

Blockchain-Based Vehicular Collaborative Computing

Qibin Zheng¹, Rongqing Zhang^{1,2,*}, Shengjie Zhao^{1,*}
{zhengqb_2018@tongji.edu.cn, rongqingz@tongji.edu.cn, shengjiezha@tongji.edu.cn}

School of Software Engineering, Tongji University, Shanghai 201804, China¹
Shandong Provincial Key Laboratory of Wireless Communication Technologies, Shandong University, China²

Abstract. Collaborative computing among vehicles on the road has been regarded as a promising application for connected and intelligent vehicles as new computing and communication technologies emerge. Instead of using cloud computing services, a vehicle can complete computing-intensive tasks by making use of neighboring vehicles' idle computing resources and meanwhile, pay for this service to the assisting vehicles to increase their incentive to participate in such collaborative computing tasks. In collaborative computing among vehicles, how to balance the profits of both sides need to be investigated. What's more, the reliability of transactions is also a concerning problem. In this paper, we first introduce a collaborative computing scenario and propose a two-layer blockchain architecture to enhance the reliability of transactions. Then, we propose a coalition formation game-based collaborative computing algorithm to efficiently achieve effective computing coalition formation in a distributed manner. Simulation results show that the proposed algorithm can significantly improve the vehicles' utilities as well as the task performance.

Keywords: Internet of vehicles, Blockchain, collaborative computing.

1 Introduction

With the rapid development of computing and communication technologies, vehicles become more intelligent and are expected to support a rich and varied onboard applications, such as real-time navigation and vehicular entertainment. On the one hand, vehicles can get a precise and broad perception of their driving environment by the growing links with neighboring vehicles (vehicle-to-vehicle (V2V) links) or roadside units (vehicle-to-infrastructure (V2I) links). On the other hand, vehicles equipped with a more powerful on board computing unit (OBU) can perform independent computation better than before. In addition, a recent report shows that the global number of connected vehicles is increasing rapidly, and more than 286 million connected vehicles will be added globally during the 2019-2025 period [1]. These trends reflect the growing intelligence and complexity of vehicles.

Although the vehicles become more capable of computing, some emerging vehicular applications, such as augmented reality (AR) and virtual reality (VR), require completing a large amount of computations in a relatively short time, which cannot be supported by current computing ability of a single vehicle. One promising solution to this problem is to deliver the computation-intensive task from the vehicle (called task requesting vehicle) to other objects

* Corresponding authors: Rongqing Zhang and Shengjie Zhao.

with higher computing ability through the network to achieve higher performance. The target objects can be a cloud center [2], [3]. The cloud center has strong computing ability and can effectively reduce the computing overhead of requesting vehicles, but the long distance to a far-away center will also cause an extra delay. Another kind of target object is the roadside unit (RSU) or base station along with edge server which can complete the computation near to requesting vehicles [4], [5].

Different from the above two cases, a new solution tries to make use of idle resources of other vehicles to complete the task. That is, the vehicles around the requesting vehicle accept the task together and perform the computation in a collaborative computing way [6]-[10]. In [7], a learning-based task offloading framework was proposed using the multi-armed bandit (MAB) theory, which enables vehicles to learn the potential task offloading performance of its neighboring vehicles with excessive computing resources and minimizes the average offloading delay. In [8], a vehicular cloud network architecture was proposed to offload the computational burden of a centralized cloud by assigning more tasks to vehicular clouds formed near traffic lights. While [7], [8] both only considered the task performance (eg. latency, cost) on behalf of the requesting vehicle, [9], [10] further considered the utilization of neighboring vehicles. An optimal computation resource allocation scheme was proposed in [9] to maximize the total long-term expected reward of the overall vehicles. In [10], a novel computation offloading marketplace was established in vehicular networks, where a Vickrey–Clarke–Groves based reverse auction mechanism was exploited, and a unilateral-matching-based algorithm was proposed to implement polynomial computational complexity.

However, these literature work only consider one side's interest, lacking in making a tradeoff between task performance and utility of neighboring vehicles. What's more, the increasing connectivity and decentralization of vehicular networks pose new challenges to the reliability of transactions when vehicles make deals with each other for tasks. It can be possible risky to make deal with strange vehicles. Blockchain is a promising technology to guarantee transaction reliability among multiple agents in a decentralized manner. To overcome the above challenges, we adopt the blockchain technology to create a reliable transaction mechanism for collaborative computing process.

