Optimal Task Assignment With Delay Constraint for Parked Vehicle Assisted Edge Computing: A Stackelberg Game Approach

Junxing Zhang, Xumin Huang, *Member, IEEE*, and Rong Yu, *Member, IEEE*

***Abstract*— Parked vehicle assisted edge computing is a paradigm to employ parked vehicles for processing workloads in vehicular networks. They collaborate with edge servers for joint task execution within task deadline and parking durations. We study task assignment problem with Stackelberg game, where the overall cost of a task publisher in terms of monetary cost and subjective dissatisfaction caused by un-offloaded workloads is minimized. To reach Stackelberg equilibrium, a dedicated algorithm is performed among players in a distributed manner for privacy preservation. Numerical results indicate that the cost is decreased about 25% in our scheme compared to that in existing work.**

***Index Terms*— Vehicular edge computing, game theory, privacy protection.**

I. Introduction

VEHICULAR Edge Computing (VEC) is a great application of mobile edge computing in transportation domains to serve vehicles with lower latency, fast response, and mobility support. Nowadays, various intelligent transportation system applications have been developed with computational-intensive and latency-sensitive requirements. A tremendous number of computation resources are still required in VEC while the massive deployment of edge servers will give rise to significant capital expenditure and time expenses, particularly in the initial phase of VEC. To cope with the dilemma, researchers have suggested Parked Vehicles (PVs) as dynamic network infrastructures to exploit idle and opportunistic resources for normal offloading service provision [1]. PVs become lightweight edge computing nodes to make use of under-utilized and accessible resources for task execution, leading to a novel paradigm called by Parked Vehicle assisted Edge Computing (PVEC) [2].

Current literature has investigated network optimization problems for facilitating PVEC. Work in [2] studied how to leverage PVs to coordinate with VEC servers for cooperative task execution. But necessary constraints, e.g., the task deadline, were not considered. Similarly, Han *et. al* [3] presented a dynamic pricing strategy to tackle with the cost-performance tradeoff in the PVEC system. Nevertheless, the scheme may depend on strong assumptions. Each PV was assumed to have the identical computation capability with an VEC server. The work in [4] first introduced container-based virtualization for improving task execution among PVs and studied task scheduling from the viewpoint of social welfare. Compared with the work above, Li *et. al* [5] focused on a Service Provider (SP)-centric scheme to assign tasks among PVs. A contract-based incentive mechanism was designed to recruit PVs for handling tasks to maximize the utility of an SP as an employer.

In this letter, we study the optimal task assignment under delay constraint in PVEC based on full considerations. First, we aim to minimize the total cost of a task publisher, which is in terms of a weighted sum of monetary cost and subjective dissatisfaction caused by the remaining un-offloaded workloads for individual undertaking. The goal is to explore unused resources from PVs to adapt to diverse offloading environment. Second, the task deadline constraint and special feature of PVs are exactly considered. To leverage the opportunistic resources, a predicted probability is proposed to evaluate the serviceability of PVs. PVs are independent to reply whether to participate in PVEC and determine their strategies in workload undertaking for utility maximization. Third, a distributed algorithm is elaborately designed to achieve the Stackelberg equilibrium with privacy protection, which can be implemented by each PV separately without revealing any private information to the SP. Existing work seldom considers network optimizations and security considerations simultaneously.

More specifically, we utilize the Stackelberg game to formulate and solve the essential task assignment problem in PVEC. A set of PVs is recruited by an SP to accomplish workloads. In the game, the SP acts as a leader to offer incentives while the PVs become followers to respond with the participation levels in computation offloading with respect to the given rewards. After that, Stackelberg equilibrium is theoretically analysed via the backward induction method. Particularly, a privacy-preserving algorithm is presented to reach the equilibrium in a distributed manner. The distributed algorithm permits PVs to mutually interact with the SP, without disclosing individual private information to the SP, thus ensuring privacy protection.

II. System Model and Game Formulation

A. Parked Vehicle Assisted Edge Computing Network

As shown in Fig. 1, we introduce the system model of PVEC. The system is consisted of PVs as task performers,

Manuscript received November 24, 2019; accepted December 11, 2019. Date of publication December 23, 2019; date of current version March 10, 2020. The work is supported in part by program of NSFC under Grant no. 61971148, the Science and Technology Program of Guangdong Province under Grant no. 2015B010129001, Natural Science Foundation of Guangxi Province under Grant no. 2018GXNSFDA281013, and Foundation for Science and Technology Project of Guilin City under Grant no. 20190214-3. The associate editor coordinating the review of this letter and approving it for publication was M. H. Cheung. *(Corresponding author: Rong Yu.)*

Junxing Zhang and Xumin Huang are with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of IoT Information Technology, Guangdong University of Technology, Guangzhou 510006, China (e-mail: junxingzhang@mail2.gdut.edu.cn; huangxu_min@163.com).

