
Task Offloading Optimization in Vehicular Edge Computing Based on Vehicle Mobility Analysis

Yarui Li, Feng Zeng*, Qiao Chen

School of Computer Science and Technology, Central South University, Changsha Hunan 410083, China
fengzeng@csu.edu.cn

Abstract: Vehicular Edge Computing (VEC), as a promising new paradigm, can improve the QoS of vehicular applications through computation offloading. However, with the emergence of more and more computation-intensive and delay-sensitive vehicular applications, VEC servers are facing the challenge of resource limitation, and the fast movement of vehicles will lead to the switching of task uploading. In this paper, based on the analysis of the task offloading switching between VEC servers due to the moving of vehicles, we propose a nonlinear programming model for joint optimization of delay, energy consumption, payment for vehicle users. Then, we solve the problem based on the KKT conditions to obtain the task offloading optimization strategy. Simulation results show that the proposed strategy can ensure that the vehicles perform task offloading switching efficiently between the VEC servers, and can reduce the average completion time of task offloading and the total cost of vehicle users.

Keyword: Vehicular edge computing, Task offloading, Switching, Mobility

1 Introduction

In recent years, there are more and more emerging vehicular applications such as autopilot, smart cockpits, and so on. These vehicular applications need high computing power, high network bandwidth, and low latency. Since cloud servers are far away from vehicles and large amounts of data should be uploaded and processed, cloud computing are difficult to meet the Quality of Service (QoS) requirements for those smart vehicular applications. To this end, Vehicular Edge Computing (VEC) emerges as the important technology to deal with the limitation of the on-board computing, and the main idea is to provide computing services for vehicles at the edge of vehicle network [1]. With the support of the adjacent edge servers, the vehicles can obtain timely and efficient computing services.

In a vehicular edge network, the vehicle offloads tasks to the edge server via wireless communication with the Road Side Unit (RSU) for efficient computing services. Due to the fast moving of vehicles and the limitation of wireless communication range, vehicles need to go through multiple RSUs to complete task offloading. When the vehicle which is uploading the task leaves the wireless signal coverage of current RSU, the vehicle will stop the current uploading and switch to the next VEC server to continue the uploading. The uploading interruption caused by the mobility of the vehicle may lead to the failure of task offloading or the delay of task processing, which is a challenge research topic in VEC.

In this paper, we consider the task offloading delay, energy consumption and the payment of vehicle users, analyze the switching process of task uploading between multiple VEC servers, and propose an optimization model for task offloading with minimum cost. Based on the KKT conditions, we solve the problem and obtain the task offloading optimization strategy. Simulation results show that the proposed



scheme can ensure that vehicle users perform switching of task uploading efficiently, reduce the whole delay of task offloading and the total cost of vehicle users.

2 Related Work

In recent years, as a promising technology, vehicular edge computing (VEC) can extend the computing power to the edge of vehicular network and provide computation service for vehicles, which has abstracted the attention of many scholars.

In VEC, in order to obtain high quality of service, vehicle users can offload the computation tasks to the VEC servers. There are many research works on the problem of task offloading in VEC. Sun et al [2] introduced one kind of decision-making and scheduling problem of task offloading, and described the problem as a mixed integer nonlinear programming problem. The heuristic and genetic algorithms were proposed to solve the problem, which reduced the delay and energy consumption of task processing and improved the offloading efficiency in VEC. A context-aware offloading scheme in an opportunistic vehicular fog computing framework was presented by Rahman et al [3], which took into account the variation of vehicle speed, direction and position, allowing vehicles to take advantage of nearby opportunistic offloading to support a vehicle-to-vehicle (V2V) computation offloading which effectively solved the capacity limitation problem at RSU and provided a sustainable computing environment for vehicle users. In order to decrease the task execution delay, Zeng et al [4] proposed a new vehicle edge computing framework based on software defined networks, which introduced the reputation to measure the contribution of each vehicle. They designed the interaction process as a kind of incentive mechanism based on reputation via using Stackelberg game modeling, and proposed a genetic optimization algorithm to quickly obtain the optimal strategy for both sides of the game.

