Correspondence

Dissatisfaction Feedback and Stackelberg Game-Based Task Offloading Mechanism for Parked Vehicle Edge Computing

Songxin Lei, Xinyao Guo, Junyi Li, Yixiao Wang, Yu Zhang, Lu Zheng, and Huaming Wu, *Senior Member, IEEE*

***Abstract*—Considering the limited computing power of Mobile Edge Computing (MEC) servers and the emergence of Vehicular Ad-Hoc Networks (VANETs), we employ the computing paradigm known as Parked Vehicle Edge Computing (PVEC) to leverage the computational capabilities of idle vehicles, thereby enhancing the overall computing performance of these vehicles. We establish a multi-stage Stackelberg game model, which captures the interactions among the requester (RV), the service provider (SP), and the parked vehicles (PV). In order to incentivize parked vehicles to provide computing power, we design a dissatisfaction feedback mechanism. To optimize the system and maximize the relative benefits of all stakeholders, we formulate an optimization problem to find an optimal pricing scheme that guides task allocation and resource utilization. We employ reverse induction and gradient descent to solve this problem. Simulation results demonstrate the effectiveness of the dissatisfaction feedback mechanism and provide insights into the changing trends of optimal strategies at each stage as the task density increases. These findings contribute to the understanding of PVEC and offer guidance for real-world task offloading scenarios.**

***Index Terms*—Parked Vehicle Edge Computing (PVEC), multi-stage Stackelberg game, task offloading, dissatisfaction feedback.**

I. Introduction

With the popularization of smart mobile devices and the development of computationally intensive applications, the task of enhancing device computing performance poses a significant challenge given the limitations of mobile device computing power. In recent years, Mobile Edge Computing (MEC) has offered a promising solution by offloading computationally intensive tasks to MEC servers, harnessing the benefits of heightened computational efficiency and reduced communication latency [1]. Nonetheless, the computing resources of edge servers remain limited, making it difficult to ensure the provision of Quality of Service (QoS) when offloading compute-intensive tasks to these constrained servers. Thus, it becomes imperative to augment the resource capacity of edge servers by effectively utilizing the idle resources of existing network entities. A noteworthy observation is the emergence of Vehicular Ad-Hoc Networks (VANETs), wherein vehicles possess abundant idle computing resources that can be harnessed to establish a novel computing paradigm for task offloading, known as Parked Vehicle Edge Computing (PVEC) [2]. This paradigm serves to enhance the communication and computing capabilities of urban areas.

Unfortunately, the process of offloading computing tasks to vehicles still poses several daunting challenging issues [3]. One prominent challenge arises from the fact that vehicles are not obliged to share their private resources, posing a significant obstacle to resource coordination. Additionally, in order to enable the widespread deployment of edge computing, energy optimization becomes a crucial consideration. As a result, effectively coordinating resources among parked vehicles (PV), service providers (SP), and requesters (RV) like vehicles and mobile device users, as well as designing transaction mechanisms that benefit all entities, present arduous tasks due to the intricate and multifaceted nature of their interactions.

In the field of MEC, numerous research advancements have been achieved through diverse research endeavors, employing a wide range of technical approaches [4], [5], [6], [7], [8]. For instance, Li et al. [9] proposed a contract-based incentive mechanism method, while Zhang et al. [10] put forth a method based on a multi-leader multi-follower Stackelberg game model. Ma et al. [11] developed a time-related trajectory prediction model based on the random forest model. Li et al. [12] explored an energy-efficient PVC paradigm, and designed a contract-based incentive mechanism. Nevertheless, there is room for optimization in these methods, particularly with regard to computational efficiency.

To tackle the challenges mentioned above, we introduce the PVEC paradigm to enhance its practical applicability. Regarding modeling, our goal is to construct a more effective task allocation model, thereby enabling RV, SP, and PV to achieve higher returns following the allocation process. Hence, we establish a multi-stage Stackelberg game model to effectively coordinate the allocation of computing resources and maximize the respective benefits of RV, SP and PV. Furthermore, within the context of incentivizing parked vehicles to contribute their computing capacities, we introduce a dissatisfaction feedback mechanism to fulfill this role. This mechanism exhibits a lower algorithmic complexity, enhancing the efficiency of task allocation when compared to the approach of establishing contracts with parked vehicles. The main contributions of this work are three-fold:

- The PVEC computing paradigm is formulated as a multi-stage Stackelberg game model, which takes into account a range of influencing factors. Moreover, a rational pricing strategy is developed aimed at guiding the task offloading process, ensuring the preservation of modeling accuracy while simultaneously reducing the overall computational complexity associated with the task offloading process.
- A dissatisfaction feedback mechanism is designed to incentivize parked vehicles to contribute computing power and reduce the algorithm complexity, while simultaneously ensuring the smooth progress of the game. By employing the reverse induction and gradient descent methods, this study offers analytical solutions for three stages.

