An Efficient Collaboration and Incentive Mechanism for Internet-of-Vehicles (IoVs) with Secured Information Exchange Based on Blockchains

Bo Yin, Yulei Wu, Tianshi Hu, Jiaqing Dong, Zexun Jiang*

***Abstract*—With the rapid development of Internet-of-Things (IoT), mobile crowdsensing, i.e., outsourcing sensing tasks to mobile devices or vehicles, has been proposed to address the problem of data collection in the scenarios such as smart city. Despite its benefits for a wide range of applications, mobile crowdsensing lacks an efficient incentive mechanism, restricting the development of IoT applications, especially for Internet-of-Vehicles (IoV) – a typical example of IoT applications; this is because vehicles are usually reluctant to participate these sensing tasks. Moreover, in practice some sensing tasks may arrive suddenly (called an emergent task) in the IoV environment, but the resources of a single vehicle may be insufficient to handle, and thus multi-vehicles collaboration is required. In this case, the incentive mechanisms for the participation of multiple vehicles and the task scheduling for their collaborations are collectively needed. To address this important problem, we firstly propose a new model for the scenario of two vehicles collaboration, considering the situation of emergent appearance of a task. In this model, for a general sensing task, we propose a bidding mechanism to better encourage vehicles to contribute their resources, and the tasks for those vehicles are scheduled accordingly. Secondly, for an emergent task, a novel time-window based method is devised to manage the tasks among vehicles and to incent the vehicles to participate. Finally, we develop a blockchain framework to achieve the secured information exchange through smart contract for the proposed models in IoV.**

***Index Terms*—Mobile crowdsensing, Incentive mechanism, Internet of Things, Internet of Vehicles, Blockchain**

I. Introduction

WITH the rapid development of smart mobile devices and embedded sensors, Internet-of-Things (IoT) has become an indispensable part of people's lives. As suggested by Gartner [1], IoT is and will still be the fastest-growing, the largest market potential, and the most attractive emerging economy. IoT has revolutionized a wide range of fields, and its applications, such as Internet-of-Vehicles (IoVs) [2], in the context of smart cities are of particular interest in the community. IoV is an open and integrated networking system composed of vehicles, users and networks, and a vehicle possesses the computation, sensing and storage resources. To better improve every aspects of people's lives and to make the smart city a reality, more sophisticated data needs to be collected through IoV.

A promising way to collect such a huge volume of data in IoVs is through mobile crowdsensing (MCS) [3]–[11], which outsources the sensing tasks to the sensors of vehicles. MCS usually involves an IoT center for receiving data collection requests and delegating the data sensing tasks to the participating devices like the vehicles in the IoV environment. It takes advantage of the computation, sensing and communication resources of vehicles and avoids to deploy a large number of task-specific sensors [3]. However, vehicles may be reluctant to contribute their resources to complete MCS tasks [12]. Thus, an effective incentive mechanism is urgently needed to encourage vehicles to participate the data sensing so as to promote the development of IoV and smart cities.

Many incentive mechanisms [13]–[23] have been reported in the literature. Most of them fail to consider the situation of an emergent sensing task in IoV. This kind of task has the characteristic of delay-sensitive nature and needs to be handled timely. In addition, there has not been any mechanism in IoV to secure the information exchange for incentive mechanisms, which is a critical issue in IoV [12], [24]. Blockchain as the most popular distributed ledger technology [25] has enabled vehicular applications for secured authentication [26] and communication [27]. It is promising to integrate blockchain to handle the information exchange of IoV in an MCS system.

Aiming at these shortcomings of the existing incentive mechanisms and MCS task allocation problems in IoV, we propose a new MCS incentive mechanism with regard to timing constraints. In addition, a new model for multi-vehicles collaboration and delay-sensitive task assignment is developed and analyzed. The main contributions of this paper are summarized as follows:

- We distinguish between a general task and a delay-sensitive emergent task, and innovatively combine the two types of tasks into one unified scheduling problem. To solve the general task assignment problem, we model the MCS task allocation as a budget constraint bidding problem.
- To handle the delay-sensitive emergent task whilst processing general tasks, we develop a new multi-vehicles task assignment model. The model elaborates the attributes of emergent tasks and considers the resource limitation nature of vehicles due to the processing of general tasks. A multi-vehicles collaboration method based on idle time window is proposed, which can effectively

B. Yin, J. Dong and Z. Jiang are with Research Institute of Information Technology, Tsinghua University, Beijing, China (email: yinbo@tsinghua.edu.cn; djq13@mails.tsinghua.edu.cn; jiangzx14@mails.tsinghua.edu.cn)

Y. Wu is with College of Engineering, Mathematics and Physical Science, University of Exeter, Exeter, UK (email: Y.L.Wu@exeter.ac.uk)

T. Hu is with International School, Beijing University of Posts and Telecommunications, Beijing, China (email: ertyoii@bupt.edu.cn)

ensure the real-time and effectiveness of multi-vehicles resource allocation and improve the vehicle resource utilization rate.
- To improve the efficiency and transparency of vehicle collaborations, a blockchain framework is envisioned to realize the bidding and secured information exchange between vehicles and the IoT center. The novelty is that we take into account how the participants are actually paid with incentive mechanisms, which is largely ignored in previous works.
- Experiment results demonstrate the advancement of the proposed method in terms of more efficient task allocation and processing, leading to shorter task running time, and extra profits gained due to the processing of emergent tasks.

The rest of the paper is organized as follows. The related work is summarized in Section II. The multi-vehicles collaboration problem is described in Sections III - V. Specifically, in Section III, we distinguish two types of tasks: the general task and the delay-sensitive emergent task; we describe the general task assignment problem and the emergent task assignment problem in Sections IV and V, respectively. A time-window based method to solve multi-vehicles collaboration issue is proposed in Section VI. The proposed method is validated and evaluated by extensive experiments in Section VII. Finally, Section VIII concludes this study.

II. Related Work

A. Mobile Crowdsensing

Extensive studies have been carried out to investigate the incentive mechanisms to encourage the participation of mobile devices in MCS. They can be generally grouped into three categories, namely, location based, social network based and time based.

For location based incentive methods [15], [16], [18], the location and the coverage of mobile devices are of great importance, and they become part of the constraints for optimization. As pointed out by Huang and Tseng [28], the coverage is a fundamental issue in wireless sensor networks to reflect how well an area is monitored. Zheng et al. [13] utilized the idea of coverage and considered the task allocation as a coverage problem. The authors modeled the scenario that the service provider publishes a number of points of interests and the data providers bid with a pair of the task and its cost. Wang, Wei and Qi [15] studied the vehicle MCS and took into account not only the current location but also the future location of vehicles when recruiting vehicles for MCS tasks. Tao and Song [16] studied the location of tasks with regard to a clustering effect. They only modeled the reward for data providers as the combined cost of sensing and travelling to a certain location, but the participatory and selfishness nature of data providers are omitted. Ko, Pack and Leung [18] proposed a coverage guaranteed and energy efficient participant selection model for MCS. They modeled the sensing tasks of static users and focused on reducing the energy consumption for devices, and they assumed that delay is allowed so that data can be sent in batch. For all the methods discussed above, where only the location or coverage requirements of MCS tasks are considered, the sensing is performed only once in a task. However, several tasks, such as traffic monitoring in [29], [30], require repeated sensing in an area. In addition, the time constraint of delay-sensitive tasks is largely ignored.

As for the social network based incentive mechanisms [17], [20]–[22], they considered social cost of participation and utilized social platform to recruit. Jiang et al. [17] proposed a social network based MCS model and discussed the prevention of sybil attack after introducing the idea of social networks. They also considered time-sensitiveness of tasks, but they failed to consider the repeated sensing in a task. Chen et al. [21] proposed a three-layer incentive structure involving the social applications, so as to taking advantage of the user base from social applications instead of finding and stimulating users per sensing tasks. Nie et al. [20], [22] considered the network effect, which refers to the phenomenon that public goods or services are more valuable if it is adopted by more users, with incentive. They modeled the interaction between service providers and users as a game. These studies [20]–[22] can be generally treated as platform-centric solutions where the rationality and selfishness nature of users is ignored.

