IC constraints \( C_5 \), we can get the optimal solution \( p_i^\star, \forall i \in \{1, 2, \cdots, M\} \), which is shown in (40).

However, the analytical solution of \( c \) can not be obtained directly. The gradient method can be used to update the Lagrangian multipliers to approximate to the optimal \( c \), i.e.,

\[ c^{(l)} = c^{(l-1)} - \frac{\partial L}{\partial c^{(l-1)}} T_k^{-1}. \]  
(72)

REFERENCES

[1] C. Wang, Y. Li, D. Jin, and S. Chen, “On the serviceability of mobile vehicular cloudlets in a large-scale urban environment,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2960–2970, 2016. DOI: [10.1109/TITS.2016.2561293](http://dx.doi.org/10.1109/TITS.2016.2561293).

[2] R. Yu, J. Ding, X. Huang, M. Zhou, S. Gjessing, and Y. Zhang, “Optimal resource sharing in 5g-enabled vehicular networks: A matrix game approach,” IEEE Transactions on Vehicular Technology, vol. 65, no. 10, pp. 7844–7856, 2016. DOI: [10.1109/TVT.2016.2536441](http://dx.doi.org/10.1109/TVT.2016.2536441).

[3] C. Shao, S. Leng, Y. Zhang, A. Vinel, and M. Jonsson, “Performance analysis of connectivity probability and connectivity-aware mac protocol design for platoon-based vanets,” IEEE Transactions on Vehicular Technology, vol. 64, no. 12, pp. 5596–5609, 2015. DOI: [10.1109/TVT.2015.2479942](http://dx.doi.org/10.1109/TVT.2015.2479942).

[4] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, and Y. Zhang, “Energy-efficient offloading for mobile edge computing in 5g heterogeneous networks,” IEEE Access, vol. 4, pp. 5896–5907, 2016. DOI: [10.1109/ACCESS.2016.2597169](http://dx.doi.org/10.1109/ACCESS.2016.2597169).
[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 Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016. DOI: [10.1109/TVT.2016.2532863](https://doi.org/10.1109/TVT.2016.2532863)

[6] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013. DOI: [10.1109/TWC.2013.072513.121842](https://doi.org/10.1109/TWC.2013.072513.121842)

[7] S. Sardellitti, G. Scutari, and S. Barbarossa, “Joint optimization of radio and computational resources for multicell mobile-edge computing,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015. DOI: [10.1109/TSIPN.2015.2448520](https://doi.org/10.1109/TSIPN.2015.2448520)

[8] C. You, K. Huang, H. Chae, and B. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017. DOI: [10.1109/TWC.2016.2633522](https://doi.org/10.1109/TWC.2016.2633522)

[9] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, “Joint computation offloading and user association in multi-task mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12313–12325, 2018. DOI: [10.1109/TVT.2018.2876804](https://doi.org/10.1109/TVT.2018.2876804)

[10] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016. DOI: [10.1109/JSAC.2016.2611964](https://doi.org/10.1109/JSAC.2016.2611964)

[11] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, “Delay-optimal computation task scheduling for mobile-edge computing systems,” in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1451–1455. DOI: [10.1109/ISIT.2016.7541539](https://doi.org/10.1109/ISIT.2016.7541539)

[12] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, “Joint computation offloading and interference management in wireless cellular networks with mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017. DOI: [10.1109/TVT.2017.2672701](https://doi.org/10.1109/TVT.2017.2672701)

[13] T. X. Tran and D. Pompili, “Joint task offloading and resource allocation for multi-server mobile-edge computing networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019. DOI: [10.1109/TVT.2018.2881191](https://doi.org/10.1109/TVT.2018.2881191)

[14] T. Koketsu Rodrigues, K. Suto, and N. Kato, “Edge cloud server deployment with transmission power control through machine learning for 6G internet of things,” *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2019. DOI: [10.1109/TETC.2019.2963091](https://doi.org/10.1109/TETC.2019.2963091)

[15] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, “Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective,” *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 38–67, 2020. DOI: [10.1109/COMST.2019.2943405](https://doi.org/10.1109/COMST.2019.2943405)

[16] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, “A scalable and quick-response software defined vehicular network assisted by mobile edge computing,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, 2017. DOI: [10.1109/MCOM.2017.1601150](https://doi.org/10.1109/MCOM.2017.1601150)
[17] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, *Vehicular edge computing and networking: A survey*, 2019. arXiv: [1908.06849](https://arxiv.org/abs/1908.06849) [eess.SP]

[18] V. Huy Hoang, T. M. Ho, and L. B. Le, “Mobility-aware computation offloading in mec-based vehicular wireless networks,” *IEEE Communications Letters*, vol. 24, no. 2, pp. 466–469, 2020. DOI: [10.1109/LCOMM.2019.2956514](https://doi.org/10.1109/LCOMM.2019.2956514)

[19] J. Zhao, Q. Li, Y. Gong, and K. Zhang, “Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019. DOI: [10.1109/TVT.2019.2917890](https://doi.org/10.1109/TVT.2019.2917890)