Manuscript received 15 July 2023; revised 21 September 2023; accepted 21 October 2023. Date of publication 25 October 2023; date of current version 14 March 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62071327 and in part by Tianjin Science and Technology Planning Project under Grant 22ZYYYJC00020. The review of this article was coordinated by Prof. Shahid Mumtaz. *(Corresponding author: Huaming Wu.)*

Songxin Lei, Xinyao Guo, Junyi Li, and Lu Zheng are with the School of Mathematics, Tianjin University, Tianjin 300350, China (e-mail: lei_songxin@tju.edu.cn; 2897857464@qq.com; 1356065031@qq.com; 2871291139@qq.com).

Yixiao Wang and Huaming Wu are with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: wang_yixiao@tju.edu.cn; whming@tju.edu.cn).

Yu Zhang is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: 1198944008@qq.com).

Digital Object Identifier 10.1109/TVT.2023.3327395

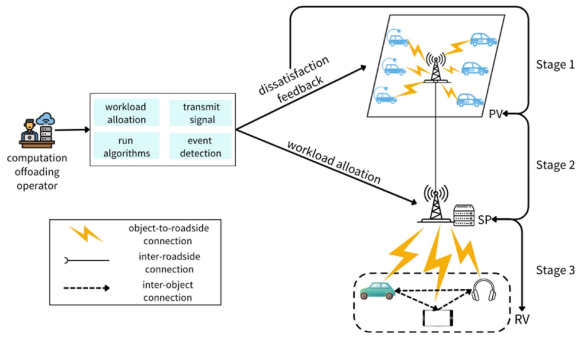


Fig. 1. Overall architecture of our PVEC system.

TABLE I
SYMBOL DEFINITIONS

Symbol	Definition
f_i^l	The local computing resources of user i
f_e	The computing resources of SP used for task offloading
f_i	The quantity of resources that user i purchases from SP
f_i^t	Threshold to determine whether user i should offload tasks
f_{PV}	The computing resources provided by parked vehicles
c	The price of computing resources collected by PV
p	The price of SP's computing resources
U_i	The utility of user i

- Through simulation experiments, we demonstrate the effectiveness of our approach by observing the trends of energy consumption and offloading decisions for users with varying parameters, illustrating its superior efficiency improvement and benefit maximization.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. Application Scenario

We focus on a parking lot that is equipped with wireless communication capabilities, which includes several key components, namely, the SP, the edge crowdsourcing platform integrated into PV and RV. The network architecture of the PVEC system that we have established is depicted in Fig. 1.

Suppose there are N mobile device users in this area, encompassing various devices such as smartphones, headphones, tablets, and smart cars. Each user is denoted by the index i, where $i \in \{1, \ldots, N\}$. We assume that the map of the parking lot is based on a grid system [2]. At each grid point, there exists a computationally intensive task described by a tuple $(c_i, d_i, t_i^{\mathrm{m}})$, where c_i represents the computational workload of the task, d_i represents the size of the input data, and t_i^{m} represents the maximum allowable time delay for completing the task. For convenience. Table I provides a summary of the key symbols used in our study.

We design a multi-stage Stackelberg game model alongside an incentive mechanism based on dissatisfaction feedback [13], for coordinating computing resources allocation and maximizing relative benefits among PV, SP, and RV. The model considers strategic decision-making, resource allocation, and pricing strategies at different stages of the PVEC process.