Although time-based incentive mechanisms have been proposed in [14], [19], they focused only on the time constraint aspect without considering the repeated sensing nature in a task. Zhan et al. [14] assumed the selfishness of mobile users and modeled the data collection process as a cooperative game between data provider and requester. They took into account the time constraint of a task which needs to be finished within a limited time. Xu et al. [19] proposed a novel scenario where the platform needs the data collection to be completed in a requested time window. Besides, they showed that the data collected in the time window has sufficient integrity. However, these incentive mechanisms focus on the fact that tasks need to be finished within a time limit; they failed to plan the sensing tasks for several time slots, which is the case for the tasks requiring repeated monitoring. Duan et al. [12] studied vehicle monitoring scenarios and proposed two different modes, i.e., an offline mode and an online mode, with regard to the lack of fairness, unconsciousness and randomness of mobile devices of the current incentive mechanisms. The novelty of their work is that they modeled the bidding with not only the demanding price but also the location-time pair of the vehicle, so that repeated sensing tasks can be allocated on time. However, they omitted the resource requirements for each bidding.

In summary, the existing literature lacks the attention to the time constraint of MCS tasks in two perspectives. On the one hand, a number of studies failed to take the delay-sensitive emergent tasks into account. On the other hand, for those who considered the time constraints of a task, they performed the sensing only once in a task. In the environment with heterogeneous sensing requests, time-sensitive emergent tasks should be considered on top of the fact that devices are carrying out repeated sensing tasks. This combined framework is not shown in any of the reviewed literatures. Moreover, for all the works investigated above, how the payment is securely dealt with is not considered.

B. *Blockchains*

The blockchain has become a promising solution to tackle trust and privacy issues of information sharing in MCS. Feng and Yan [31] proposed an MCS-chain to solve privacy and fault-tolerance of existing MCS networks. They focused on an innovative consensus algorithm and trust evaluation, but the execution of the bidding and information exchange are ignored. Chatzopoulos et al. [32] studied the usage of blockchains in MCS by proposing an effective incentive auction. The bidding and information exchange problems are addressed, but it is for location-based MCS and only location privacy is preserved. Zhao et al. [33] also discussed the blockchains in MCS focusing on preventing malicious nodes that publish false information or intentionally not provide data after accepting a task.

III. An Overview of Blockchains

Blockchains, which are developed as the key technology behind Bitcoin [34], have gained popularity in various areas of applications. It is composed of blocks, and each block contains a hash of the previous block along with a time stamp. The blocks combined to form a distributed ledger is transparent in a way that all the transactions made since the creation of the blockchain can be available to check. It is tamper-proof, meaning that the data recorded on the blockchain cannot be modified and with no fraud [35]. It can be seen as a log whose records are batched into time-stamped blocks [36], and it can be used by various parties to efficiently record transactions between each other in a verifiable and permanent way [37].

A. *Blockchain Networks*

A blockchain network [38] consists of a set of nodes which are the entry point for multiple users or devices to interact with each other. The nodes make executions on behalf of these users or devices, and they keep both a replicated copy of the ledger of the blockchain, which will be updated when transaction is made on the blockchain, and the smart contract for execution of the transactions [39]. The transaction encompasses a variety of data which is valuable in the blockchain network, such as the information collected by the IoT devices [40], bitcoin, etc. Before being recorded on a block, a transaction needs to be signed off by a node using its private key, and its contents can be examined by other nodes using the corresponding public key. After the transaction is validated and accepted by nodes in the blockchain network, the record of the new block containing the information about this transaction will be broadcasted to the whole network where each node updates its ledger by adding this block.

B. *Smart Contracts*

Smart contract can ensure a secure, efficient and automatic data exchange between different parties without the traditional contracting process such as search, negotiation and commitment [38]. The basic idea behind smart contracts is that many contractual clauses (such as collateral, bonding, delineation of property rights, etc.) can be embedded in the hardware and software, in which case the breach of contract is expensive [41]. A smart contract is a predefined set of rules that agreed by multiple parties on how a transaction affects their status, such as their account. For example, a smart contract can be used to query a certain data on the blockchain with regard to some kind of information, and it can also be used to move 10 certain assets from node A to node B. The execution of a transaction will result in a status which will be recorded in the ledger by each party in the blockchain network.

IV. Mobile Crowdsensing in Internet-of-Vehicles

As classified by Ganti, Ye and Lei in [3], MCS applications can be classified into three categories, namely, environmental, infrastructural, and social applications.

For environmental applications like noise monitoring [42], and infrastructure applications like traffic congestion monitoring [29], as well as social applications like BikeNet [43] relying on individuals to contribute the location and bike route quality, continuous data collection is needed. In other words, for a single task in these applications, repeated sensing is required. We name this type of sensing task as *general sensing tasks* or *general tasks* in short. For safety related applications suggested in [44], data collection is delay-sensitive, and we term this kind of task as *emergent tasks*.

For a general task, e.g., noise monitoring that requires data to be collected continuously, the IoT center will publish the task descriptions including task name, required resource and time constraint through smart contract to the blockchain. The task name can be noise monitoring; required resource can be feedback on noisy level; and time constraint can be a time period. Then, data providers can use smart contracts to view tasks and offer biddings if they intend to participate the sensing task. The related bidding information including participant ID and its bidding and resource pair will be stored on the blockchain. Upon receiving bidding information after a predefined period of time, smart contract will be executed on behalf of IoT center to select an optimal vehicle set and publish the result on blockchain. Next, the vehicles will be asked to provide the required data to store on the blockchain when being notified that they are selected. The IoT center retrieves the provided data from the blockchain, and then it will offer reward to the participants in terms of tokens through the use of smart contract.

While the vehicles are carrying out the general tasks in this area, suddenly, suppose an ambulance carrying a patient plans to go through this area, which publishes the task of inquiring the area traffic congestion onto the blockchain. The IoT center then publishes the tasks and does the same thing as with general tasks. In this situation, the tasks are delay-sensitive and emergent, and the vehicles need to utilize their idle resources such as camera and acceleration sensors to help with the emergent tasks.

In what follows, we propose a bidding mechanism with regard to time and resource constraints to schedule the vehicles with general tasks in Section IV. Then, a time-window based algorithm is proposed in Section V, to schedule the delay-sensitive emergent tasks. All the communications between vehicles and the IoT center are handled by a blockchain platform [25]–[27], including bidding, payment, and scheduling information.

V. Multi-vehicles General Task Assignment

A. Problem Description

The problem solved in this section is for the general task scheduling in multi-vehicles collaboration environment. In this scenario, tasks are scheduled by an IoT center through a permissioned blockchain [26], where all the nodes have already been registered on it. In this phase, it is assumed that there are M tasks, denoted by $T = \{T_1, T_2, \ldots, T_M\}$, to be scheduled by the IoT center, and each task T_i has a requirement with the pair (t_i, d_i), where t_i represents all the required time periods to carry out the task and it is defined as $\{t_i^1, t_i^2, \ldots, t_i^n\}$, and d_i, defined by the quality parameters $\{d_i^1, d_i^2, \ldots, d_i^n\}$, is the minimum data quality requirement. For example, for a noise monitoring task which needs the data to be collected from the road, the sensing time can be every single hour from 7:00 to 10:00 and from 16:00 to 19:00. The data quality requirements can be the noisy level (a vague value, e.g., OK and not noisy, or accurate decibel which can be mapped to a certain value representing the quality) and continuous sensing time. The IoT center publishes tasks on the blockchain and waits for the response from the vehicles offering biddings; the publishing and bidding can be performed on the blockchain through a smart contract [27]. Assume there are N vehicles, denoted by $A = \{A_1, A_2, \ldots, A_N\}$, willing to finish the tasks in order to gain rewards, and each vehicle A_j will offer a bidding B_j which comprises a pair of bidding price along with their data quality (B_j, D_{ij}, τ_j). For the task T_i, the quality D_{ij} contains n quality parameters $\{D_{ij}^1, D_{ij}^2, \ldots, D_{ij}^n\}$, and τ_i contains all the possible time periods $\{\tau_i^1, \tau_i^2, \ldots, \tau_i^n\}$ that a vehicle is able to contribute their resources.

