

## Research Article

# An Incentive Mechanism for Computation Offloading in Satellite-Terrestrial Internet of Vehicles

Xingyu Zhang <sup>1</sup>, Heyang Zhang <sup>1</sup>, Sida Dai,<sup>2</sup> and Yang Liu <sup>2</sup>

<sup>1</sup>International School, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>2</sup>State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Yang Liu; [liu.yang@bupt.edu.cn](mailto:liu.yang@bupt.edu.cn)

Received 30 September 2021; Accepted 7 December 2021; Published 19 January 2022

Academic Editor: Chi Lin

Copyright © 2022 Xingyu Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of satellite-terrestrial technology and the popularity of the Internet of Vehicles, how to improve the efficiency of mobile cloud computing (MCC) has become the next concern. However, the resource of cloudlets is not sufficient to perform large-scale computation tasks, or some applications designed to run on vehicles have more efficiency executed on vehicles than executed on cloudlets. Additionally, it is still challenging for platforms to motivate mobile vehicle owners to join in the process while the existing mechanisms cannot provide all the desired properties in cloudlet scenarios. To this end, we design a satellite-terrestrial IoV based on an incentive mechanism for computation offloading (IMCO) in mobile edge computing to motivate vehicle owners to perform computation offloading tasks so as to offload certain kinds of tasks to the mobile vehicles. By optimizing the MCC model, we integrate auction theory into the mechanism to ensure individual rationality, budget balance, system efficiency, and truthfulness for both sellers and buyers. Through rigorous theoretical analysis, we prove that our mechanism can achieve computational efficiency under the condition that all algorithm outputs be computed in polynomial time. Both theoretical derivations and numerical calculations prove that all the desired properties of the mechanism hold.

## 1. Introduction

Along with advances in computing and communications technology, the Internet of Vehicles (IoV), which is gradually replacing conventional Vehicle Ad-hoc Networks, is becoming popular worldwide and creating a promising vision for transport systems. According to a recent report by Allied Market Research, the global IoV market is expected to exceed 200 billion by 2024. IoV is a distributed network [1] of huge commercial and research value that supports the use of data created by connected cars and vehicular ad-hoc networks (VANETs). Today, the design, implementation, and management of IoV technology rely heavily on modern cloud computing technologies, which play a central role in linking isolated vehicles into an organic network in which data are being processed primarily. However, considering limitations such as the size of the data and the stability of the channel, cloud computing struggles to process effi-

ciently and could cause fatal latency [2]. According to a McKinsey & Company estimate, connected cars create up to 25 GB of data per hour. With such huge amounts of data, we need to consider new computing technologies [3]. Mobile edge computing is an emerging paradigm in IoT systems [4] that offloads data processing to the mobile which localizes data processing and enabling services to be functional [5] even if connectivity between the vehicle and cloud is absent. In addition, some applications which are suitable for running on vehicles have more efficiency executed on powerful mobile vehicles than executed on cloudlet.

However, implementing mobile edge computing for computation offloading still faces several challenges [6]. First, requisitioning a viable and suitable mobile vehicle for computing takes up a user's personal resources, and it is difficult to convince users to volunteer their mobile vehicles without rewards. Therefore, we need to design a flexible and reasonable incentive mechanism to encourage users to

willingly provide their vehicles as computation resources [7]. Second, the complex physical environment in which cars travel [8], such as unstable signals and frequent congestion or crashes, places high demands on the stability and capacity of communications. Therefore, we design to adopt a satellite-terrestrial network [9] on IoV [8] to overcome the implementation difficulties. With seamless accessibility to overcome the coverage and distribution limitations of RSUs and BSs, it could provide macroscopic management to improve overall network efficiency and resource utilization at a low cost. Third, resource allocations [10] and computation offloading transactions require low time delay or there is no necessity to conduct it. Therefore, we design costumed algorithms during different stages which guarantee system and computation efficiency [11]. Lastly, the battery capacity is still a limitation of supporting energy consumption [12]. Luckily, with the popularity of fast charging technology [13], the charging efficiency of vehicles is greatly improved [14], since the development of low delay wireless charging technology [15] and the convenience of using wireless charging [16]. By using wireless charging technology [17], users can charge their mobile vehicles anytime and anywhere [18]. Therefore, with the development of fast charging technology and wireless charging technology, the problem of energy constraints can be relieved [19].

In order to overcome the above challenges, we propose an incentive mechanism for computation offloading (IMCO) that maximizes the motivation of vehicle owners to perform computation offloading tasks while guaranteeing the system stability and the privacy of participants. More specifically, we first investigate the auction theory and integrate it into Mobile Cloud Computing (MCC) scenarios for computing offloads, where the problem of resource allocation was redesigned according to demand. Then, we propose the sealed-bid auction model which achieves consistent asks of mobile vehicles and variational offer of cloudlets as well as many desirable properties such as individual rationality, budget balance, truthfulness, and computational efficiency. To further illustrate the mechanism of IMCO, we design three algorithms for different phases within an auction and further use an example to demonstrate the process in detail. By rigorous theoretical analysis, our proposed mechanism is proved to achieve the four proposed desirable properties and close-to-optimal performance. Finally, we carry out extensive simulations of our proposed mechanism by testing the performance on a dataset generated by distribution functions by analyzing six indexes of different aspects.

The main contributions of this paper are highlighted below.

- (1) We propose an incentive mechanism for computation offloading (IMCO) based on satellite-terrestrial IoV. The feasibility and the advantages of our design are also discussed
- (2) IMCO solves the problem that remote cloud server is not that sufficient to perform large scale computation tasks which will greatly encourage users to join

in with computation offloading even further helps to shape a good user environment and provides the basis for widespread commercialization

- (3) We apply the auction theory in economics to the scenarios in which computation tasks are offloaded to the vehicle owners. Since the existing auction algorithms cannot achieve all the properties when they are applied to computation offloading in MCC with cloudlets and vehicles as homogeneous items
- (4) We theoretically prove that our proposed mechanism is more suitable for real-life scenarios as it achieves truthfulness. Extensive simulations verify that all previous proposed desirable properties are satisfied

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes our system model and formulates problems. Section 4 designs the algorithm and the mechanism as well as carries out a theoretical analysis. Section 5 presents simulation and performance results of our proposed mechanism. Section 6 concludes the paper.

## 2. Related Work

In this part, the related work is divided into two groups: incentive mechanisms especially for mobile cloud computing and auction mechanisms used in economics.

Mobile edge computing is an emerging paradigm in the current IoT system. Based on the concepts of cloud computing and mobile computing, MCC was built relying on wireless networks and focusing on the construction and interaction of physical devices and virtual resources. So, there are a number of studies and researches about different applications in MCC, such as online translation service with MCC and big data storage strategy for MCC. Nevertheless, there is little research about the incentive mechanism. In the scenario, the auction must deal with items with complementary and replacement relationships.

As what we illustrated in Section 1, even if the auction theory is well developed and applied, the existing auction algorithms are not suitable to apply in computation offloading from cloudlets to mobile vehicles. Because of incompletely satisfying the desirable properties, Vickrey auction [20] has the properties of individually rational, budget-balanced. However, it cannot deal with cheating behavior; therefore, it is susceptible to market manipulation. McAfee in [21] focuses on the auction of nonidentical items. In other words, buyers do not lean to auction commodities. When the auction begins, every buyer makes their offer only once and each seller asks for a price. After that, the auctioneer ranks the offers in a nonascending way and asks in a nondescending order to find out the minimum possible offer and maximal possible ask; then, determining the winners in buyers according to their bids. That is, their bids are larger than or equal to the maximum possible ask. And also determining the winning sellers relies on the same rule. Next, the auctioneer generates two prices: clearing price and clearing

payment. Every buyer must submit the clearing price to the auctioneer. In the meantime, each seller gets rewards that are equal to clearing payment. Notwithstanding, McAfee cannot be applicable to the computation offloading scenario, since the mechanism aims at auction of homogeneous items. Even if it achieves all the desirable properties, in the scenario, each mobile vehicle will be rated differently from cloudlets rely on different factors, such as latency [6], quality of service, communication overhead, and computational capabilities.

Considering the homogeneous commodities, TASC [22] includes two phases: winner determination phase and pricing phase. In the first stage, the auctioneer filtrates inapposite sellers and buyers to determine the first-stage winners. In the second stage, the auctioneer implements the McAfee mechanism to determine the clearing price and payment. Although the scenario of TASC is close to mobile cloud computing with cloudlets, there are still some defects by applying TASC in the cloudlet scenario. Because of the distinct feature of cloudlets, TASC cannot support truthfulness for buyers even if it can guarantee truthfulness for sellers and other properties.