- *Stage 1:* PV acts as the leader and SP acts as the follower. Leveraging the dissatisfaction feedback mechanism, parked vehicles voluntarily contribute their computing resources to maximize their rental income.
- *Stage 2:* SP acts as the follower when engaging with PV as the leader. Simultaneously, SP takes on the role of the leader in the game where RV acts as the follower. In this stage, SP is responsible for determining the quantity of computing resources to be procured from PV. Additionally, SP communicates the unit price of computing resources to the requesting users.
- *Stage 3:* RV acts as the follower, and SP acts as the leader. All requesting users evaluate and determine whether to offload their computing tasks. Additionally, they make decisions regarding the quantity of computing resources to be acquired from the roadside server. Within our global game model, we manage the rental funds, ensuring control and oversight. In addition, the prices of computing resources are communicated to the roadside server by the PV.

For the practical implementation of our approach, we assume that all parked or slow-moving vehicles within the parking lot have the ability to establish a wireless connection with SP. Each vehicle is equipped with onboard units that possess storage, communication, and computing capabilities. It is important to note that the computing capability of these onboard units is uniform across all vehicles. To ensure fairness and efficiency in the transaction process, we introduce computing brokers. These brokers play a crucial role in collecting and transmitting information, executing algorithms, and scheduling tasks. They act as intermediaries between the users and the resource providers, facilitating smooth coordination and resource allocation. We aim to optimize the resource scheduling process and assist users in making informed decisions regarding the offloading of their computing tasks. This emphasis on transparency and fairness enhances the overall efficiency and effectiveness of the system.

B. Multi-Stage Stackelberg Game-Based Model

1) Stage 1: PV serves as the leader in the parked vehicle-edge server Stackelberg game and sets the price c for its computational resources provided to the edge server. The utility function can be defined as follows:

$$U_{PV}(c) = cf_{PV} - \frac{f_{PV}}{f_{PV}^{\max}}\alpha,\ c \geqslant 0,\ 0 \leqslant f_{PV} \leqslant f_{PV}^{\max}. \tag{1}$$

where the first term represents the rental income obtained by the SP, which corresponds to the benefit of the PVs participating in edge computing, the second term represents the loss in utility caused by user dissatisfaction, where α represents the impact factor of user dissatisfaction, and $f_{PV}^{\max}$ represents the maximum computational capacity of the PVs. $c \geq 0$ ensures that PV has the computational capacity and $0 \leq f_{PV} \leq f_{PV}^{\max}$ ensures that the computational capacity of PV should not exceed the maximum value. As the computational capacity of the PVs approaches its maximum value, the functionality of the PVs gradually saturates and decreases. This decrease in functionality leads to increased user dissatisfaction and has a negative impact on the utility of the SP. However, in the context of the Stackelberg game, the vehicles themselves, driven by the desire to maximize their own benefits, will inevitably attempt to increase the maximum value of their computational capacity. By doing so, they aim to mitigate the decrease in utility caused by the saturation of PV functionality. This behavior aligns with the concept of self-motivation, where the vehicles actively strive to enhance their capabilities to achieve their individual goals.

2) Stage 2: We define the utility function [3] as:

$$U_{SP} = \sum_{i=1}^{n} pf_i - ekf_e{}^2 - cf_{PV}, \tag{2}$$

where e represents the cost of unit energy, f_e represents the computing power, k is the energy conversion factor, and $ekf_e{}^2$ represents the total energy consumed by the SP, reflecting the utility consumed from an energy perspective [14], [15].

In summary, it determines f_e and f_{PV} based on the unit computing power pricing by the PV and passes the unit price p to RV. The problem can be described as follows:

$$\boldsymbol{\mathcal{P}_1}: \max_{p, f_e, f_{PV}} U_{SP}, \tag{3}$$

$$s.t.\ \ p \geqslant 0, \tag{4}$$

$$f_{PV} \geqslant 0, \tag{5}$$

$$0 \leqslant f_e \leqslant f_e^{\max}, \tag{6}$$

$$f_e + f_{PV} = \sum_{i=1}^{n} f_i. \tag{7}$$

3) Stage 3: To capture the various factors and dynamics involved in the task offloading process, such as the relationship between computational power and benefits, the cost of resource acquisition, and the impact of time delay, we consider the utility function for requesting vehicle i as:

$$U_i(f_i) = \frac{\theta}{t_i^m} \ln\left(\frac{f_i}{f_i^l} + \delta\right) - pf_i. \tag{8}$$

which consists of two terms, i.e., benefit value and cost of requesting tasks from SP [3], [16].

