2020 16th International Conference on Mobility, Sensing and Networking (MSN) | 978-1-7281-9916-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/MSN50589.2020.00020

# A Game Theoretical Pricing Scheme for Vehicles in Vehicular Edge Computing

Chaogang Tang*†, Chunsheng Zhu‡§, Huaming Wu¶, Xianglin Wei‖, Qing Li**, Joel J. P. C. Rodrigues††‡‡
*School of Computer Science and Technology, China University of Mining and Technology, 221116, Xuzhou, China
†Mine Digitization Engineering Research Center of the Ministry of Education,
China University of Mining and Technology, Xuzhou 221116, China
‡SUSTech Institute of Future Networks, Southern University of Science and Technology, 518055, Shenzhen, China
§PCL Research Center of Networks and Communications, Peng Cheng Laboratory, 518055, Shenzhen, China
¶ Center for Applied Mathematics, Tianjin University, Tianjin 300072, China
‖The 63rd Research Institute, National University of Defense Technology, 410073, Changsha, China
** Department of Computing, The Hong Kong Polytechnic University, Hong Kong
†† Federal University of Piauí, Teresina - Pi, Brazil
‡‡ Instituto de Telecomunicações, Portugal
cgtang@cumt.edu.cn, chunsheng.tom.zhu@gmail.com, whming@tju.edu.cn, wei_xianglin@163.com,
csqli@comp.polyu.edu.hk, joeljr@ieee.org

***Abstract*—Vehicular edge computing (VEC) brings the computing resources to the edge of the networks and thus provisions better computing services to the vehicles in terms of response latency. Meanwhile, the edge server can earn their revenues by leasing the computing resources. However, a higher price does not always bring forth more benefits for the edge server in VEC, since it may discourage vehicles from renting more computing resources from VEC. To the best of our knowledge, few of previous works have focused on the real-time pricing problem for VEC. We investigate in this paper the pricing problem from the viewpoints of both vehicles and the edge server, so as to optimize the utility values and revenues of vehicles and the edge server, respectively. We resort to the Stackelberg game for modeling the interactions between vehicles and edge server, and a distributed algorithm for this pricing problem is proposed in the paper. Experimental results have displayed the efficiency and effectiveness of the proposed algorithm.**

***Index Terms*—Vehicular edge computing, Stackelberg game, distributed, edge server, pricing**

## I. Introduction

For intelligent transportation, significant progress has been made in event monitoring and data acquisition, owing to the wide deployment of in-car sensors. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) [1] communication technologies enable the real-time information dissemination and data delivery. Vehicular operating systems are also developing rapidly in recent years. For example, BYD (short for Build Your Dreams), one of the recent rise of Chinese carmakers, unveils lately the world's first 5G-equipped electric car running on HarmonyOS which is a self-developed and widely supported cross platform operating system from Chinese smartphone maker Huawei.

Against this background, the number of in-car applications is also explosively increasing, which has demonstrated ravenous maw for computing resources, e.g., in-car interactive gaming and nature language understanding and processing [2]. Due to the limited capabilities of vehicle loaded computers, some resource-hungry applications needs to be outsourced to a third party (e.g., remote cloud) for execution [3]. In spite of unlimited and powerful computing resources in cloud computing, one drawback of cloud computing for time sensitive vehicular application outsourcing is that the response latency could be extremely long due to the data transmission via the backhaul links in the core networks.

Vehicular edge computing (VEC) [4] was recently proposed to cope with vehicular applications featured by strict delay requirements by bringing the computing resources to the edge of the networks. Many works on VEC have been conducted [2][5][6][7][8][9], which focuses on application outsourcing and resource allocation from the viewpoint of response latency reduction or energy optimization, with an assumption that the computing resources in VEC can be used unconditionally. However, this assumption does not always hold in reality since vehicles need to pay for the computing resources rented from VEC. Meanwhile, resource providers in VEC try to maximize their revenues by leasing computing resources at a reasonable price.