Our previous work proposes a computation offloading incentive mechanism [23]. Compared to [23], this paper discusses how to unload some types of applications to mobile vehicles by encouraging vehicle owners to perform computing unload tasks. By optimizing the MCC model, we integrate auction theory into the mechanism to ensure individual rationality, budget balance, system efficiency, and truthfulness for both sellers and buyers. IMCO maximizes the motivation of vehicle owners to perform computation offloading tasks. In addition, IMCO can assign more than one mobile vehicle for a single cloudlet in a temporary state.

### 3. System Model and Problem Formulation

In our system architecture shown in Figure 1, there are three main parts: the satellite-terrestrial networks (STNs), the vehicles, and the cloudlets. STNs are the integration of satellite networks and terrestrial networks [24], consisting of one or more satellite systems and land-based stations, which efficiently disseminate information through wireless channels [25]. In the architecture, computational tasks on cloudlets can be offloaded to vehicles within coverage [26], while vehicles are simultaneously free to choose to perform computational tasks from all cloudlets covered by their current locations via STNs.

There are lots of vehicles around one cloudlet, the performance of some vehicles may be too weak to carry out computation-intensive tasks. To lift efficiency, the cloudlet can use auction mechanisms to offload computation tasks to other high-performance vehicles so as to reduce the communication overhead and improve efficiency [27]. Each vehicle will be rated differently from cloudlets based on different factors, such as latency, quality of service, communication overhead, and computational capabilities [28]. On the other side, vehicle owners can be rewarded for providing computation resources and compensated for communication costs [29]. As in most real cases in the auction market, both the buyers and the sellers can get benefits from the

transactions, that is, car owners can get paid to be incentivized to share its resource [30]. When the computation offloading is applied in the mobile edge computing, every cloudlet bids when it needs extra computational performance, and it submits the offers to the auctioneer [31]. Then, the asks of vehicles are also submitted. The auctioneer, also known as a control center, will allocate  $m$  vehicles among  $n$  cloudlets according to the outcome of IMCO, consequently reducing the communication overhead and the latency so that the services will be affordable.

*3.1. Resource Allocation for Mobile Vehicles.* As the structure shown in Figure 1, there are lots of mobile vehicles around one cloudlet, and some performance of vehicles may be too weak to carry out some computation-incentive applications. The cloudlet can provide service to running applications for those vehicles. However, some tasks rely on particular hardware such as DSP, ISP, and NPU. The virtual machines may solve the problem by hardware virtualization. But the efficiency of virtualization is very low due to the structure of vehicles. The integration level of vehicles is much higher than cloudlet, so there are some applications using heterogeneous computing combining the advantages of CPU, GPU, NPU, ISP, and DSP. And lots of hardware are designed to process specific tasks. Hence, using virtualization is not a suitable solution. One cloudlet may serve several low-end mobile vehicles at one time; to lift efficiency, cloudlet can use auction mechanism to offload computation tasks to other high-performance mobile vehicles. The close range of high-performance mobile vehicles can be exploited to reduce the communication overhead and improve efficiency. Each mobile vehicle will be rated differently from cloudlets rely on different factors, such as latency, quality of service, communication overhead, and computational capabilities. Some cloudlet may need more powerful computation capabilities to achieve computation-intensive tasks, and others may prefer a low communication latency to respond to their clients more quickly.

On the other side, vehicle owners can be rewarded for providing computation resource and compensated for communication cost. As most real cases in the auction market, both the buyers and sellers can get benefits from the trading, that is, vehicle owners can get paid such as bitcoins to be incentivized to share its resource, and the tasks which are offloaded to the mobile vehicles should be done in given times. Especially, cloudlet must pay no less than the cost of the vehicle owner providing such computation resource.

In an area, the more computation offloading tasks, the higher the resource utilization of the mobile vehicles. In order to maximize the use of resources, the auctioneer should accurately allocate the mapping mobile vehicles' resources to the assigned tasks from cloudlets. Mobile vehicles only care about the cost of renting their resources. And the cloudlets' offers are adjustable. The performance of mobile vehicles varies from field to field and according to the auction model will be calculated to determine the right winning bidder.

*3.2. Auction Model.* The computation offloading is applied in the MCC scenario which is illustrated in Figure 1. Every

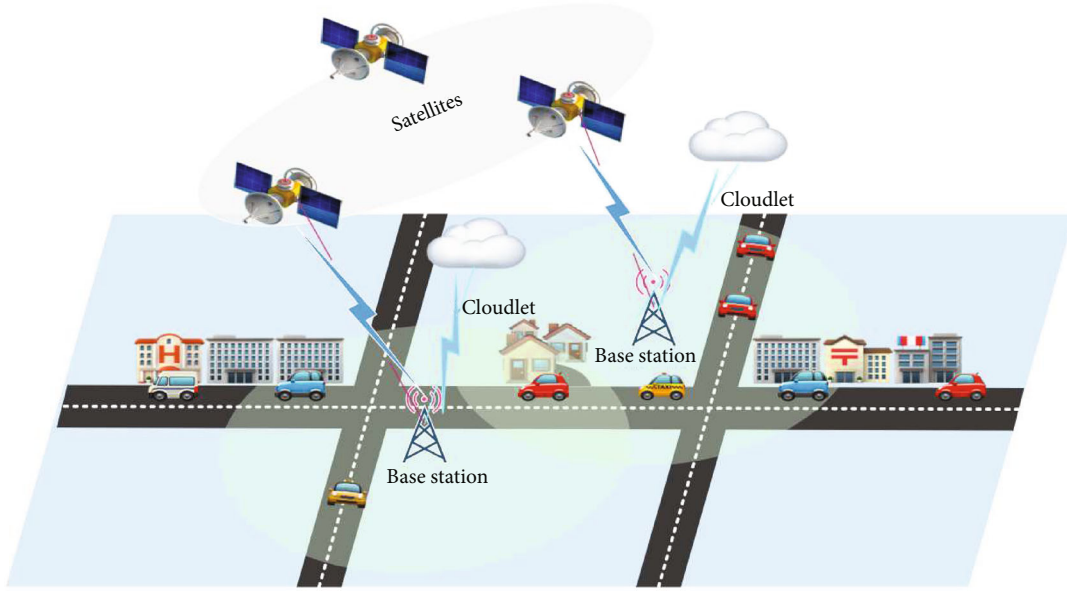


FIGURE 1: The system architecture of satellite-terrestrial IoV.

cloudlet bids when they need extra computational performance, and they submit their offers to the auctioneer. Then, the asks of mobile vehicles are also submitted. The auctioneer also known as the control center will allocate  $m$  mobile vehicles among  $n$  cloudlet according to IMCO, consequently reducing the communication overhead and the latency so that the services will be affordable.

To achieve truthfulness, the sealed-bid auction model is necessary. That is to say, each cloudlet (mobile vehicle) can upload its bid (ask) secretly to the auctioneer so that everyone's information including buyers' bids and sellers' asks is invisible to others. As Table 1 depicted:

- (i) For each cloudlet  $c_i \in C$ ,  $C = \{c_1, c_2, \dots, c_n\}$ .  $B_i = \{B_i^1, B_i^2, \dots, B_i^m\}$  is defined as bid vector, where  $B_i^j$  is the bid for mobile vehicle  $m_j \in M$ ,  $M = \{m_1, m_2, \dots, m_m\}$ . The matrix of bids including all the bid vectors of all cloudlets is denoted by  $B = \{B_1; B_2; \dots; B_m\}$ .
- (ii) For each mobile vehicle  $m_j \in M$ , its ask vector is defined as  $A = \{A_1, A_2, \dots, A_m\}$ , where  $A_j$  is the ask of mobile vehicles  $m_j \in M$

As you can see, the asks of mobile vehicles are consistent no matter which cloudlets are since the mobile vehicles only care about how much payment for renting their resources.

On the contrary, the offer of cloudlets is variational respecting different mobile vehicles due to the weights of different tasks are distinct, and the performance of different mobile vehicles of different fields is different.

The auctioneer will calculate to determine the winning mobile vehicles  $C_w \in C$  and winning cloudlets  $M_w \in M$ , also define a mapping  $\sigma$  between  $C_w$  and  $M_w$ , the price  $P_i^c$  is charged from winning cloudlets  $M_w \in M \in M$ , and the reward  $P_j^m$  to  $C_w \in C$ . In particular,  $P_{ij}^c$  is denoted as the price charged

to cloudlet  $c_i$  from mobile vehicle  $m_j$ , and  $P_{ij}^m$  is defined as payment rewarded to mobile vehicle  $m_j$  from cloudlet  $c_i$ .