The requesting vehicle user i is a follower in the user-edge server Stackelberg game and determines the amount of computational power to purchase from the edge server based on the price p to maximize its own benefit. This problem can be described as:

$$\boldsymbol{\mathcal{P}_2}: \max_{f_i} U_i(f_i), \tag{9}$$

$$s.t.\ \ f_i > f_i^t, \tag{10}$$

$$\frac{c_i}{f_i^l} + \frac{d_i}{r_i} < t_i^m. \tag{11}$$

where $f_i^t = \frac{c_i}{c_i/f_i^l + d_i/r_i}$ is the offloading threshold, r_i represents the transmission rate of task i to the edge server, c_i/f_i^l and d_i/r_i are the computation time and the transmission time, respectively. Constraint (11) ensures that the total required delay for solving task i is less than the maximum tolerable delay.

III. Solutions

In this section, we solve the three-stage Stackelberg game model using a reverse inductive approach.

A. Solution for Stage 3

Proposition 1: $U_i(f_i)$ is a concave function, the optimal solution f_i^* of problem $\boldsymbol{\mathcal{P}_2}$ is

$$f_i^* = \begin{cases} \frac{\tau_i}{p} - f_i^l \delta & p < \frac{\tau_i}{f_i^l\delta + f_i^t} \\ 0 & p \geqslant \frac{\tau_i}{f_i^l\delta + f_i^t} \end{cases} \tag{12}$$

Proof: By taking the derivative of the objective function $U_i(f_i)$, it can be observed that it is a concave function. Therefore, we obtain the optimal solution f_i^* by solving for the zero points of the first derivative of the objective function.

By taking the derivative of the objective function $U_i(f_i)$,

$$\frac{\partial U_i(f_i)}{f_i} = \tau_i \frac{f_i^l}{f_i + f_i^l \delta} - p, \tag{13}$$

$$\frac{\partial^2 U_i(f_i)}{f_i^2} = -\tau_i \frac{f_i^l}{(f_i + f_i^l\delta)^2} < 0. \tag{14}$$

According to (13), it can be observed that U_i is a concave function. Therefore, we obtain the optimal solution f_i^* by solving for the zero points of the first derivative of the objective function in (14). ■

We can observe that if the price p becomes excessively high, users will no longer choose to upload their tasks for offloading and instead perform the computations locally.

B. Solution for Stage 2

By considering the task volume f_i assigned to the computational resources acquired by each user, we substitute the result obtained from (12) into the utility function of the SP. Given that the result can be expressed as a piecewise function, we can represent it using the indicator function denoted as χ_i. Consequently, we obtain an equivalent optimization problem as follows:

$$\boldsymbol{\mathcal{P}_3}: \max_{p, f_e, f_{PV}} U_{SP} = p\sum_{i=1}^{n} \chi_i f_i^l \delta + f_i^t + ekf_e^2 - cf_{PV} \tag{15}$$

$$s.t.\ \ p \geqslant 0, \tag{16}$$

$$f_{PV} \geqslant 0, \tag{17}$$

$$0 \leqslant f_e \leqslant f_e^{\max}, \tag{18}$$

$$f_e + f_{PV} = \sum_{i=1}^{n} \chi_i f_i^l \delta + f_i^t, \tag{19}$$

$$\chi_i = \begin{cases} 1, & p < \frac{\tau_i}{f_i^l\delta + f_i^t} \\ 0, & p \geqslant \frac{\tau_i}{f_i^l\delta + f_i^t} \end{cases} \tag{20}$$

where we can observe that χ_i is a discontinuous function of p, but once the values of χ_i are determined, the above problem becomes a convex programming problem with respect to (p, f_e, f_{PV}). Here, let us assume that the users can be sorted according to the following rules:

$$\frac{\tau_1}{f_1^l\delta + f_1^t} \geqslant \frac{\tau_2}{f_2^l\delta + f_2^t} \geqslant \cdots \geqslant \frac{\tau_n}{f_n^l\delta + f_n^t}. \tag{21}$$