In this paper, we pay attention to the real-time pricing scheme for VEC. On one hand, the pricing for computing resources in VEC can be set based on the demands of vehicles; on the other hand, the pricing also needs to consider the utilities of vehicles, since inappropriate pricing could dampen enthusiasm of vehicles. To the best of our knowledge, few of previous works have focused on the real-time pricing problem for VEC. The contributions of our paper are outlined as follows:

1) We investigate the pricing issue from the viewpoints of both vehicles and the edge server, with the purpose of optimizing the utility values and revenues of vehicles and the edge server, respectively.
2) To model the interactions between the vehicles and the edge server, we resort to an one-leader multi-followers

978-1-7281-9916-0/20/$31.00 ©2020 IEEE
DOI 10.1109/MSN50589.2020.00020

Stackelberg game model.

3) The existence and uniqueness of the proposed game is proven and the experimental results have shown the effectiveness and efficiency of our distributed pricing scheme.

The remainder of this paper is organized as follows. The related works are reviewed in Section II, followed by the description of the system model in Section III. The problem is mathematically formulated and we resort to a Stackelberg game model for analyzing the interactions between them in Section IV. A distributed algorithm for the pricing problem in VEC is presented in Section V. Experimental results are shown in Section VI and the conclusion is given in Section VII.

## II. Related Works

Extensive attention has been increasingly paid to the newly emerging computing paradigms such as vehicular fog computing (VFC)[10][11][12][13] and VEC. VFC and VEC are similar in the sense that both of them endeavor to reduce the response latency by pushing the computing resources to the edge of networks. In the context of smart vehicles, the edge of networks usually refers to the road side units (RSUs) or the vehicles themselves. VEC is considered to be one of the key enablers to promote the development of smart cities and even the 5G era [14].

In our previous works such as [2][3], either RSU or vehicles themselves are treated as the edge of networks, which have been empowered with powerful computing capabilities with an aim to optimize the response latency or energy consumption in the dynamic vehicle ad hoc network (VANET).

In one-server VEC systems, large number of tasks are offloaded to the server which may degrade the performance of VEC systems such as the increasing response latency, to a great extent. To address this issue, authors in [8] propose to investigate the task offloading from the viewpoint of load balancing and research resource allocation in the context of a multi-server VEC systems. Similar works can also be found in [6][14].

Despite the benefits anticipated from VEC, one of the main drawbacks is that vehicles are highly dynamic which renders that the service time is limited in the situation where vehicles are either resource consumers or resource providers. To cope with this issue, authors in [9] resort to one particular vehicles – buses in the city. Owing to its characteristics such as moving slowly and fixed routes, buses can be enhanced with computing resources to serve as the edge servers in the edge of networks, with an aim to provision computing services to the vehicles in vicinity.

On another hand, many works focus on task offloading and resource allocation from the game theoretical viewpoint such as [15][16]. For example, the Heterogeneous Vehicular NETwork (HetVNET) is developing fast recently, driven by their potential benefits as well as the drawbacks faced by the current wireless networks. Authors in [15] try to define a congestion game for information congestion control issue in the paper.

Vehicular mobile cloud computing targets at provisioning computing and communication resources to the mobile terminals such as mobile clients. One of the main drawbacks as analyzed in [16] is that large number of user requests from vehicles or mobile devices can overload the virtual machines (VM) of the vehicular mobile cloud computing systems. Thus, inappropriate usage of virtual machines can lead to several issues such as energy wasting.

However, few researches in the present works have paid attention to the pricing issue in VEC, although the pricing scheme has an important influence over the server provisioning in VEC. In this paper, we strive to investigate the pricing scheme in VEC from game theoretical viewpoint, and a Stackelberg game is used to model the interactions between the involved parties (i.e., vehicles and the edge server).

