



# Optimized Workload Allocation in Vehicular Edge Computing: A Sequential Game Approach

Dongdong Ye, Maoqiang Wu, Jiawen Kang, and Rong Yu(✉)

Guangdong University of Technology, Guangzhou, China  
dongdongye8@163.com, maoqiang.wu@vip.163.com,  
kjwx886@163.com, yurong@ieee.org

**Abstract.** With the development of Vehicle-to-Everything (V2X) communication technologies, Vehicular Edge Computing (VEC) is utilized to speed up the running of vehicular computation workload by deploying VEC servers in close proximity to vehicular terminals. Due to resource limitation of VEC servers, VEC servers are unable to perform a large number of vehicular computation workloads. To improve the performance of VEC servers, we propose a new workload allocation framework where vehicular terminals are divided into Resource Provision Terminals (RPTs) and Resource Demand Terminals (RDTs). In this framework, we design an optimized workload allocation strategy through a sequential Stackelberg game. With the sequential Stackelberg game, a VEC server, RDTs, and RPTs achieve an efficient coordination of the workload allocation. The sequential Stackelberg game is proven to reach two sequential Nash Equilibriums. The simulation results validate the efficiency of the optimized workload allocation strategy.

**Keywords:** Vehicular edge computing · Workload allocation  
Sequential Stackelberg game

## 1 Introduction

With the development of Vehicle-to-Everything (V2X) communication technologies, vehicular networks have gained extensive attention in recent years [1]. Meanwhile, vehicular terminals in vehicular networks can run various new applications such as real-time navigation, interactive gaming and augmented reality. However, vehicular terminals have relatively limited resources due to the physical size constraint. Therefore, it is difficult for vehicular terminals to support real-time and low-latency applications.

In order to meet the requirement of resource-constraint vehicular terminals, Vehicular Edge Computing (VEC) was proposed for vehicular network. VEC is similar to the concept of vehicular fog computing which has been proposed in [2]. VEC locally deploys light-weight cloud servers in close proximity to vehicular

terminals. Therefore, vehicular terminals can get realtime interaction and low delay services from VEC servers. Due to resource limitation of VEC servers, VEC servers are unable to perform a large number of vehicular computation workloads [3]. Thus, an optimized workload allocation strategy is essential for a VEC server in vehicular network.

Many researchers engaged in studying the efficient workload allocation strategies. In [3], the authors studied delay constrained offloading for vehicular edge computing in cloud-enabled vehicular networks and designed an efficient computation offloading scheme with a contract theoretic approach. In [4], the authors proposed to combine the vehicular cloud with the infrastructure-based cloud to expand the current available resources for task requests from smartphones, and designed an algorithm to select the suitable cloud service provider to perform the requested task. In [5], the authors proposed a non-cooperation matrix game to balance the workload of multiple local servers for vehicular terminals. In [6], the authors used Semi-Markov Decision Process method to optimize computation resource allocation scheme in vehicular cloud computing.

Vehicular terminals, having idle computation resources, are normally distributed within the coverage of the VEC server. However, few work has considered utilizing idle computation resources of vehicular terminals as the compensation of the VEC server. In addition, the mobility of vehicular terminals has been ignored in those work. In this paper, we propose a new workload allocation framework in vehicular network. The main contributions of this paper are summarized as follows.

- We propose a new workload allocation framework. In this framework, vehicular terminals are divided into Resource Provision Terminals (RPTs) and Resource Demand Terminals (RDTs). We design an optimized workload allocation strategy for the combination of RPTs, RDTs, and a VEC server.
- We elaborately use a sequential Stackelberg game to analyze and solve the optimized workload allocation. With the sequential Stackelberg game, the VEC server, RPTs and RDTs can maximize their utilities, respectively.
- The sequential Stackelberg game is proven to reach two unique sequential Nash Equilibriums. We propose a sequential algorithm to find out the unique solution for the Nash Equilibriums.