Proposition 2: When $A_k \leq c < A_{k-1}$, corresponding to k users choosing to purchase computational resources, i.e., $\chi_i = 1$ (for $i \leq k$), the analytical expression for the optimal solution to (15) can be calculated as follows:

$$p^* = \sqrt{\frac{c\Gamma_k}{\delta F_k}}, \tag{22}$$

$$f_e^* = \min \frac{c}{2ke}, f_e^{\max}, \tag{23}$$

$$f_{PV}^* = \begin{cases} \left[\sqrt{\frac{\delta\Gamma_k F_k}{c}} - \delta F_k - \frac{c}{2ke}\right]^+, & c \leqslant 2ke \\ \left[\sqrt{\frac{\delta\Gamma_k F_k}{c}} - \delta F_k - f_e^{\max}\right]^+, & c > 2ke \end{cases} \tag{24}$$

where $\Gamma_k = \sum_{i=1}^{k} \tau_i$, $F_k = \sum_{i=1}^{k} f_i^l$, and $A_k = \frac{\tau_k\sqrt{\delta F_k}}{(f_k^l + f_k^t)\Gamma_k}$.

Proof: We can simplify the analysis by considering a specific scenario where $k = n$, implying that all users demonstrate a willingness to purchase resources.

Algorithm 1: Optimal Solution for Stage 2.

1 *Initialization:* k=N
2 *Sort all users according to Eq.* (21)
3 *Compute* $A_k = \frac{\tau_k \sqrt{\delta F_k}}{(f_k^l + f_k^t)\Gamma_k}$
4 *Compare* c *with* A_k
5 **if** $c \geqslant A_k$ **then**
6 set $k = k - 1$
7 *Compute* p^*, f_e^* *and* f_{PV}^* *based on Eqs.* (22), (23) *and* (24)

The Lagrangian function of the concave optimal programming function in (15) is defined as

$$\mathcal{L} = -\Gamma_n + \delta F_n p + ekf_e^2 + \alpha\left(\frac{\Gamma_n}{p} - \delta F_n - f_e - f_{PV}\right) + cf_{PV} + \beta(f_e - f_e^{\max}) - \gamma p - \mu f_e - \lambda f_{PV}, \tag{25}$$

where α, β, γ, μ and λ are the Lagrange multipliers.

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

$$\frac{\partial l}{\partial p} = \frac{\partial l}{\partial f_e} = \frac{\partial l}{\partial f_{PV}} = 0, \tag{26}$$

$$\frac{\Gamma_n}{p} - \delta F_n - f_e - f_{PV} = 0, \tag{27}$$

$$\beta(f_e - f_e^{\max}) = \gamma p = \mu f_e = \lambda f_{PV} = 0, \tag{28}$$

$$p, f_e, f_{PV}, \alpha, \beta, \gamma, \mu, \lambda \geqslant 0. \tag{29}$$

According to (26)–(29), we have

$$\gamma = \mu = \lambda = 0. \tag{30}$$

Then we can easily obtain the optimal solutions as shown in (22) and (23). It should be noted that the pricing of computing resources must adhere to a certain condition, as follows:

$$p < \min\left\{\frac{\tau_1}{f_1^l\delta + f_1^t}, \ldots, \frac{\tau_n}{f_n^l\delta + f_n^t}\right\} = \frac{\tau_n}{f_n^l\delta + f_n^t}. \tag{31}$$

Then from (22), we have $\sqrt{(c\Gamma_n)/(\delta F_n)} < \tau_n/(f_n^l\delta + f_n^t)$, which follows $c < A_n$. ■

The algorithmic process is as described in Algorithm 1.

C. Solution for Stage 1

1) Availability of Opportunistic Resources: In the PVEC paradigm, it is crucial to maintain a relatively stable state for vehicles participating in edge computing, ensuring that they remain parked in the parking lot for a specific duration of time. This enables them to fulfill the tasks they have accepted and downloaded. To address privacy concerns, the SP cannot directly access the precise departure time of a vehicle. However, it can monitor the duration of the vehicle's parking in the parking lot until a certain point in time. We can employ the probability of the vehicle remaining parked beyond a duration of T_0, which is denoted as the time required for a vehicle to complete edge computing.

We define $F(t) = \int_0^t f(x)dx (t < T^{\max})$ as the probability distribution function of the vehicle's parking time in the parking lot. Here,

Algorithm 2: Optimal Solution for Stage 1.