Rong Yu is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong–HongKong–Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China (e-mail: yurong@ieee.org).

Digital Object Identifier 10.1109/LCOMM.2019.2961663

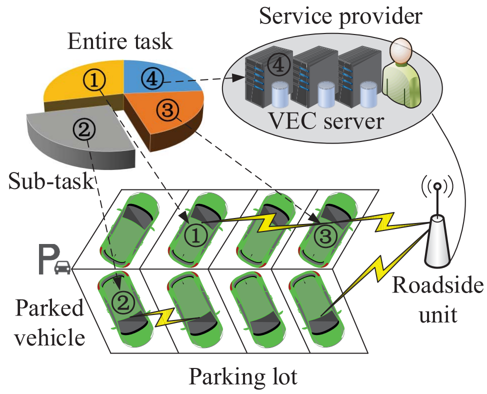

Fig. 1. System model of PVEC.

an SP with renting VEC servers for publishing tasks, and a roadside unit to be deployed for message forwarding. *i) PV*: Owing to the idle state and opportunistic computation resources, PVs become significant performers to execute tasks within the parking durations. *ii) SP*: In the system, the SP directly hosts the offloading services at the vehicular network edge. A task with large workloads is split into serval sub-tasks and assigned to multiple PVs for cooperative execution. *iii) VEC server*: The decision-making capability of the SP is enhanced by using VEC servers. Besides, some of them may also be scheduled for task execution when participating PVs cannot handle the entire task. *iv) Roadside unit*: As a local router, it enables mutual communications between the SP and PVs but also collects necessary status information of the PVs to proximal VEC servers via wired connections.

B. Parked Vehicle Selection

For stable and reliable offloading provision, the SP will select the adequate PVs as competent performers, according to their serviceability. Based on [2], the SP evaluates the serviceability of the PVs through the predicted probability that the PVs will continue to remain parked in a task assignment period. The task assignment period is denoted as T. PV's probability density function about parking durations is defined as $y(t)$, where t is the parking duration and constrained by $t \in [0, t^{\max}]$. The accumulated parking duration of the PV i up to now is denoted as bp_i. Hence the predicted probability of PV i mentioned above is calculated by

$$P_i = \frac{1}{1 - Y(bp_i)} \int_{bp_i + T}^{t^{\max}} y(t) dt = \frac{1 - Y(bp_i + T)}{1 - Y(bp_i)} \tag{1}$$

where $Y(t)$ is the cumulative distribution function of $y(t)$. In short, the SP will employ the qualified PVs that meet the condition $P_i \geq P_t$, in which P_t is a presetting threshold, and arrange them into a set $\mathcal{I}$. The size of the set $\mathcal{I}$ is influenced by the distribution of P_t.

C. Revenue Function of Parked Vehicle

A PV i belonging to $\mathcal{I}$ is incentivized for processing workloads. Each PV has a predefined maximal workload that can be processed, $w_i^{\max}$. PV i decides the amount of undertaking workloads assigned by the SP, $w_{i,s} \in [0, w_i^{\max}]$. $w_i^{\max} - w_{i,s}$ is the residual workload capacity left for satisfying any possible individual demand. Due to the reserved workloads for self use, there exists a satisfactory degree $S(\cdot)$ for the PV, which is interpreted as the state gain. The state gain of PV i is achieved from retaining residual workload capacity for potential incoming internal workloads. $S(\cdot)$ is monotonically decreasing with $w_{i,s}$ as conducting extra workloads for the SP may hinder personal services [6]. We define $S(\cdot)$ as a logistic function of $w_{i,s}$ as

$$S(w_{i,s}) = \log(1 + w_i^{\max} - w_{i,s}). \tag{2}$$

PV i also acquires rewards from the SP for undertaking the workloads. For fairness, the rewards are linear with $w_{i,s}$ and computed by $pw_{i,s}$, where p is a pricing parameter of the SP to encourage PVs for processing per workload. To summarize, the revenue function of PV i is

$$u_i = \alpha_i S(w_{i,s}) + pw_{i,s} \tag{3}$$

where α_i is a tradeoff parameter between internal satisfactory degree and external monetary rewards.