## III. System Model

A vehicular edge computing system considered in this paper consists of one RSU $R$ and a set of $V = \{v_1, v_2, ..., v_n\}$ vehicles. We assume that $R$ is enhanced with edge server with powerful computing capacities and the computing resources are sufficient in VEC to satisfy the demands of vehicles in vicinity. The V2I communication technologies are needed for the wireless communications between vehicles and the edge server $R$. Thus, the communication range is limited. Both service provisioning and service consuming should take place within the communication range. Let $p$ denote the unit price of the computing resources set by VEC and $s_i$ denote the computing resource demands of vehicle $v_i$. We denote by $s_{min,i}$ and $s_{max,i}$ the lower bound and the upper bound of resource for $v_i$, respectively. Thus, for arbitrary vehicle $v_i(1 \leq i \leq n)$ , we have $s_{min,i} \leq s_i \leq s_{max,i}$.

The set of applications is denoted by $A = \{a_1, a_2, ..., a_n\}$ where $a_i(1 \leq i \leq n)$ represents the application that $v_i$ wants to outsource to VEC for execution. Note that $a_i$ can be denoted by a tuple $(s_{max,i}, l_i)$ where $s_{max,i}$ denotes the amount of computing resources when $a_i$ is totally outsourced and $l_i$ denotes the maximal response latency that $v_i$ can tolerate.

Due to the high mobility of vehicles, they can move out of the wireless communication of $R$ in a short time. We need to estimate the dwell time of $v_i$ within the communication range of $R$ as follows:

$$dt_i = \frac{2D\sin(\theta/2)}{r_i} \tag{1}$$

where $D$ is the communication distance of $R$, $r_i$ is the velocity of $v_i$, and $\theta$ is the angle formed between $R$ and the points where $v_i$ enters and leaves the communication range of $R$, respectively. Given the price $p$ set by VEC, the utility of $v_i$ is defined as:

$$\mathcal{U}_i = \alpha_i \log(1 + s_i) - p \cdot s_i \tag{2}$$

where $\alpha_i$ as a constant denotes the satisfaction level on the rented computing resources. Note that different $v_i$ has different

$\alpha_i$. Instinctively, the more the computing resources are rent, the higher the satisfaction level is. The first term $\alpha_i \log(1+s_i)$ represents the utility value earned by renting $s_i$, while the second term $p \cdot s_i$ denotes the payments for renting resources from VEC.

The edge server strives to pursue its own profits by leasing its computing resources. Given the unit price for the resources $p$, its utility function can be defined as:

$$\mathcal{U}_e = p\sum_{i=1}^{n} s_i - c\sum_{i=1}^{n} s_i \tag{3}$$

where the first term $p\sum_{i=1}^{n} s_i$ represents the profits obtained by leasing computing resources to vehicles in vicinity and the second term $c\sum_{i=1}^{n} s_i$ denotes the costs by contributing resources; $c$ is the cost for leasing unit computing resources.

## IV. Problem Formulation

The appropriate pricing for the computing resources in VEC can reach a win-win situation where both the edge server and vehicles can achieve their goals, e.g., utility optimization and revenues maximization, respectively. To that end, the interactions between the edge server and vehicles can be sketched out as follows. At the beginning, the edge server $R$ determines a suitable price $p$ and broadcasts it to the vehicles in vicinity together with the beacon information. After receiving the beacons, each vehicle adjusts its own computing resource demands to rent from VEC based on the given price $p$. Then the amount of the computing resources after adjustment is disseminated to $R$ by each vehicle. Upon receiving the demands for computing resources from each vehicle, $R$ recalculates the price $p$ based on its utility function and then broadcasts the updated $p$ to vehicles. This procedure will not stop until $p$ and the utility values of vehicles do not change anymore.

To model the interactions detailed above, a one-leader multi-followers Stackelberg game is introduced, where $R$ is the leader and vehicles are the followers. The Stackelberg game model is widely applied to the multilevel decision-making processes where multiple independent followers (e.g., vehicles) respond to the decision maker, i.e., the leader of the game, by selecting their own strategies after observing the leader's strategy. In this paper, the pricing scheme in VEC is modeled as a noncooperative game where the edge server acts as the leader and vehicles act as the followers. To be specific, the game of the pricing scheme in this paper can be defined as : $G = (V \cup \{R\}, \boldsymbol{p}, \mathcal{U}_i\{p_i\}_{i\in \boldsymbol{N}})$, where $V \cup \{R\}$ is the set of players, consisting of the set of vehicles (i.e., followers) plus $R$ (i.e., the leader), $\boldsymbol{p}$ denotes the strategies of the edge server, i.e., the set of prices set by $R$, and $\mathcal{U}_i\{p_i\}_{i\in \boldsymbol{N}}$ denotes the utility values of vehicles $\mathcal{U}_i\{p_i\}_{i\in \boldsymbol{N}}$.