[20] H. Ke, J. Wang, L. Deng, Y. Ge, and H. Wang, “Deep reinforcement learning-based adaptive computation offloading for mec in heterogeneous vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7916–7929, 2020. DOI: [10.1109/TVT.2020.2993849](https://doi.org/10.1109/TVT.2020.2993849)

[21] Z. Xiao, X. Dai, H. Jiang, D. Wang, H. Chen, L. Yang, and F. Zeng, “Vehicular task offloading via heat-aware mec cooperation using game-theoretic method,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020. DOI: [10.1109/JIOT.2019.2960631](https://doi.org/10.1109/JIOT.2019.2960631)

[22] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, and D. Zhao, “A game-based computation offloading method in vehicular multiaccess edge computing networks,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4987–4996, 2020. DOI: [10.1109/JIOT.2020.2972061](https://doi.org/10.1109/JIOT.2020.2972061)

[23] B. Gu and Z. Zhou, “Task offloading in vehicular mobile edge computing: A matching-theoretic framework,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 100–106, 2019. DOI: [10.1109/MVT.2019.2902637](https://doi.org/10.1109/MVT.2019.2902637)

[24] G. Qiao, S. Leng, K. Zhang, and Y. He, “Collaborative task offloading in vehicular edge multi-access networks,” *IEEE Communications Magazine*, vol. 56, no. 8, pp. 48–54, 2018. DOI: [10.1109/MCOM.2018.1701130](https://doi.org/10.1109/MCOM.2018.1701130)

[25] J. Feng, Z. Liu, C. Wu, and Y. Ji, “Mobile edge computing for the internet of vehicles: Offloading framework and job scheduling,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 28–36, 2019. DOI: [10.1109/MVT.2018.2879647](https://doi.org/10.1109/MVT.2018.2879647)

[26] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang, “Reliable computation offloading for edge-computing-enabled software-defined iov,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7097–7111, 2020. DOI: [10.1109/JIOT.2020.2982292](https://doi.org/10.1109/JIOT.2020.2982292)

[27] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, “Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16067–16081, Dec. 2020, ISSN: 1939-9359. DOI: [10.1109/TVT.2020.3041929](https://doi.org/10.1109/TVT.2020.3041929)

[28] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, “Future intelligent and secure vehicular network toward 6g: Machine-learning approaches,” *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2020. DOI: [10.1109/JPROC.2019.2954595](https://doi.org/10.1109/JPROC.2019.2954595)

[29] F. Malandrino, C. Casetti, C. Chiasserini, C. Sommer, and F. Dressler, “The role of parked cars in content downloading for vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4606–4617, 2014. DOI: [10.1109/TVT.2014.2316645](https://doi.org/10.1109/TVT.2014.2316645)

[30] Z. Su, Q. Xu, Y. Hui, M. Wen, and S. Guo, “A game theoretic approach to parked vehicle assisted content delivery in vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6461–6474, 2017. DOI: [10.1109/TVT.2016.2630300](https://doi.org/10.1109/TVT.2016.2630300)
[31] G. Sun, M. Yu, D. Liao, and V. Chang, “Analytical exploration of energy savings for parked vehicles to enhance vanet connectivity,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1749–1761, 2019. DOI: [10.1109/TITS.2018.2834569](https://doi.org/10.1109/TITS.2018.2834569).

[32] J. Grover, “Vehicular fog computing paradigm: Scenarios and applications,” in *Vehicular Cloud Computing for Traffic Management and Systems*, IGI Global, 2018, pp. 200–215.

[33] X. Wang, Z. Ning, and L. Wang, “Offloading in internet of vehicles: A fog-enabled real-time traffic management system,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4568–4578, 2018. DOI: [10.1109/TII.2018.2816590](https://doi.org/10.1109/TII.2018.2816590).

[34] C. Li, S. Wang, X. Huang, X. Li, R. Yu, and F. Zhao, “Parked vehicular computing for energy-efficient internet of vehicles: A contract theoretic approach,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6079–6088, 2019. DOI: [10.1109/JIOT.2018.2869892](https://doi.org/10.1109/JIOT.2018.2869892).

[35] H. Sami, A. Mourad, and W. El-Hajj, “Vehicular-obus-as-on-demand-fogs: Resource and context aware deployment of containerized micro-services,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 778–790, 2020. DOI: [10.1109/TNET.2020.2973800](https://doi.org/10.1109/TNET.2020.2973800).

[36] Z. Hou, H. Chen, Y. Li, Z. Han, and B. Vucetic, “A contract-based incentive mechanism for energy harvesting-based internet of things,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6. DOI: [10.1109/ICC.2017.7997316](https://doi.org/10.1109/ICC.2017.7997316).

[37] B. Johansson, P. Soldati, and M. Johansson, “Mathematical decomposition techniques for distributed cross-layer optimization of data networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1535–1547, 2006. DOI: [10.1109/JSAC.2006.879364](https://doi.org/10.1109/JSAC.2006.879364).

[38] D. P. Bertsekas, “Nonlinear programming,” *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.