Some researchers analyze the impact of the mobility of vehicles on task offloading. Zhang et al [5] described task offloading as a finite level Markov decision process (MDP). Considering the uncertain transition probability under real circumstances, they presented a concrete expression of transition probability and proposed a robust time-aware task offloading algorithm, then further proved that the proposed algorithm can reduce the delay of task offloading even under the uncertain high transition probability. Considering the mobility of vehicles, Liu, Li and Sun [6] studied the problem of task offloading in a dynamic environment with certain resource and delay constraints. Based on one-to-one and one-to-many matching algorithms, task assignment was studied on three different speed models (straight road, urban road with traffic lights and curved road). Their research work reduced the total network delay of task offloading. Li et al [7] proposed an auxiliary slice network structure, which utilized Mobile Edge Computing (MEC) to host some network services. They presented a traffic scheduling strategy, designed a new flow scheduling mechanism, considering the requirements of high mobility and reliability between vehicles.

The VEC servers mentioned above are always assumed to have sufficient resources, however, during peak demand periods, VEC servers tend to become congested due to the limited resources, resulting in a decline of the QoS. To this end, some researchers took into account the dynamic management of resources, and studied resource allocation optimization issues. Salahuddin et al [8] proposed a resource allocation scheme based on vehicle cloud reinforcement learning technology to minimize the supply cost of vehicle cloud resources. In order to improve the computing power of vehicle users, some researchers also considered the factors of user behavior relationship between vehicles. For example, Zhang et al [9] used unmanned aerial vehicle (UAV) to assist social networking of vehicles, and the combination of social content caching and wireless resource scheduling was considered to explore the dynamic resource allocation problem of energy perception, the dynamic power allocation of fixed vehicles was optimized by using Search Algorithm.

In order to take the use of idle vehicular resources, Zeng et al [10] studied how to effectively and economically utilize the idle resources in volunteer vehicles to handle the overloaded tasks in VEC servers, and proposed a fast searching algorithm based on genetic algorithm to find the best pricing strategy for the VEC server. Wang et al [11] mentioned that a large number of parking vehicles in the parking lots have underutilized resources which can be used to assist the content provider with collaborative caching popular content. However, because of the mobility of the vehicle in real time, the communication between the requesting vehicle and the parked vehicles may be unsustainable. In contrast, it is considered in this paper that stable communication can be maintained between short distance vehicles under current technical conditions. While due to the high mobility of the vehicle, if the vehicle users move out of the current RSU's coverage and do not complete the uploading, it will cause the interruption of task offloading and decrease the QoS of VEC. Therefore, we need to deeply analyze the process of task offloading while the moving of vehicles, and design the optimal scheduling of task.

3 System Description

We assume that the system deploys m RSUs on the road side, and each RSU is connected with a VEC server through the wired link. The set of VEC servers is denoted as $R = \{r_1, r_2, \dots, r_m\}$. Each server $r_i (1 \leq i \leq m)$ has undertaken some computation tasks from vehicle users within its communication range, and the set of vehicle users is denoted as $V = \{v_1, v_2, \dots, v_n\}$, as is shown in figure 1. It is assumed that each vehicle user has large data volumes and latency-sensitive tasks, these tasks can be offloaded to the VEC servers or executed on local devices. It is supposed that all RSUs have the same wireless transmission range, but as the location changing of the vehicles, the vehicles should switch from one RSU to another for data uploading, since each RSU's wireless signal can cover the limited area. Therefore, it is necessary to consider the switching problem between the servers while task offloading. That is, when the vehicle user travels from the coverage of the VEC server r_i to the coverage of the VEC server r_j , the vehicle user's task offloading process will change accordingly. This model will be described in more detail below.