D. Cost Function of Service Provider

A task submitted to the SP is described as (D, H, W, τ), where D is the input data size of the task and H is an application-centric parameter related to the feature of the task. The total amount of workloads required to carry out the task is $W = HD$ [7]. τ indicates the task deadline.

When PVs may not undertake the entire task, the SP has to schedule a dedicated VEC server to perform the residual workloads $W - \sum_{i \in \mathcal{I}} w_{i,s}$. Note that the residual workloads exactly violate the willingness of the SP that provides the offloading services by recruiting PVs. This leads to a subjective dissatisfaction formulated as $\beta_s \left(W - \sum_{i \in \mathcal{I}} w_{i,s}\right)$, where β_s is the dissatisfaction level predetermined by the SP. As a result, the overall cost function of the SP is consisted of individual dissatisfaction and monetary cost, and expressed by

$$C = p \sum_{i \in \mathcal{I}} w_{i,s} + \beta_s \left(W - \sum_{i \in \mathcal{I}} w_{i,s} \right). \tag{4}$$

For task assignment among the PVs, there exist some constraints needed to be considered. First, the delay constraint of the task should be satisfied. In the local computing, we calculate the execution time at the SP side by

$$T_s = \frac{W - \sum\limits_{i \in \mathcal{I}} w_{i,s}}{f_s} \tag{5}$$

where f_s is the computation capability of the VEC server.

In the remote computing, the sub-task is transferred to PV i for execution. The data transmission time of the sub-task from the SP to PV i is acquired by

$$T_i^t = \frac{d_{i,s}}{r_i} = \frac{w_{i,s}}{Hr_i} \tag{6}$$

where $d_{i,s}$ is the data size of the sub-task assigned to PV i, and r_i is the downlink rate from the SP to the PV i. For ease of analysis, we consider r_i is fixed and uniform. The execution time performed by PV i is

$$T_i^e = \frac{w_{i,s}}{f_i} \tag{7}$$

where f_i is the computation capability of PV i. Similar to [8], we neglect the delay to send back the output results to the SP. Hence, the overall task execution time performed by PV i in the offloading service is $T_i^o = T_i^t + T_i^e$. By considering the parking duration and task delay constraint, we have

$$\begin{cases} \forall i \in \mathcal{I}, \ \ T_i^o \le T \\ \forall i \in \mathcal{I}, \ \ T_i^o \le \tau, T_s \le \tau \end{cases} \tag{8}$$

E. Stackelberg Game

The objectives of each PV and the SP are to maximize the revenue function in Eqn. (3) and to minimize the cost function in Eqn. (4), respectively. To this end, game theory is a well-suited solution. We formulate the interaction between the SP and PVs as a two-stage Stackelberg game, where the SP acts as a leader to determine reward policy p and PVs become followers to respond with $w_{i,s}$. So the Stackelberg game is defined by the strategic form as

$$\Omega = \left\{ \left(SP \cup \{i\}_{i\in\mathcal{I}} \right), \left(p, w_{i,s} \right), \left(C, u_i \right) \right\}. \tag{9}$$

The strategic form is composed of several parts to indicate player set, strategy set and utility set accordingly.

III. Stackelberg Game Solution

A. Stackelberg Equilibrium Analysis

The goal of the proposed Stackelberg game is to find the Stackelberg equilibrium. Here, we denote $\boldsymbol{w^*}$ as the optimal strategies of PVs, i.e., $\boldsymbol{w^*} = [w_{1,s}^*, \ldots, w_{i,s}^*, \ldots, w_{I,s}^*]$ and regard p^* as the optimal decision of SP. The equilibrium is defined as follows.

Definition 1: A series of strategies $(\boldsymbol{w^*}, p^*)$ can be regarded as the Stackelberg equilibrium, if and only if it meets the following set of inequalities.

$$\begin{aligned} \forall w_{i,s}, \ u_i(w_{i,s}^*, p^*) \ge u_i(w_{i,s}, p^*), \\ \forall p, \ C(\boldsymbol{w^*}, p^*) \le C(\boldsymbol{w^*}, p). \end{aligned} \tag{10}$$

We use the typical backward induction method. The optimal decision of PV i is analysed. We take the first and second derivatives of u_i with respect to $w_{i,s}$, which can be written as

$$\begin{aligned} \frac{\partial u_i}{\partial w_{i,s}} &= \frac{-\alpha_i}{1 + w_i^{\max} - w_{i,s}} + p, \\ \frac{\partial^2 u_i}{\partial w_{i,s}^2} &= \frac{-\alpha_i}{\left(1 + w_i^{\max} - w_{i,s}\right)^2} < 0. \end{aligned} \tag{11}$$