We are also concerned about the utilities of cloudlets and mobile vehicles in the mechanism. There are two factors that affect the utilities. The first one is the evaluation of cloudlets toward the services provided by mobile vehicles, and the next one is the cost of providing such computation resources. So, we defined  $V_i^j$  as the valuation toward mobile vehicle  $m_j$  rated by  $c_i$ .  $V_i = (V_i^1, V_i^2, \dots, V_i^m)$ . Hence, we can have the utility of  $c_i$  and  $m_j$  as below.

$$U_c^i = \begin{cases} V_i^j - P_i^c, & \text{if } c_i \in C_w, \\ 0. & \text{otherwise,} \end{cases} \quad (1)$$

$$U_m^j = \begin{cases} P_j^m - \text{Cost}_j, & \text{if } m_j \in M_w, \\ 0. & \text{otherwise.} \end{cases} \quad (2)$$

As what (1) and (2) expressed, the utility  $U_c^i > 0$  only when cloudlet  $c_i$  is allocated to a mobile vehicle  $m_j$ . The higher  $U_c^i$  is, the more satisfying the cloudlet will be. On the other side, the utility  $U_m^j$  illustrates the profit the mobile vehicles received, since it is equals to the remainder of the received reward over its cost.

**3.3. Desired Properties.** As what we have depicted, the auction model has several inputs:  $C$ ,  $M$ ,  $B$ , and  $A$ . Correspondingly, the auctioneer adopts an auction mechanism to calculate a series of results:  $C_w$ ,  $M_w$ , and  $\sigma$ . A well-designed auction mechanism must have the properties as follows.

- (i) Individual rationality:

TABLE 1: Notions.

Symbol	Definition
$c_i$	Cloudlet
$c_{ij}$	Cloudlets with positive evaluation toward mobile vehicles
$n$	Total number of cloudlets
$m$	Total number of mobile vehicles
$m_j$	Mobile vehicle
$C$	Set of cloudlets
$C^+$	Cloudlets with positive evaluation
$M$	Set of mobile vehicles
$C_c$	Set of winning cloudlets candidates
$M_c$	Set of winning mobile vehicles candidates
$C_{bf}$	Set of winning cloudlets before filtering
$M_{bf}$	Set of winning mobile vehicles before filtering
$C_w$	Set of winning cloudlets
$M_w$	Set of winning mobile vehicles
$C$	Nonincreasing order of $C^+$
$M$	Nondecreasing order of mobile vehicles
$\sigma'()$	A function that maps from $M_{bf}$ to $C_{bf}$
$\sigma()$	A function that maps from $M_w$ to $C_w$
$B_i^j$	Cloudlet $c_i$ bidding on mobile vehicle $m_j$
$B_i$	Cloudlet $c_i$ 's bid vector
$B$	Bid matrix of $C$
$A_j$	Ask of mobile vehicles $m_j$
$A$	Ask vector of $M$
$A_{-j}$	Ask vector of all sellers not including $m_j$
$V_i^j$	Valuation of $m_j$ rated by $c_i$
$V_i$	Valuation vector of $c_i$
$Cost_j$	Cost of $m_j$ performing computation offloading tasks
$P_i^c$	Price that is charged to $c_i$
$P_j^m$	Reward price paid to $m_j$
$P_{ij}^c$	Price that is charged to $c_i$ for $m_j$
$P_{ij}^m$	Reward price paid to $m_j$ with allocated $c_i$
$U_i^c$	Utility of $c_i$
$U_j^m$	Utility of $m_j$
$U_{ij}^c$	Utility of $c_i$ with allocated $m_j$
$U_{ij}^m$	Utility of $m_j$ with allocated $c_i$
$P_w^c$	The set of clearing price that is charged to $C_w$
$P_w^m$	The set of clearing rewards to $M_w$

$$P_i^c \leq B_i^j, \forall c_i \in C_w, \quad (3)$$

$$P_j^m \geq A_j, \forall c_i \in C_w, \forall m_j \in M_w. \quad (4)$$

The formula shown in (3) means that the winning cloudlet is charged less than or equal to its offer. The formula in (4) shows that the winning vehicle is paid more or equal to its ask.

(ii) Budget balance:

$$\sum_{c_i \in C_w} P_i^c \geq \sum_{m_j \in M_w} P_j^m. \quad (5)$$

The formula above shows that the total rewards that the auctioneer gives to all winning vehicles are more or equal to the sum of prices that all winning buyers submit to the auctioneer.

(iii) *Truthfulness*. An auction algorithm is said to be truthful if and only if

$$U_i^c(c_i', B_{-i}) \geq U_i^c(c_i, B_{-i}), \quad (6)$$

$$U_j^m(m_j', A_{-j}) \geq U_j^m(m_j, A_{-j}). \quad (7)$$

$B_{-i}$  means that the bid vector without the cloudlet  $c_i$  and  $A_{-j}$  means that the ask vector of all sellers not including  $m_j$ . Such that a cloudlet cannot improve its utility by faking a bid different from the true valuation of a vehicle and no vehicle can fake the ask different from its cost.  $U_i^c$  achieves its maximum only if

$$B_i = V_i, \forall c_i \in C. \quad (8)$$

The  $U_j^m$  achieves its maximum only if

$$A_j = C_j, \forall m_j \in M. \quad (9)$$

(iv) *Computational Efficiency*. The outputs of the auction mechanism, i.e.,  $C_w$ ,  $M_w$ ,  $\sigma$ ,  $P_w^c$ , and  $P_w^m$  should be calculated in a polynomial time concerning  $n$  and  $m$

As what we have depicted, the auction model has those input:  $C$ ,  $M$ ,  $B$ , and  $A$ . Correspondingly, the auctioneer adopts auction algorithms to calculate a series of result:  $C_w$ ,  $M_w$ ,  $\sigma$ , and (referring to Table 1). A well-designed auction mechanism must have the properties as follows.

**3.4. Technical Challenges.** As the what we argue in Section 2, the existing auction algorithms cannot achieve all the properties when they are applied to computation offloading in MCC with cloudlets and mobile vehicles as homogeneous items. The example shown below illustrates that TASC auction mechanism in [22] cannot support truthfulness for buyers despite the support of the other desirable properties.

Table 2 shows a matrix of bid in which the value of different mobile vehicles rated by different cloudlets are

TABLE 2: Bid matrix.

	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$
$c_1$	0	7	0	0	10	5	0
$c_2$	0	1	2	2	0	0	8
$c_3$	0	0	0	6	9	0	0
$c_4$	10	0	0	0	0	0	6
$c_5$	6	0	9	8	0	0	0

different. And Table 3 shows the ask vector of mobile vehicles which has the same value of the cost of providing computation resources. To illustrate the lacking truthfulness of TASC applied in this scenario, supposing that the auctioneer uses an algorithm to maximize the overall QoS. The TASC mechanism has two stages: the first step is to determine the winner and the next one is to determine the price. The mapping of winning cloudlet candidates and winning mobile vehicle candidates is shown in Figure 2.

In the second stage, we get the result of final winning cloudlet:  $c_1$  and  $c_4$ , the winning mobile vehicles:  $m_1$  and  $m_5$ , the  $P_w^c = 8$  and  $P_w^m = 6$ . It is clear to see that the utility of  $c_3$  is 0 according to equation (1).

If cloudlet  $c_3$  cheats the auctioneer by submitting bid dishonestly. That is,  $c_3$  increases its offer  $B_3^5$  by adding a variable  $\Delta$  which is larger than 1, since  $9 + \Delta > 10$ , the new result of winner mapping is shown in Figure 3. The winning cloudlets are changed to  $c_4$  and  $c_3$  which is still the same value as in Figure 2. The utility of  $c_3$  is changed to  $9 - 8 = 1$ . Hence,  $c_3$  can improve its utility by faking its bid. It means that TASC auction mechanism is not suitable to the scenario that we are interested in. In Section 4, we propose a truthful double auction mechanism called IMCO, which can achieve all the desirable properties illustrated in Section 1.

#### 4. Mechanism Design

As we illustrated in the background, truthful Vickrey auction [20] cannot guarantee truthfulness and other properties at the same time. However, the McAfee auction mechanism cannot be directly used in MCC with cloudlets and mobile vehicles as heterogeneous items. TASC auction mechanism is extending the McAfee mechanism in [22]. Unlike the McAfee auction, TASC is a heterogeneous auction mechanism. While TASC cannot achieve truthfulness to the scene as the example we present in the technical challenges. The result shows that TASC cannot counter market manipulation of untruthful cloudlets in stage 1.

To solve this problem, we propose IMCO. In IMCO, we change the order of the general stage. The auctioneer firstly selects the winning candidates both the cloudlets and the mobile vehicles referring to some rules. And then, each mobile vehicle that is the winning candidate will be allocated to one winning cloudlet candidate. And the  $P_w^c$  and  $P_w^m$  will be determined as well. In addition, IMCO can assign more than one mobile vehicle for a single cloudlet in a temporary state; in the last stage, the winner redundant mobile vehicles will be eliminated to ensure the efficiency of the mechanism.