The rest of this paper is organized as follows. System model is introduced in Sect. 2. Problem formulation is presented in Sect. 3. We present the workload allocation strategy and propose a sequential algorithm for the sequential Stackelberg game in Sect. 4. The simulation results are presented in Sect. 5. We conclude the paper in Sect. 6.

## 2 System Model

Figure 1 shows the new workload allocation framework. The framework consists of a VEC server and vehicular terminals within the communication radius of the VEC server. In one cycle, when vehicular terminals are executing real-time and

low-latency applications, they become Resource Demand Terminals (RDTs) who require computation resources for computation services. We denote  $N$  as the set of RDTs. When vehicular terminals are under low-loaded condition, they become Resource Provision Terminals (RPTs) who have idle computation resources. We denote  $M$  as the set of RPTs. We denote  $K$  as the set of total vehicular terminals. The roles of the RDTs and RPTs may be converted under different conditions ( $K = N + M$ ).

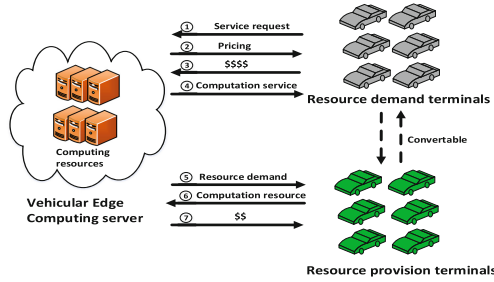


Fig. 1. The workload allocation framework.

RDT  $n \in N$  wants to purchase computation amount  $x_n$  from the VEC server. The unit price charged by the VEC server for computation resource is denoted as  $p_n$ . Taking the payment into consideration, each RDT adjusts its required computation amount. After collecting the required computation amount from RDTs, the VEC server adjusts the unit price for its own maximum utility. Finally, the VEC server and RDTs reach a coordination with mutually satisfactory unit price and computation amount. Further, to improve the performance of the VEC server, the VEC server offers an incentive  $R_0$  to recruit idle computation resources from  $M$  to serve RDTs. The computation amount that the RPT  $m \in M$  provides is denoted as  $y_m$ . Considering the cost and obtained incentive, each RPT adjusts its offered computation amount. After collecting the offered computation amount from RPTs, the VEC server adjusts the incentive. Finally, the VEC server and RPTs reach a coordination with mutually satisfactory incentives and computation amount.

### 3 Problem Formulation

#### 3.1 Utilities of RDTs and RPTs

The satisfaction function monotonically increases on  $x_n$ . Therefore, we define the satisfaction function of RDT  $n$  as  $\varphi_n \log(x_n + 1 - x_n^{\min})$ , where  $\varphi_n$  is a parameter and  $x_n^{\min}$  is the minimum demand computation amount. The cost of RDT  $n$  is the payment to the VEC server, which is given by  $p_n x_n$ . Therefore, the utility of RDT  $n$  is denoted as

$$u_n = \varphi_n \log(x_n + 1 - x_n^{\min}) - p_n x_n. \tag{1}$$

Following the principle of proportional sharing, the incentive amount of RPT  $m$  is proportional to its offered computation amount. For convenience, the incentive of RPT  $m$  is given by  $\frac{y_m}{\sum_{m \in M} y_m} R_0$ . Therefore, the utility of RPT  $m$  is denoted as

$$u_m = \frac{y_m}{\sum_{m \in M} y_m} R_0 - e_m y_m, \tag{2}$$

where  $e_m$  is unit energy consumption. According to the realistic measurements in [7,8], we can set  $e_m = 10^{-11}(s_m)^2$ , where  $s_m$  is computation capability of RPT  $m$ .

### 3.2 Utility of VEC Server

The total cost includes the computation energy consumption and transmission energy consumption of the VEC server.