Before proving the uniqueness of the Nash Equilibrium (NE) of this game, we mathematically formulate this problem from the viewpoints of vehicles and the edge server $R$, respectively.

### A. Vehicle Utility Optimization

The purpose of each vehicle is to maximize its own utility value by adjusting its computing resource renting from VEC. Therefore, regarding each vehicle, the problem can be mathematically defined as:

$$\mathcal{P}: \quad \max_{s} \quad \mathcal{U}_i \tag{4}$$

s.t.:

$$s_{min,i} \le s_i \le s_{max,i} \qquad \forall i \in [1,n] \tag{5}$$

$$l_i \le dt_i \qquad \forall i \in [1,n] \tag{6}$$

where inequation (5) guarantees that the amount of computing resources rented from VEC is bounded, and inequation (6) ensures that the deadline for application outsourcing must be shorter than dwell time of vehicle within the communication range of the edge server.

Let's focus on the utility function of vehicles $\mathcal{U}_i$ and assume that $s_i$ is a continuous variable. The first derivative of $\mathcal{U}_i$ with regards to (w.r.t.) $s_i$ is given by

$$\frac{d\mathcal{U}_i}{ds_i} = \frac{\alpha_i}{(1+s_i)\ln 2} - p \tag{7}$$

The second derivative of $\mathcal{U}_i$ w.r.t. $s_i$ is given by

$$\frac{d^2\mathcal{U}_i}{ds_i^2} = \frac{-\alpha_i}{(1+s_i)^2\ln 2} < 0 \tag{8}$$

The second derivative of $\mathcal{U}_i$ is always negative in the feasible domain, so the utility function, i.e., $\mathcal{U}_i$ is convex. Therefore, problem $\mathcal{P}$ is a convex optimization problem. Thus the maximal value of $\mathcal{U}$ exists. By letting $d\mathcal{U}_i/ds_i = 0$, we can get it as

$$s_i = \frac{\alpha_i}{p\ln 2} - 1 \tag{9}$$

Note that the satisfaction level $\alpha_i$ should be large enough to guarantee that $s_i$ is positive. Thus, given the price $p$ set by VEC, a maximal value of $\mathcal{U}$ can be obtained. However, from equation (9), we can observe that as the price $p$ increases, the amount of computing resources rented by vehicles decreases. Thus, the pricing scheme should be designed reasonably to retain the enthusiasm of vehicles in vicinity.

### B. Edge Server Profits Maximizataion

The fact that VEC brings the computing resources much closer to the resource requesters, on one hand drastically reduces the response latency and, on the other hand, benefits itself by leasing these computing resources. Thus, in this paper the purpose of the edge server is to maximize the revenues defined in Eq. (3). Thus, the optimization problem for the edge server can be mathematically formulated as:

$$\mathcal{Q}: \quad \max_{p} \quad \mathcal{U}_e \tag{10}$$

Consider the utility function $\mathcal{U}_e$ and assume $p$ is a continuous variable. Let the first derivative of $\mathcal{U}_e$ w.r.t., $p$ equal zero, i.e., $d\mathcal{U}_e/dp = 0$. Substituting $s_i$ using equation (9), $d\mathcal{U}_e/dp = 0$ then can be rewritten as:

$$\frac{d\mathcal{U}_e}{dp} = \frac{d(\frac{1}{\ln 2}\sum_{i=1}^{n}\alpha_i - pn - \frac{c}{p\ln 2}\sum_{i=1}^{n}\alpha_i - nc)}{dp} = 0 \tag{11}$$

Then, we can derive:

$$p = \sqrt{\frac{c}{n\ln 2}\sum_{i=1}^{n}\alpha_i} \tag{12}$$

From the equation above we can observe that the optimal price set by $R$ totally depends on the number of vehicles requesting the resources, the unit cost of edge server, and the satisfaction level of each vehicle.

