Noname manuscript No.
(will be inserted by the editor)

Contract-theoretic Approach for Delay Constrained Offloading in Vehicular Edge Computing Networks

Ke Zhang · Yuming Mao · Supeng Leng · Sabita Maharjan · Alexey Vinel · Yan Zhang

Received: date / Accepted: date

Abstract Mobile Edge Computing (MEC) is a promising solution to improve vehicular services through offloading computation to cloud servers in close proximity to mobile vehicles. However, the self-interested nature together with the high mobility characteristic of the vehicles make the design of the computation offloading scheme a significant challenge. In this paper, we propose a new Vehicular Edge Computing (VEC) framework to model the computation offloading process of the mobile vehicles running on a bidirectional road. Based on this framework, we adopt a contract theoretic approach to design optimal offloading strategies for the VEC service provider, which maximize the revenue of the provider while enhancing the utilities of the vehicles. To further improve the utilization of the computing resources of the VEC servers, we incorporate task priority distinction as well as additional resource providing into the design of the offloading scheme, and propose an efficient VEC server selection and computing resource allocation algorithm. Numerical results indicate that our proposed schemes greatly enhance the revenue of the VEC provider, and concurrently improve the utilization of cloud computing resources.

K. Zhang, Y. Mao and S. Leng
School of Communication and Information Engineering, University of Electronic Science and Technology of China
E-mail: {zhangke, ymmao, spleng}@uestc.edu.cn

S. Maharjan
Simula Research Laboratory, Norway
E-mail: sabita@simula.no

A. Vinel
Halmstad University, Sweden
E-mail: alexey.vinel@hh.se

Y. Zhang
University of Oslo, Norway
E-mail: yanzhang@ieee.org

Keywords Vehicular network · Cloud · Mobile edge computing · Contract theory

1 Introduction

The advancements in Internet of Things (IoT) and wireless technologies bring us pervasive smart devices such as smart vehicles that can facilitate realizing many novel and powerful mobile applications [1]. Some of such applications include interactive infotainment, traffic cognition and automatic driving [2]-[6]. However, with the drastically increasing needs for resources along with stricter requirements on performance for advanced vehicular applications, supporting large computing intensive applications is a big challenge for the resource constrained vehicles [7][8].

To cope with the explosive application demands of the vehicular terminals, cloud-based vehicular networking is widely considered as a promising approach to improve the performance of the services [9]-[11]. In cloud-enabled networks, computation processing and storage for vehicular applications are provided as services on the cloud [12]. Thus, the complicated computing tasks can either run locally on the vehicular terminals or be offloaded to the remote computation cloud.

By leveraging rich computational resources from cloud, both the performance of the vehicular applications as well as the resource utilization of the cloud can be improved. However, the long distance between mobile vehicles and remote cloud servers may incur significant network transmission delay as well as considerable overhead [13][14]. This latency and overhead seriously impairs the performance of delay-sensitive mobile applications and the computation offloading efficiency. A new architecture and technology known as Mobile Edge Computing (MEC) has emerged to address the above challenges, which pushes cloud services to the edge of the radio network, and provides a cloud-based computation offloading in close proximity to the mobile vehicular terminals [15].

Due to proximity to mobile vehicles, the network latency accessing to cloud computing services can be greatly reduced, which enables MEC to provide fast interactive response in the computation offloading service. However, compared to the traditional cloud servers located in the backbone network with powerful computation platforms, MEC servers may suffer from resource limitation. Furthermore, unlike handheld mobile devices that always move slowly within a relatively small range, smart vehicles have a unique feature in terms of their high mobility [16]. Considering the limited coverage area of each RoadSide Unit (RSU), a vehicle may pass by several RSUs within the delay tolerance time of its applications. In MEC systems, servers are always equipped with the RSUs. Thus, in order to improve the computation offloading efficiency while ensuring the performance constraints, it is imperative to jointly investigate MEC server selection strategies for offloading data transmission as well as the MEC cloud resource allocation schemes.

Intuitively, mobile computations can be migrated effectively in MEC systems with well-designed offloading mechanisms. However, in practice, MEC service is always provided by operators, and vehicles should pay to use this service. Since vehicle users are self-interested and want to maximize their own profit, it is unrealistic to assume that they follow the control instructions from the MEC service providers unconditionally. This factor poses a significant challenge on the design of an optimal offloading scheme.

To address this challenge, we turn to contract theory, which is a powerful framework from economics and makes the rational trading participants act according to the contractual arrangements [17]. In this paper, we investigate the characteristics of edge computing in vehicular cloud networks, and design an optimal contract theoretic computation offloading scheme, which maximizes the revenues gained by the cloud service providers. The main contributions of this paper are listed below:

- We propose a new Vehicular Edge Computing (VEC) offloading framework, where both the computing resource capacity of the VEC servers and the high-speed mobility of vehicles are considered.
- We model the VEC offloading process with a contract theoretic approach, and design a contract-based offloading scheme, which maximizes the revenue of the VEC provider while also improving the utilities of the vehicles.
- To further improve the computing resource utilization of the VEC servers while also ensuring the delay constraints of the vehicular applications, we propose an efficient algorithm for VEC server selection and computing resource assignment.

The rest of the paper is organized as follows. In Section II, we review related works. The system model is presented in Section III. The contract-based VEC offloading schemes are described in Section IV. In Section V, we propose an effective delay-constrained VEC server selection and computing resource allocation algorithm. We present performance evaluation in Section VI and conclude the paper in Section VII.

2 Related Work

Vehicular cloud networks, an integration of vehicular communication and mobile computing, possesses the potential to handle massive computing in a flexible and virtualized manner [18]-[20]. The authors in [21] formulated the computation resource allocation of a vehicular cloud computing system as a semi-Markov decision process, and proposed an optimal decision-making scheme to maximize the expected reward. In [22], the authors introduced a collaborative traffic information sharing system, where traffic images provided by a vehicular cloud are utilized to assist drivers for route planning and route decisions. The authors in [12] proposed a backoff-based wireless resource scheduling method in mobile cloud computing networks, where the average queueing delay of the multiuser applications is minimized. The authors in [16] proposed a coalition game for resource management among cloud service providers, where the resources of the cloud-enabled vehicular network are effectively utilized.

Compared with core-based cloud platform, MEC can provide lower computing delay and higher data throughput, which makes it a promising cloud networking solution [23]-[25]. Driven by this fact, MEC has recently been explored for various applications. The authors in [26] formed a tree hierarchical distributed edge cloud server deployment, which improves the utilization of the cloud resources serving the peak loads from mobile users. To optimize energy consumption of mobile devices while minimizing application latency, the authors in [27] incorporated dynamic voltage scaling technology into partial computation offloading for MEC, and proposed a locally optimal algorithm with the univariate search technique. In [28], the authors focused on the tradeoff between the gain and the cost of live avatars migration, and proposed a profit maximization avatar placement scheme for the edge cloudlet networks. In [29], the authors exploited fog computing and mobile edge computing to detect abnormal or critical events, and demostrated that these technologies are helpful in improving the services readiness required in case of time-critical events. In [30], the authors utilized mobile edge computing to manage smart grid data, where the large data sets generated by various smart grid devices are disseminated efficiently with the aid of cloud mechanisms. In [31], the authors formulated edge computation offloading decision as multi-user game, and designed a distributed algorithm to achieve a Nash equilibrium.