Firstly, the VEC server serves RDTs with computation amount  $\sum_{n \in N} x_n$ . To reduce the energy consumption, the VEC server pays  $R_0$  to recruit computation amount  $\sum_{m \in k} y_m$  from RPTs. From [9], the computation energy consumption of the VEC server is denoted as  $c_0 = \sigma_0(z_0)^2 + \eta_0 z_0 + \alpha_0$ , where  $z_0 = \sum_{n \in N} x_n - \sum_{m \in k} y_m$ . It is provided by the VEC server.

Secondly, the transmission energy consumption of the VEC server is related to the mobility of RPTs. From [4,10], we know that the wireless channel would fade when RPT  $m$  is driving at high speed. When the speed of RPT  $m$  is less than a threshold speed, the bandwidth resource of  $m$  is  $B_m^{in}$ . When the speed of RPT  $m$  exceeds the threshold speed, the fading rate of RPT  $m$  is proportional to its speed and denoted as  $r_m = wv_m$ , where  $w$  is a constant factor. Therefore, the bandwidth resources consumed by RPT  $m$  are denoted as  $B_m = B_m^{in} + \int r_m dt$ , where  $t$  is the residence time of RPT  $m$  within the communication coverage of the VEC server. From [11], the transmission rate of RPT  $m$  can be obtained by  $R_m = B_m \log_2(1 + \frac{P_m d_m^{-\eta} |h_0|^2}{N_0})$ , where  $\eta$  is a parameter and  $d_m$  is the average distance between the VEC server and RPT  $m$ .  $P_m$  is the transmit power of the VEC server.  $h_0$  is the complex Gaussian channel coefficient that follows the complex normal distribution  $CN(0, 1)$ .  $N_0$  is the additive white Gaussian noise (AWGN) at the RPT receivers. Therefore, the transmission energy consumption is denoted by  $q_m = \frac{y_m}{R_m} P_m$ . The utility of the VEC server is denoted as

$$u_0 = \sum_{n \in N} p_n x_n - c_0 - R_0 - \beta_0 \sum_{m \in M} q_m, \tag{3}$$

where  $\beta_0$  is unit cost per energy consumption.

All the RPTs, the VEC server, and the RDTs are assumed to be rational and want to make multilevel independent decision to maximize their utilities. It is impossible that the utilities of three-party are satisfied simultaneously, due

to the heterogenous framework of the workload allocation. A sequential stackelberg game studies the sequential decision of the players. Therefore, we model the sequential decision as the sequential stackelberg game [12]. The problem is formulated to

$$\max_{R_0}(u_0), \tag{4}$$

$$s.t. \max_{x_n \in X}(u_n | p_n, R_0), \tag{5}$$

$$\max_{y_m \in Y}(u_m | R_0). \tag{6}$$

The optimal decisions of the three-party including the VEC server, the RDTs and the RPTs are analyzed based on their utilities. In the first stage, the VEC server is the leader and the RDTs are the followers. Their optimal decisions are charged prices and requested computation amount respectively. In the second stage, the VEC server is the leader and the RPTs are followers. Their optimal decisions are optimal incentives and offered computation amount.

## 4 Solution and Algorithm

In this section, we use the backward induction method to prove the existence and uniqueness of the sequential Nash Equilibriums for problem (6). We propose a sequential algorithm for the sequential Stackelberg game.

### 4.1 Stackelberg Equilibrium Solution

#### 4.1.1 Nash Equilibrium Between VEC Server and RDTs

**Theorem 1.** A unique Nash Equilibrium exists between the RPTs and the VEC server.

*Proof:* by taking the derivative of the utility function  $u_n$  with respect to  $x_n$ , we obtain

$$\frac{d^2u_n}{dx_n^2} = -\frac{\varphi_n}{(x_n + 1 - x_n^{\min})^2} < 0. \tag{7}$$

Clearly, the utility function  $u_n$  of  $n$  is concave function, which indicates that the maximum value of the utility function exists. Using first order optimality condition  $\frac{du_n}{dx_n} = 0$ , we get the optimal computation amount  $x_n^* = \frac{\varphi_n}{p_n} + x_n^{\min} - 1$ .