1 *Initialization:* $M = 0$, $k = N$
2 **for** $j = 1$ **to** M_0 **do**
3 Compute P_j;
4 **if** $P_j > P_0$ **then**
5 $M = M + 1$;
6 *Compare* A_k *with* $2kef_e^{max}$*;*
7 *Compute* c^**;*
8 **if** $A_{k-1} \leqslant c^* < A_k$ **then**
9 continue;
10 **else**
11 $k = k - 1$;

$f(t)$ represents the probability density function, and $T^{\max}$ denotes the maximum duration the vehicle can stay parked in the parking lot. This process typically follows a Poisson distribution [17]. Given the accumulated parking duration T_j for vehicle j up to the current moment, its conditional probability distribution function can be expressed:

$$F(t|t > T_j) = \frac{F(t, t > T_j)}{1 - F(T_j)} = \frac{F(t)}{1 - F(T_j)} \quad (t > T_j). \tag{32}$$

By taking the derivative with respect to t, we can derive the conditional probability density function as follows:

$$f(t|t > T_j) = \frac{f(t)}{1 - F(T_j)}. \tag{33}$$

Therefore, the probability P_j that vehicle j remains in the parking lot after a duration of T_0 is given by:

$$P_j = \int_{T_j}^{T_j + T_0} \frac{f(t)}{1 - F(T_j)} dt = \frac{1 - F(T_j + T_0)}{1 - F(T_j)}. \tag{34}$$

Setting P_0 as the threshold, when $P_j > P_0$, vehicle j is selected to join edge computing.

2) Optimal Solution of Stage 1: We assume that after the screening process mentioned earlier, M vehicles are selected to participate in edge computing. We distribute the task volume equally among the vehicles, denoted as $f_j = \frac{f_{PV}}{M}$. Considering the condition $A_{k-1} \leqslant c < A_k$, we substitute the result obtained from (22) into the utility function in (8) of each vehicle.

$$U_j(c) = c\frac{\sqrt{\frac{\delta\Gamma_k F_k}{c}} - \delta F_k - \frac{c}{2ke}}{M} - \alpha\frac{\sqrt{\frac{\delta\Gamma_k F_k}{c}} - \delta F_k - \frac{c}{2ke}}{f_{PV}^{\max}}. \tag{35}$$

By taking the derivative of this convex function, we can obtain its optimal solution:

$$c^* = ke\left(\frac{\alpha}{f_j^{\max}} + \sqrt{\frac{\delta\Gamma_k F_k}{c}} - \delta F_k - \frac{c}{2ke}\right). \tag{36}$$

We just need to verify that $A_{k-1} \leqslant c^* < A_k$.

The algorithmic process is as shown in Algorithm 2. For Stage 1, the time complexity depends on Step 2, namely, $\mathcal{O}(M)$. For Stage 2, according to (21), the time complexity of sorting users is $\mathcal{O}(N \log N)$. In the worst case, Step 4 will take $\mathcal{O}(N)$ time. For Stage 3, according to (14), the time complexity of solving f_i is a constant, i.e., $\mathcal{O}(1)$. This approach leads to a noteworthy reduction in computational complexity when compared to alternative iterative methods commonly used for solving the model.

Algorithm 3: Optimal Solution for the Multistage Stackelberg Game.

Set the necessary parameters in the environment;
Compare A_k *with* $2kf_e^{max}$ *according to* ***Algorithm 2****;*
Compute c^* *according to Eq.* (36)*;*
Compute p^*, f_e^*, f_{PV}^* *according to* ***Algorithm 1*** *;*
for $i = 1$ **to** N **do**
  Compute χ_i according to Eq. (20) ;
  if $\chi_i = 1$ **then**
    Compute f_i^* according to Eq. (12) ;

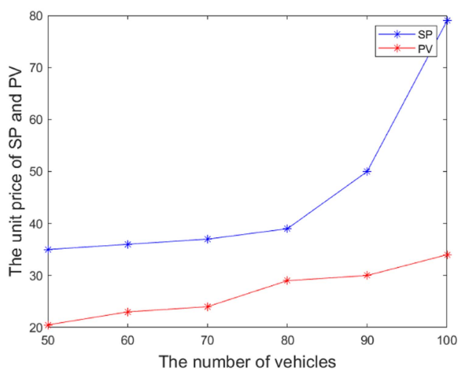


Fig. 2. Unit price of SP and PV vs. the number of vehicles.