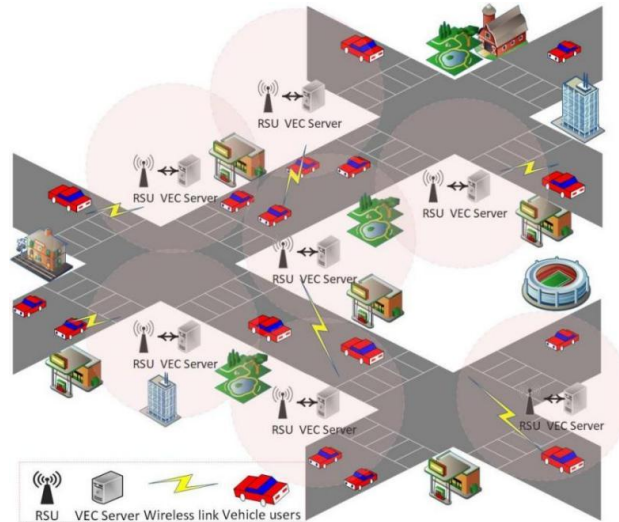


Fig.1. Vehicular Edge Network

1) VEC Group

Every RSU is installed around the road, and is equipped with a VEC server. When the vehicle user offloads the task to the server, the server will assign resources to the vehicle user based on the requirements. Multiple VEC servers can share their idle resources and cache content, constitute a resource sharing pool, a VEC group. If the current server does not have enough resources to complete the task, it can request other

servers in the group to obtain idle resources to handle overloaded tasks. At the same time, when the task requested by the vehicle user is cached by another server, the current server can also get cache content from other servers.

2) Vehicle User

The vehicle is equipped with wireless communication devices such as global positioning, Bluetooth, WiFi, with certain task processing capabilities. When the vehicle does not have enough resources to handle the task, it can request the VEC server to provide computation service. If the server accepts task processing request from the vehicle user, it will assign a certain amount of resources for the vehicle user, and the vehicle user needs to pay a certain fee to the server to obtain a good task offloading service [12].

3) Vehicular Task

For task w_k of vehicle user v_k ($1 \leq k \leq n$), it can be represented by $w_k = \{u_k, c_k\}$, where u_k is the amount of task size, and c_k is the execution ability required to process the task, measured with the CPU cycles. Taking the image identification task as an example, w_k is the image file size, and c_k is the amount of CPU cycles required to finish the image processing. Vehicular task w_k can be executed on a local vehicular device or a VEC server, but cannot be executed simultaneously on the two.

4 Task Offloading Process

Considering the data uploading switching between VEC servers, we introduce the process of task offloading. The uploading delay, energy consumption, payment and cache policy are taken into account in our analysis of task offloading.

1) Task Uploading Delay

When the vehicle user moves from the cover area of the current VEC server r_i to the cover area of the next VEC server r_{i+1} , it is necessary to perform a switching of the vehicular task. Vehicle users unload tasks through RSU to the edge server which is directly connected with the RSU via wired link, and the wireless channel between the vehicle and the RSU is affected by radio interference and the noise. According to radio theory, the signal-to-interference-plus-noise ratio (SINR) between the vehicle users v_k and the RSU can be represented as

$$\Lambda_{k,i} = \frac{p_{k,i}h_{k,i}}{\sum_{v_x \in \{V \setminus v_k\}} p_{x,i}h_{x,i} + I_{k,i}}, \quad (1)$$

in which $\sum_{v_x \in \{V \setminus v_k\}} p_{x,i}h_{x,i}$ is the interference of other vehicle users (v_x) to v_k , $p_{k,i}$ represents wireless link transmission power of vehicle user v_k , and $I_{k,i}$ represents communication background noise. It is assumed that the task unloading bandwidth B_i is the same for every vehicle within the same RSU coverage, therefore, the transmission rate of vehicle user v_k in RSU i can be represented by $g_{k,i} = B_i \log(1 + \Lambda_{k,i})$. The transmission delay between the VEC server and its connected RSU is negligible, so the transmission delay of vehicle user v_k directly unloading the task to VEC server r_i is

$$t_{k,i}^{up} = \frac{\varphi_{k,i}u_k}{g_{k,i}}, \quad (2)$$

where $\varphi_{k,i} \in [0,1]$ represents the proportion of the task required to handle within the current VEC server coverage, and the current amount of task unloaded at server r_i is $\varphi_{k,i}u_k$.