From the above equation, we know the relationship between the price and the optimal computation amount. We derivative the optimal price, denoted as  $p_n^* = \frac{\varphi_n}{x_n + 1 - x_n^{\min}}$ , based on the interaction between RDT  $n$  and the VEC server. We substitute  $p_n^*$  into Eq. (3) and get

$$u_0 = \sum_{n \in N} \frac{\varphi_n}{x_n + 1 - x_n^{\min}} x_n - c_0. \tag{8}$$

The second derivative of the function can be obtained as

$$\frac{\partial^2 u_0}{\partial x_n^2} = -2 \frac{\varphi_n}{(x_n + 1 - x_n^{\min})^3} - 2\beta_0\sigma_0 < 0. \quad (9)$$

Clearly, the second-order derivative of the utility of the VEC server is negative and strictly convex function. Therefore a unique Nash Equilibrium exists in the game. ■

#### 4.1.2 Nash Equilibrium Between VEC Server and RPTs

In the stage, the VEC server is the leader and the RPTs are the followers. We prove that it exists an Nash Equilibrium. The Nash Equilibrium is unique.

**Definition 1.** When the other followers' strategies  $\mathbf{y}_{-m}$  are given, the best response function  $\mathbf{f}_m(y_m, \mathbf{y}_{-m})$  of RPT can be defined by

$$\mathbf{f}_m(y_m, \mathbf{y}_{-m}) = \arg \max u_m(y_m, \mathbf{y}_{-m}). \quad (10)$$

**Theorem 2 (Existence).** An Nash Equilibrium exists among RPTs.

*Proof:* give the second order condition of the RPT's utility  $u_m(y_m, \mathbf{y}_{-m})$  as

$$\frac{\partial^2 u_m(y_m, \mathbf{y}_{-m})}{\partial y_m^2} = -2 \left( \frac{1}{y_m + \sum_{j \in k \setminus m} y_j} \right)^3 R_0 < 0. \quad (11)$$

Since the second-order derivative of  $u_m$  is negative, the utility  $u_m$  is a strictly convex function in  $y_m$ . Therefore an Nash Equilibrium exists in the game. ■

The Nash Equilibrium is unique. The key of the Nash Equilibrium is to prove that the best response function of each RPT is a standard function [14].

**Definition 2.**  $\mathbf{f}(\mathbf{p}) = (\mathbf{f}_1(\mathbf{p}), \mathbf{f}_2(\mathbf{p}), \dots, \mathbf{f}_M(\mathbf{p}))$ , where  $\mathbf{p} = (p_1, \dots, p_M)$ .  $\mathbf{f}(\mathbf{p})$  is said to be standard if it satisfies the following properties for all  $\mathbf{p} \geq 0$

- *Positivity:*  $\mathbf{f}(\mathbf{p}) > 0$ .
- *Monotonicity:* for all  $\mathbf{p}$  and  $\mathbf{p}'$ , if  $\mathbf{p} > \mathbf{p}'$  then  $\mathbf{f}(\mathbf{p}) > \mathbf{f}(\mathbf{p}')$ .
- *Scalability:* for all  $\mu > 1$ ,  $\mu \mathbf{f}(\mathbf{p}) \geq \mathbf{f}(\mu \mathbf{p})$ .

**Theorem 3.** The best response function  $\mathbf{f}_m(y_m, \mathbf{y}_{-m})$  of RPT  $m$  is a standard function of  $\mathbf{y}_{-m}$ .

*Proof:* from the Theorem 1, we know that the utility function  $u_0(y_m, \mathbf{y}_{-m})$  is strictly concave. Let  $\frac{\partial u_0}{\partial x_i} = 0$ , we get the best function  $\mathbf{f}_m(y_m, \mathbf{y}_{-m})$  of  $m$ ,

$$\mathbf{f}_m(y_m, \mathbf{y}_{-m}) = \begin{cases} \sqrt{\frac{\sum_{i \in k/m} y_i R_0}{e_m}} - \sum_{i \in k/m} y_m, \\ 0, R_0 \leq e_i \sum_{i \in k/m} y_m. \end{cases} \quad (12)$$

Next, we prove that the best function  $\mathbf{f}_m(y_m, \mathbf{y}_{-m})$  satisfies the three properties of a standard function.