D. Overall Three-Stage Game Model

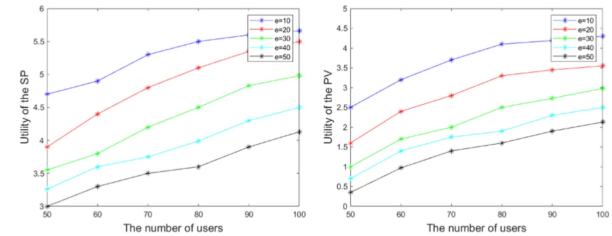
Indeed, the operational sequence of the entire game model is reversed compared to its construction process. To clarify, when provided with essential environmental parameters such as the task vehicle set, our initial step involves inserting the optimal solution c obtained from the optimization algorithm in Stage 1 into the Stage 2 algorithm. This enables us to further refine the solution and derive the optimal values p^*, f_e^*, f_{PV}^*. Likewise, the optimal solution obtained in Stage 2 is integrated into Stage 3, resulting in the determination of the optimal solution for f^*. This comprehensive approach enables us to establish the optimal pricing strategy and the optimal resource allocation scheme under this pricing strategy for the entire process.

IV. Performance Evaluation

A. Model Training

We consider a parking lot with limited space. The total number of available parked vehicles, denoted as M_0, is determined by filtering based on parking duration and can be acquired through statistical analysis of parked vehicle behavior [12]. Other parameters are adopted from [18]. The time interval is set to $t = 10$ min and $\delta = 1$. The computing frequency of the SP server is $f_e^{\max} = 300$ GHz, while the local computing frequency is 1 GHz. The maximum delay tolerance t_i^{m} for user i is randomly assigned within the range of [0.5, 2] ms.

Fig. 2 illustrates the optimal pricing strategies for the unit computing resources of SP and PV, taking into account various levels of traffic pressures. It is evident that as the number of vehicles rises while the server capacity remains unchanged, the unit prices of computing resources for both SP and PV also experience an increase. Conversely, in scenarios where the number of vehicles remains constant, the pricing strategy employed by SP surpasses that of PV in terms of cost.



(a) (b)

Fig. 3. Utility of SP and PV vs. the number of users. (a) SP. (b) PV.

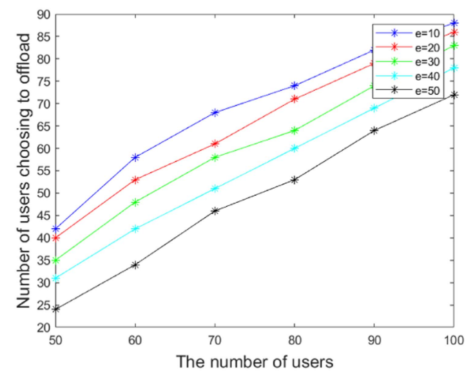
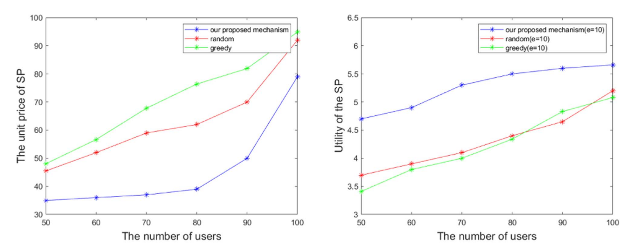


Fig. 4. Number of users choosing to offload.



(a) (b)

Fig. 5. Comparison of the unit price and utility of SP under three different algorithms with $e = 10$. (a) Price. (b) Utility.

Fig. 3 illustrates the relationship between the utility of the SP or PV and the number of users. With a fixed energy parameter e, it can be observed that as the number of users rises, the utility of both the SP and PV also improves. However, when the number of users remains fixed, the utility of the server declines as the energy parameter e increases. In addition, we also found that when the number of request users rises to a certain value, the benefit value of SP and PV will tend to a stable value.

Fig. 4 depicts the relationship between the number of users choosing to offload and the energy coefficient e. With a fixed number of users, the energy consumption decreases as e increases, and users tend to choose not to offload.

B. Comparative Experiment

To establish the superiority of our proposed algorithm, we conducted a comparative analysis against both the greedy algorithm and the randomized algorithm (Monte Carlo) in terms of pricing and utility. As depicted in Fig. 5, our algorithm exhibits a tendency toward lower pricing and higher utility when the number of requested users remains constant. This observation suggests the advantages and excellence of our algorithm in these aspects.