TABLE 3: Ask vector of mobile vehicles.

Mobile vehicle	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$
Ask	2	3	9	5	1	4	7

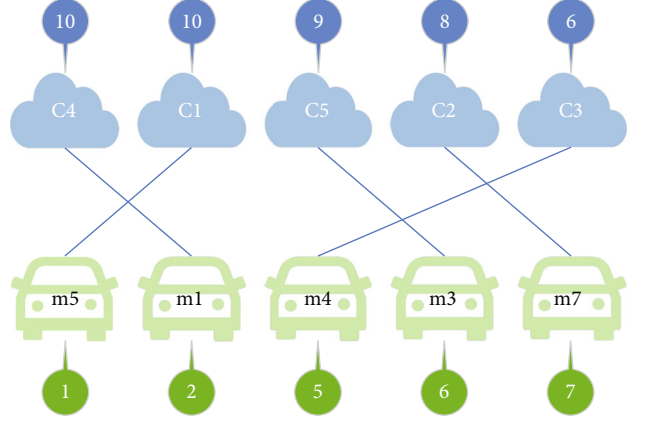


FIGURE 2: The bidirectional mapping graph of winning mobile vehicles and cloudlets.

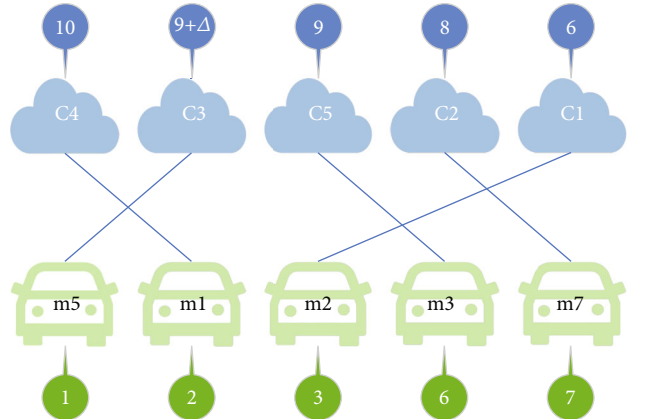


FIGURE 3: Different results of bidirectional mapping graph of winning mobile vehicles and cloudlets.

In the next part of this paper, we illustrate the mechanism of IMCO minutely. Then, we give an example to present the IMCO more clearly. The properties of IMCO will be discussed in the following chapter.

**4.1. Details of IMCO.** The IMCO consists of three phases: winner candidates determination phase, assignment and pricing phase, and winner elimination phase to achieve the preceding desirable properties.

To be specific, Algorithm 1 is used to filter some unqualified cloudlets and mobile vehicles so as to select winner candidates. To begin with, Algorithm 1 creates  $C^+$  from the original cloudlet set  $C$  which consists of bidders whose bid is greater than zero in accordance to  $B$ . That is to say, a cloudlet can appear several times in regard to the mobile vehicles for which the cloudlet's offer is larger than zero. Next, the  $C^+$  is sorted to  $C$  in which the bids are sorted in a nonascending sequence. Also, mobile vehicles are sorted

```

Input: C, M, B, A;
Output:  $C_c, M_c, B_{p\alpha}^{q\alpha}, A_{j\beta}$ ;
1:  $C_c \leftarrow \emptyset, M_c \leftarrow \emptyset$ ;
2: Create  $C^+$  which is consist of bidders whose bid is greater than zero in accordance to B, i.e.,  $C^+ = \{c_p^q: B_p^q > 0, c_p \in C\}$ ;
3: Sort all cloudlets in  $C^+$  to a non-ascending sequence  $C = (c_{p1}^{q1}, c_{p2}^{q2}, \dots, c_{p1y}^{qy})$  in which  $B_{p1}^{q1} \geq B_{p2}^{q2} \geq \dots \geq B_{p1y}^{qy}$ ;
4: Sort all mobile vehicles in M into a non-descending order  $M = (m_{j1}, m_{j1}, \dots, m_{jm})$  such that  $A_{j1} \leq A_{j2} \leq \dots \leq A_{jm}$ ;
5: Find the median ask  $A_{j\beta}$  of M, where  $\beta = \lceil 1 + m/2 \rceil$ ;
6: Find the minimized  $\alpha$ , so that  $B_{p\alpha+1}^{q\alpha+1} < A_{j\beta}$  and  $B_{p\alpha}^{q\alpha} \geq A_{j\beta}$ ;
7:  $C_c \leftarrow$  the first  $\alpha$  cloudlets in C;
8: For  $c_p^q \in C_c$  do
9:   If  $A_q > A_{j\beta}$  then
10:      $C_c \leftarrow C_c \setminus \{c_p^q\}$ ;
11:   Else
12:     If  $m_q \notin M_c$ 
13:        $M_c \leftarrow M_c \cup \{m_q\}$ ;
14:     End if
15:   End if
16: End for
17: Return  $(C_c, M_c, B_{p\alpha}^{q\alpha}, A_{j\beta})$ ;

```

ALGORITHM 1: Winner candidates determination phase.

in a nondescending list according to ask A, denoted by M. Particularly, the  $A_{j\beta}$  is the median ask of mobile vehicles in M, where  $\beta = \lceil 1 + m/2 \rceil$ , is to find the minimize value  $\alpha$ , so that  $B_{p\alpha+1}^{q\alpha+1} < A_{j\beta}$  and  $B_{p\alpha}^{q\alpha} \geq A_{j\beta}$ . The  $A_{j\beta}$  and  $B_{p\alpha}^{q\alpha}$  are two standards to identify the winning candidates. For example, if  $c_p^q$  is a winning cloudlet candidate in  $C_c$ , it should satisfy  $A_q < A_{j\beta}$  and  $B_{p\alpha}^{q\alpha} \geq A_{j\beta}$ . The mobile vehicle  $m_\alpha$  is said to be winning candidate if and only if and at least one cloudlet from C has a bid for  $m_\alpha$ . The  $\beta$  can be other value no necessary the median value. The value of  $\beta$  directly affect the number of  $M_c$  so to affect the number of  $C_c$ .

In the next stage, to protect the system from bidding and asking untruthfully, we closely combine assigning and pricing. The procedure is shown above in Algorithm 2. First, the control center assigns the winning cloudlet candidates for winning mobile vehicle candidates. There are three situations needed to be discussed. The first situation is that there is only one cloudlet bidding for one mobile vehicle, so the cloudlet is added to the  $C_{bf}$  waiting for processing Algorithm 2 and charged  $B_{p\alpha}^{q\alpha}$ . The second situation is that there are two or above cloudlets scrambling for one mobile vehicle, the auctioneer would choose the cloudlet with the highest bid from the cloudlet candidates and add it to  $C_{bf}$ . The last situation is when there are two or more cloudlets that offer the highest bids, the control center will randomly choose one cloudlet among the cloudlets with the highest bid and add to  $B_{p\alpha}^{q\alpha}$ . Different from the first situation, the cloudlet is charged for the 2<sup>nd</sup> highest bid. To make it easier to understand, supposing that the clearing price  $B_{p\alpha}^{q\alpha}$  is 6 and two highest bids are  $B_a^j = B_b^j = 12$ , and the next bid in nonascending order is  $B_c^j = 10$ , then,  $c_{aj}$  and  $c_{bj}$  have a 50-50 chance of becoming member of  $C_{bf}$ . And the winner will

```

Input:  $C_c, M_c, B_{p\alpha}^{q\alpha}, A_{j\beta}, B$ ;
Output:  $C_{bf}, M_{bf}, \sigma^j, P_{bf}^m, P_{bf}^c$ ;
1:  $C_{bf} \leftarrow \emptyset, M_{bf} \leftarrow M_c, P_{bf}^m \leftarrow \emptyset, P_{bf}^c \leftarrow \emptyset$ ;
2: For  $m_j \in M_{bf}$  do
3:    $P_j^m = A_{j\beta}, P_{bf}^m \leftarrow P_{bf}^m \cup \{P_j^m\}$ ;
4:    $C^j = \{c_{ij} : c_{ij} \in C_c\}$ ;
5:   If  $|C^j| = 1$  then
6:      $C_{bf} \leftarrow C_{bf} \cup \{c_{ij}\}, \sigma^j(j) = i$ 
7:      $P_{bf}^c = B_{p\alpha}^{q\alpha}, P_{bf}^c \leftarrow P_{bf}^c \cup \{P_{ij}^c\}$ 
8:   Else
9:     Sort  $C^j$  into a non-ascending list  $C^j$  such that
10:     $B_i^j(1) \geq B_i^j(2) \geq \dots \geq B_{p\alpha}^{q\alpha}$ 
11:    If the first  $k$  ( $k \geq 2$ ) bids in  $C^j$  are equal then
12:      Randomly choose one buyer from the first  $k$ 
13:      Buyers in  $C^j$ ;
14:    Else
15:      Select the 1st buyer of  $C^j$  whose bid is the
16:      Highest;
17:    End if
18:     $C_{bf} \leftarrow C_{bf} \cup \{c_{ij}\}, \sigma^j(j) = i$ ;
19:     $P_{bf}^c = B_i^j(2), P_{bf}^c \leftarrow P_{bf}^c \cup \{P_{ij}^c\}$ ;
20:  End if
21: End for
22: Return  $(C_{bf}, M_{bf}, \sigma^j, P_{bf}^m, P_{bf}^c)$ ;

```

ALGORITHM 2: Assignment and pricing phase.

be charged 10 to avoid untruthfully bidding since the 2<sup>nd</sup> highest bid is 10.