The revenue function u_i is strictly concave. Solving the first order optimality condition $\partial u_i / \partial w_{i,s} = 0$, we acquire the optimal decision denoted as $w_{i,s}^*$ which is

$$w_{i,s}^* = 1 + w_i^{\max} - \frac{\alpha_i}{p}. \tag{12}$$

We can get the lower and upper bounds of p regarding to each PV by setting $w_{i,s} = 0$ and $w_{i,s} = w_i^{\max}$, represented by p_{i-} and p_i^-. Based on p_{i-} and p_i^-, the best response of PV i is updated by

$$w_{i,s}^* = \begin{cases} 0, & p \le p_{i-} \\ 1 + w_i^{\max} - \dfrac{\alpha_i}{p}, & p_{i-} < p < p_i^- \\ w_i^{\max}, & p \ge p_i^- \end{cases} \tag{13}$$

We first consider all the PVs belonging to the set $\mathcal{I}$ would like to be recruited by the SP. By substituting $w_{i,s}^*$ into Eqn. (4), C is replaced with $\sum_{i\in\mathcal{I}} (p + w_i^{\max} p - \alpha_i) + \beta_s W - \beta_s \sum_{i\in\mathcal{I}} (1 + w_i^{\max} - \alpha_i/p)$. The optimization problem of the SP with feasible constraints is reformulated as

$$\begin{aligned} &\min_p \ C \\ &\text{s.t. } C1: \ p\left(\sum_{i\in\mathcal{I}} (1 + w_i^{\max}) - W \right) \le \sum_{i\in\mathcal{I}} \alpha_i \\ &\qquad C2: \ p\left(\tau f_s + \sum_{i\in\mathcal{I}} (1 + w_i^{\max}) - W \right) \ge \sum_{i\in\mathcal{I}} \alpha_i \\ &\qquad C3: \ p\left(1 + w_i^{\max} - \frac{f_i H r_i \min(T, \tau)}{f_i + H r_i} \right) \le \alpha_i, \quad \forall i \in \mathcal{I} \end{aligned} \tag{14}$$

Taking the first and second derivatives of C with respect to p, we acquire

$$\begin{aligned} \frac{\partial C}{\partial p} &= \sum_{i\in\mathcal{I}} (1 + w_i^{\max}) - \sum_{i\in\mathcal{I}} \frac{\beta_s \alpha_i}{p^2} \\ \frac{\partial^2 C}{\partial p^2} &= \sum_{i\in\mathcal{I}} \frac{2\alpha_i \beta_s}{p^3} > 0, \end{aligned} \tag{15}$$

which indicate the objective function in the problem is strictly convex in terms of p. In addition, there exist several linear constraints. As a consequence, the problem in (14) is a typical convex optimization problem and can be solved with an optimal solution of p (denoted as p^*) by using existing convex optimization tools, e.g., CVX. By comparing p^* with p_{i-}, PV i determines whether to receive the employment from the SP. Then $\mathcal{I}$ is updated by deleting the PVs with negative responses from the set. Next, the problem in (14) is reformulated and solved again to acquire p^*. The final value of p^* is obtained only when p^* is accepted by all the members in the set $\mathcal{I}$.

Theorem 1: There exists a unique Stackelberg equilibrium $(\boldsymbol{w^*}, p^*)$ in the proposed game Ω.

Proof: Based on the given reward policy p, each PV has an optimal strategy $w_{i,s}^*$, which is unique due to the concave feature of the revenue function u_i, $\partial^2 u_i / \partial w_{i,s}^2 < 0$. Next, the SP has a unique optimal strategy under the best responses from all the PVs, with $\partial^2 C / \partial p^2 > 0$. Both the leader and followers are satisfied since the final strategies $(\boldsymbol{w^*}, p^*)$ have minimized the overall cost and maximized the revenues, respectively. Therefore, the unique equilibrium is achieved. ■

B. Distributed Algorithm With Privacy Protection

As mentioned above, the SP can optimize the pricing parameter to minimize its overall cost in a centralized manner, when the SP has global private information by all the PVs, i.e., α_i, $w_i^{\max}$, f_i, and r_i. This may not be feasible in reality. The rational PVs are not willing to disclose their private information for privacy concerns. As a consequence, the optimal strategies of the PVs cannot be recognized by the SP and the problem in (14) is not formulated thereafter. To address the challenge, we design a distributed algorithm to implement the Stackelberg game and preserve privacy for the PVs simultaneously. In this letter, we focus on privacy protection of PVs that participate in offloading services in PVEC as follows.