For the tasks to be scheduled, the IoT center chooses the vehicle, considering its own budget limit and the combined data quality provided by the vehicle. Next, the IoT center schedules the tasks and waits for the data to be collected from the vehicle which will be rewarded with their corresponding bidding and task completion. These processes are recorded and are also performed on the blockchain through a smart contract. We consider the token as the monetary reward in this study, and the processes are shown in Fig. 1.

In the blockchain network, the devices from data providers and IoT center are both peer nodes, and they can execute smart contracts to store or retrieve information from the blockchain of their network.

B. Basic Assumptions

1) Resource Awareness: The vehicles are aware of the consumption of their resources such as battery and computation capability when finishing the allocated tasks, so they can bid with a price higher than the cost of the resources they need to spend out for their own interest [12], [17].

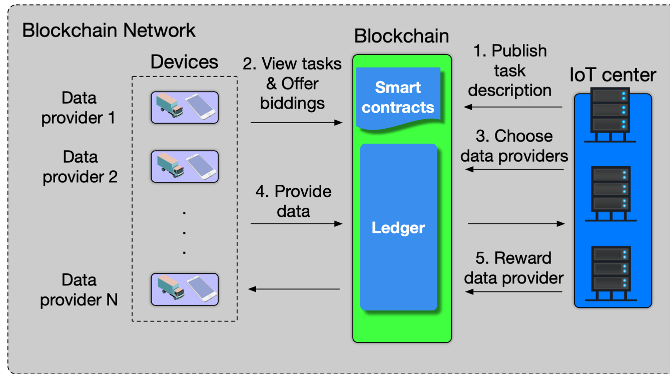


Fig. 1. The senario of vehicles mobile crowdsensing: bidding and selecting processes

2) Quality and Value: A value function $V(d_i^1, d_i^2, \ldots, d_i^n)$ is proposed to compute the value of the provided data, and it satisfies Equation (1).

$$\frac{\partial V(d_i^1, d_i^2, \ldots, d_i^n)}{\partial d_i^k} \geq 0 \tag{1}$$

The above equation shows that the value of data will increase with the value of data quality. Therefore, the higher data quality per quality parameter is needed, the higher value of the data is for the IoT center. We can simply represent the value of data by combining the quality parameters.

3) Malicious Bidding: Vehicle nodes could bid with unreasonable price and false data quality. For the high price, the IoT center can ignore bidding requests, because it has a budget limit and is for the benefit of itself; it selects a set of vehicles with lowest price and highest data quality. For the nodes reporting false data quality parameters, when their data is gathered by the IoT center during data collection phase, the IoT center will spot the difference between the real data quality and the reported one. Then, these nodes will no longer be able to participate data collection and have no reward. This can be done by removing these nodes from the network. However, vehicles will not maliciously use their resources to attack other vehicles [45], [46].

C. Quality Requirement

To finish a task T_i, all the parameters of its quality requirements must be satisfied by:

$$d_i^a \geq \sum_{j=1}^{m} D_{ij}^a, \forall a \in \{1, 2, \ldots, n\} \tag{2}$$

where D_{ij}^a is provided by the vehicle A_j belonging to the selected set of vehicles. All the quality parameters ranging from 1 to n must be satisfied.

D. Budget Requirement

The IoT center has a budget limit b which means the combined bidding price of the selected vehicles must satisfy:

$$b \geq \sum_{j=1}^{m} B_j \tag{3}$$

where B_j is the bidding price of vehicle A_j. Therefore, for those malicious nodes who bid with unreasonable price, their bid will not be accepted by the IoT center when other nodes, with similar resources, bid with reasonable price due to the price limit of the IoT center.

E. Time Availability

For a bidding vehicle A_i, the time availability variable $L_i = \{L_i^1, L_i^2, \ldots, L_i^n\}$ is formed by comparing the bidding contribution time of the vehicle and the task requirement time. For the value of $L_i^\alpha = 1$ where the vehicle can participate in the required time, the vehicle can continuously contribute to sensing data; $L_i^\alpha = 0$, is for the situation that the vehicle cannot participate during that period of time.

F. Optimal Set Selection

Combining with Equations (1) and (2), the optimal set for the general tasks is selected for the maximal benefit of the IoT center:

$$\max \frac{\sum_{t=1}^{K} \sum_{j=1}^{N} \sum_{k=1}^{n} D_{ij}^k L_t}{\sum_{j=1}^{m} B_j} \quad (4)$$

where N vehicles are selected for the set with the n quality parameters of M satisfied tasks, and K is the number of time slots.

VI. Multi-vehicles Emergent Task Assignment

A. Problem Description

The problem to be solved in this section is for real-time scheduling of the burst of tasks in multi-vehicles collaboration environment. Because the tasks to be dealt with are emergent tasks, meeting the task resource requirement is of great importance. Thus, the data quality can be ignored, while ensuring the idle resources of vehicles. The bidding process can be omitted, and the IoT center makes vehicles cooperate and rewards the selected optimal set after the tasks are completed. Because of the time constraint, the emergent tasks have both resource attribute and time attribute. In this problem, we take the time requirement as X-axis and the resource demand as Y-axis. We consider a typical scenario that there are M tasks $T = \{T_1, T_2, \ldots, T_M\}$ in the range $Q = [0, s] \times [0, s]$, where each task is represented by a pair of resource requirement and time attribute. Here, the time attribute is the duration of the time needed to be taken for processing the task. The type and quantity of resources required to process T_i are expressed as $R_i = \{R_i^1, R_i^2, \ldots, R_i^m\}$, where R_i^k represents the demand for the k-th resource for handling the emergent task T_i. The N types of heterogeneous vehicles $A = \{A_1, A_2, \ldots, A_N\}$ are used to indicate the running vehicles in the current area Q, and the vehicles in the area where the current burst of tasks are located can cooperate to complete the task processing. Corresponding to the attribute requirements in the target task, the remaining resources $R_j = \{R_j^1, R_j^2, \ldots, R_j^n\}$ of different types of vehicles are different.

For the burst of tasks to be processed, the goal of multi-vehicles collaborative task processing includes two aspects: a) within the range Q, allocate vehicle resources to maximize the benefits of task processing for both the IoT center and vehicles, where vehicles can gain more profits by bidding with higher price than the original cost, and the IoT center can have their tasks finished; b) for a burst of tasks, it is necessary to respond to and process the task request in time, and process as many tasks as possible within the shortest time, which can improve the efficiency of overall task processing.

B. Resource Requirements

The vehicle task assignment module performs the initial task assignment for the tasks within the range Q, and multiple types of vehicles perform the assigned tasks; we have discussed this in the previous section, i.e., Section V. During the period when a vehicle has idle resources, if a new task T_j arrives, the vehicle processes it with the following two scenarios:

Scenario 1: The vehicle is constrained by its own resource limit. There is a situation where the resources of a single vehicle are not enough to handle a new task, as shown in Equation (5):

$$x_{ij}^k r_i^k \leq R_i^k, \forall k \in \{1, 2, \ldots, m\} \quad (5)$$

where x_{ij}^k indicates whether the k-th attribute of the vehicle i satisfies the resource requirement corresponding to the target task T_j, and r_i^k represents the amount of the k-th resource of a vehicle i. $x_{ij}^k = 1$ indicates that the task is assigned to the vehicle i for processing, otherwise, $x_{ij}^k = 0$.