V. Conclusion and Future Work

This paper introduces the PVEC paradigm for task offloading through the utilization of parked vehicles. The approach is formulated as a multi-stage Stackelberg game model, which incorporates a feedback mechanism based on user dissatisfaction. Subsequently, we put forth a model-solving algorithm that leverages the reverse induction and gradient descent methods. Finally, we derive the pricing and computational resource allocation strategies for each stage. The experimental simulation results vividly illustrate how the optimal pricing strategy dynamically evolves with increasing task request intensity. Furthermore, our findings offer valuable insights into the interplay between task offloading preferences and the utilities of PV and SP, serving as a guide for real-world task offloading scenarios.

From a practical perspective, our model considers numerous influencing factors and employs various functions to characterize their influence on efficiency. This leads to complex expressions for benefit functions, necessitating the use of gradient descent for model optimization. Additionally, the absence of physical experiments contributes to a limited amount of experimental data, making it challenging to employ deep learning algorithms. In future work, we intend to improve the model and incorporate federated learning strategies to further boost the algorithm's performance. Simultaneously, as hardware capabilities continue to advance, we plan to explore the feasibility of creating a virtual parking lot and deploying the algorithm on edge servers for conducting physical experiments.

References

[1] W. Fan, J. Liu, M. Hua, F. Wu, and Y. Liu, "Joint task offloading and resource allocation for multi-access edge computing assisted by parked and moving vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5314–5330, May 2022.

[2] F. Zeng, R. Rou, Q. Deng, and J. Wu, "Parked vehicles crowdsourcing for task offloading in vehicular edge computing," *Peer-to-Peer Netw. Appl.*, vol. 16, pp. 1803–1818, 2023.

[3] Y. Li, B. Yang, H. Wu, Q. Han, C. Chen, and X. Guan, "Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-Stackelberg approach," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15969–15982, Sep. 2022.

[4] Q. He et al., "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 3, pp. 515–529, Mar. 2020.

[5] J. Du et al., "Resource pricing and allocation in MEC enabled blockchain systems: An A3C deep reinforcement learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 33–44, Jan./Feb. 2022.

[6] X. Ren et al., "AI-bazaar: A cloud-edge computing power trading framework for ubiquitous AI services," *IEEE Trans. Cloud Comput.*, vol. 11, no. 3, pp. 2337–2348, Jul.–Sep. 2023.

[7] Q. He et al., "A game-theoretical approach for mitigating edge DDoS attack," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2333–2348, Jul./Aug. 2022.

[8] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9517–9529, Oct. 2020.

[9] Y. Li, B. Yang, Z. Chen, C. Chen, and X. Guan, "A contract-Stackelberg offloading incentive mechanism for vehicular parked-edge computing networks," in *Proc. IEEE 89th Veh. Technol. Conf.*, 2019, pp. 1–5.

[10] J. Zhang, X. Huang, and R. Yu, "Optimal task assignment with delay constraint for parked vehicle assisted edge computing: A Stackelberg game approach," *IEEE Commun. Lett.*, vol. 24, no. 3, pp. 598–602, Mar. 2020.

[11] C. Ma, J. Zhu, M. Liu, H. Zhao, N. Liu, and X. Zou, "Parking edge computing: Parked-vehicle-assisted task offloading for urban VANETs," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9344–9358, Jun. 2021.

[12] 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 Things J.*, vol. 6, no. 4, pp. 6079–6088, Aug. 2019.

[13] R. Yu, J. Ding, X. Huang, M.-T. Zhou, S. Gjessing, and Y. Zhang, "Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7844–7856, Oct. 2016.

[14] 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 Trans. Netw.*, vol. 28, no. 2, pp. 778–790, Apr. 2020.

[15] 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 Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6461–6474, Jul. 2017.

[16] 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.

[17] K. Nguyen, S. Drew, C. Huang, and J. Zhou, "Collaborative container-based parked vehicle edge computing framework for online task offloading," in *Proc. IEEE 9th Int. Conf. Cloud Netw.*, 2020, pp. 1–6.

[18] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: Exploiting opportunistic resources for distributed mobile applications," *IEEE Access*, vol. 6, pp. 66649–66663, 2018.