- *Positivity*: when  $y_m > 0$  and  $R_0 \leq e_i \sum_{i \in k/m} y_i$  are satisfied, we get

$$f_m(y_m, \mathbf{y}_{-m}) = \sqrt{\frac{\sum_{i \in k/m} x_i R_0}{e_m}} - \sum_{i \in k/m} x_i > 0. \tag{13}$$

- *Monotonicity*: given the first order condition of  $f_m(y_m, \mathbf{y}_{-m})$  with respect to  $y_i$ , we obtain

$$\frac{\partial f_m(y_m, \mathbf{y}_{-m})}{\partial y_i} = \sqrt{\frac{\sum_{i \in k/m} y_i R_0}{4e_m}} - 1 > 0. \tag{14}$$

We get the constrain  $\sum_{i \in k/m} y_i > \frac{4E_m^{consu}}{R_0}$ . When it is satisfied, the monotonicity is satisfied.

- *Scalability*: based on (12), we obtain

$$\lambda f_m(y_m, \mathbf{y}_{-m}) - f_m(\lambda y_m, \mathbf{y}_{-m}) = (\lambda - \sqrt{\lambda}) \sqrt{\frac{R_0 \sum_{i \in k/m} y_i}{e_m}}. \tag{15}$$

For  $\forall \lambda > 1$ , we have  $(\lambda - \sqrt{\lambda}) > 0$ . Therefore, it is positive.

Based on Eq.(12), we get the total amount computation of the RPTs as  $\sum_{m \in k} y_m = \frac{(k-1)}{\sum_{m \in k} e_m} R_0$ . We substitute  $\sum_{m \in k} y_m$  into Eq.(6). The problem can be written as

$$\min(c_1), \tag{16}$$

$$s.t. k > 1, \tag{17}$$

$$R_0 > 0, \tag{18}$$

$$\beta_0, \sigma_0, \eta_0, \alpha_0 > 0, \tag{19}$$

where

$$c_1 = R_0 + \beta_0 \sigma_0 \left( \sum_{n \in N} x_n - \frac{(k-1)}{\sum_{m \in k} e_m} R_0 \right)^2 + \beta_0 \alpha_0 + \beta_0 \eta_0 \left( \sum_{n \in N} x_n - \frac{(k-1)}{\sum_{m \in k} e_m} R_0 \right) + \frac{(k-1)R_0}{\sum_{m \in k} e_m} \sum_{m \in k} \frac{P_m}{R_m}. \tag{20}$$

We give the second order condition and get  $\frac{\partial^2 c_1}{\partial R_0^2} = 2A^2 \sigma_0 \beta_0 > 0$ , where  $A = \frac{k-1}{\sum_{m \in k} e_m}$ . Clearly, the cost of the VEC server is convex, which indicates that the

minimum value of this function exists. Therefore, given first order optimality condition  $\frac{dc_1}{dR_0} = 0$ , we get the optimal  $R_0^*$

$$R_0^* = \frac{2A \sum_{n \in N} x_n + \beta_0 \eta_0 A - 1 - A \sum_{m \in k} \frac{P_m}{R_m}}{2A^2 \sigma_0 \beta_0}. \quad (21)$$

## 4.2 Sequential Distributed Algorithm

To reach two sequential Nash Equilibrium, we propose a sequential algorithm. Details are given in Algorithm 1.