Scenario 2: The vehicle may not be constrained by its own resource limit, and thus Equation (5) does not need to be satisfied.

In both scenarios, in order to complete the processing of the task T_j, it is necessary to form a multi-vehicle set I_A together with the other vehicle A_i possessing the idle resources. The total resources of I_A must meet the resource requirements of the target T_j, and thus the Equation (6) holds.

$$\sum_{A_i \in I_A} x_{ij}^k r_i^k \geq R_i^k, \forall k \in \{1, 2, \ldots, m\} \quad (6)$$

C. Execution Profit

To encourage more vehicles to participate the data collection process for emergent tasks, the IoT center rewards the vehicles with execution profit based on the contribution that each vehicle makes. The vehicle gains the execution profit after processing a task. Firstly, the task execution profit function is defined as follow. When a vehicle processes the target task, different types of tasks are processed, and thus different benefits are obtained. Therefore, when the vehicle A_i is allocated to process the task T_j, the net income that can be obtained is expressed as

$$C_{ij}^k = \sum_{k=1}^{I} G_{ij}^k - I(X_{ij}^D = 0), I = 1 \text{ iff } X_{ij}^D = 0 \quad (7)$$

where G_{ij}^k denotes the profit gained by the vehicle A_i from processing the task T_j.

$$G_{ij}^k = p_{ij}^k V_j^k, (p_{ij}^k = f_i^k \frac{R_i^k}{R_j^k}) \quad (8)$$

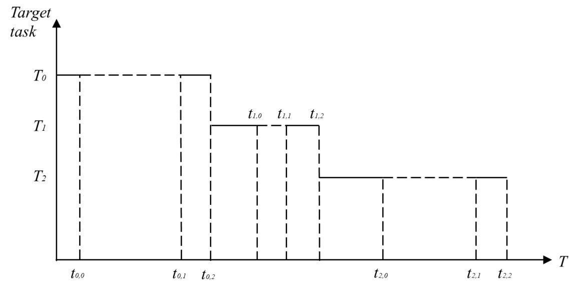


Fig. 2. The task processing sequence of a vehicle

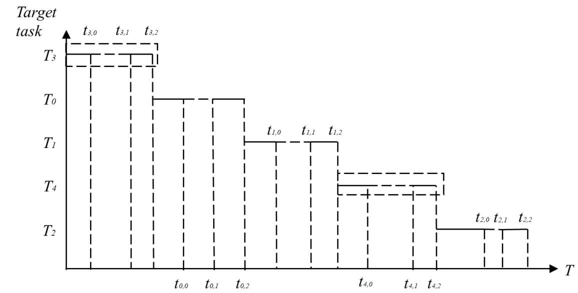


Fig. 3. The task processing sequence of a vehicle with idle time window

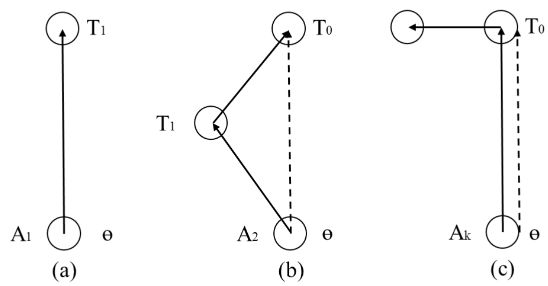


Fig. 4. The type of the starting time of a vehicle

where p_{ij}^k represents the completion probability of assigning the vehicle A_i to process the task T_j. f_i^k denotes the probability that the vehicle A_i can satisfy the k-th resource demand of T_j. For each resource provided by a vehicle, the profit V_j^k is a constant which is set by the IoT center.

Given the profits described above, the total profit of a multi-vehicles task assignment problem can be expressed as:

$$C_A = \sum_{i=1}^{n} \sum_{k=1}^{m} x_{ij}^k C_{ij}^k \tag{9}$$

where x_{ij}^k denotes whether the k-th resource of vehicle A_i can process the task T_j.

D. Task Processing Time

When multiple vehicles are working with their assigned tasks, due to the occurrence of emergent tasks, the running vehicles which have the available remaining resources and time of each resource will collaborate to process the emergent task. In this case, a multi-vehicle alliance is formed to jointly complete the processing of an emergent task. When the multi-vehicles jointly process a task, it is required that the vehicles in the respective vehicle alliance can achieve the same idle time and meet the resource requirements of the emergent task. That is to say, each vehicle has appropriate idle resources in the current time period, which can meet the task requirements. Due to the current operating state of a vehicle, and the devices may be heterogeneous, the remaining resources and time attributes of the vehicles and devices that make up the alliance are different. This requires the time and resource coordination for each vehicle and device. Therefore, the vehicles with matching resources will have a waiting time, and when the resources and time meet the conditions, the vehicle alliance will be formed to coordinate the tasks.

In what follows, we explain the multi-vehicles cooperation problem from the time dimension. As shown in Fig. 2, each vehicle has a time sequence to process tasks. Each task in vehicles is processed sequentially. We assume $(t_{j-1,i}, t_{j,0})$ is the adjustment time the vehicle resets to process the task T_j from the previous task T_{j-1}, and $(t_{j,1}, t_{j,2})$ is the processing time of the next task T_j. Then, the idle time window of a vehicle is $(t_{j,0}, t_{j,1})$, i.e., the vehicle is available between task $t_{j,0}$ and task $t_{j,1}$.

Let us first illustrate the process of collaborative task processing for multiple vehicles using the idle time window. As shown in Fig. 3, we insert task T_3 before task T_0, and insert task T_4 between task T_1 and T_2. Then, we can see that multi-vehicles in IoV can process more tasks than the one shown in Fig. 2

Let us define, at the moment, the new task that the vehicle can process is t_k^f, which is the starting time of an idle time window. According to different situations of vehicles, there are three types of starting time of an idle time window, as shown in Fig. 4. Let T_0 denote the next processing task, T_j be the new arrival task, and θ represent the current situation of vehicles. As shown in Fig. 4(a), when there is no previous task assigned to the current vehicle, we can assign the emergent task T_j to it. The starting time of task T_j is shown in Equation 10:

$$t_k^j = t_k^{cur-j} + t^{cur} \tag{10}$$

where t^cur denotes the current time, t_k^{cur-j} represents the waiting time that the vehicle A_k is assigned to process task T_j.

In Fig. 4(b), the initial waiting time of a vehicle to process a task is T_0, i.e., the vehicle can process T_j before T_0. Then, the task T_0 is assigned to the vehicle before the task T_j is in the initial waiting time. The starting time of task T_j is the same as Equation (10). The starting time of T_0 is changed because of inserting the task T_j before it, and it is shown in Equation (11).

$$t_k^{0'} = t_k^{jwait} + t_k^{j-0} + t_k^j \tag{11}$$

where t_k^{jwait} is the waiting time of a vehicle processing task T_j, and t_k^{j-0} denotes the adjustment time of processing task T_j to task T_0.