Assumed that the vehicle user completes the task uploading after driving from the first RSU to the j th RSU, if the server r_j does not store any data related to task w_k (not caching), the server r_j needs to execute the task. Let the VEC server r_j can provide the processing power as $f_{k,j}$, then the time of task execution is

$$t_k^{ex} = \frac{c_k}{f_{k,j}}. \quad (3)$$

Since the data of output is much less than the amount of task uploaded by the vehicle user, the time taken by the vehicle user to download the execution results can be ignored. Hence, the total processing time for vehicle users to complete task offloading is

$$t_k^{off} = t_k^{up} + t_k^{ex} = \sum_{i=1}^j \left(\frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + \frac{c_k}{f_{k,j}}. \quad (4)$$

2) Energy Consumption of Vehicle Users

The energy consumption of the vehicle user v_k comes from wireless transmission while task unloading to the server, which is denoted as

$$Q_{k,i} = p_{k,i} t_{k,i}^{up} = p_{k,i} \frac{\varphi_{k,i} u_k}{g_{k,i}}. \quad (5)$$

3) Payment for Task Offloading

The vehicle should pay the data uploading and task execution for VEC servers. The uploading payment has relation with transmission rate, and we assume the unit cost of the server r_i to receive the uploaded data is $q_{k,i}$, the unit task executing cost is $s_{k,j}$. The total payment of v_k for task offloading can be expressed as:

$$E_k = \sum_{i=1}^j (q_{k,i} g_{k,i}) + s_{k,j} f_{k,j}. \quad (6)$$

4) Cache Policy

In order to provide a better service experience for vehicle users, the VEC servers will cache some popular and task-related data. When the task requested by the vehicle user is cached on the server, the result can be directly got without any processing operation. At this time, the vehicular task execution time $t_k^{ex} = 0$, and the vehicle user v_k pays the execution fee for VEC server $s_{k,j} f_{k,j}$ is also 0, shown in (7):

$$\eta_{k,j} = \begin{cases} 1 & w_k \text{ is cached} \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where $\eta_{k,j} \in \{0,1\}$ indicates whether there is a cache w_k on the VEC server r_j .

5) Local Processing Model

If the total cost of the task offloading is higher than that of the local computation processing, the vehicle user will perform the vehicular task locally. The local task execution time, energy, and currency consumption are

$$t_{k,loc} = \frac{\sum_{i=1}^j (\varphi_{k,i}) c_k}{f_{k,loc}}, \quad (8)$$

$$Q_{k,loc} = \zeta_k \sum_{i=1}^j (\varphi_{k,i}) c_k f_{k,loc}^2, \quad (9)$$

$$E_{k,loc} = s_{k,loc} f_{k,loc}, \quad (10)$$

where ζ_k is coefficient related to the hardware architecture of vehicular device [13], $f_{k,loc}$ is local processing capacity of vehicle user, $s_{k,loc}$ is unit cost for vehicle users.

6) Objective (Cost) of Task Offloading

In this paper, we consider the joint optimization of the delay, energy consumption and currency payment of task offloading for the vehicle users, and the objective of task offloading for vehicle users is to minimize the delay, energy consumption and the payment for VEC servers. Let $\alpha, \beta, \gamma \in [0,1]$ respectively indicate that the vehicle user's attention to delay, energy and payment. For vehicle user v_k and its task w_k , we define the objective function U_k as (11), and U_k is also called as the cost of task offloading hereafter.