Definition 2: Privacy preservation for PVs: each PV is not necessary to reveal individual parameters, including preference values and resource status, to any others. Besides, the decision-making strategy of a PV still remains confidential to other PVs.

The distributed algorithm is performed by each PV separately without exposing any private information to the SP. Each PV independently determines the best response $w_{i,s}^*$ with given reward policy and task requirements, and feedbacks it to the SP. In this way, private information remains unknown to the SP and the other PVs. Moreover, their strategies are just realized by the SP instead of arbitrarily being shared among each other.

The algorithm is consisted of a limited number of iterations. The decision variable of p is increased by Δp iteratively. At each iteration, there exists an inquiry-response stage between the SP and each PV. PV i computes the best response $w_{i,s}^*$ based on Eqn. (13), and then submits the response to the SP with the reward policy p. After collecting all the responses $\boldsymbol{w}^*$, the SP evaluates whether the basic delay constraints and workload constraint can be met for responding PVs, and adds qualified PVs to a set $\mathcal{P}$. According to the valid responses of the PVs belonging to the set $\mathcal{P}$, the SP calculates C and judges whether there exists a smaller value of C. After that, the optimal strategy p^* is updated when necessary. The above procedure is repeated until $\frac{p_{\max}-p_{\min}}{\Delta p}$ iterations are completed. Hence, the total computational complexity of Algorithm 1 to search p^* is $\mathcal{O}\left(\frac{p_{\max}-p_{\min}}{\Delta p}+1\right)$.

The distributed algorithm is an approximate algorithm to solve the equilibrium. As a step size, Δp is a critical parameter to determine the search speed but also influences approximation effect of the final result. When the value of Δp becomes smaller, the final result will be more accurate but more executed iterations are required. Thus, the selection of Δp is a speed-accuracy tradeoff problem.

Algorithm 1 Distributed Algorithm to Reach the Stackelberg Equilibrium Without Privacy Leakage

Input: Prior knowledge about the task: (D, W, τ).
Output: p^*.

Initialization: C^* is given with a larger value and a temporary set $\mathcal{P} = \mathcal{I}$.
for $p = p^{\min} : \Delta p : p^{\max}$ **do**
  for Each PV $i \in \mathcal{P}$ **do**
    PV i responds with $w_{i,s}^*$ according to Eqn. (13).
  end for
  if $\sum_{i\in\mathcal{P}} w_{i,s}^* \le W$ **then**
    if $w_{i,s}^*$ meets the delay constraints in Eqn. (8) **then**
      $\mathcal{P} = i \rightarrow \mathcal{P}$.
    end if
    if $T_s \le \tau$ **then**
      According to the set $\mathcal{P}$, C is calculated by $C = p\sum_{i\in\mathcal{P}} w_{i,s}^* + \beta_s\left(W - \sum_{i\in\mathcal{P}} w_{i,s}^*\right)$.
      if $C \le C^*$ **then**
        $C^* = C$, $p^* = p$.
      end if
    end if
  end if
  Reset the set $\mathcal{P} = \emptyset$.
end for
return p^*

TABLE I
PARAMETER SETTING IN THE SIMULATION

Parameter	Setting
The amount of parked vehicles I	50
Tradeoff parameter α	$[100, 200]$
Parking duration T	12 minutes
Computation capability of parked vehicles f	$[1, 2]$ GHz
A task: D, H, W and τ	$[30, 44]$ MB, 0.3 giga CPU cycles/KB, $[9000, 13000]$ giga CPU cycles, 15 minutes
Dissatisfaction level β_s	6
Computation capability of VEC server f_s	8 GHz
Downlink rate r	3 Mbps

IV. NUMERICAL RESULTS

We evaluate the performance of the scheme via Matlab simulations. The parameters are summarized in Table I. In the system, the tasks are compute-intensive, such as inference of a batch of images. Moreover, the tasks are scheduled and executed sequentially.

In Fig. 2, we compare the proposed algorithm with two baseline algorithms in terms of the overall cost of the SP, when coping with different number of offloading tasks from the SP. CVX is a centralized optimization algorithm which can obtain the global optimum, i.e., the theoretical value, from Eqn. (14). But the algorithm needs to collect all the private information of the PVs for centralized calculation, leaving the PVs susceptible to privacy leakage. As a result, CVX cannot preserve PVs privacy while getting the optimal solution.