In Fig. 4(c), the waiting time of a vehicle is T_0, and the processing time of the task T_j is longer than the current

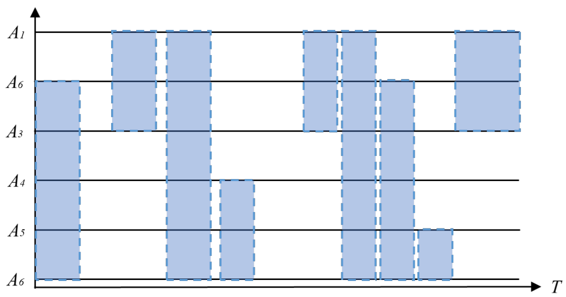


Fig. 5. Idle time window of multi-vehicles collaboration

waiting time. Then, the starting time of a vehicle to process task T_j is shown in Equation (12):

$$t_k^j = t_k^{cur-0} + t_k^{0_{wait}} + t_k^{0-j} + t^{cur} \tag{12}$$

Let us define the waiting time between the finish time of task T_j and the new task as the termination time of the idle time window. Let $t_k^{0_{process}}$ be the initial appointment time of assigning the vehicle to task T_0. Then, the finish time of the idle time window $t_k^{j'}$ is shown in Equation (13):

$$t_k^{j'} = t_k^{0_{process}} - t_k^{j-0} \tag{13}$$

With Equations (10) - (13), we can calculate the start time and termination time of processing an emergent task. Each vehicle decides if inserting an emergent task to the current task list according to the time constraint shown as follows. When the k-th vehicle is assigned a new task T_j, the starting time of this new task, t_k^j , needs to satisfy Equation (14) :

$$t_k^{pre} + t_k^{pre-j} \leq t_k^j \leq t_k^{follow} - t_k^{j-follow} \tag{14}$$

where t_k^{pre} and t_k^{follow} are the starting time of the previous task and the following task, with respect to the new task T_j.

When multiple vehicles collaborate to process an emergent task, the idle time window of multi-vehicles and the resource demand of the emergent task should be satisfied. As shown in Fig. 5, the multiple vehicles with the idle time window overlapped can be collaborated to process the emergent task. Let us use w_i to denote the idle time window of multi-vehicles collaboration A_i. The idle time window should satisfy the following constraint shown in Equation (15).

$$\forall A_i, A_j \in I_A, w_i \cap w_j \neq \emptyset \tag{15}$$

E. Multi-vehicles Collaboration Model

After modelling the multi-vehicles collaboration of task processing issues mentioned above, we evaluate the multi-vehicles collaboration in three aspects: a) maximizing the total profit of processing tasks by multi-vehicles, b) when an emergent task occurs, the real-time requirement is another evaluation parameter, and c) minimizing the number of vehicles in multi-vehicles collaboration. In other words, using less number of vehicles in multi-vehicles collaboration, there will be more concentrations on the resources of the involved vehicles, which will be more conducive to improving resource utilization.

Because the starting time of multi-vehicles collaboration is determined by the start time of previous idle time window T'_A, and the total number of vehicles in the multi-vehicles collaboration N_A, we use the total waiting time of multi-vehicles collaboration and the total number of vehicles in the collaboration to indicate the timeliness of task processing. With regard to the *Scenario 1* of Section VI.B, where the vehicle is constrained by its own resource limit, combined with the total profit of processing tasks, the objective function of multi-vehicles task assignment collaboration is shown by Equation (16).

$$\max \frac{\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ij}^k C_{ij}^k}{\sum_{i=1}^{n} x_{ij}^k v T_A}$$
$$\text{s.t.} \begin{cases} x_{ij}^k r_{ij}^k \leq R_l^k, \forall k \in \{1,2,\dots,m\}, \\ \sum_{i=1}^{n} x_{ij}^k r_i^k \geq R_j^k, \forall k \in \{1,2,\dots,m\}, \\ w_A = \bigcap_{i=1}^{n} w_i \neq \emptyset, \forall A_i \in I_A, \\ T_A = \sup(\min w_A) \end{cases} \tag{16}$$

As for the *Scenario 2* of Section VI.B, where the vehicle is not constrained by its own resource limit, the objective function of multi-vehicles task assignment collaboration is shown by Equation (17).

$$\max \frac{\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ij}^k C_{ij}^k}{\sum_{i=1}^{n} x_{ij}^k v T_A}$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} x_{ij}^k r_i^k \geq R_j^k, \forall k \in \{1,2,\dots,m\}, \\ w_A = \bigcap_{i=1}^{n} w_i \neq \emptyset, \forall A_i \in I_A, \\ T_A = \sup(\min w_A) \end{cases} \tag{17}$$

Since the time window constraint has been introduced, the existing integer programming method is not suitable for solving the model. In what follows, a new method is designed to solve the above model.

VII. The Proposed Multi-vehicles Collaboration Method

We propose a multi-vehicle time-coordinated task assignment method to solve the above-mentioned time window-based multi-vehicles task assignment problem model. First, when a vehicle among multiple vehicles finds an emergent task, it immediately broadcasts the set of tasks and the previously assigned tokens within the current vehicles. The role of the token is to determine that the vehicle in the network reserves the token, and the vehicle that does not want to be networked automatically discards the token and broadcast information. The purpose of this is to ensure that the competition of multi-vehicle resources is avoided on the basis of task broadcast in the whole network, avoiding the deadlock situation of resources and the waste of resources. After the vehicle A_i gets a token and obtains the team qualification, the task T_j is auctioned in the format $R_j = \{A_j, T_j\}$. Then, the other vehicles in the IoV calculate the time window $w_j = \{w_j^1, w_j^2, \dots, w_j^m\}$ after inserting the task T_j into the current task list $E_j = \{E_j^1, E_j^2, \dots, E_j^m\}$ according to Equations (12) - (16). Then, we calculate the profit $C_j = \{C_j^1, C_j^2, \dots, C_j^m\}$

after processing the task T_j using Equations (7) - (9). After that, we provide the time window and profit results to vehicle A_i. The vehicle A_i calculates the best collaboration, based on Equation (16), to process task T_j. It then sends the message to the vehicles of this collaboration. They insert task T_j to their task lists and process it accordingly.

The task request initiator adopts the time window based multi-vehicles collaboration method to form a set of vehicles to process tasks. In order to eliminate the influence of time window overlap on the problem solving, the task request initiator first sorts the vehicles in ascending order according to the start time of the vehicle's idle time window to form a set of candidate vehicles. Then, the requirements of the resources are matched according to the task. Once the resources required by the current task are met, the candidate vehicle set is formed. It is then possible to determine the earliest start time of a vehicle in this set to perform the task processing. Therefore, the time window constraint in the objective function is removed, and the model is transformed into a standard integer programming problem. The optimal set of vehicle candidates can be then readily solved.

Let d_i denote the resource contribution degree of a vehicle. The vehicle, in the candidate set, with the biggest contribution is selected using a greedy algorithm according to d_i, thereby forming a final task processing set. d_i can be expressed by the contribution ratio of the resources that a vehicle i can provide to process the target task.

$$d_j = \sum_{j=0}^{l} w^j \frac{D_i^j}{R_A^j} \tag{18}$$

where w^j represents the weight of the j-th resource, R_A^j denotes the number of j-th resources that the candidate set is still missing under the condition that the existing vehicle has been added, and D_i^j is the j-th resource of the vehicle i to contribute to task processing.

$$D_i^j = \begin{cases} R_i^j, R_i^j \leq R_A^j \\ R_A^j, \text{otherwise} \end{cases} \tag{19}$$

When optimizing the solution, we need to keep the last vehicle in the candidate set added. Otherwise, the final task collaboration will not meet the resources required by the target task.

Algorithm 1 The TWA time-window based algorithm

Input: Candidate vehicle set I_A
Output: Multi-vehicles collaboration task processing set I_C

Initial candidate vehicle set $I_B \leftarrow \emptyset$
if $w_i < w_j, \forall A_k \in I_A$ **then**
  select A_j to form the preparation set I_B
  delete A_j from I_A
end if
calculate v_n the number of vehicles in I_A
for $i_n = 1$ to v_n **do**
  if $\sum_{i=1}^{n} x_{ij}^k r_i^k \geq R_j^k, \forall k \in \{1, 2, \ldots, m\}$ **then**
    break
  end if
  if $w_a < w_k, \forall A_k \in I_A$ **then**
    if $\forall A_j \in I_A, W_k \cap W_j \neq \emptyset$ **then**
      add A_a to preparation set I_B
    end if
  end if
end for
if I_B contains only A_j or $\sum_{i=1}^{n} x_{ij}^k r_i^k \leq R_j^k, \forall k \in \{1, 2, \ldots, m\}$ **then**
  the collaboration fails
  a new vehicle is added to I_A
  goto 7
end if
initialize the task collaboration set $I_C \leftarrow \emptyset$
select last added A_j in I_B to task set I_C
if $\sum_{i=1}^{n} x_{ij}^k r_i^k \geq R_j^k, \forall k \in \{1, 2, \ldots, m\}$ **then**
  goto 33
else
  while $\sum_{i=1}^{n} x_{ij}^k r_i^k \leq R_j^k, \forall k \in \{1, 2, \ldots, m\}$ **do**
    if $d_a < d_k, \forall A_a \in I_B$ **then**
      add A_a to task set I_C
    end if
  end while
end if
return I_C

VIII. Experimental Results and Analysis

In this section, the proposed solution is validated through extensive experiments using a simulator developed in MATLAB. To show the merit of our method of adding critical resource constraints as discussed in Section V, we compare it with the related work in [12]. In addition, the efficiency of the proposed time window based mechanism is validated and evaluated.