Among the aforementioned studies, only [21] and [30] have taken into account the impacts of vehicle mobility on MEC server choosing and recently in operation management. In addition, the effectiveness of incentive-based approaches in the design of MEC offloading mechanisms has been ignored in these work. As the offloading nodes are rational and self-interested, they will perform computation offloading in a way that maximizes their revenues. The nodes may not follow the optimal offloading strategies without any incentive. To improve the offloading efficiency, an incentive-based MEC offloading scheme is required.

Contract theory incentivizes two rational entities in a trading system to reach agreements, and has been widely applied in economic control, resource allocation and operation management recently. For example, in [32], the authors modeled the spectrum trading process in cognitive radio networks as a monopoly market, and designed an optimal quality-price contract, which maximizes the utility of spectrum owners. Observing the dominant position of content providers in publish-subscribe networks, the authors in [33] adopted contract theory to optimize the distribution strategies for the content delivery process. In [34], the authors proposed a contract theoretic approach for motivating user equipments to participate in device-to-device communications in cellular networks. In [35], contract theory is introduced as a economic incentive approach for energy trade in the smart grid. However, none of these work have incorporated contract-based strategies into the VEC offloading schemes.

Different from these studies, in this paper we concentrate on the computation offloading process in a vehicular edge cloud network and propose the

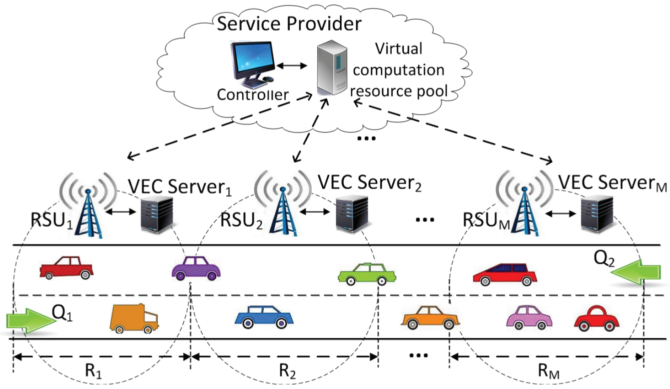

Fig. 1 The computation offloading in a cloud-based vehicular network.

optimal contract-based offloading schemes to improve the revenues of the cloud service providers while guaranteeing the required delay constraints.

3 System Model

Figure 1 shows our proposed VEC network. We consider a two-way road, which has M Road Side Units (RSUs). Each RSU is equipped with a VEC server. We denote the set of these VEC servers as $\mathcal{M} = \{1, 2, ..., M\}$. The computation resources of each VEC server are limited [36]. The amount of the computation resources for VEC server m is b_m units, $m \in \mathcal{M}$. All the RSUs and the VEC servers along the road belong to the same service provider. The resources of the VEC servers together constitute a virtual computation resource pool, which is charged by the provider, along with the wireless access of the vehicular terminals.

Due to the variation in transmission power and the wireless environment, the RSUs may have different wireless coverage areas [37]. Each vehicle accesses the RSU with the strongest signal. Thus, the road can be divided into M segments, whose length is $\{R_1, R_2, ..., R_M\}$, respectively. The vehicles running within the mth segment can only access RSU m and offload their tasks to VEC server m.

We consider that there are Q_1 and Q_2 vehicles arriving at the two ends of the road, respectively. All the vehicles move at a constant speed h, and have identical onboard computation resource, which is denoted as d_0. Each vehicle has a computation task, which is denoted as $T = \{d, t^{max}\}$. Here, d is the amount of computing resources required to accomplish the task, and t^{max} is the maximum allowed end-to-end latency for the task [38][39]. Each computation task can be accomplished either locally on the vehicle or remotely on a VEC server through task offloading.

In our model, there are N types of computation tasks of these vehicles, whose resource requirements are denoted as $\{d_1, d_2, ..., d_N\}$, respectively [40]-

[42]. Thus, the tasks of the vehicular terminals can be presented as $T_i = \{d_i, t_i^{max}\}$, $i \in \mathcal{N} = \{1, 2, ..., N\}$. Without loss of generality, we assume that $d_1 < d_2 < ... < d_N$.

According to their computation task types, these vehicles can be correspondingly classified into N types. Let γ_i be the proportion of vehicles with task T_i in all the arriving vehicles, and $\sum_{i=1}^{N} \gamma_i = 1$. Each vehicle knows its own type. However, as the types of the vehicles are private information of each vehicle, the computation offloading service provider may not be well aware of this information [43]. Thus, we can see that an information asymmetry has occurred between the service provider and the vehicles. Although the service provider can not accurately determine the type of each vehicle, it can obtain the probability distribution of the vehicle types through some statistical information.

Both the service provider and the vehicles are rational and self-interested [44]. Each vehicle can offload its computation task to a VEC server with payment to the service provider. Thus, to maximize its revenue, the service provider derives the optimal amount of the computation resources allocated to offload the tasks and determines the corresponding payment. The information of these services is broadcasted to the vehicles running on the road in contract forms via wireless communication. According to the contract information, each vehicle offloads its task to maximize its own utility.

4 Contract Theoretic Computation Offloading

In this section, we first formulate the computation offloading process as a contract problem. Then we derive the optimal contract solutions, which maximize the utility of the VEC service provider while satisfying the requirement of the vehicles.

4.1 Contract Problem Formulation

As there are N types of vehicles according to their computation tasks, the provider needs to offer N kinds of contracts correspondingly. Let (q_i, p_i) denote the contract designed for the vehicle that belongs to type i ($i \in \mathcal{N}$). Here q_i is the amount of computation resources provided by the VEC servers for offloading the computation task from a type i vehicle. p_i is the payment that type i vehicle should pay to the provider for using the offloading service.

Each vehicle can obtain the contract information through the wireless broadcast from the provider. After that, the vehicle chooses to accept the contract that brings maximal utility to it. Here we define the utility of a type i vehicle gained from offloading its task based on the contract (q_i, p_i) as

$$\begin{aligned} U_v^i(q_i, p_i) &= \lambda \tfrac{d_i}{d_0} \ln(\alpha q_i + \beta) + (1-\lambda) e_0 q_i - p_i, \\ q_i &\le d_i, \quad i \in \mathcal{N}. \end{aligned} \tag{1}$$

In (1), the first item represents the utility of saving computation resource by task offloading. The utility is affected by the original resource utilization d_i/d_0. α and β are coefficients, and $\alpha > 0$, $\beta > 0$. In the second item, e_0 is the utility gained from saving energy for running a task on a remote unit computation resource. λ is the weight factor, and $0 < \lambda < 1$. The inequality $q_i \leq d_i$ ensures that the amount of the resources defined by the contract is less than or equal to the total resources required by the task. Let $G(d_i, q_i) = \lambda \frac{d_i}{d_0} \ln(\alpha q_i + \beta) + (1-\lambda) e_0 q_i$. Then, (1) can be rewritten as

$$U_v^i(q_i, p_i) = G(d_i, q_i) - p_i. \tag{2}$$

As the vehicles are rational, they will not choose the contract which brings negative utilities to them. This property is called Individual Rationality (IR), and can be mathematically expressed as $U_v^i \geq 0$ $(i \in \mathcal{N})$. Besides this property, according to the contract theory, a feasible contract should satisfy the Incentive Compatible (IC) constraint [45]. This constraint indicates that type i vehicles are incentivized to choose the contract specifically designed for their own type, but not the contracts of the others. The IC constraint can be represented as $U_v^i(q_i, p_i) \geq U_v^i(q_j, p_j)$, $i \neq j$, $i, j \in \mathcal{N}$.