$$U_k = (1 - \theta_k) \left(\alpha \frac{\sum_{i=1}^j (\varphi_{k,i}) c_k}{f_{k,loc}} + \beta \zeta_k \sum_{i=1}^j (\varphi_{k,i}) c_k f_{k,loc}^2 + \gamma s_{k,loc} f_{k,loc} \right) + \theta_k \left(\alpha \left(\sum_{i=1}^j \left(\frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + (1 - \right.$$

$$\eta_{k,j}) \frac{c_k}{f_{k,j}}) + \beta \sum_{i=1}^j \left(p_{k,i} \frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + \gamma \left(\sum_{i=1}^j (q_{k,i} g_{k,i}) + (1 - \eta_{k,j}) s_{k,j} f_{k,j} \right) \quad (11)$$

In (11), $\theta_k = 0$ indicates that the task is executed in local device, $\theta_k = 1$ indicates that the vehicle user offloads the task to the VEC server, and due to the moving of vehicle v_k , it is supposed that the task offloading is processing from RSU l area to RSU j area.

5 Problem Description and Solution

5.1 Problem Description

In VEC, the computing capability and storage space of the VEC server are limited, and we assume that the maximum computing capability of the VEC server r_j is F_j^{max} . For ease of understanding, Table I shows the main parameters used herein. The joint optimization of offloading delay, energy consumption and the payment can be modeled as:

$$\min_{\{f_{k,j}, \varphi_{k,i}, \theta_k\}} \sum_{k=1}^n \left((1 - \theta_k) \left(\alpha \frac{\sum_{i=1}^j (\varphi_{k,i}) c_k}{f_{k,loc}} + \beta \zeta_k \sum_{i=1}^j (\varphi_{k,i}) c_k f_{k,loc}^2 + \gamma s_{k,loc} f_{k,loc} \right) + \theta_k \left(\alpha \left(\sum_{i=1}^j \left(\frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + (1 - \eta_{k,j}) \frac{c_k}{f_{k,j}} \right) + \beta \sum_{i=1}^j \left(p_{k,i} \frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + \gamma \left(\sum_{i=1}^j (q_{k,i} g_{k,i}) + (1 - \eta_{k,j}) s_{k,j} f_{k,j} \right) \right) \right) \quad (12)$$

$$\text{subject to C1: } f_{k,j} > 0 \quad (13\text{-a})$$

$$\text{C2: } \sum_{k=1}^n f_{k,j} \leq F_j^{max} \quad (13\text{-b})$$

$$\text{C3: } \sum_{i=1}^j \varphi_{k,i} = 1 \quad (13\text{-c})$$

$$\text{C4: } \theta_k = \{0,1\} \quad (13\text{-d})$$

$$\text{C5: } \eta_{k,j} = \{0,1\} \quad (13\text{-e})$$

$$\text{C6: } \alpha + \beta + \gamma = 1 \quad (13\text{-f})$$

In above model, constraints C1 and C2 ensure that the computing capability $f_{k,j}$, which is required by task execution, is a positive value, but the sum of computing capability cannot exceed the maximum capability F_j^{max} . C3 means that the vehicle user can handle the vehicular task after switching over j VEC servers, C4 means that the vehicle user can only execute the task in local device or offload the task to a VEC server, and C5 represents whether the VEC server caches the task.

Table I Main Notation Used In This Paper

Parameter	Meaning
R	the set of VEC servers
V	the set of Vehicle users
u_k	the size of task w_k
c_k	the CUP cycles for the execution of task w_k
$\varphi_{k,i}$	proportion of the data uploaded by v_k to r_i
$g_{k,i}$	transmission rate of v_k to r_i
$f_{k,j}$	the CUP cycles required by r_j to execute w_k
$f_{k,loc}$	v_k 's local processing power
$q_{k,i}$	unit uploading cost
$s_{k,j}$	unit computation cost

$\eta_{k,j}$	whether w_k cached on the VEC server r_j
θ_k	whether to perform tasks locally
α, β, γ	weight variable