In the final phase, if one cloudlet in  $C_{bf}$  who wins at least two mobile vehicles in  $M_{bf}$ , the auctioneer relies on the

efficiency of utilization to decide only one mobile vehicle for a cloudlet. For example, if  $c_{i\theta}$  and  $c_{i\vartheta}$  are both in the  $C_{bf}$ , in other words,  $c_i$  as a cloudlet wins 2 mobile vehicles,  $m_\theta$  and  $m_\vartheta$  in  $M_{bf}$ . The auctioneer would choose one cloudlet according to whose utility is higher. Just as we illustrated above, if there are at least 2 utilities equal, the auctioneer will select only one mobile vehicle stochastically, when algorithm 3 is finished, there should be a one-to-one mapping relationship between the mobile vehicle in  $M_w$  and cloudlet in  $C_w$  since the redundant mobile vehicles can join another auction.

The idea that cloudlet offloads one task to multiple mobile vehicles seems an excellent way to raise the efficiency of cloudlets, but it may be too ideal to adopt. The major tasks offloaded to mobile vehicles are applications run on mobile vehicles, and the applications may not design for running parallelly or running on heterogeneous hardware.

**4.2. A Walk through Example.** In this section, we use an example to demonstrate the process of IMCO by using cloudlets' bid in Table 2 and mobile vehicles' ask in Table 3.

The first stage is to determine the winner candidates.

- (i) Create the set  $C^+$  from Table 2  $C^+ = \{c_{12}, c_{15}, c_{16}, c_{22}, c_{23}, c_{24}, c_{27}, c_{34}, c_{35}, c_{41}, c_{47}, c_{51}, c_{53}, c_{54}\}$
- (ii) Sequencing the cloudlets in  $C^+$  in a nonincreasing order to get  $\mathbb{C} = \{c_{15}, c_{41}, c_{35}, c_{53}, c_{27}, c_{54}, c_{12}, c_{34}, c_{47}, c_{51}, c_{16}, c_{23}, c_{24}, c_{22}\}$
- (iii) Sorting the mobile vehicles in  $M$  in a nondecreasing sequence to obtain  $\mathbb{M} = \{m_5, m_1, m_2, m_6, m_4, m_3, m_7\}$
- (iv) Drawing the mapping relationship graphic between cloudlets in  $\mathbb{C}$  and mobile vehicles in  $\mathbb{M}$  in Figure 4
- (v) Calculating the two standards:  $B_{p\alpha}^{q\alpha} = B_1^6 = 5$  and  $A_{j\beta} = A_6 = 4$
- (vi) Determining the winner cloudlet and mobile vehicle candidates:

$$C_c = \{c_{15}, c_{41}, c_{35}, c_{53}, c_{27}, c_{54}, c_{12}, c_{34}, c_{47}, c_{51}, c_{16}\};$$

$$M_c = \{m_5, m_1, m_2\}$$

- (vii) Drawing the mapping relationship graphic between cloudlets in  $C_c$  and mobile vehicles in  $M_c$  in Figure 4

- (viii) According to Algorithm 2, we can obtain

$$C_{bf} = \{c_{15}, c_{41}, c_{12}\}; M_{bf} = \{m_5, m_1, m_2\};$$

so we can see  $\sigma'(5) = 1, \sigma'(1) = 4, \sigma'(2) = 1$

- (ix) Computing the price of cloudlets and determining the clearing price:

$$P_{bf}^c = \{P_{15}^c = B_3^5 = 9, P_{41}^c = B_5^1 = 6, P_{12}^c = B_{p\alpha}^{q\alpha} = 5\}$$

- (x) Determining the payment paid to winning mobile vehicles:

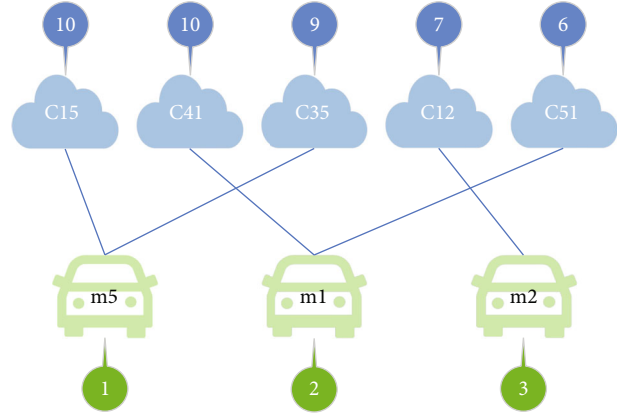


FIGURE 4: Mapping relationship graphic between  $C_c$  and  $M_c$ .

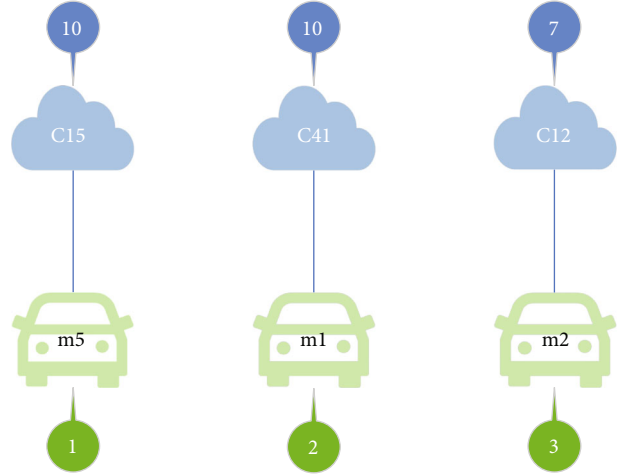


FIGURE 5: Mapping relationship graphic between  $C_{bf}$  and  $M_{bf}$ .

$$P_{bf}^m = \{P_5^m = P_1^m = P_2^m = A_{j\beta} = 4\}$$

- (xi) Drawing the mapping relationship graphic between cloudlets in  $C_{bf}$  and mobile vehicles in  $M_{bf}$  in Figure 5

However, there still has some mobile vehicles server one cloudlet at the same time; auctioneer needs to filter the redundant vehicles.

So, auctioneer executes the elimination algorithm depending on the utilities of cloudlet according to different mobile vehicles.

- (i) Calculating the utilities of cloudlet  $c_1$  according to mobile vehicle  $m_5$  and  $m_2$ :

$$U_{15}^c = B_1^5 - P_{15}^c = 10 - 9 = 1,$$

$$U_{12}^c = B_1^2 - P_{12}^c = 7 - 5 = 2.$$

- (ii) It is clear to see that  $U_{12}^c > U_{15}^c$ , so mobile vehicle  $m_5$  is filtered to achieve higher utility of cloudlet  $c_1$  with  $m_2$ . The winning cloudlet is  $C_w = \{c_1, c_4\}$



(iii) Determining the winning mobile vehicles:  $M_w = \{m_5, m_1\}$

$$P_w^c = \{P_1^c = 5, P_4^c = 6\}, P_w^m = 4, \sigma(1) = 4, \sigma(2) = 1.$$

Remember what we illustrate about the TASC mechanism which fails to satisfy the truthfulness. The IMCO overcomes the shortcomings of TASC. As what you can see below.

If  $C_3$  increases its offer  $B_3^5$  from 9 to  $9 + \delta$  to cheat auctioneer. ( $\delta > 1$ ). Then, Figure 6 shows the situation when assignment and pricing phase finish. In consequence, the utility of  $C_3$  is  $9 - 10 = -1$  which is lower than 0. So, it will be filtered in the next stage. To win the auction,  $C_3$  should bid truthfully.

## 5. Theoretical Analysis

In this chapter, we analyze the IMCO that we designed with the attributes that we are interested in, including computational efficiency, individual rationality, budget balance, and truthfulness. If and if only those properties are achieved, the auction mechanism can be applied to the scenario.