A. Simulation Setup

Initially, the vehicle number, task number and resource types are initialized. In the default scenario there are 5 types of resource types, and the number of vehicles ranges from 100 to 1000 as compared to [12], while we study the effect of the number of vehicles, the number of tasks, and the benefits introduced by our model. It is worth noting that the number of tasks indicates the number of tasks allocated in a general task scheduling. Then, an emergent task is assigned to vehicles at the time duration with no task occupying, according to Equations (6) - (10). The tasks are represented by a tuple with its start time and duration. After assigning all the tasks to vehicles using Equations (2) - (4), if an emergent task appears, we assign it to the vehicles. The resources for each individual vehicle follow a uniform distribution in [1, 20]. Besides, whether the resources of each vehicle can satisfy a certain resource requirement is generated. We randomly generate an emergent task represented by its start time, finish time and resource demand. Then, we use the proposed algorithm to find the optimal vehicle set.

Each experiment is obtained by repeatedly running 500 times and being averaged. The upper limit of each treatment time is 1000. The unit processing speed of a vehicle type

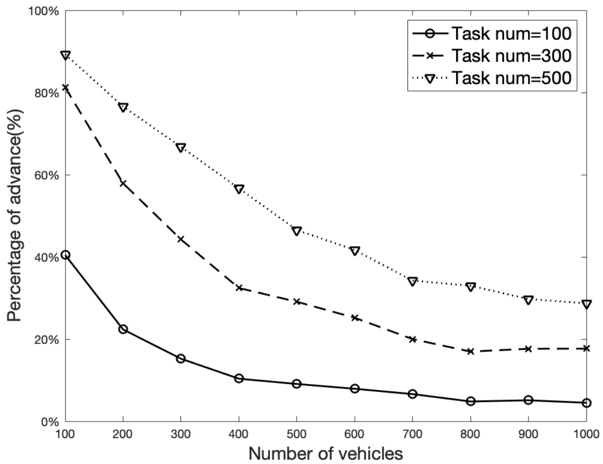


Fig. 6. The percentage of time advance with regard to the number of vehicles and the number of tasks

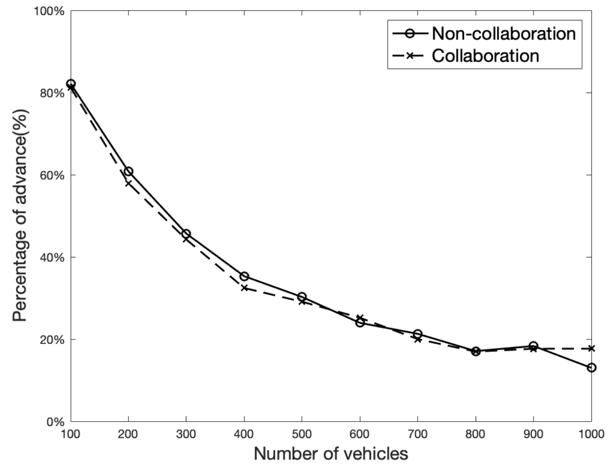


Fig. 7. The percentage of time advance of collaboration and non-collaboration vehicles

is the value uniformly distributed within the range of [10, 90], and the processing speed of each vehicle is different. In addition, the resource requirements and time points of each vehicle and each emergent task are randomly generated, and the generation curves of the respective resource demand are subject to a normal distribution.

B. Experimental Results and Analysis

1) Time Advance: We evaluate the effectiveness of our collaboration mechanism in two ways. Firstly, we demonstrate that the proposed method can save time comparing to the scheduling mechanisms proposed in [12], [17], [19]. With their models, the emergent task can only be added to the end of the task queue because their models either can only process a complete task or do not consider the case of idle resources while carrying out a general task. The time saved can be measured with regard to how much time earlier an emergent task can be handled by the proposed mechanism. We use an index l called the percentage of time advance as shown in Equation (20), where $t_{emergentStart}$ is the start processing time of an emergent task, and t_{finish} is the finish time of the last task for each vehicle, so that we can measure how much time saved.

$$l = \frac{t_{emergentStart}}{t_{finish}} \tag{20}$$

As shown in Fig. 6, regardless of the number of tasks, the percentage of time advance is decreasing as the number of vehicles increases. This is because for a fixed number of tasks, if there are more vehicles, due to the setting that tasks are assigned randomly to vehicles, each vehicle has fewer tasks in a fixed time interval. When each vehicle has only a relatively small number of tasks, the attribute of an emergent task is dimmed, which means that the task is no longer urgent because there is a large probability that the starting time of the emergent task is located in the idle time duration of a vehicle. On the contrary, if the number of tasks is fixed, for a small number of vehicles, each vehicle has a larger number of general tasks assigned. Therefore, if the collaboration mechanism is not used, the vehicle has to wait until all their tasks finish in order to process this task, which takes longer time, so the percentage of advance is significant. We can also obtain that when the number of vehicles is fixed, the more tasks there are, the more effective that this mechanism proves to be in the sense that the more time is saved. The reason is similar to the explanation above. For a fixed number of vehicles, if there are less tasks, an emergent task will no longer be urgent, and the time advance is less significant. It is worthy to be noted that when the vehicles are heavily loaded with tasks (e.g., 100 vehicles and 500 tasks), the time advance is nearly 90% which shows that the vehicles make full use of the idle time window.

We also compare the percentage of time advance in collaboration and non-collaboration scenarios, as shown in Fig. 7. The non-collaboration represents that the vehicles' resources are not constrained, and it is possible that the resource of a single vehicle can meet the requirement.

We can intuitively make a conclusion from Fig. 7 of whether vehicles need to collaborate to meet the resource requirement, given that the percentage of time advance is almost the same. This is because even if one vehicle's resource is enough for the requirement, part of the resource may not satisfy the need of the task as explained in Equation (5); there is a parameter x_ij^k which is randomly generated in the experiment to simulate the real-world situation. In addition, the final objective function takes into consideration that the time window needs to be as small as possible, while the time window of a single vehicle processing the task is not necessarily smaller than that of multi-vehicles collaboration.

2) Profit Gain: Secondly, we show that our proposed mechanism is effective because it can bring profits. If an emergent task is handled within a required time interval, the vehicle which has contributed its resources will be rewarded a profit. The profit index shows the profit defined in Equation (8).