The utility of the provider providing computation task offloading service based on these contracts is defined as

$$U_{SP} = Q \sum_{i=1}^{N} \gamma_i (p_i - c q_i), \tag{3}$$

where c is the cost for the VEC servers running a computation task on a unit resource.

Based on the utility functions of the vehicles and the service provider while considering the IR and IC constraints, the contract theoretic optimization problem of the computation offloading can be formulated as follows,

$$\begin{array}{ll} \max\limits_{\{p_n, q_n\}} & U'_{SP} = \sum\limits_{n=1}^{N} \gamma_n (p_n - c q_n) \\ s.t. & \mathrm{C1}: G(d_i, q_i) - p_i \geq 0, i \in \mathcal{N} \\ & \mathrm{C2}: G(d_i, q_i) - p_i \geq G(d_i, q_j) - p_j \\ & \qquad i \neq j, \quad i, j \in \mathcal{N} \\ & \mathrm{C3}: 0 \leq q_i \leq d_i, \quad i \in \mathcal{N} \end{array}. \tag{4}$$

It is noteworthy that, compared to (3), Q is omitted in (4), as this parameter does not affect the design of the contracts.

4.2 Optimal Contract Design

As there are $N(N+3)/2$ constraints, it may be complex to solve (4), especially for a large N. Thus, to solve (4) more efficiently, some constraints should be removed [46]. In this subsection, we first present the steps to simplify the optimization problem. Then, we propose an efficient algorithm to get the optimal contract solutions.

Lemma 1 *Monotonicity: Both the computation resources q_i and the offloading payment p_i of the feasible contract $\{q_i, p_i\}$ designed for type i vehicles are monotonically increasing in terms of i, $i \in \mathcal{N}$, i.e., for contracts $\{q_i, p_i\}$ and $\{q_j, p_j\}$, we have $q_i \geq q_j$ and $p_i \geq p_j$, if and only if $i > j$, $i, j \in \mathcal{N}$.*

Proof According to the IC constraints of type i and type j vehicles, where $i \neq j$ and $i, j \in \mathcal{N}$, we have

$$G(d_i, q_i) - p_i \geq G(d_i, q_j) - p_j, \tag{5}$$

and

$$G(d_j, q_j) - p_j \geq G(d_j, q_i) - p_i. \tag{6}$$

Adding (5) and (6), we can get $G(d_i, q_i) + G(d_j, q_j) \geq G(d_i, q_j) + G(d_j, q_i)$. By substituting $G(\cdot, \cdot)$, the inequation is equally changed to $(d_i - d_j)\ln(\frac{\alpha q_i + \beta}{\alpha q_j + \beta}) \geq 0$. Given $i > j$, according to the vehicle type definition, we have $d_i > d_j$, which implies that between the feasible contracts, $q_i \geq q_j$ if $i \geq j$. Next we will prove that $q_i \geq q_j$ should hold whenever $p_i \geq p_j$. Given the condition $p_i \geq p_j$, according to (5), we have $\lambda \frac{d_i}{d_0} \ln(\alpha q_i + \beta) + (1-\lambda) e_0 q_i - p_i \geq \lambda \frac{d_i}{d_0} \ln(\alpha q_j + \beta) + (1 - \lambda) e_0 q_j - p_j$. By rearranging the inequation, we can get $\lambda \frac{d_i}{d_0} \ln(\frac{\alpha q_i + \beta}{\alpha q_j + \beta}) + (1 - \lambda) e_0 (q_i - q_j) \geq p_i - p_j \geq 0$. Due to $0 < \lambda < 1$, we can get $q_i \geq q_j$. Similarly, the inequality $p_i \geq p_j$ can be proved under the condition $q_i \geq q_j$.

Lemma 2 *IR Constraints Reduction: If the IR constraint of type 1 vehicles is satisfied, then other IR constraints of type i, $1 < i \leq N$, do automatically hold [45].*

Proof Based on the IC constraints, we can get $G(d_i, q_i) - p_i > G(d_i, q_1) - p_1$. Furthermore, according to the definition of the vehicle types, we have $d_i > d_1$, where $1 < i \leq N$. As $G(d_i, q_i)$ is a monotonically increasing function in terms of d_i, we get $G(d_i, q_1) - p_1 > G(d_1, q_1) - p_1$. The inequality indicates that the IR constraint of type i vehicles, $1 < i \leq N$, is automatically satisfied whenever IR constraint of type 1 vehicles holds.

Definition 1 Local Downward Incentive Constraint (LDIC) and Downward Incentive Constraint (DIC): Considering two adjacent types, namely type i and type $(i - 1)$, the IC constraint of the contracts between these two types is defined as LDIC, which can be formally presented as

$$U_v^i(q_i, p_i) \geq U_v^i(q_{i-1}, p_{i-1}), \quad i \in \{2, 3, ..., N\}. \tag{7}$$

By extending the definition of LDIC to the IC constraints between type i and type j, where $j \in \{1, ..., i - 1\}$, we have the DICs, which are shown as

$$U_v^i(q_i, p_i) \geq U_v^i(q_j, p_j), \quad i \in \{2, 3, ..., N\}, j \in \{1, ..., i - 1\}. \tag{8}$$

Definition 2 Upward Incentive Constraint (UIC) and Local Upward Incentive Constraint (LUIC): Similar to **Definition** 1, the definition of UIC and LUIC are formally given as $U_v^i(q_i, p_i) \geq U_v^i(q_j, p_j)$, where $i \in \{1, 2, ..., N - 1\}, j \in \{i+1, ..., N\}$, and $U_v^i(q_i, p_i) \geq U_v^i(q_{i+1}, p_{i+1})$, where $i \in \{1, 2, ..., N-1\}$, respectively.

Lemma 3 *Given the LDICs hold, all DICs do automatically hold and can be reduced. Similarly, under the condition that LUIC holds, all the UICs can be removed.*

Proof According to the definition of $G(\cdot, \cdot)$, it is easy to get $G(d_{i+1}, q_i) - G(d_{i+1}, q_{i-1}) \geq G(d_i, q_i) - G(d_i, q_{i-1})$. Given that LDIC holds, according to (7), we have $G(d_i, q_i) - G(d_i, q_{i-1}) \geq p_i - p_{i-1}$. From these two inequations, we get

$$G(d_{i+1}, q_i) - G(d_{i+1}, q_{i-1}) \geq p_i - p_{i-1}. \tag{9}$$

Based on (9) and the LDIC definition between type $(i+1)$ and type i contracts, we have

$$\begin{aligned} G(d_{i+1}, q_{i+1}) - p_{i+1} &\geq G(d_{i+1}, q_i) - p_i \\ &\geq G(d_{i+1}, q_{i-1}) - p_{i-1}. \end{aligned} \tag{10}$$

The inequation (10) can be extended to prove that all the DICs do hold, i.e.,

$$\begin{aligned} G(d_{i+1}, q_{i+1}) - p_{i+1} &\geq G(d_{i+1}, q_{i-1}) - p_{i-1} \geq ... \\ &\geq G(d_{i+1}, q_1) - p_1, \quad i \in \{1, 2, ..., N-1\}. \end{aligned} \tag{11}$$

Hence, we come to the conclusion that all the DICs hold and can be reduced. The feature that all the UICs hold can be proved in a similar way.