The main contribution of this paper is to propose a two-layer vehicular blockchain architecture, which enables the neighboring vehicles to store their transaction records in both temporary and permanent manners to enhance the transaction reliability. What's more, to balance both sides' interests, we further employ the coalition formation game to model such a collaborative computing process and propose a distributed computing coalition formation algorithm of the vehicles that have the incentive to participate in collaborative computing. Simulations results verify the efficiency of our proposed scheme compared with other baselines in terms of utility and computing performance.

The rest of this paper is organized as follows: Section 2 presents the system model. In Section 3, we introduce the coalition formation game concepts and then provide the proposed algorithm. Section 4 presents the simulation results. Finally, the conclusion is drawn in Section 5.

2 System model

2.1 System architecture

As shown in **Figure 1**, the system architecture consists of the vehicular layer and the edge layer. The vehicular layer contains vehicles along the road. These vehicles equipped with OBU can perform specific computations and communicate with their neighboring vehicles through V2V communication links. The edge layer contains two types of nodes that are placed at the edge of the network compared to the cloud center: roadside units (RSUs) and base stations. The edge nodes have the stronger computing power and more storage resources than the vehicles. They communicate with the vehicles in the vehicular layer through V2I links, which can transfer and store the data generated during vehicle driving and interaction.

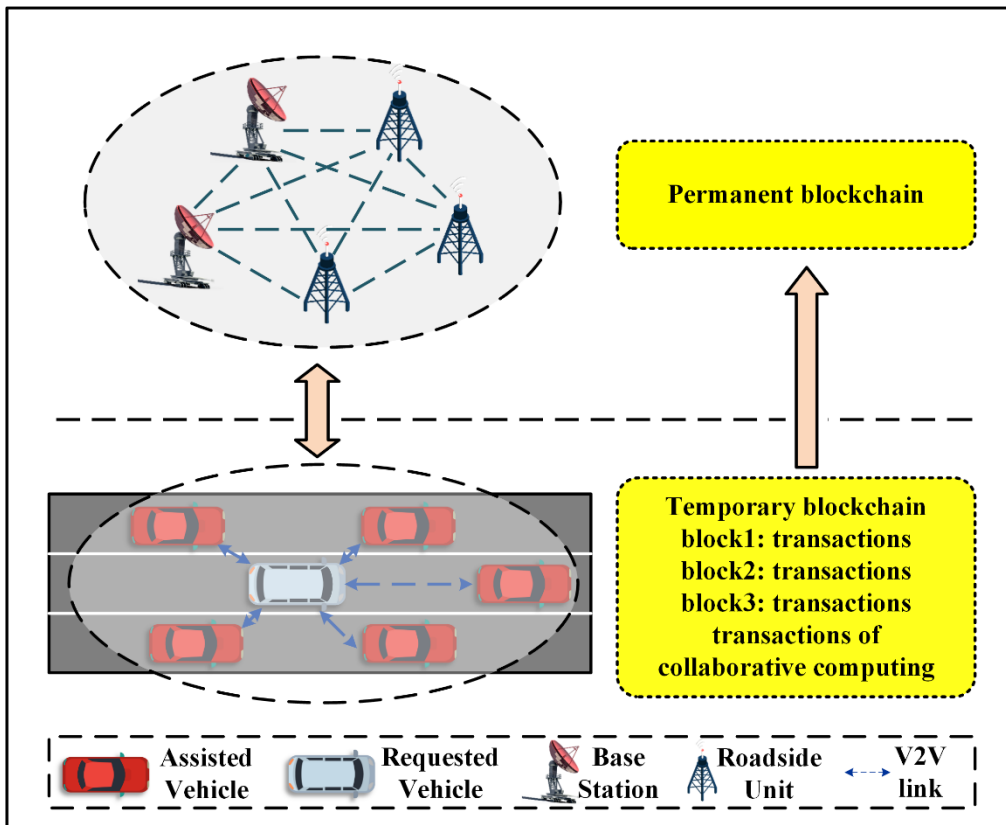


Fig.1. System architecture.

Since the vehicles also have computing capability, it is a feasible and efficient mechanism to complete a large computing task through collaborative computing among multiple vehicles. Therefore, the vehicles in the vehicular layer are further divided into two types in terms of different roles: requesting vehicles and assisting vehicles. The requesting vehicles issue computing tasks, whereas the assisting vehicles form coalitions to complete computing tasks in a cooperative manner. To ensure the reliability of transactions, a small scale blockchain network will be established by the formed vehicle coalition to record transaction data of the collaborative computing process. The data recorded on the blockchain include the payment of the requested

task, the vehicle execution records of computing results, and the transactions of payment allocation.