As shown in Fig. 8, the profit is increasing when the

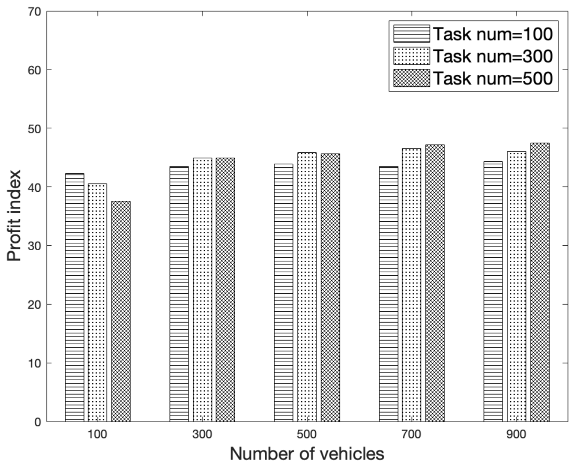


Fig. 8. The profit of different number of vehicles and tasks

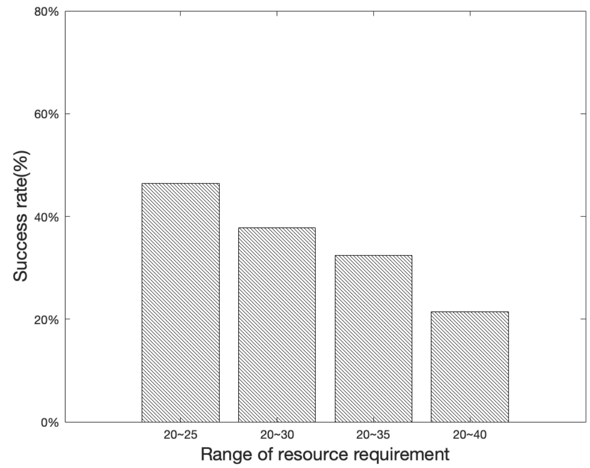


Fig. 10. The success rate of allocating a task

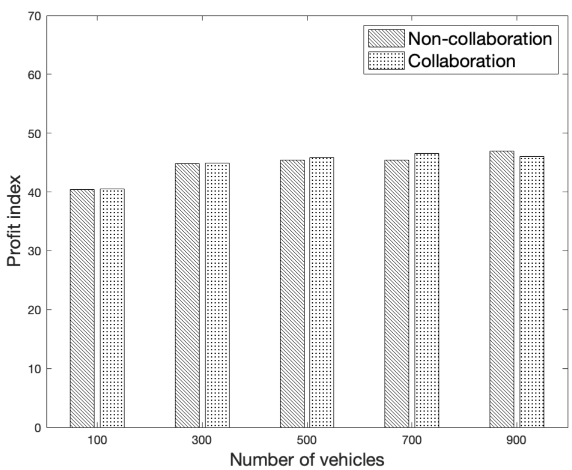


Fig. 9. The profit index of non-collaboration and collaboration vehicles

number of vehicles increase regardless of the number of tasks. However, the changing of the slope of profit with regard to the changes of the number of vehicles is different for different number of tasks. The reason is that the profit increases with the increasing number of vehicles; given a number of tasks, when there are more vehicles, more combinations of vehicles can be formed to the candidate set, because more vehicles will be in an idle state when an emergent task comes.

In Fig. 9, it can be seen that the profit made by vehicles is also irrelevant to whether the vehicle's resources are constrained, and the profit increases as the number of vehicles goes up. The reason is the same as explained above in the percentage of advance between collaboration and non-collaboration vehicles.

3) Successful Rate: We also investigate the success rate of our time-window based algorithm. We make 500 times of random resource generation, and when the number of tasks and the number of vehicles are set to 300 and 500, respectively, in the collaboration scenario we obtain the success rate as shown in Fig. 10. Given that the resources of each vehicle is uniformly distributed in [1, 20], and assume that each individual vehicle's resources cannot satisfy the resource requirement, we choose 4 different ranges of resource requirements. The result is expected, as indicated by the figure, and when the range increases, the success rate decreases. That means it is harder to find a set of vehicles that not only have a relatively high profit but also a small time-window.

4) Performance Improvement: Apart from the aforementioned advantages of using our time-window based method, we compare the efficiency of our model with the online model proposed in [12] in terms of running time. Specifically, we implement their online incentive mechanisms with our experimental settings, where an emergent task appears when the vehicles are carrying out general tasks. The comparison results are shown in Fig. 11.

From the figure 11, we can observe that the running time decreases with the increasing number of vehicles. This is because more vehicles can provide more resources, and it is easier to find idle time window from a larger pool of participants. Besides, our proposed method runs faster because, their methods do not consider the use of idle resources of vehicles, so it is harder for their models to find optimal vehicles to allocate emergent tasks. In addition, our method performs multi-vehicles collaboration while their methods can only accept one vehicle for a task.

IX. CONCLUSIONS

This paper has presented a set of solutions to vehicular mobile crowdsensing in IoV. Repeated general sensing tasks and delay-sensitive emergent tasks have been discussed and defined. An incentive mechanism with time and resource constraints has been proposed to handle the repeated general tasks allocation. A time-window based method has been developed to assign emergent tasks to the vehicles while they are processing general tasks but have idle resources. The method has made full use of the idle time duration between assigned tasks, and it is capable of handling emergent tasks by utilizing these idle resources of multiple vehicles. An objective function

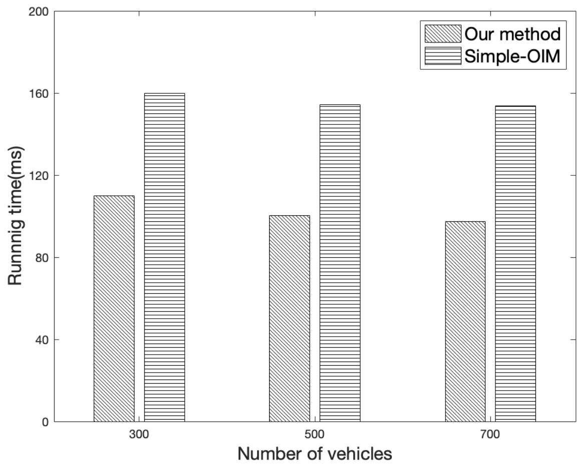


Fig. 11. Performance in terms of running time

aiming to reduce the processing time while increasing the profit generated by the task processing has been proposed, and an algorithm has been designed to sort time window of the available vehicles, so that the objective function can be solved as an integer programming problem. Extensive experimental results have demonstrated the effectiveness of the proposed mechanisms by showing that using our time-window based method, the amount of time of processing emergent tasks can be saved, and more profits can be gained. Finally, a blockchain framework has been proposed so that secured information exchange can be handled among participated vehicles in the mobile crowdsensing network.

Acknowledgment

This work was supported in part by the National Key Research and Development Program under Grant no. 2016YFB1000102, in part by the National Natural Science Foundation of China under Grant no. 61972222, 61672318, and in part by Purple Mountain Laboratory: Networking, Communications and Security in part by the Academician Expert Workstation of Bitvalue Technology (Hunan) Company Limited (2017RS3064).

References

[1] (2019) Top 10 strategic technology trends for 2019. [Online]. Available: Available: http://www.gartner.com

[2] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE world forum on internet of things (WF-IoT)*. IEEE, 2014, pp. 241–246.

[3] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[4] A. Jian, G. Xiaolin, Y. Jianwei, S. Yu, and H. Xin, "Mobile crowd sensing for internet of things: A credible crowdsourcing model in mobile-sense service," in *2015 IEEE International Conference on Multimedia Big Data*. IEEE, 2015, pp. 92–99.

[5] S. K. Datta, R. P. F. Da Costa, C. Bonnet, and J. Härri, "onem2m architecture based iot framework for mobile crowd sensing in smart cities," in *2016 European conference on networks and communications (EuCNC)*. IEEE, 2016, pp. 168–173.

[6] A. Capponi, C. Fiandrino, C. Franck, U. Sorger, D. Kliazovich, and P. Bouvry, "Assessing performance of internet of things-based mobile crowdsensing systems for sensing as a service applications in smart cities," in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2016, pp. 456–459.

[7] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "Prism: platform for remote sensing using smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 63–76.

[8] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 2012, pp. 337–350.

[9] L. Ravindranath, A. Thiagarajan, H. Balakrishnan, and S. Madden, "Code in the air: simplifying sensing and coordination tasks on smartphones," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. ACM, 2012, p. 4.