Lemma 4 *All the LDICs are binding at the optimal contracts obtained from problem* (4).

Proof For the contract designed for type i vehicles, an LDIC is not binding, when $G(d_i, q_i) - p_i > G(d_i, q_{i-1}) - p_{i-1}$. Under this condition, the service provider can adapt the contract by raising all p_j $(j \geq i)$ to make the LDIC binding. This method can improve the maximum utility of the provider while not affecting the LDICs for the contracts of the other types of vehicles.

The fact that all the LDICs are binding for the optimal contracts, together with the monotonicity proved in Lemma 1, leads all the LUICs to be satisfied [34]. Thus, we can rewrite optimization problem (4) as

$$\begin{aligned} &\max_{\{p_n, q_n\}} U'_{SP} = \sum_{n=1}^{N} \gamma_n (p_n - c q_n) \\ &s.t. \quad \mathrm{C1}: G(d_1, q_1) - p_1 = 0 \\ &\qquad\ \ \mathrm{C2}: G(d_i, q_i) - p_i = G(d_i, q_{i-1}) - p_{i-1}, 1 < i \leq N\,. \\ &\qquad\ \ \mathrm{C3}: 0 \leq q_1 \leq q_2 \leq ... \leq q_N \\ &\qquad\ \ \mathrm{C4}: 0 \leq q_i \leq d_i, \quad i \in \mathcal{N} \\ &\qquad\ \ \mathrm{C5}: p_i \geq 0, \quad i \in \mathcal{N} \end{aligned} \tag{12}$$

Let $\Delta_k = G(d_k, q_k) - G(d_k, q_{k-1})$, where $1 < k \leq N$ and $\Delta_1 = 0$. Combining constraints C1 and C2 in (12), p_i can be expressed as

$$p_n = G(d_1, q_1) + \sum_{k=1}^{n} \Delta_k, \quad n \in \mathcal{N}. \tag{13}$$

Now replacing p_n in (12) with (13), the objective function can takes the form

$$\begin{aligned}
\max_{\{q_n\}} U'_{SP} &= \sum_{n=1}^{N} \gamma_n (G(d_1, q_1) + \sum_{k=1}^{n} \Delta_k - cq_n) \\
&= G(d_1, q_1) \sum_{n=1}^{N} \gamma_n - G(d_2, q_1) \sum_{n=2}^{N} \gamma_n \\
&\quad + G(d_2, q_2) \sum_{n=2}^{N} \gamma_n - G(d_3, q_2) \sum_{n=3}^{N} \gamma_n + ... \\
&\quad + G(d_{N-1}, q_{N-1}) \sum_{n=N-1}^{N} \gamma_n - G(d_N, q_{N-1}) \gamma_N \\
&\quad + G(d_N, q_N) \gamma_N + c \sum_{n=1}^{N} \gamma_n q_n
\end{aligned} . \tag{14}$$

Then (12) can be rewritten as follows, where the value of the objective function is only determined by the variable set $\{q_n\}$.

$$\begin{aligned}
\max_{\{q_n\}} U'_{SP} &= \sum_{n=1}^{N} \{ G(d_n, q_n) \sum_{i=n}^{N} \gamma_i \\
&\qquad - G(d_{n+1}, q_n) \sum_{j=n+1}^{N} \gamma_j - c\gamma_n q_n \} . \\
s.t. \quad & \mathrm{C1}: 0 \le q_1 \le q_2 \le ... \le q_N \\
& \mathrm{C2}: 0 \le q_i \le d_i, \quad i \in \mathcal{N}
\end{aligned} \tag{15}$$

Let $S_n = G(d_n, q_n) \sum_{i=n}^{N} \gamma_i - G(d_{n+1}, q_n) \sum_{j=n+1}^{N} \gamma_j - c\gamma_n q_n$. The objective function of (15) can be presented as $U'_{SP} = \sum_{n=1}^{N} S_n$. As S_n is independent from S_i, $n \ne i$, and S_n is only related to the amount of computation resources q_n, the optimal $\{q_n^*\}$ that maximizes U'_{SP} can be obtained separately by letting $q_n^* = \arg\max_{q_n} S_n$.

Lemma 5 *If $(d_{n+1} - d_n)/d_n > \gamma_n / \sum_{i=n+1}^{N} \gamma_i$, S_n is a concave function in terms of q_n, $n \in \mathcal{N}$.*

Proof We have $\partial^2 S_n / \partial q_n^2 = ((d_{n+1} - d_n) \sum_{i=n+1}^{N} \gamma_i - d_n \gamma_n) \alpha^2 \lambda / (\alpha q_n + \beta)^2 d_0$. If $(d_{n+1} - d_n)/d_n > \gamma_n / \sum_{i=n+1}^{N} \gamma_i$, we can get $\partial^2 S_n / \partial q_n^2 > 0$, which indicates that S_n is a concave function.

According to Fermat's theorem, q_n^* can be derived by solving $\partial S_n / \partial q_n |_{q_n = q_n^*} = 0$. Considering constraint C2 in (15), if the obtained q_n^* is a negative number or exceeds d_n, q_n^* should be set as 0 or d_n. Here $q_n^* = 0$ means that the contract is set to $\{0, \mathtt{Na}\}$. According to this contract, there is no computation offloading between type n vehicles and the VEC servers.

Besides constraint C2, the obtained optimal q_n^* should satisfy constraint C1 of (15). As each q_n^* is derived separately from the corresponding S_n, there may exist some sub-sequences not following the increasing order, which is described in constraint C1. Noting that when $\{Sn, n \in \mathcal{N}\}$ are concave functions, the

Algorithm 1 The substitution algorithm for the infeasible sub-sequences.

Initialization: Let $q_n^* = \arg\max_{q_n} S_n$, $n \in \mathcal{N}$.

1: **while** The set $\{q_n^*\}$ is not in the increasing order, **do**

2: In the set $\{q_n^*\}$, search for the infeasible sub-sequence $\{q_i^*, q_{i+1}^*, ..., q_j^*\}$, where $q_i^* \geq q_{i+1}^* \geq ... \geq q_j^*$, $1 \leq i < j \leq N$;

3: Set $q_k^* = \arg\max_{\{q\}} \sum_{x=i}^{j} S_x(q)$, $k \in \{i, i+1, ..., j\}$;

4: **end while**

5: **return** The feasible set $\{q_n^*\}$, $n \in \mathcal{N}$.

problem of these infeasible sub-sequences can be solved by an iterative substitution algorithm, which is presented as Algorithm 1 [32].

Based on the derived feasible optimal computation resources $\{q_n^*\}$, we can get the corresponding payments $\{p_n^*\}$ through (13). Thus, we can obtain the optimal contract set $\{q_n^*, p_n^*\}$ under the condition that $\{S_n\}$ are concave functions. If the concave condition is not satisfied, we can first solve the optimization problem (12) without constraint C3 by using Lagrange multiplier. Then, we can check whether the solution to this relaxed problem satisfies constraint C3.