### C. Stackelberg Equilibrium

The purposes of vehicles and the edge servers are to solve the problems $\mathcal{P}$ and $\mathcal{Q}$, respectively. We now investigate the solutions from the game theoretical perspective. To be specific, the reasonable solution in $G$ for vehicles and the edge server is the existence and uniqueness of the Stackelberg Equilibrium (SE), which guarantees that the edge server gets the optimal rental price and vehicles can get the optimal amount of computing resources. Let $\boldsymbol{s} = \{s_1, s_2, ..., s_n\}$ and $\boldsymbol{p}$ denote the strategy of vehicles and the strategies of the edge server, respectively. Then the SE of the game $G$ can be defined as follows:

*Definition 1:* If the strategy profile $\boldsymbol{s}^* = \{s_1^*, s_2^*, ..., s_n^*\}$ and $p^*$ are the SE of $G$, then no vehicles can increase their utility values anymore by adjusting their strategies unilaterally and for the edge server, it cannot get more profits by adjusting the price $p$ unequal to $p^*$, i.e.,

$$\mathcal{U}_i(s_i^*, s_{-i}^*, p^*) \geq \mathcal{U}_i(s_i', s_{-i}^*, p^*) \qquad \forall i \in [1, n] \tag{13}$$

$$\mathcal{U}_e(s_i^*, s_{-i}^*, p^*) \geq \mathcal{U}_e(s_i^*, s_{-i}^*, p') \qquad \forall i \in [1, n] \tag{14}$$

In this definition, $s_{-i}^* = \{s_1^*, ..., s_{i-1}^*, s_{i+1}^* ..., s_n^*\}$ denotes the strategies of other vehicles except $v_i$.

However, in the non-cooperative Stackelberg game between vehicles and the edge server, an equilibrium does no always exist. Therefore, we need to prove the existence and uniqueness of this game proposed in this paper.

*Theorem 1:* A unique SE of game $G = (V \cup \{R\}, \boldsymbol{p}, \mathcal{U}_i\{p_i\}_{i\in\boldsymbol{N}} \cup \mathcal{U}_e\{s_i\}_{i\in\boldsymbol{N}})$ always exists between vehicles and the edge server.

*Proof* It is obvious that the followers in the game (i.e., the vehicles) are independent of each other, and the utility function of vehicles only depends on their own amount of computing resources rented from $R$ and the price $p$ set by $R$. As proved earlier, the utility function $\mathcal{U}_i$ is strictly convex for $\forall i \in [1, n]$, due to the fact that $d^2\mathcal{U}_i/ds_i^2 < 0$ always holds, for $\forall i \in [1, n]$. Accordingly, given a price $p(> 0)$ set by $R$, each vehicle $v_i$ can get their optimal computing resource consumption that can maximize the utility value. Considering the interval $[s_{min,i}, s_{max,i}]$, there are only three candidates for maximizing the utility function, i.e., $s_i \in \{s_{min,i}, s_{max,i}, \alpha_i/p\ln 2 - 1\}$. In Definition 1, we have defined that $G$ can achieve SE if and only if each player including each vehicle $v_i$ and $R$, reaches the maximal utility and profits, respectively. Accordingly, the game $G$ can reach the SE as long as the edge server finds the optimal price $p$ meanwhile $v_i$ obtains an optimal amount of computing resources.