**Theorem 1.** *IMCO is computational efficiency.*

*Proof.* In Algorithm 1, there are not more than  $n \times m$  cloudlet customers in the filtered set of cloudlets with positive valuation  $C^+$ . Merge-sort sorts the cloudlets from  $C^+$  into  $C$  using  $O(nm \log(nm))$  times and sorting the mobile vehicles in  $M$  using  $O(m \log(m))$  time. Then when auctioneer sorts the cloudlets into  $C_c$ , the number of cloudlets in  $C_c$  is  $n \times \lceil 1 + m/2 \rceil$ . The remain part of the algorithm, the following for loops, takes  $O(\lceil 1 + m/2 \rceil \times (1 + n))$  times. Thus, the time complexity of Algorithm 1 is  $O(nm^2 + nm \log n)$ .

In Algorithm 2, there exist not more than  $\lceil 1 + m/2 \rceil$  cloudlet customers in  $C_c$  and mobile vehicles in  $M_c$ . To find out the subset of  $C_c C_j$ , the auctioneer needs to take  $O(nm)$  time, since auctioneer has already sort the cloudlets in non-increasing order, so the auctioneer can save some time without sorting again. There are no more than  $n$  mobile vehicles, so calculate the winning mobile vehicles only takes  $O(n)$  time. The following loops take  $O(\lceil 1 + m/2 \rceil) = O(n^2 m^2)$ .

In Algorithm 2, there exist no more than  $\lceil 1 + m/2 \rceil$  cloudlets in  $C_{bf}$  and mobile vehicles in  $M_{bf}$ , since  $|C_{bf}| = |M_{bf}|$ . The loop has a time complexity of  $O(|C_{bf}| \times |C_{bf} - 1/2|) = O(m^2)$ .

All in all, since the algorithms in IMCO is sequential execution, so the time complexity of IMCO mechanism is  $O(nm^2 + nm \log n)$ . It means that, IMCO is computational efficiency.  $\square$

**Theorem 2.** *IMCO is individually rational.*

*Proof.* Each winning mobile vehicle  $m_j \in M_w$ , the price offered to mobile vehicles  $m_j$  is  $P_w^j$  which is larger than the ask of  $m_j$  on the basis of IMCO. Therefore, mobile vehicles accommodate the demand of desirable property.

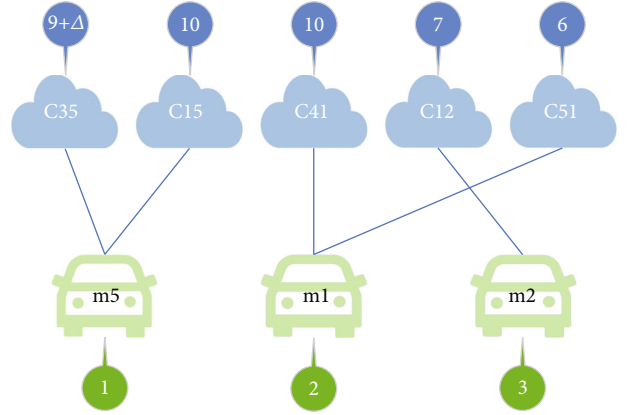


FIGURE 6: Mapping relationship graphic between  $C_c$  and  $M_c$  with untruthful bidding.

In Algorithm 2, for every cloudlet  $c_{ij} \in C_{bf}$ , there are two different situations needed to be discussed.

In the former situation, there is only one cloudlet  $c_{ij}$  selects one mobile vehicle  $m_j$ . According to Algorithm 2, the price that cloudlet  $c_i$  needed to submit to auctioneer is the clearing price  $B_{p\alpha}^{q\alpha}$  which is less than or equal to the price  $B_i^j$ .

In the last case, there are more than one cloudlet competing for one mobile vehicle. Supposing that cloudlet  $c_i$  wins the auction; in other words, it has the highest bid valued  $B_i^j$ . Referring to Algorithm 2, the  $c_{ij}$  is charged the 2<sup>nd</sup> highest bid in  $C_j$ . It is easy to see that  $P_{ij}^c$  is no more than  $B_i^j$ .

As what we illustrated above, both the cloudlets and mobile vehicles meet the desirable property. In the next stage, auctioneer filters some redundant mobile vehicles and according to the utility of cloudlet. Algorithm 3 will not affect the  $P_{ij}^m$  and  $P_{ij}^c$ , so it can still has the desirable property.  $\square$

**Theorem 3.** *IMCO is budget-balanced.*

*Proof.* After the auctioneer filters all redundant mobile vehicles, only one  $c_i \in C_w$  mapped to one  $m_j \in M_w$ . According to the mapping relationship  $\sigma(j) = i$ , it is easy to see that

$$P_i^c \geq B_{p\alpha}^{q\alpha} \geq A_{j\beta} = P_j^m. \quad (10)$$

That is

$$\sum_{c_i \in C_w} P_i^c - \sum_{m_j \in M_w} P_j^m \geq 0. \quad (11)$$

$\square$

**Lemma 4.** *IMCO is truthful for mobile vehicles.*

Referring to Algorithm 2, all the winning mobile vehicles are rewarded  $A_{j\beta}$ . According to the utility formula (2)

```

Input:  $C_{bf}, M_{bf}, \sigma', P_{bf}^m, P_{bf}^c$ ;
Output:  $C_w, M_w, \sigma, P_w^m, P_w^c$ ;
1:  $C_w \leftarrow C_{bf}, M_w \leftarrow M_{bf}, \sigma \leftarrow \sigma', P_w^m \leftarrow P_{bf}^m, P_w^c \leftarrow P_{bf}^c$ ;
2: For any two cloudlets  $c_{\sigma(a),a}, c_{\sigma(b),b} \in C_w$  &  $a \neq b$  do
3:   If  $\sigma(a) = \sigma(b)$  then
4:      $U_{\sigma(j)j}^c = B_{\sigma(j)}^j - P_{\sigma(j),j}^c, j = \{a, b\}$ 
5:     If  $U_{\sigma(a),a}^c = U_{\sigma(b),b}^c$  then
6:        $j' = \text{select from } \{a, b\} \text{ randomly};$ 
7:     Else
8:        $j' = \min(U_{\sigma(a),a}^c, U_{\sigma(b),b}^c);$ 
9:     End if
10:     $C_w \leftarrow C_w \setminus \{c_{\sigma(j'),j'}\}, M_w \leftarrow M_w \setminus \{s_{j'}\};$ 
11:     $P_w^c \leftarrow P_w^c \setminus \{P_{\sigma(j'),j'}^c\}, P_w^m \leftarrow P_w^m \setminus P_{j'}^m,$ 
12:     $\sigma(j') = \emptyset;$ 
13:  End if
14: End for
15: Return  $(C_w, M_w, \sigma, P_w^m, P_w^c);$ 

```

ALGORITHM 3: Winner elimination phase.

of mobile vehicles:

$$U_j^m = P_j^m - \text{Cost}_j. \quad (12)$$

Since the cost is regarded as constant and the reward is also constant, the utility of winning mobile vehicles can be improved, in the contrary, if the ask is higher than its cost in some degree (higher than  $A_{j\beta}$ ), the utility will decrease to 0.

**Lemma 5.** *IMCO is truthful for cloudlets.*

According to Algorithm 2, cloudlet cannot fake bidding higher than the actual value of mobile vehicle to compete the mobile vehicle, because the price charging to winning cloudlets is variable. So, the utility of faking cloudlet is below zero. From formula (1), it is easy to obtain

$$U_i^c = V_i^j - P_i^c < 0. \quad (13)$$

Since there are competition in the auction, the winning cloudlet cannot improve its utility by increasing its bid, because the price charged to it depends on the second-highest bidding.

**Theorem 6.** *IMCO is truthful.*

*Proof.* According to the above lemma, we can get that IMCO is truthful for both mobile vehicles and cloudlets, that is to say, IMCO is truthful.  $\square$

## 6. Performance Evaluation

In this part, we use the dataset to analyze the IMCO to find out whether it satisfies the desirable properties that are

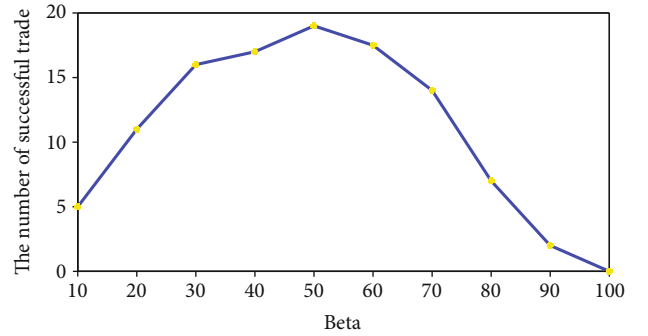


FIGURE 7: The relationship between the number of successful trades and different  $\beta$ .

proved in the section above. Also, we validate the effectiveness of the proposed approach. Because there has no statistics about the cost of mobile vehicles and the demands of cloudlets. So we use the built-in uniform distribution function to generate the bids of cloudlet and ask of mobile vehicles in the scope from 0 to 1. That is the conclusions hold true no matter what dataset is used.