5 Computing Resource Management for Task Offloading

Considering the different delay tolerances of the vehicular computation tasks and the mobilities of the vehicles, there may exist different set of optional VEC servers, which can be chosen as the offloading target by each type of vehicles. Furthermore, various locations and computing capabilities of the VEC servers make them heterogeneous available cloud resources to different types of vehicles. To ensure efficient utilization of the VEC resources, in this section , based on the obtained optimal offloading contracts, we further propose efficient VEC offloading schemes that decide which VEC servers are chosen by each type of vehicles and how much computing resources are allocated to execute the vehicular applications.

5.1 Characteristics of various offloading types

In a practical scenario, the sum of the task offloading demands from the arriving vehicles is a variable. The resource requirement of the demands may exceed the total computation resource of the MEC servers. Thus, a practical implementation of the contract-based offloading scheme is required.

Theorem 1 *According to the optimal contracts $\{q_n^*, p_n^*\}$, the revenue gained by the service provider from offering a unit resource to serve a lower type vehicle is more than utility gained from a higher type vehicle.*

Proof Let $u(n)$ denote the revenue of the service provider, which is gained from offering a unit resource to offload the task of a type n vehicle. Here, we have $u(n) = (p_n^* - cq_n^*)/q_n^*$. Thus, we get the difference between $u(n+1)$ and $u(n)$ as

$$D = u(n+1) - u(n) = p_{n+1}^*/q_{n+1}^* - p_n^*/q_n^*. \tag{16}$$

Using (13), we have $q_n^* p_{n+1}^* - q_{n+1}^* p_n^* = q_n^*(G(d_1, q_1^*) + \sum_{k=1}^{n+1} \Delta_k) - q_{n+1}^*(G(d_1, q_1^*) + \sum_{k=1}^{n} \Delta_k) = (q_n^* - q_{n+1}^*)(G(d_1, q_1^*) + \sum_{k=1}^{n} \Delta_k) - q_n^* \Delta_{n+1}$. Due to the monotonicity proved in Lemma 1, we have $q_n^* < q_{n+1}^*$. Thus, we get $\Delta_k = G(d_k, q_k^*) - G(d_k, q_{k-1}^*) = \lambda \frac{d_k}{d_0} \ln(\frac{\alpha q_k^* + \beta}{\alpha q_{k-1}^* + \beta}) + (1-\lambda) e_0 (q_k^* - q_{k-1}^*) > 0, k = \{1, 2, ..., n, n+1\}$. Then, we can find $q_n^* p_{n+1}^* < q_{n+1}^* p_n^*$, and come to the conclusion that $u(n+1) < u(n)$.

Theorem 1 indicates that the vehicles of lower type should be served with a higher priority, so as to make the limited resource more profitable.

Due to their mobility, the vehicles may reach different RSUs at different times as they move along the road. Since both the vehicles' traveling and the computing execution cost time, there exits a VEC server for each type of vehicles, which is the last server on the road that the vehicles can offload tasks to under the delay constraints of the vehicular applications. We denote the id of the last VEC server for type n vehicles as m_n^{max}. The definition of m_n^{max} is given as

$$m_n^{\max} = \arg\max_k \left(\sum_{m=1}^{k} R_m/h + d_n/q_n^* \le t_n^{\max} \right), \quad n \in \mathcal{N}. \tag{17}$$

Theorem 2 *For the vehicles traveling in the same direction, under the concave condition stated in Lemma 5, the last VEC server of the higher type vehicles is farther than that of the lower type ones.*

Proof According to Lemma 5, S_n is a concave function, and the optimal amount of the computation resources provided for a type n vehicle can be obtained by solving equation $\partial S_n / \partial q_n |_{q_n = q_n^*} = 0$. Substituting $G(d_n, q_n)$, we have $\alpha\lambda(d_n \gamma_n + \sum_{j=n+1}^{N} \gamma_j (d_n - d_{n+1}))/d_0 \gamma_n (c + (1-\lambda) e_0) = \alpha q_n^* + \beta$. Let $Q = \beta d_0 (c + (1-\lambda) e_0)/\alpha$. Thus, we get $q_n^* = \beta(\lambda(d_n + \sum_{j=n+1}^{N} \gamma_j (d_n - d_{n+1})/\gamma_n) - Q)/\alpha Q$. Given the optimal VEC computation resources for each type of vehicles, the difference between the VEC execution time of a type n task and a type $n+1$ task can be written as $\Delta_t = d_n/q_n^* - d_{n+1}/q_{n+1}^*$. Here, we have $\Delta_t' = q_n^*/d_n - q_{n+1}^*/d_{n+1} = \frac{\beta}{\alpha Q} \{ \lambda \gamma_{n+1} (1 - d_{n+1}/d_n)/\gamma_n - \lambda \sum_{j=n+2}^{N} \gamma_j (1 - d_{n+2}/d_{n+1})/\gamma_{n+1} + \lambda \sum_{j=n+2}^{N} \gamma_j (1 - d_{n+1}/d_n)/\gamma_n + Q/d_{n+1} - Q/d_n \}$. Since $d_{n+1} > d_n$, we have $\lambda \gamma_{n+1}(1 - d_{n+1}/d_n)/\gamma_n < 0$ and $Q/d_{n+1} - Q/d_n < 0$. Then, we focus on the remain part of Δ_t, which is given as $\lambda \sum_{j=n+2}^{N} \gamma_j ((1 - d_{n+1}/d_n)/\gamma_n - (1 - d_{n+2}/d_{n+1})/\gamma_{n+1})$. According to the concave condition stated in Lemma 5, we find $(d_{n+1} - d_n)/d_n > (d_{n+2} - d_{n+1})\gamma_n / d_{n+1}\gamma_{n+1}$. Thus, the remain part is negative. Since $\frac{\beta}{\alpha Q} > 0$, we get $\Delta_t' < 0$ and $\Delta_t > 0$, which means that the VEC execution time of the higher type vehicles is less than that of the lower type ones. Thus, under the delay constraints of vehicular applications, the maximum available travel time for lower type vehicles is shorter than that of the higher type ones. Thereafter, the last VEC server of the lower type vehicles is closer to their starting point.

5.2 VEC Server Selection and Computing Resource Allocation Schemes

The delay in the VEC offloading process is the sum of two components: the vehicle travel time and the computation execution time. Under the offloading delay tolerance of each type of applications, the tolerance in the acceptable delay due to the travel can be higher because of the savings in computation time. Thus, the range of the available VEC servers for vehicles can be extended by providing the vehicles with additional computing resources.

Lemma 6 *For type n vehicles ($n \in \mathcal{N}$), with the additional computing resource $\Delta_{r,n}$ apart from the optimal contract resource q_n^*, they can choose to offload to a VEC server with index no more than $m_n^{\max'}$.*

Proof Due to the rationality of the VEC service provider, the provider's revenue should be nonnegative with the extra resource allocation, i.e., $p_n^* - c(q_n^* + \Delta_{r,n}) \geq 0$. Considering the total time cost of the VEC offloading process with the additional resource, we have $d_n/(q_n^* + \Delta_{r,n}) + \sum_{j=1}^{m_n^{\max'}} R_j/h \leq t_n^{\max}$, and here $m_n^{\max'}$ is the extended last VEC server that type n vehicles can offload to. Combining the above two inequalities, we get $m_n^{\max'} = \arg\max_k \left(\sum_{m=1}^{k} R_m/h + d_n c/p_n^* \leq t_n^{\max} \right)$.