---

### Algorithm 1. Reach the Nash Equilibrium

---

**Input:**  $\bar{e} = (e_1, e_2, \dots, e_n)$   
**Output:**  $\bar{X} = (x_1, x_2, \dots, x_n)$ ,  $\bar{Y} = (y_1, y_2, \dots, y_m)$   
**Initialize** parameters  $e$ ,  $\bar{N}$ ,  $\bar{M}$ ,  $U_0^{up}$ ,  $U_0$ ,  $R_0$ ,  $k'$ ,  $error = (u_o^{up} - u_o)^2$   
**for** *RPT*  $m \leftarrow 1$  **to**  $M$  **do**  
    **if**  $E_i^{consu} < \frac{\sum_{m \in k} E_m^{consu}}{(k-1)}$  **then**  
        |  $k' \leftarrow k' \cup \{m\}$ ,  $m \leftarrow m + 1$   
    **end**  
**end**  
**for** *RPT*  $m \leftarrow 1$  **to**  $k'$  **do**  
    |  $y_i^* = \frac{(k-1)}{\sum_{m \in k} E_m^{consu}} R_0 (1 - \frac{(k-1)}{\sum_{m \in k} e_m} e_i)$   
**end**  
**for** *RDT*  $n \leftarrow 1$  **to**  $n$  **do**  
    **if**  $x_n = \frac{\varphi_n}{p_n} + x_n^{\min} - 1 > 0$  **then**  
        |  $n' \leftarrow n' \cup \{n\}$ ,  $n \leftarrow n + 1$   
    **end**  
**end**  
**return**  $(y_1^*, y_2^*, \dots, y_j^*)$   $j \in k'$   
**return**  $(x_1^*, x_2^*, \dots, x_q^*)$   $q \in n'$   
**return**  $\sum_{z \in N'} x_z$   
**while**  $(u_o^{up} - u_o)^2 > error$  **do**  
    The VEC server calculates the total utility.  
    **if**  $(u_o^{up} - u_o)^2 < error$  **then**  
        | The algorithm ends.  
        | **Output:**  $\bar{X} = (x_1, x_2, \dots, x_n)$ ,  $\bar{Y} = (y_1, y_2, \dots, y_m)$   
        |  
    **end**  
     $R_0 \leftarrow R_0 + 1$ ;  $u_o^{up} = u_o$ .  
**end**

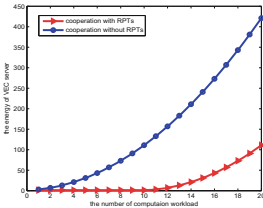
---



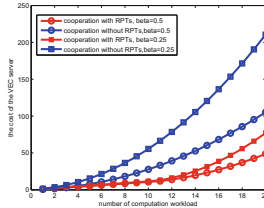
## 5 Numerical Results

We evaluate the performance of the proposed workload allocation strategy by simulations. The VEC server is assumed to be located in (500, 500) (in  $m$ ). We use the following parameter settings as that in [4, 5, 7, 9]:  $\beta \in [2, 50]$ ,  $\sigma = 2$ ,  $\eta \in (0, \frac{3}{2})$ ,  $\alpha \in [1, 10]$ ,  $\varphi \in [10, 20]$ ,  $v \in [1, 10]$ ,  $s_m = 50$  to 100 GZ,  $B_m = 6$  to 10 Mbps.

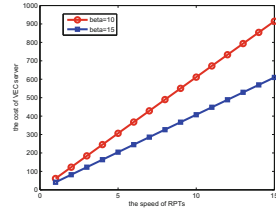
Figure 2 shows the total energy consumption of the VEC server with respective to the number of computation resource. The energy consumption of the VEC server increases as the number of computation resource increases. The energy consumption of the VEC server by our proposed workload allocation strategy consumes less energy for the same computation resource. It is because the VEC server utilizes idle computation resource of the RPTs to reduce its energy consumption. Figure 3 shows the cost of the VEC server in computation resources for two values of  $\beta_0$ . The VEC server has higher cost with larger  $\beta_0$ . With our proposed strategy, the VEC server consumes lower cost in computation resources. It is because the VEC server consumes lower cost and cooperates with the RPTs. Figure 4 shows the impact of the speed of RPTs on the cost of the VEC server. The cost of the VEC server  $n$  increases as the speed of RPTs increases. It is because that RPTs provide less computation resources as the cost of RPTs increases.