**6.1. The Influence of Parameter  $\beta$ .** First, we randomly generate 100 cloudlets and 100 mobile vehicles, which means there are 10000 bids in the auction. The number of winning cloudlet candidates and winning mobile vehicle candidates is closely related to the value of  $\beta$ . We found out that the largest number of winning cloudlet candidates appears when  $\beta$  is closed to the  $\lceil 1 + m/2 \rceil$ . So, we are interested in successful trades. Figure 7 shows the relationship between the number of successful trades and different  $\beta$ .

From what we observe from Figure 7 when the  $\beta$  is small, the threshold  $B_{p\alpha}^{q\alpha}$  is small enough. However, the number of sellers is too small, so it limited the total number of cloudlet buyers. When the  $\beta$  is large, the threshold  $B_{p\alpha}^{q\alpha}$  is

TABLE 4: Computation time.

$n = 100$	$m$	50	100	150	200	250	300
Time(s)		0.160	0.456	0.742	1.034	1.325	1.617

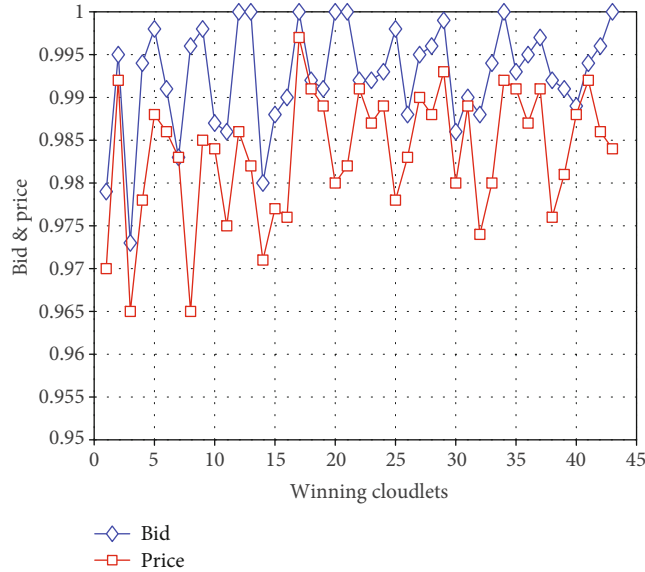


FIGURE 8: The relationship between the bids and prices of winning cloudlets.

so large that there are few cloudlet buyers who meet the conditions. And the  $\beta$  value is closely related to different values of  $n$  and  $m$ .

**6.2. Computational Efficiency.** We randomly generate different numbers of cloudlets and vehicles to test the running time. The computation time is shown in Table 4.

**6.3. Individual Rationality.** To verify this property, we present both the winning cloudlets' bids and the prices, which are shown in Figure 8. In addition, Figure 9 shows the asks and the payments of winning vehicles. We can see that the bids are always no less than the price charged to cloudlets, and the rewards to vehicles are no less than the asks of vehicles. So, IMCO meets individual rationality. In other words, both the cloudlets and the vehicles can achieve positive utilities. The auctioneer can stimulate vehicles to carry out computation offloading tasks by paying them no less than their asks while the cloudlets can make use of the computation resources of the vehicles.

As what we observe from the figures, the bids are always no less than the price charged to cloudlets, and the payment rewards to mobile vehicles are no less than the asks of mobile vehicles. So, the IMCO meets the individual rationality. In other words, both the cloudlets and mobile vehicles can achieve positive utilities, and it is a win-win game. The auctioneer can stimulate mobile vehicles to carry out computation offloading tasks by paying them no less than their asks, and the cloudlets can make use of the computation resources of mobile vehicles.

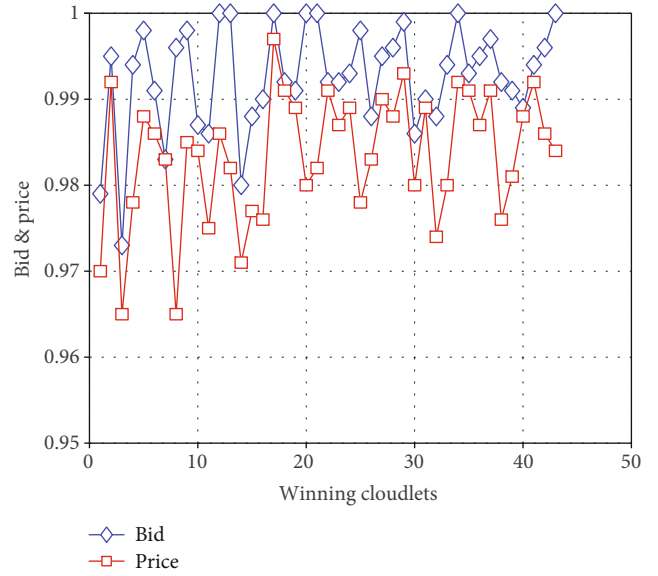


FIGURE 9: The relationship between the asks and payments of winning mobile vehicles.



FIGURE 10: The relationship between the prices and payments of IMCO.

**6.4. Budget Balance.** IMCO is only of budget balance when the total fund that the cloudlets submit to the auctioneer is larger than or equal to the payments which are rewarded to the winning vehicles. We set the number of total cloudlets to 100 consistently. By changing the number of vehicles from 50 to 150, we obtain the total prices and payments. The result is shown in Figure 10 showing each time the total number of vehicles increases by 10. It is clear that the total prices are always larger than the total payments.

As the result shows, the total prices are always larger than the total payments.

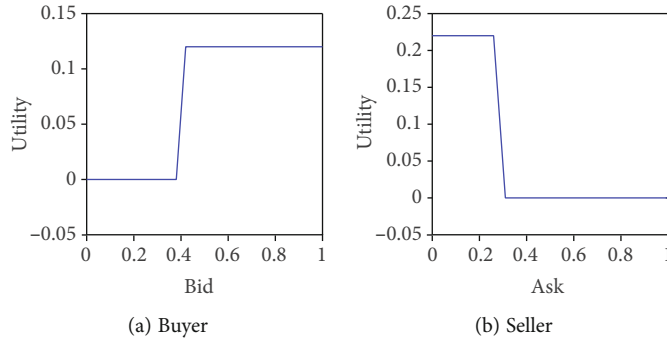


FIGURE 11: The utilities of vehicles and cloudlets in the winning set.

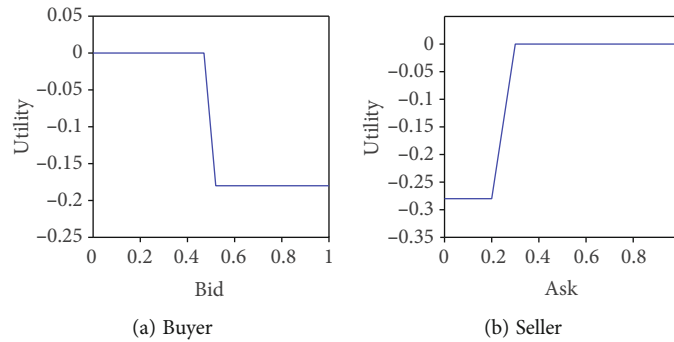


FIGURE 12: The utilities of vehicles and cloudlets not in the winning set.

**6.5. Truthfulness.** We focus on two different cloudlets and mobile vehicles groups which contain both winners and losers. Then, by changing their bids and asks from 0 to 1, we obtain the changing trends of utilities. The result of those in the winning set is shown in Figure 11. Figure 11(a) shows that when the cloudlet offers a truthful bid of 0.42, it has a utility of 0.12. However, no matter how higher a bid the cloudlet selects, the utility still stays the same, unless it reduces the bid under the original price. Figure 11(b) shows that the utility of mobile vehicles cannot be improved by asking untruthfully. Offering asks lower than 0.25 cannot achieve higher utilities while the utilities are always positive. The result of those not in the winning set is shown in Figure 12. Figure 12(a) shows that the utility of the losing cloudlet is always no larger than 0 since it loses the auction. However, even if the losing cloudlet fakes its bid, it still cannot make the utility larger than 0. Figure 12(b) shows that the utility of a losing mobile vehicle can only be 0 or below 0. Overall, IMCO can achieve truthfulness for both cloudlets and mobile vehicles.

In conclusion, IMCO can achieve truthfulness for both cloudlets and mobile vehicles. They cannot improve their utilities by lying.