Although a lower type vehicle brings more revenue to the provider than a higher type one, which is stated in Theorem 1, a unit computing resource allocated to lower type vehicles may not always generate more revenue than being allocated to higher ones with the providing of additional resources. However, there exists relationships of the revenues gained from the vehicles of different types under a given condition.

Lemma 7 *Given the allocated additional computing resources $\Delta_{r,n}$, if the revenue $p_n^*/(q_n^* + \Delta_{r,n})$ gained from providing a unit resource to type n vehicles is higher than that gained from a unit resource providing to type m vehicles ($m > n$, $m, n \in \mathcal{N}$), serving type n vehicles is more profitable than serving type m vehicles with additional resources.*

Proof Given $p_n^*/(q_n^* + \Delta_{r,n}) > p_m^*/q_m^*$, since $\Delta_{r,m} > 0$, we have $p_m^*/q_m^* > p_m^*/(q_m^* + \Delta_{r,m})$. Thus, there has $p_n^*/(q_n^* + \Delta_{r,n}) > p_m^*/(q_m^* + \Delta_{r,m})$.

With the additional allocated computing resources, the vehicles can offload their tasks to more distant VEC servers as long as the delay constraints of their tasks can be satisfied. As a result, more cloud resources can be utilized. Considering the limited computing capacity of each VEC server, to raise the

VEC provider's revenue, the computing resources should be allocated to more profitable vehicle. When several types of vehicles want to offload their tasks to the same VEC server, a collision occurs. Thus, a VEC computing resource management is required.

Here we propose a contract-based efficient VEC server selection and cloud computing resource allocation scheme for the vehicles. According to Theorem 1 and Theorem 2, the lower type vehicles are more profitable and their available VEC servers are nearer to the starting point of the traveling vehicles. Therefore, in the proposed scheme, the computing resource management begins from the VEC servers located nearest to the starting point, and the computing resources are first allocated to the lowest type vehicles. In the system model, the road is bidirectional, and there are vehicles traveling in opposite directions. Thus, the starting points of the road from both sides need to be considered. When there is a competition for the same VEC resources between different types of vehicles, a revenue comparison is adopted to decide the resource subscribers. The details of the proposed VEC resource management scheme are described in Algorithm 2.

Algorithm 2 The contract-based VEC resource management algorithm.

Initialization: The arriving vehicles Q_1 and Q_2; The computation task $T_n = \{d_n, t_n^{max}\}$, $n \in \mathcal{N}$; The available computation resource $\{b_m\}$ for MEC server m, $m \in \mathcal{M}$.

Derive the optimal contract set $\{q_n^*, p_n^*\}$ ($n \in \mathcal{N}$) following the steps described in Section IV;
Obtain the maximum id $m_n^{max'}$ ($n \in \mathcal{N}$) with the additional allocated VEC resources according to Lemma 6;
For $j = 1 : 1 : \lceil M/2 \rceil$ **do**
  Loop
    Based on $m_n^{max'}$ ($n \in \mathcal{N}$), obtain the set of candidate vehicle types for offloading tasks to server j (server $M - j + 1$), which is denoted as Set_j (Set_{M-j+1}).
    Compute the revenues gained from providing a unit VEC resource to the types of vehicles belonging to the sets Set_j (Set_{M-j+1}).
    Searching from the lowest type vehicles in set Set_j (Set_{M-j+1}), find type n_1^* (n_2^*), which brings the highest revenues to server j (server $M - j + 1$).
    Allocate VEC resource to type n_1^* (n_2^*) vehicles from server j (server $M - j + 1$).
    Update the remain computation resources b_j of server j (b_{M-j+1} of server $M - j + 1$).
    Remove the type of vehicles, whose offloading requirements have been fully satisfied, from Set_j (Set_{M-j+1}).
    If $Set_j == \emptyset \mid\mid b_j == \emptyset$ ($Set_{M-j+1} == \emptyset \mid\mid b_{M-j+1} == \emptyset$) **then**
      End loop;
    End if
  End loop
End For

6 Numerical Results

In this section, we evaluate the proposed contract-based VEC computing resource management schemes. We consider a scenario where $M = 6$ RSUs ran-

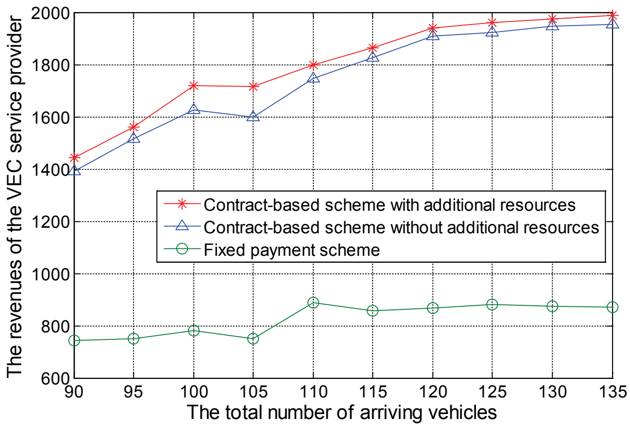

Fig. 2 The revenues of the service provider with different schemes.

domly located in a bidirectional road. The computing resources of these VEC servers are randomly distributed within the range of (100, 500) units [47]. For each RSU, the cost for computing for a unit resource is set $c = 0.3$. There are 100 arriving vehicles on the road, and they run at the speed 100 km/hr. The vehicles choose the directions of travel with equal probabilities. The arriving vehicles are classified into $N = 5$ types in terms of their computation tasks with the probabilities $\gamma = \{0.17, 0.23, 0.28, 0.18, 0.16\}$. For each type of vehicular computation tasks, the resource requirement is $d = \{12, 13, 15, 18, 21\}$, respectively.

Figure 2 evaluates the performance of the proposed contract-based computing resource management scheme. In the resource allocation process of this scheme, both the delay tolerance and the priorities of different types of vehicles are considered. In addition, the additional VEC resources providing strategies are adopted to further improve cloud utilization. We compare the performance of our proposed scheme with two other schemes. One is the contract-based scheme without additional resource allocation. The other is the non-contract scheme adopting fixed offloading payment of the vehicles. It can be seen that both the contract-based schemes yield higher revenues to the VEC service provider than the fixed payment one. The reason is that in the contract theoretic approach, each contract is designed for the corresponding computation task type. Thus, the revenues gained from providing offloading service to the vehicles can be improved by making the LDICs binding as described in Lemma 4. Furthermore, we can find that the revenues obtained by our proposed scheme is higher compared to the contract-based scheme without additional resource. This can be explained as follows. Due to the vehicles' travel time cost, the computing resources of the VEC servers far away from the road starting points can not be used by the vehicles within the delay tolerance of the vehicular applications. However, through the adoption of our proposed additional resource allocation, the range of the available VEC servers for the

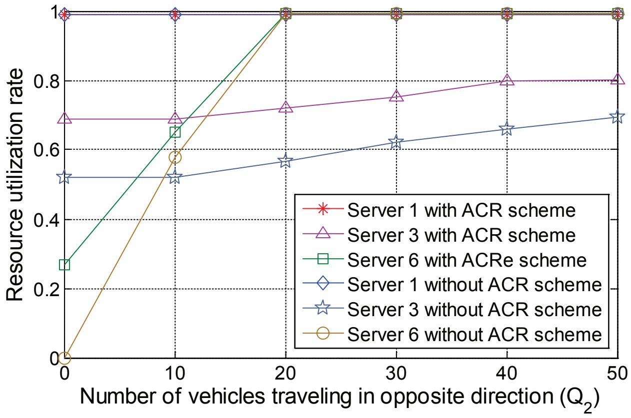

Fig. 3 The resource utilization rate of the VEC servers with different schemes.

vehicles has been extended. Thus, more VEC resources can be utilized, which brings higher revenue to the service provider.