**Fig. 2.** Impact of computation resources on the VEC server



**Fig. 3.** Cost of the VEC server with different  $\beta_0$



**Fig. 4.** Impact of the speed of RPTs on the VEC server.

## 6 Conclusion

In this paper, we propose a workload allocation framework in VEC. In the framework, RPTs are recruited to improve the performance of computation services which serve the RDTs. An optimized workload allocation strategy is proposed to maximize the utilities of the VEC server, the RPTs and the RDTs. We use a sequential Stackelberg game to design the strategy. The sequential Stackelberg game is proven to reach two sequential Nash Equilibriums. The simulations validate the efficiency of the optimized workload allocation strategy.

**Acknowledgment.** This work was supported in part by programs of NSFC under Grant nos. 61422201, 61370159 and U1301255, U1501251, the Science and Technology Program of Guangdong Province under Grant no. 2015B010129001, Special-Support Project of Guangdong Province under grant no. 2014TQ01X100, High Education Excellent Young Teacher Program of Guangdong Province under grant no. YQ2013057, Science and Technology Program of Guangzhou under grant no. 2014J2200097.

## References

1. Yu, R., Zhang, Y., Gjessing, S., Xia, W., Yang, K.: Toward cloud-based vehicular networks with efficient resource management. *IEEE Netw.* **27**(5), 48–55 (2013)
2. Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., Chen, S.: Vehicular fog computing: a viewpoint of vehicles as the infrastructures. *IEEE Trans. Veh. Technol.* **65**(6), 3860–3873 (2016)
3. Zhang, K., Mao, Y., Leng, S., Vinel, A., Zhang, Y.: Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 288–294. IEEE (2016)
4. Zhang, H., Zhang, Q., Du, X.: Toward vehicle-assisted cloud computing for smartphones. *IEEE Trans. Veh. Technol.* **64**(12), 5610–5618 (2015)
5. Yu, R., Ding, J., Maharjan, S., Gjessing, S., Zhang, Y., Tsang, D.: Decentralized and optimal resource cooperation in geo-distributed mobile cloud computing. *IEEE Trans. Emerg. Top. Comput.* (2017)
6. Zheng, K., Meng, H., Chatzimisios, P., Lei, L., Shen, X.: An SMDP-based resource allocation in vehicular cloud computing systems. *IEEE Trans. Ind. Electron.* **62**(12), 7920–7928 (2015)
7. Chen, X.: Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. Parallel Distrib. Syst.* **26**(4), 974–983 (2015)
8. Wen, Y., Zhang, W., Luo, H.: Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. In: INFOCOM, 2012 Proceedings IEEE, pp. 2716–2720. IEEE (2012)
9. Deng, R., Lu, R., Lai, C., Luan, T.H., Liang, H.: Optimal workload allocation in fog-cloud computing towards balanced delay and power consumption. *IEEE Internet Things J.* **3**(6), 1171–1181 (2016)
10. Yu, R., Huang, X., Kang, J., Ding, J., Maharjan, S., Gjessing, S., Zhang, Y.: Cooperative resource management in cloud-enabled vehicular networks. *IEEE Trans. Ind. Electron.* **62**(12), 7938–7951 (2015)
11. Zhang, Y., Pan, E., Song, L., Saad, W., Dawy, Z., Han, Z.: Social network aware device-to-device communication in wireless networks. *IEEE Trans. Wire. Commun.* **14**(1), 177–190 (2015)
12. Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, R., Han, Z.: Fog computing in multi-tier data center networks: a hierarchical game approach. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2016)
13. Lee, J., Guo, J., Choi, J.K., Zukerman, M.: Distributed energy trading in microgrids: a game-theoretic model and its equilibrium analysis. *IEEE Trans. Ind. Electron.* **62**(6), 3524–3533 (2015)
14. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: *Algorithmic Game Theory*, vol. 1. Cambridge University Press, Cambridge (2007)