Oppositely, our distributed algorithm not only acquires an approximately optimal solution, but also provides privacy preservation to the PVs by keeping their private information localized. As can be seen, the result under the distributed algorithm is close to that under the privacy-violating CVX. Moreover, the approximated solution is still effective with the increasing amount of processed tasks. This demonstrates that our proposal has a good approximation on

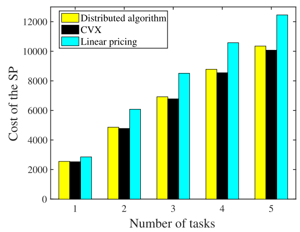

Fig. 2. Comparison of the cost of the SP with various schemes and different number of tasks.

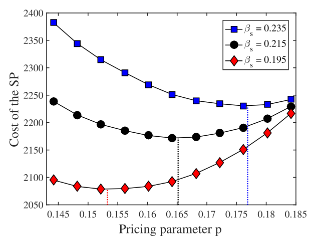

Fig. 3. Comparison of the cost of the SP with different p and β_s.

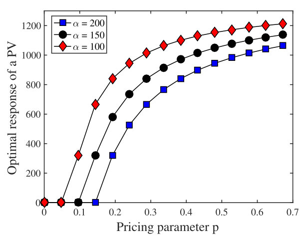

Fig. 4. Comparison of the optimal response of a PV with different p and α.

the theoretical value in a distributed fashion with privacy protection.

Another baseline is linear pricing scheme, where the rewards given to the PVs are linear to their computation capabilities, i.e., kf_i, as in [9]. Simulation result proves that the distributed algorithm can reduce 25% of the overall cost of the SP, compared with the linear pricing scheme. The cost reduction is more obvious when having more tasks to be processed.

We investigate the impacts of pricing parameter p and dissatisfaction level β_s on the cost of the SP as shown in Fig. 3. The optimal strategy p^* can be found in Fig. 3 which is marked by the dashed. p^* is increased with the enhancement of β_s. The reason is that a larger value of β_s indicates the greater dissatisfaction caused by the un-offloaded workloads. Therefore, the SP is willing to offer a higher p to motivate the PVs to process more workloads.

Fig. 4 illustrates the optimal response w_s^* of a PV with diverse pricing parameter p and tradeoff parameter α. w_s^* decreases with the increase of α under the same p. With the larger value of α, PVs can gain a higher revenue from the internal satisfactory degree. Thus, PVs have a higher preference to reserve residual workload capacities to meet any possible individual demand, rather than undertake workloads for the SP. When the value of p is extremely small, w_s^* is equal to 0, which indicates the PVs will not join PVEC and

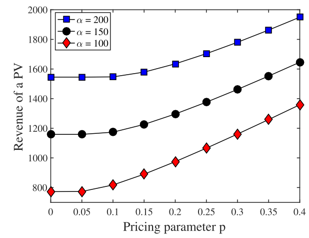

Fig. 5. Comparison of the revenue of a PV with different p and α.

execute tasks for the SP. This is because the rational PVs will not participate in the offloading services if the p given by the SP does not exceed the lower bound p_- of PVs, as in Eqn. (13).

Fig. 5 plots the revenue u of a PV with variable pricing parameter p and tradeoff parameter α. When p becomes smaller, the revenue u is fixed and will not be affected by the p. As mentioned above, the rational PV will not take part in the offloading services, until the p is larger than the lower bound p_-.

V. Conclusion

In this letter, we investigate the optimal task assignment problem with delay constraint and special feature of PVs in PVEC. The problem is solved within the Stackelberg game framework, where the SP minimizes the overall cost while the PVs achieve revenue maximization. To reach the Stackelberg equilibrium, we elaborately design a privacy-preserving algorithm which can be implemented among PVs in a distributed fashion. The distributed algorithm allows PVs to interact with the SP with privacy protection. Finally, numerical results demonstrate our proposed scheme for PVEC is effective and efficient.

References

[1] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[2] 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.

[3] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 420–423, Apr. 2019.

[4] X. Huang, P. Li, and R. Yu, "Social welfare maximization in container-based task scheduling for parked vehicle edge computing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1347–1351, Aug. 2019.

[5] 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.

[6] X. Wang, X. Chen, W. Wu, N. An, and L. Wang, "Cooperative application execution in mobile cloud computing: A Stackelberg game approach," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 946–949, May 2016.

[7] Y. Mao, J. Zhang, Z. Chen, 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.

[8] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[9] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.