Figure 3 shows the computing resource utilization rates of different VEC servers adopting various resource allocation schemes. Here, the vehicles traveling from server 1 to server 6 is denoted as set $\mathcal{Q}_1$ with the vehicle number $Q_1 = 50$. Due to its proximity to the starting point of vehicles $\mathcal{Q}_1$, server 1 can be chosen as the available server by all types of vehicles. Thus, the resource utilization of server 1 is always close to 1. In contrast, server 6 is the farthest one from $\mathcal{Q}_1$, but the nearest to the starting point of vehicles $\mathcal{Q}_2$, which travel in the opposite direction of $\mathcal{Q}_1$. When the number of vehicles $\mathcal{Q}_2$ is small, the resource utilization of server 6 is low, since the long travel delay obstructs vehicles $\mathcal{Q}_1$ to offload tasks to server 6 under delay constraints. In this case, by implementing our proposed Additional Computing Resource (ACR) scheme, server 6 is turned to be an available offloading target of $\mathcal{Q}_1$ with the extended traveling delay tolerance. As a result, the resource utilization of server 6 improves. Server 3 is located in the middle of the road. Thus, a large part of the low type vehicles can not offload tasks to this server by adopting the scheme without ACR. However, as can be seen from Figure 3, our proposed ACR scheme greatly increases the utilization rate of server 3 with an average rate of 24.9%.

Figure 4 indicates the revenues of the VEC service provider gained from executing an offloading task to a unit resource. We can see that by utilizing a unit computing resource, the provider gains higher profit from offloading tasks of the lower type vehicles. This result corroborates Theorem 1. In addition, the revenue gained from each type of vehicles first increases and then decreases. This can be explained as follows. According to Theorem 1, the revenues gained from running a task T_i on a unit resource is defined as $u(i) = p_i/q_i - c$. In the contract-based offloading scheme, the provider raises price p_i to gain higher profit with the growth of cost c. To cope with the increasing price, vehicles

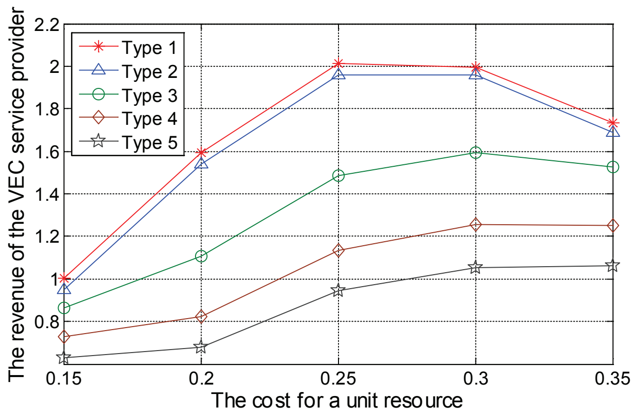

Fig. 4 The revenues of the service provider gained from an unit computation resource.

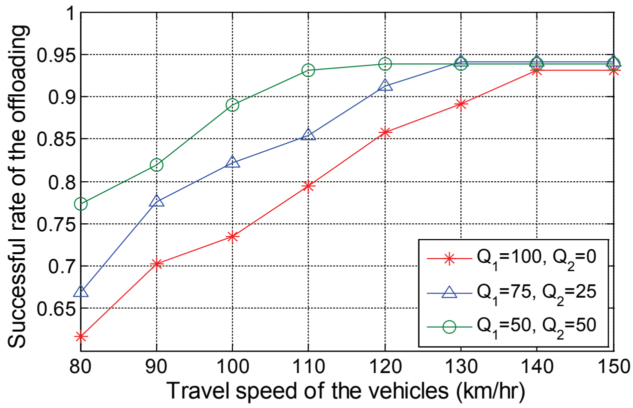

Fig. 5 The successful rate of the offloading service in terms of the vehicles' travel speed.

will reduce the offloading resource consumption q_i. The rate of the demand reduction is higher at lower c.

Figure 5 shows successful rate of computation offloading service for vehicular applications in the cases with different numbers of vehicles traveling in the opposite directions. It can be seen that the rates improve with increase in the speed of the vehicles, in all cases, until the rates reach their maximum thresholds. In addition, we can see that in the case where the distribution of the vehicles' travel directions is more dispersed, the maximum threshold is reached at a slower speed. With a higher speed, vehicles can access more VEC servers within their task delay constraints. Thus, they have more chances to offloading their computation task. When the speed is higher than the threshold, which enables the vehicles to offload tasks to the farthest servers, the successful rate will not increase with the speed acceleration. In the case where the vehicles are more dispersed, the average distance between the farthest VEC server and the starting points of the traveling vehicles is shorter. Thus, the speed threshold is lower in this case.

7 Conclusion

In this paper, we have presented a vehicular edge computing framework for studying computation offloading process. By implementing contract-theoretic approach, we design optimal computation offloading strategies for the cloud service provider in terms of computing resource allocation and service pricing. To further improve the utilization of the edge cloud servers while maximizing the revenues of the service provider, we propose an efficient cloud resource management scheme with offloading priority distinction and additional resource allocation. Numerical results demonstrate that our proposed schemes greatly improve the performance of the vehicular application offloading process.

References

1. Z. Qu, J. Keeney, S. Robitzsch, F. Zaman and X. Wang, Multilevel pattern mining architecture for automatic network monitoring in heterogeneous wireless communication networks, China Communications, 13(7):108-116 (2016)
2. Z. Zhou, Y. Wang, Q. M. J. Wu, C. Yang, X. Sun, Effective and efficient global context verification for image copy detection, IEEE Trans. Information Forensics and Security, 12(1): 48-63 (2017)
3. J. Li, X. Li, B. Yang and X. Sun, Segmentation-based image copy-move forgery detection scheme, IEEE Trans. Information Forensics and Security, 10(3): 507-518 (2015)
4. X. Chen, S. Chen and Y. Wu, Coverless information hiding method based on the chinese character encoding, Journal of Internet Technology, 18(2): 91-98 (2017)
5. Z. Fu, K. Ren, J. Shu, X. Sun and F. Huang, Enabling personalized search over encrypted outsourced data with efficiency improvement, IEEE Trans. Parallel and Distributed Systems, 27(9): 2546-2559 (2016)
6. Z. Fu, X. Wu, C. Guan, X. Sun and K. Ren, Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement, IEEE Trans. Information Forensics and Security, 11(12): 2706-2716 (2016)
7. K. Wang, Y. Shao, L. Shu, C. Zhu and Y. Zhang, Mobile big data fault-tolerant processing for ehealth networks, IEEE Network, 30(1): 36-42 (2016)
8. K. Wang, Y. Wang, X. Hu, Y. Sun, D. Deng, A. Vinel, Y. Zhang, Wireless big data computing in smart grid, IEEE Wireless Communications, 24(2): 58-64 (2017)
9. Y. Kong, M. Zhang and D. Ye, A belief propagation-based method for task allocation in open and dynamic cloud environments, Knowledge-based Systems, 115: 123-132 (2017)
10. J. Shen, J. Shen, X. Chen, X. Huang and W. Susilo, An efficient public auditing protocol with novel dynamic structure for cloud data, IEEE Trans. Information Forensics and Security, accepted
11. Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu and N. Linge, A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment, Security and Communication Networks, 9(17): 4002-4012 (2016)
12. X. Liu, Y. Li and H. Chen, Wireless resource scheduling based on backoff for multiuser multiservice mobile cloud computing, IEEE Trans. Vehicular Technology, 65(11): 9247-9259 (2016)
13. J. Shen, S. Chang, J. Shen, Q. Liu and X. Sun, A lightweight multi-layer authentication protocol for wireless body area networks, Future Generation Computer Systems (2016)