5.2 Problem Solving

When offloading the task, the vehicle user determines the task offloading policy (θ_k^*), and the vehicle user v_k can complete task offloading after traveling through the VEC servers $r_1, r_2, \dots, r_i, \dots, r_j$. With the given policy θ_k^* , the joint optimization problem can be shown as follows:

$$\min_{\{f_{k,j}, \varphi_{k,i}, \theta_k^*\}} \sum_{k=1}^n \left((1 - \theta_k^*) \left(\alpha \frac{\sum_{i=1}^j (\varphi_{k,i}) c_k}{f_{k,loc}} + \beta \zeta_k \sum_{i=1}^j (\varphi_{k,i}) c_k f_{k,loc}^2 + \gamma s_{k,loc} f_{k,loc} \right) + \theta_k^* \left(\alpha \left(\sum_{i=1}^j \left(\frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + (1 - \eta_{k,j}) \frac{c_k}{f_{k,j}} \right) + \beta \sum_{i=1}^j \left(p_{k,i} \frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + \gamma \left(\sum_{i=1}^j (q_{k,i} g_{k,i}) + (1 - \eta_{k,j}) s_{k,j} f_{k,j} \right) \right) \right) \quad (14)$$

$$\text{subject to C1: } f_{k,j} > 0 \quad (15\text{-a})$$

$$\text{C2: } \sum_{k=1}^N f_{k,j} \leq F_j^{max} \quad (15\text{-b})$$

$$\text{C3: } \sum_{i=1}^j \varphi_{k,i} = 1 \quad (15\text{-c})$$

$$\text{C4: } \alpha + \beta + \gamma = 1 \quad (15\text{-d})$$

The above optimization problem is a multivariate nonlinear programming problem, we have the following Lemma.

Lemma 1 Given the uploaded ratio of VEC servers ($\varphi_{k,i}^*$), the optimization problem (14) is a convex optimization problem about $f_{k,j}$. Assumed that the vehicle user's policy set is \mathbf{f}_k , the method of the KKT condition can be used to obtain the optimal solution.

Proof: First, according to (14), the Lagrange function of the above problem can be obtained, as shown in (16).

$$\begin{aligned} L(\mathbf{f}_k, \mu) = & \sum_{k=1}^n \left((1 - \theta_k^*) \left(\alpha \frac{\sum_{i=1}^j (\varphi_{k,i}) c_k}{f_{k,loc}} + \beta \zeta_k \sum_{i=1}^j (\varphi_{k,i}) c_k f_{k,loc}^2 + \gamma s_{k,loc} f_{k,loc} \right) + \right. \\ & \left. \theta_k^* \left(\alpha \left(\sum_{i=1}^j \left(\frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + (1 - \eta_{k,j}) \frac{c_k}{f_{k,j}} \right) + \beta \sum_{i=1}^j \left(p_{k,i} \frac{\varphi_{k,i} u_k}{g_{k,i}} \right) + \gamma \left(\sum_{i=1}^j (q_{k,i} g_{k,i}) + (1 - \eta_{k,j}) s_{k,j} f_{k,j} \right) \right) \right) + \\ & \mu \left(\sum_{k=1}^n f_{k,j} - F_j^{max} \right) \end{aligned} \quad (16)$$

In (16), μ is the variable of non-negative Lagrange multiplier, the \mathbf{f}_k and μ in the above optimization problem must meet the following equations:

$$\frac{dL(\mathbf{f}_k, \mu)}{df_{k,j}} = -\frac{\alpha(1-\eta_{k,j})c_k}{(f_{k,j})^2} + \gamma(1-\eta_{k,j})s_{k,j} + \mu = 0 \quad (17)$$

$$\frac{dL(\mathbf{f}_k, \mu)}{d\mu} = \sum_{k=1}^n f_{k,j} - F_j^{max} = 0 \quad (18)$$

Therefore, the optimal solution of the allocated computing capability of the VEC server to the vehicle user is:

$$f_{k,i}^* = \frac{\sqrt{\alpha(1-\eta_{k,j})c_k(F_j^{max})^2}}{\sum_{k=1}^n \sqrt{\alpha(1-\eta_{k,j})c_k}} \quad (19)$$

$$\mu^* = \left(\frac{\sum_{k=1}^n \sqrt{\alpha(1-\eta_{k,j})c_k}}{F_j^{max}} \right)^2 - \gamma(1-\eta_{k,j})s_{k,i} \quad (20)$$