[10] B. Cao, J. Zhao, P. Yang, P. Yang, X. Liu, and Y. Zhang, "3-d deployment optimization for heterogeneous wireless directional sensor networks on smart city," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1798–1808, 2018.

[11] B. Cao, X. Kang, J. Zhao, P. Yang, Z. Lv, and X. Liu, "Differential evolution-based 3-d directional wireless sensor network deployment optimization," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3594–3605, 2018.

[12] Z. Duan, L. Tian, M. Yan, Z. Cai, Q. Han, and G. Yin, "Practical incentive mechanisms for iot-based mobile crowdsensing systems," *IEEE Access*, vol. 5, pp. 20 383–20 392, 2017.

[13] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, "A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2392–2407, 2016.

[14] Y. Zhan, Y. Xia, Y. Liu, F. Li, and Y. Wang, "Incentive-aware time-sensitive data collection in mobile opportunistic crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 7849–7861, 2017.

[15] X. Wang, W. Wu, and D. Qi, "Mobility-aware participant recruitment for vehicle-based mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4415–4426, 2017.

[16] X. Tao and W. Song, "Location-dependent task allocation for mobile crowdsensing with clustering effect," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1029–1045, 2018.

[17] L. Jiang, X. Niu, J. Xu, Y. Wang, Y. Wu, and L. Xu, "Time-sensitive and sybil-proof incentive mechanisms for mobile crowdsensing via social network," *IEEE Access*, vol. 6, pp. 48 156–48 168, 2018.

[18] H. Ko, S. Pack, and V. C. Leung, "Coverage-guaranteed and energy-efficient participant selection strategy in mobile crowdsensing," *IEEE Internet of Things Journal*, 2018.

[19] J. Xu, J.-R. Fu, D.-J. Yang, L.-J. Xu, L. Wang, and T. Li, "Fimi: A constant frugal incentive mechanism for time window coverage in mobile crowdsensing," *Journal of Computer Science and Technology*, vol. 32, no. 5, pp. 919–935, 2017.

[20] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2018.

[21] Y. Chen, H. Chen, S. Yang, X. Gao, Y. Guo, and F. Wu, "Designing incentive mechanisms for mobile crowdsensing with intermediaries," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[22] J. Nie, Z. Xiong, D. Niyato, P. Wang, and J. Luo, "A socially-aware incentive mechanism for mobile crowdsensing service market," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[23] J. Cui, X. Zhang, H. Zhong, Z. Ying, and L. Liu, "Rsma: Reputation system-based lightweight message authentication framework and protocol for 5g-enabled vehicular networks," *IEEE Internet of Things Journal*.

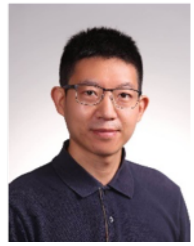
[24] L. Cheng, J. Liu, G. Xu, Z. Zhang, H. Wang, H. N. Dai, Y. Wu, and W. Wang, "Sctsc: A semicentralized traffic signal control mode with attribute-based blockchain in iovs," *IEEE Transactions on Computational Social Systems*.

[25] M. Iansiti and K. R. Lakhani, "The truth about blockchain," *Harvard Business Review*, vol. 95, no. 1, pp. 118–127, 2017.

[26] L. Mendiboure, M. A. Chalouf, and F. Krief, "Towards a blockchain-based sd-iov for applications authentication and trust management," in *International Conference on Internet of Vehicles*. Springer, 2018, pp. 265–277.

[27] M. Singh and S. Kim, "Branch based blockchain technology in intelligent vehicle," *Computer Networks*, vol. 145, pp. 219–231, 2018.
[28] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," *Mobile networks and Applications*, vol. 10, no. 4, pp. 519–528, 2005.
[29] Y. Liu, X. Weng, J. Wan, X. Yue, H. Song, and A. V. Vasilakos, "Exploring data validity in transportation systems for smart cities," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 26–33, 2017.
[30] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based iot: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.
[31] W. Feng and Z. Yan, "Mcs-chain: Decentralized and trustworthy mobile crowdsourcing based on blockchain," *Future Generation Computer Systems*, vol. 95, pp. 649–666, 2019.
[32] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui, "Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2018, pp. 442–450.
[33] K. Zhao, S. Tang, B. Zhao, and Y. Wu, "Dynamic and privacy-preserving reputation management for blockchain-based mobile crowdsensing," *IEEE Access*, vol. 7, pp. 74 694–74 710, 2019.
[34] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[35] M. E. Peck, "Blockchains: How they work and why they'll change the world," *IEEE spectrum*, vol. 54, no. 10, pp. 26–35, 2017.
[36] N. Rifi, N. Agoulmine, N. Chendeb Taher, and E. Rachkidi, "Blockchain technology: Is it a good candidate for securing iot sensitive medical data?" *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
[37] R. Chamarajnagar and A. Ashok, "Opportunistic mobile iot with blockchain based collaboration," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.
[38] M. Swan, *Blockchain: Blueprint for a new economy*. " O'Reilly Media, Inc.", 2015.
[39] Y. Hao, D. Guo, W. Kai, Z. Jiang, Y. Lyu, and X. Ju, "Hyperconnected network: A decentralized trusted computing and networking paradigm," *IEEE Network*, vol. 32, no. 1, pp. 112–117, 2018.
[40] W. Kai, Y. Hao, Q. Wei, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.
[41] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
[42] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. ACM, 2010, pp. 105–116.
[43] F. Bouhafs, M. Merabti, and H. Mokhtar, "The bikenet mobile sensing system for cyclist experience mapping," *SenSys. ACM, New york*, pp. 87–101, 2007.
[44] M. Alam, B. Fernandes, L. Silva, A. Khan, and J. Ferreira, "Implementation and analysis of traffic safety protocols based on etsi standard," in *2015 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2015, pp. 143–150.
[45] T. Ning, Z. Yang, X. Xie, and H. Wu, "Incentive-aware data dissemination in delay-tolerant mobile networks," in *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2011, pp. 539–547.
[46] T. Ning, Z. Yang, H. Wu, and Z. Han, "Self-interest-driven incentives for ad dissemination in autonomous mobile social networks," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2310–2318.

Bo Yin received her Ph.D in Beijing University of Posts and Telecommunications 2013. She works as a post-doctoral in RITT Tsinghua university,Beijing, China.Her research interests include cloud computing,Blockchain,wireless networks,big data,and software-defined networking



Yulei Wu is a Senior Lecturer in the Department of Computer Science with the University of Exeter, United Kingdom. He received the Ph.D. degree in Computing and Mathematics and the B.Sc. degree (1st Class Hons.) in Computer Science from the University of Bradford, United Kingdom, in 2010 and 2006, respectively. His expertise is on networking and his main research interests include intelligent networking technologies, network slicing and softwarization, future Internet architecture and technologies, Green networking, wireless networks, network security and privacy, and analytical modelling and performance optimization. He is an Editor of IEEE Transactions on Network and Service Management, Elsevier Computer Networks and IEEE Access. He contributes to major conferences on networking as various roles including a Steering Committee Chair, a General Chair, a Program Chair, and a Technical Program Committee Member. His research has been supported by Engineering and Physical Sciences Research Council (EPSRC) of United Kingdom, National Natural Science Foundation of China, University's Innovation Platform and industry. He is a Senior Member of the IEEE, and a Fellow of the HEA (Higher Education Academy).

Tianshi Hu is currently pursuing the B.S. degree in Electrical Engineering from Beijing University of Posts and Telecommunications. His research interests include blockchain, iov and cloud computing.

Jiaqing Dong received the B.S. degree in computer science from Peking University, Beijing, China, in 2013. He is currently pursuing the Ph.D. degree in Department of Computer Science at Tsinghua University, Beijing, China. His research interests include data center networks and distributed systems.

Zexun Jiang received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 2014. He is currently pursuing the Ph.D. degree in Department of Computer Science at Tsinghua University. His research interests include network measurement and information retrieval.