14. B. Gu and V. S. Sheng, A robust regularization path algorithm for ν-support vector classification, IEEE Trans. Neural Networks and Learning Systems, accepted
15. K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, G. Zhang, S. Maharjan and Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogenous networks, IEEE Access, 4: 5896-5907 (2016)
16. R. Yu, X. Huang, J. Kang, J. Ding, S. Maharjan, S. Gjessing and Y. Zhang, Cooperative resource management in cloud-enabled vehicular networks, IEEE Trans. Industrial Electronics, 62(12): 7938-7951 (2015)
17. L. Werin and H. Wijkander, Contract Economics, Blackwell (1992)
18. X. Huang, R. Yu, J. Kang, N. Wang, S. Maharjan and Y. Zhang, Software defined networking with pseudonym systems for secure vehicular clouds, IEEE Access, 4:3522-3534 (2016)
19. Z. Xia, X. Wang, X. Sun and Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel and Distributed Systems, 27(2): 340-352 (2016)
20. Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun and K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing,IEEE Trans. Information Forensics and Security, 11(11): 2594-2608 (2016)
21. K. Zheng, H. Meng, P. Chatzimisios, L. Lei and X. Shen, An SMDP-based resource allocation in vehicular cloud computing systems, IEEE Trans. Industrial Electronics, 62(12): 7920-7928 (2015)
22. D. Kwak, R. Liu, D. Kim, B. Nath and L. Iftode, Seeing is believing: sharing real-time visual traffic information via vehicular clouds, IEEE Access, 4: 3617-3631 (2016)
23. Z. Pan, P. Jin, J. Lei, Y. Zhang, X. Sun, and S. Kwong, Fast Reference Frame Selection Based on Content Similarity for Low Complexity HEVC Encoder, Journal of Visual Communication and Image Representation, 40(Part B): 516-524 (2016)
24. Y. Chen, C. Hao, W. Wu and E. Wu, Robust dense reconstruction by range merging based on confidence estimation, Science China Information Sciences, 59(9): 1-11 (2016)
25. C. Yuan, Z. Xia and X. Sun, Coverless image steganography based on SIFT and BOF, Journal of Internet Technology, 18(2): 209-216 (2017)
26. L. Tong, Y. Li and W. Gao, A hierarchical edge cloud architecture for mobile computing, in Proc. IEEE International Conference on Computer Communications (INFOCOM), 1-9 (2016)
27. Y. Wang, M. Sheng, X. Wang, L. Wang and J. Li, Mobile-edge computing: partial computation offloading using dynamic voltage scaling, IEEE Trans. Communications, 64(10): 4268-4282 (2016)
28. X. Sun and N. Ansari, PRIMAL: profit maximization avatar placement for mobile edge computing, in Proc. IEEE International Conference on Communications (ICC), 1-6 (2016).
29. M. Sapienza, G. L. Torre, G. Leombruno, E. Guardo, M. Cavallo and O. Tomarchio, Solving critical events through mobile edge computing: an approach for smart cities, in Proc. IEEE International Conference on Smart Computing (SMARTCOMP), 1-5 (2016)
30. N. Kumar, S. Zeadally and J. J. P. C. Rodrigues, Vehicular delay-tolerant networks for smart grid data management using mobile edge computing,IEEE Communications Magazine, 54(10): 60-66 (2016)
31. X. Chen, L. Jiao, W. Li and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Networking*, vol. 24, no. 5, pp. 2795 -2808, Oct. 2016.
32. L. Gao, X. Wang, Y. Xu, and Q. Zhang, Spectrum trading in cognitive radio networks: a contract-theoretic modeling approach,IEEE Journal on Selected Areas in Communications, 29(4): 843-855 (2011)
33. Y. Li, P. Wang, D. Niyato and Y. Zhang, Contract-theoretic modeling for content delivery in relay-based publish-subscribe networks, in IEEE International Conference on Communications (ICC), 2678-2683 (2014)
34. Y. Zhang, L. Song, W. Saad, Z. Dawy and Z. Han, Contract-based incentive mechanisms for device-to-device communications in cellular networks, IEEE Journal on Selected Areas in Communications, 33(10): 2144-2155 (2015)
35. K. Zhang, Y. Mao, S. Leng, S. Maharjan, Y. Zhang, A. Vinel and M. Jonsson, Incentive-driven energy trading in the smart grid, IEEE Access, 4: 1243-1257 (2016)

36. C. Yuan, X. Sun and R. LV, Fingerprint liveness detection based on multi-scale LPQ and PCA, China Communications, 13(7): 60-65 (2016)
37. Y. Zhang, R. Yu, W. Yao, S. Xie, Y. Xiao and M. Guizani, Home M2M networks: architectures, standards, and QoS improvement, IEEE Communications Magazine, 49(4): 44-52 (2011)
38. Z. Fu, F. Huang, X. Sun, A. V. Vasilakos and C. Yang, Enabling semantic search based on conceptual graphs over encrypted outsourced data, IEEE Trans. Services Computing, accepted
39. Z. Pan, J. Lei, Y. Zhang, X. Sun and S. Kwong, Fast motion estimation based on content property for low-complexity H.265/HEVC encoder, IEEE Trans. Broadcasting, 62(3): 675-684 (2016)
40. B. Gu, X. Sun and V. S. Sheng, Structural minimax probability machine, IEEE Trans. Neural Networks and Learning Systems, accepted
41. Z. Pan, Y. Zhang and S. Kwong, Efficient motion and disparity estimation optimization for low complexity multiview video coding,IEEE Trans. Broadcasting, 61(2): 166-176 (2015)
42. Z. Zhou, C. Yang, B. Chen, X. Sun, Q. Liu and Q. M. J. Wu, Effective and efficient image copy detection with resistance to arbitrary rotation, IEICE Trans. information and systems, E99-D(6): 1531-1540 (2016)
43. Z. Fu, F. Huang, K. Ren, J. Weng and C. Wang, Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data, IEEE Trans. Information Forensics and Security, 12(8): 1874-1884 (2017)
44. Q. Tian and S. Chen, Cross-heterogeneous-database age estimation through correlation representation learning, Neurocomputing, 238: 286-295 (2017)
45. P. Bolton and M. Dewatripont, Contract theory, MIT press, 31-64, (2005)
46. Y. Zhang, X. Sun and B. Wang, Efficient algorithm for K-barrier coverage based on integer linear programming, China Communications, 13(7): 16-23 (2016)
47. B. Gu, V. S. Sheng, K. Y. Tay, W. Romano and S. Li, Incremental support vector learning for ordinal regression, IEEE Trans. Neural Networks and Learning Systems, 26(7): 1403-1416 (2015)