**6.6. System Efficiency.** In Section 5, we have proved that IMCO meets the desirable properties in theoretical, and Section 6 uses the numerical results shows the above properties. As what we address, the TASC auction mechanism cannot guarantee the property of truthfulness applied in the cloudlet

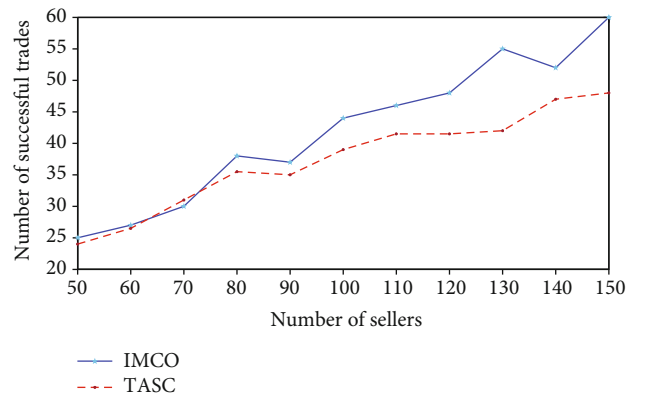


FIGURE 13: System efficiency with uniformly distributed datasets.

scenario. In Figure 13, the system efficiency of IMCO and TASC is evaluated as the number of successful trades.

Figure 14 shows the efficiency of TASC and IMCO according to a number of successful trades. The bids and asks are generated by the MATLAB built-in unified function which is used to generate uniformly distributed datasets. Some of the efficiency of IMCO is sacrificed to guarantee truthfulness. Thus, the total efficiency of IMCO still has an advantage over TASC since IMCO maintains more winning cloudlet candidates than TASC, and IMCO only filters a few winning cloudlets in the third stage. Hence, IMCO can achieve a reasonable efficiency with all the desirable properties.

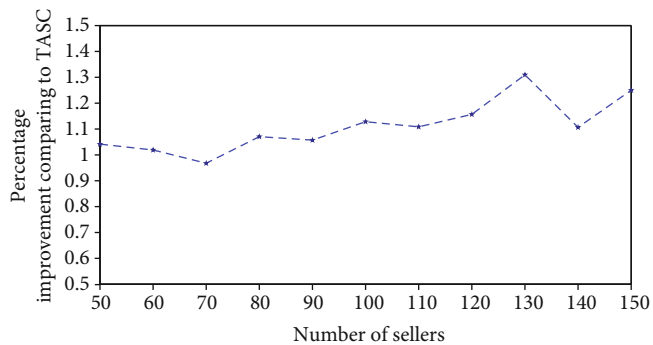


FIGURE 14: Percentage improvement of normalized system efficiency comparing to TASC with uniformly distributed datasets.

## 7. Conclusion

In this paper, we concentrate on an emerging paradigm in the current IoT systems, in which computation is offloaded to vehicle owners. Because of the distance between mobile vehicles and cloudlets and the distinct computational capabilities of mobile vehicles, the mobile vehicles can be seen as heterogeneous sellers to cloudlets, so different cloudlets may evaluate each mobile vehicle due to different demands and give different valuations toward different vehicles. In order to make mobile vehicles offer the computation offloading servers to cloudlets, we have proposed the IMCO which deals with the resource exchanging between mobile vehicles and cloudlets. IMCO can meet the properties such as individual rationality, budget balance, system efficiency, and truthfulness for both cloudlets and mobile vehicles.

There are some aspects that may be improved in future work, such as the system efficiency is not that satisfying even it is better than TASC as a whole since IMCO sacrifices some efficiency to maintain the truthfulness for both cloudlets and mobile vehicles. The other problem is that even if the time complexity of the mechanism is acceptable, the program in MATLAB could not be seen as of high-efficiency. So we could optimize the programs to execute faster as we expected in further study.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by Research Innovation Fund for College Students of Beijing University of Posts and Telecommunications.

## References

- [1] S. Xia, F. Lin, Z. Chen, C. Tang, Y. Ma, and X. Yu, "A Bayesian game based vehicle-to-vehicle electricity trading scheme for blockchain-enabled internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6856–6868, 2020.
- [2] M. Chen, S. Guo, K. Liu, X. Liao, and B. Xiao, "Robust computation offloading and resource scheduling in cloudlet-based mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2025–2040, 2021.
- [3] S. Chen, Y. Zheng, W. Lu, V. Varadarajan, and K. Wang, "Energy-optimal dynamic computation offloading for industrial iot in fog computing," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 566–576, 2020.
- [4] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.
- [5] L. Li, T. Q. Quek, J. Ren, H. H. Yang, Z. Chen, and Y. Zhang, "An incentive-aware job offloading control framework for multi-access edge computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 63–75, 2021.
- [6] A. Mukherjee, D. De, and D. G. Roy, "A power and latency aware cloudlet selection strategy for multi-cloudlet environment," *Computing*, vol. 7, no. 1, pp. 141–154, 2019.
- [7] B. Zhao, S. Tang, X. Liu, and X. Zhang, "PACE: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2021.
- [8] R. Han, Q. Guan, F. R. Yu, J. Shi, and F. Ji, "Congestion and position aware dynamic routing for the internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16082–16094, 2020.
- [9] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5871–5883, 2019.
- [10] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [11] G. Pan, J. Ye, Y. Tian, and M.-S. Alouini, "On HARQ schemes in satellite-terrestrial transmissions," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7998–8010, 2020.
- [12] C. Lin, Y. Yu, J. Xiong et al., "Shrimp: a robust underwater visible light communication system," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 134–146, New York, NY, USA, 2021.
- [13] C. Lin, F. Gao, H. Dai, J. Ren, L. Wang, and G. Wu, "Maximizing charging utility with obstacles through Fresnel diffraction model," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 2046–2055, Toronto, ON, Canada, 2020.
- [14] C. Lin, Z. Wang, J. Deng, L. Wang, J. Ren, and G. Wu, "mts: temporal- and spatial-collaborative charging for wireless rechargeable sensor networks with multiple vehicles," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018.
- [15] C. Lin, Z. Yang, H. Dai, L. Wang, and G. Wu, "Minimizing charging delay for directional charging," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2478–2493, 2021.
- [16] C. Lin, Y. Zhou, F. Ma, J. Deng, L. Wang, and G. Wu, "Minimizing charging delay for directional charging," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2478–2493, 2019.

- [17] C. Lin, J. Zhou, C. Guo, H. Song, G. Wu, and M. S. Obaidat, "TSCA: a temporal-spatial real-time charging scheduling algorithm for on-demand architecture in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 211–224, 2018.
- [18] Y. Sun, C. Lin, H. Dai et al., "Trading off charging and sensing for stochastic events monitoring in wsrns," *IEEE/ACM Transactions on Networking*, pp. 1–15, 2021.
- [19] C. Lin, Z. Shang, W. Du, J. Ren, L. Wang, and G. Wu, "Codoc: a novel attack for wireless rechargeable sensor networks through denial of charge," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 856–864, Paris, France, 2019.
- [20] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of Finance*, vol. 16, no. 1, pp. 8–37, 1961.
- [21] R. P. McAfee, "A dominant strategy double auction," *Journal of Economic Theory*, vol. 56, no. 2, pp. 434–450, 1992.
- [22] D. Yang, X. Fang, and G. Xue, "Truthful auction for cooperative communications," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1–9, Paris, France, 2011.
- [23] X. Zhang, H. Zhang, S. Dai, and Y. Liu, "An incentive mechanism for computation offloading in satellite-terrestrial internet of vehicles," in *2021 the 7th International Conference on Computer and Communications*, Chengdu, Sichuan, China, 2021.
- [24] A. Guidotti, A. Vanelli-Coralli, M. Conti et al., "Architectures and key technical challenges for 5G systems incorporating satellites," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2624–2639, 2019.
- [25] H. Guo and J. Liu, "Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4514–4526, 2018.
- [26] Z. Wang, D. Zhao, M. Ni, L. Li, and C. Li, "Collaborative mobile computation offloading to vehicle-based cloudlets," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 768–781, 2021.
- [27] V. Krishna, *Auction Theory*, Academic Press, 2009.
- [28] A. Yassine, M. S. Hossain, G. Muhammad, and M. Guizani, "Cloudlet-based intelligent auctioning agents for truthful autonomous electric vehicles energy crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5457–5466, 2020.
- [29] X. Chen, T. Zhang, W. Ye, Z. Wang, and H. H.-C. Iu, "Blockchain-based electric vehicle incentive system for renewable energy consumption," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 396–400, 2021.
- [30] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 319–333, 2019.
- [31] J. Du, E. Gelenbe, C. Jiang, Z. Han, and Y. Ren, "Auction-based data transaction in mobile networks: data allocation design and performance analysis," *IEEE Transactions on Mobile Computing*, vol. 19, no. 5, pp. 1040–1055, 2020.