6 Performance Evaluation

6.1 Parameter Setting

We use Matlab to simulate and evaluate the above solution. In the simulation, the VEC network has a total of 15 VEC servers connected to the corresponding RSU, the coverage is circular space, and each RSU has 5-50 vehicle users within its wireless coverage. The channel bandwidth between servers is set to 10-30MHz, and the computing capability of VEC servers is set to 7000-10000. The amount of request data in the task is 500-3000Mb, and the required CPU cycles are 350-2700 cycles. The local computing capability of the vehicle is randomly set between 1-5, and the values of the optimization variables α , β , and γ are 0.3, 0.45 and 0.25, respectively. The experiment parameters are shown in Table II.

Table II Simulation experiment parameters

Parameter	Value
Number of VEC server	15
Number of vehicle	5-50
Channel bandwidth	10-30MHz
Task data size	500-3000Mb
CPU cycle	350-2700Cycles
variable α , β , γ	0.3, 0.45 and 0.25

6.2 Experiment Analysis

In the experiment, we analyze the impact of different task processing types on the total cost (objective function) of vehicles. The figure 2 shows that, as the number of vehicle users requesting task offloading increases, the total cost for all vehicle users increases. When the number of vehicle users is fixed, the total cost of task offloading will be lower than that of local processing, since the VEC servers can provide efficient processing resources with low transmission delay. However, since the total amount of resources of the VEC server is fixed, when a large number of VEC servers are assigned to nearby vehicles, there will be competition between vehicles, the price for VEC server resources will increase. We can observe from figure 2 that the total cost with the support of cache processing is the lowest, because when the VEC servers have the cached task data, the offloaded task does not have to be executed by the VEC server, and the result can be directly return to requester without task execution. With the support of caching, the execution delay can be ignored, so that the total cost is lower than that of other processing types.

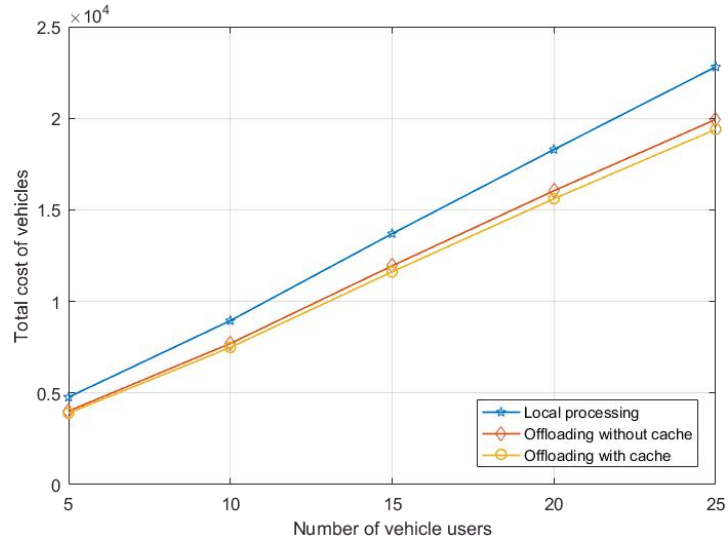


Fig.2. Impact of different task processing types on the cost of vehicle users

When task offloading, the driving vehicle users need to switch different number of VEC servers. Since the processing capability of each server is limit, with the increase of tasks, the allocated resource of each server for each task will reduce, thus the number of servers switched by vehicle users will increase accordingly, as is shown in figure 3.