Given $s_i$ of each vehicle by substituting $s_i$ in equation (3) with equation (10), we can derive the second derivative of $\mathcal{U}_e$ w.r.t. $p$ as:

$$\frac{d^2\mathcal{U}_e}{dp^2} = -\frac{2}{p^3}\frac{c}{\ln 2}\sum_{i=1}^{n}\alpha_i < 0$$

Therefore, $\mathcal{U}_e$ is strictly convex in the feasible domain with regards to $p$. Given the information on computing resource purchase of each vehicle $s_i$, we can get the optimal pricing scheme $p^*$ by letting $d\mathcal{U}_e/dp = 0$. Accordingly, there always exists a unique SE in $G$. □

## V. Distributed Algorithm

Considering the privacy of each vehicle, e.g., they may not be willing to expose the satisfaction levels to the public, a distributed algorithm is therefore needed for obtaining the optimal price. In the distributed algorithm, the edge server needs frequent interactions with vehicles so as to reach the SE of the game. At the beginning, a price randomly set by the server is sent to each vehicle requesting the computing resources. After receiving the price, each vehicle calculates its best amount of computing resources $s_i$ and reports it to the server. At the server side, a resource renting vector is constructed, i.e., $\boldsymbol{s} = [s_1, s_2, ..., s_n]$. Based on $\boldsymbol{s}$, a new price can be derived with the purpose of maximizing its own profits. This procedure continues until the vehicles and edge server iteratively reach the SE of the game $G$ in the distributed way. In other words, equations (13) and (14) are both satisfied as defined in Definition 1. The corresponding algorithms for vehicles and the edge server are shown in the following, respectively.

The process of seeking the optimal utility values for the vehicles (i.e., followers) is denoted in Algorithm 1. Upon receiving the price $p$ from $R$, the algorithm first checks whether the dwell time of each vehicle exceeds the maximal response latency. Then each vehicle calculates their own amount of renting resources given $p$. As mentioned earlier in Theorem 1, there are only three candidates for the best amount of renting resources, and the procedure checks them one by one. Finally each vehicle reports its best amount of computing resources to the edge server.

In Algorithm (2), the edge server updates its best utility value (i.e., revenues) every time when all the amounts of computing resources (i.e., $\boldsymbol{s}$) are received. According to the monotonic feature of $\mathcal{U}_e$ w.r.t. $p$ given $\boldsymbol{s}$, it is obvious that the more the pricing, the more the benefits obtained from the perspective of the edge server. Thus, the procedure updates $p$ (e.g., increasing with a step size of $\Delta p$) each time and sends it to the vehicles. The procedure repeats until the best utility value does not change anymore.

```
Algorithm 1: Distributed Algorithm for Vehicles in VEC
Input: $p$, $A$
Output: The optimal $s_i^*$ for each $v_i$
for each $v_i$ in $V$ do
    $\mathcal{U}_i = 0$;
end
for each iterative $p$ received from $R$ do
    for each $v_i$ in $V$ do
        if $l_i \leq dt_i$ then
            $s_i = \frac{\alpha_i}{p \ln 2} - 1$;
            if $s_i < s_{min,i}$ then
                $s_i = s_{min,i}$;
            end
            if $s_i > s_{max,i}$ then
                $s_i = s_{max,i}$;
            end
        end
    end
end
return $s_i^*$ of each $v_i$;
```

```
Algorithm 2: Distributed Algorithm for Edge Server in VEC
Input: $\boldsymbol{s}$, $c$, $n$
Output: The optimal price $p^*$
$\mathcal{U}_e^* = 0$;
$p^* = 0$;
$sum = 0$;
for each element $s_i$ in $\boldsymbol{s}$ do
    $sum+ = s_i$;
end
$\mathcal{U}_e = p \cdot sum - c \cdot sum$;
while $\mathcal{U}_e - \mathcal{U}_e^* > \epsilon$ do
    $\mathcal{U}_e^* = \mathcal{U}_e$;
    $p^* = p$;
    Update $p$;
    Send $p$ to each $v_i$;
end
return $p^*$;
```

## VI. Numeric Evaluation and Analysis

In this section, we strive to evaluate the proposed real-time distributed pricing scheme in VEC in terms of efficiency and effectiveness. To be specific, the main parameters involved in this experiment are shown in Table I.