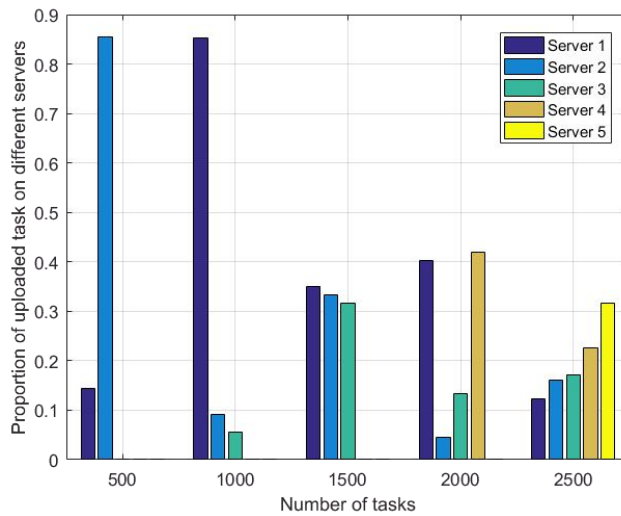


Fig.3. The number of servers switched by vehicle users in task offloading

7 Conclusion

In this paper, we focuses on the task switching and offloading scheme of vehicle users while moving, establishes an optimal model of task offloading with joint optimization of delay, energy consumption and the payment for vehicle users. In order to obtain the task offloading optimization strategy, we proposed a solution based on the KKT conditions, which shows that each server should undertake the optimal amount of tasks during task unloading considering the moving of vehicles. The simulation results show that the proposed scheme has good effect, the vehicles perform task offloading switching efficiently between the VEC servers, and the delay of task offloading and the total cost of vehicle users are decreased.

References

- [1] Zhang K , Mao Y , Leng S , et al. Mobile-Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading[J]. *IEEE Vehicular Technology Magazine*, 2017, 12(2): 36-44.
- [2] Sun J, Gu Q, Zheng T, et al. Joint Optimization of Computation Offloading and Task Scheduling in Vehicular Edge Computing Networks[J]. *IEEE Access*, 2020: 10466-10477.
- [3] Rahman A U, Malik A W, Sati V, et al. Context-aware opportunistic computing in vehicle-to-vehicle networks[J]. *Vehicular Communications*, 2020,24(1):1-9.
- [4] Zeng, F., Chen, Y., Yao, L., Wu, J. A Novel Reputation Incentive Mechanism and Game Theory Analysis for Service Caching in Software-Defined Vehicle Edge Computing. *Peer-to-Peer Networking and Applications*, 2021, 14:467–481.

- [5] Zhang X, Zhang J, Liu Z, et al. MDP-Based Task Offloading for Vehicular Edge Computing Under Certain and Uncertain Transition Probabilities[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(3): 3296-3309.
- [6] Liu P, Li J, Sun Z, et al. Matching-Based Task Offloading for Vehicular Edge Computing[J]. *IEEE Access*, 2019: 27628-27640.
- [7] Li L, Li Y, Hou R, et al. A Novel Mobile Edge Computing-Based Architecture for Future Cellular Vehicular Networks[C]. *wireless communications and networking conference*, 2017: 1-6.
- [8] Salahuddin M A, Alfuqaha A, Guizani M, et al. Reinforcement learning for resource provisioning in the vehicular cloud[J]. *IEEE Wireless Communications*, 2016, 23(4): 128-135.
- [9] Zhang L, Zhao Z, Wu Q, et al. Energy-Aware Dynamic Resource Allocation in UAV Assisted Mobile Edge Computing Over Social Internet of Vehicles[J]. *IEEE Access*, 2018: 56700-56715.
- [10] Zeng, F., Chen, Q., Meng, L., Wu, J. Volunteer Assisted Collaborative Offloading and Resource Allocation in Vehicular Edge Computing. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22(6): 3247-3257.
- [11] Wang S, Zhang Z, Yu R, et al. Low-latency caching with auction game in vehicular edge computing[C]. *international conference on communications*, 2017: 1-6.
- [12] Zeng F, Zhang R, Cheng X , et al. Channel Prediction Based Scheduling for Data Dissemination in VANETs[J]. *IEEE Communications Letters*, 2017,21(1):1409–1412.
- [13] Ye D, Yu R, Pan M, et al. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach[J]. *IEEE Access*, 2020: 23920-23935.