As introduced earlier, the SE of the game can be reached between the vehicles and servers iteratively. The followers, i.e., vehicles, update their own strategies (i.e., the amounts of computing resources rented from VEC), with an aim to find the optimal value according to equation (2), after they observed the strategy of the edge server. The first set of experiments is to investigate the utility values of vehicles following the leader' strategies. The result is shown in Figure 1 where the x-coordinate represents the number of iterations and the y-coordinate represents the utility values of these vehicles. In this set, the number of vehicles is set to 20. We only choose five vehicles from them in Figure 1. The initial unit price for computing resources is 3. From the figure, we can observe that the utility values of all these vehicles decrease with the increasing number of iterations. It is a process of game between vehicles and edge server. The initial unit price $p$ is relatively small which is good for vehicles, for the reason that according to equation (9), vehicles can get relatively large utility values. However, from the viewpoint of the edge server, it is not appropriate to lease computing resources at the current price $p$, due to that the revenues obtained are not satisfactory. Thus, the leader updates the price until the SE of the game arrives.

TABLE I
Parameter Settings

| Notations | Meanings | Values |
|---|---|---|
| $n$ | Number of vehicles | [10, 20] |
| $D$ | Communication range | 100 |
| $\theta$ | Angle | 60 |
| $c$ | The unit cost | 1 |
| $s_{min,i}$ | The lower bound of the resources | [0, 3] |
| $s_{max,i}$ | The upper bound of the resources | [120, 140] |
| $\alpha_i$ | The satisfaction level | [100, 200] |

After reaching the SE of the game, we can observe that the utility values of these vehicles do not change anymore. No vehicle can achieve a higher utility value by adjusting its own strategy unilaterally in the SE of the game.

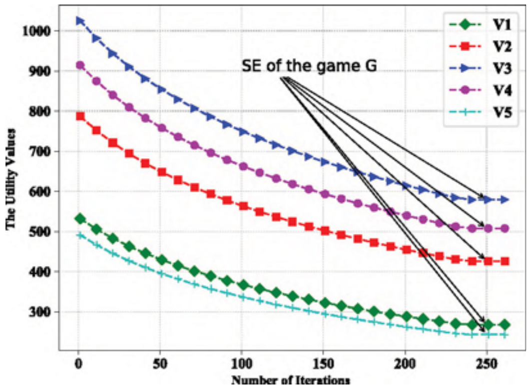

Fig. 1. The utility values of four vehicles before reaching SE of the game

For the edge server side, the revenues of the edge server with the increasing number of iterations are shown in Figure 2. Two approaches are compared, i.e., one is the distributed approach proposed; the other is the centralized approach. The centralized approach assumes that the edge server knows the satisfaction level of all vehicles. After reaching the SE of the game, the revenues of the edge server do not change anymore either.

On the other hand, the centralized approach can get the optimal price by equation (12). In contrast, the distributed approach does not know $\alpha$ for the sake of privacy protection. As a result, it takes relatively long time to converge to the SE of the game. In addition, the rate of convergence mainly depends upon the way $p$ is updated. Small values of $\Delta p$ slow down the convergence rate, while too large values of $\Delta p$ may cause the price $p$ to miss the convergence point.

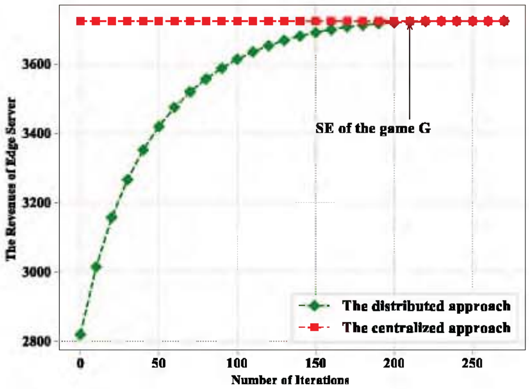

Fig. 2. The revenues of the edge server with different approaches

## VII. Conclusion

In this paper, the pricing scheme in VEC is investigated from the viewpoint of Stackelberg game where the edge server is modeled as the leader and the vehicles requesting computing resources are modeled as followers. We aim at optimizing the utility values of vehicles and maximizing the revenues of the edge server at the same time. Thus, in this paper the Stackelberg equilibrium has been proven in terms of existence and uniqueness. A distributed algorithm has been proposed to address this issue.

## Acknowledgment

This research was partially supported by the project “PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications (LZC0019)”, by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/50008/2020 and by Brazilian National Council for Scientific and Technological Development (CNPq) via Grant No. 309335/2017-5. Chunsheng Zhu is the corresponding author.

## References

[1] A. Ghosh, V. V. Paranthaman, G. Mapp, O. Gemikonakli, and J. Loo, “Enabling seamless v2i communications: Toward developing cooperative automotive applications in vanet systems,” Communications Magazine IEEE, vol. 53, no. 12, pp. 80–86, 2015.

[2] C. Tang, C. Zhu, X. Wei, H. Wu, Q. Li and J. J. P. C. Rodrigues, “Intelligent Resource Allocation for Utility Optimization in RSU-Empowered Vehicular Network,” in IEEE Access, vol. 8, pp. 94453–94462, 2020.

[3] C. Tang, X. Wei, C. Zhu, Y. Wang, W. Jia, “Mobile Vehicles As Fog Nodes For LatencyOptimization In Smart Cities,” IEEE Transactions on Vehicular Technology, online, 2020, doi: 10.1109/TVT.2020.2970763.

[4] C. Xu, J. Lei, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” IEEE/ACM Transactions on Networking, vol. 24, no. 5, pp. 2795–2808, 2016.

[5] C. Chen et al., “Delay-Optimized V2V-Based Computation Offloading in Urban Vehicular Edge Computing and Networks,” IEEE Access, vol. 8, pp. 18863-18873, 2020.

[6] J. Zhang, H. Guo, J. Liu and Y. Zhang, “Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution,” in IEEE Transactions on Vehicular Technology, vol. 69, no. 2, pp. 2092-2104, Feb. 2020.

[7] J. Zhang, H. Zhong, J. Cui, M. Tian, Y. Xu and L. Liu, “Edge Computing-Based Privacy-Preserving Authentication Framework and Protocol for 5G-Enabled Vehicular Networks,” in IEEE Transactions on Vehicular Technology, vol. 69, no. 7, pp. 7940-7954, July 2020.

[8] Y. Dai, D. Xu, S. Maharjan and Y. Zhang, “Joint Load Balancing and Offloading in Vehicular Edge Computing and Networks,” in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4377-4387, June 2019.

[9] Z. Liu, X. Zhang, J. Zhang, D. Tang and X. Tao, “Learning Based Fluctuation-aware Computation offloading for Vehicular Edge Computing System,” 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea (South), 2020, pp. 1-7.

[10] X. Hou, L. Yong, C. Min, W. Di, D. Jin, and C. Sheng, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures,” IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 3860–3873, 2016.

[11] C. Tang, X. Wei, C. Zhu, W. Chen and J. J. P. C. Rodrigues, “Towards Smart Parking Based on Fog Computing,” in IEEE Access, vol. 6, pp. 70172–70185, 2018.

[12] H. Cheng, R. Lu, and K. K. R. Choo, “Vehicular fog computing: Architecture, use case, and security and forensic challenges,” IEEE Communications Magazine, vol. 55, no. 11, pp. 105–111, 2017.

[13] Z. Ning, X. Wang, and J. Huang, “Vehicular fog computing: Enabling real-time traffic management for smart cities,” IEEE Wireless Communications, vol. 26, no. 1, pp. 87–93, 2019.

[14] Y. Deng, Z. Chen, X. Yao, S. Hassan, and J. Wu, “Task scheduling for smart city applications based on multi-server mobile edge computing,” IEEE Access, vol. 7, pp. 14410–14421, 2019.

[15] C. Chen, T. Qiu, J. Hu, Z. Ren, Y. Zhou, and A. K. Sangaiah, “A congestion avoidance game for information exchange on intersections in heterogeneous vehicular networks,” J. Netw. Comput. Appl., vol. 85, pp. 116-126, 2017.

[16] N. Kumar, S. Zeadally, N. Chilamkurti and A. Vinel, “Performance analysis of Bayesian coalition game-based energy-aware virtual machine migration in vehicular mobile cloud,” in IEEE Network, vol. 29, no. 2, pp. 62-69, March-April 2015.