After collaborative computing is completed, the vehicle group is dissolved, and the generated blockchain data is uploaded to the edge layer. Nodes at the edge layer also establish a blockchain network to permanently store the transactions.

Another class of data recorded in the permanent blockchain is the trust value of vehicles. Trust value reflects the credit conditions of vehicles during a certain period and changes based on vehicles' activities over time. To enhance the reliability, requesting vehicles can set a trust value threshold for their tasks, which makes assisting vehicles ignore the ones with low trust value during computing coalition formation.

2.2 Computing model

In this paper, we consider a collaborative computing scenario in vehicular networks, where there is a requesting vehicle (denoted as v_0) on the road, and N nearby vehicles around v_0 in its communication range, denoted as the set $\mathcal{N} = \{v_1, v_2, \dots, v_N\}$. All these vehicles can communicate through V2V links. When v_0 needs to complete a computing task $Task_{v_0}$, v_0 sends the request message to the vehicles in the set \mathcal{N} . The content of the request message is represented as $\{A_{v_0}, T_{v_0}, R_{v_0}\}$, where A_{v_0} is the computing amount of $Task_{v_0}$, T_{v_0} is the completion time limit of $Task_{v_0}$, and R_{v_0} is the reward payment for the completion of the task. It is assumed that due to the limitations of A_{v_0} and T_{v_0} , a single vehicle in \mathcal{N} usually cannot complete the whole task alone. To complete the task in time, the minimum computing power (denoted as x_0) is:

$$x_0 = \frac{A_{v_0}}{T_{v_0}}. \quad (1)$$

After receiving the message, each vehicle in \mathcal{N} starts to find nearby vehicles to form the coalition for completing the task $Task_{v_0}$ in a collaborative manner. Let $S \subset \mathcal{N}$ be a coalition consisting of $|S|$ vehicles. Assume the vehicle $v_i \in \mathcal{N}$ has the computing power x_i , and the total computing power of vehicles in S is given by x_S :

$$x_S = \sum_{v_i \in S} x_i. \quad (2)$$

In order to complete the task in a shorter time, the task payment R_{v_0} is designed as a function of x_S :

$$R_{v_0} = \begin{cases} b \cdot R_0 & x_S > b \cdot x_0 \\ \frac{x_S}{x_0} \cdot R_0 & x_S \leq b \cdot x_0 \end{cases}. \quad (3)$$

where R_0 is the given payment when S spends T_{v_0} to complete the task (that is, $x_S = x_0$). When $x_S > x_0$, the payment keeps increasing until reaching the maximum payment $b \cdot R_0$. To earn a considerable profit, the vehicles in \mathcal{N} will select suitable partners to form the coalition. The profit measurement and the method of selecting will be discussed further in section III.

Denoted $P = \{S_1, S_2, \dots, S_k\}$ is the set consisting of several disjoint coalitions formed by the vehicles in \mathcal{N} . v_0 will select the appropriate coalition (or coalitions) from P and delegate

the task to it. The selected coalition starts the collaborative computing and records related data in the temporary blockchain. When the computing task is completed, the selected coalition will allocate the payment R_{v_0} to the member vehicles through the blockchain network. The selected coalition uploads the data on the temporary blockchain to the permanent blockchain network at the edge layer for preservation periodically or before dissolution.

3 Computing coalition formation algorithm

To realize efficient coalition formation in a distributed manner, we employ the coalition formation game[11] to obtain an effective solution to the above problems, which can not only meet the computing requirements of v_0 , but also encourage the vehicles to participate in collaborative computing and gain a considerable profit.

3.1 Coalition game concept

Definition 1: *Transferable utility $v(S)$.*

$u(S)$: $u(S)$ is the revenue of the coalition S in the game. It represents the reward the whole coalition will get by forming this coalition.

$c(S)$: $c(S)$ is the cost of the coalition S in the game. Coalition formation game theory considers the presence of cost in the game. In such an assumption, the coalition will cause some kind of cost during the forming operation, e.g., the cost of exchanging information between members.

$v(S)$: $v(S)$ is the difference between the revenue and the cost of a coalition which is: $v(S) = u(S) - c(S)$. $v(S)$ is a real value that can be divided in any manner between the coalition members. We will define $v(S)$ for the collaborative computing scenario in section III-B.

Definition 2: *Coalition formation game for the vehicular collaborative computing problem.* A coalition formation game for the vehicular collaborative computing problem is defined by the pair (\mathcal{N}, v) which involves a set of assisting vehicles, denoted by $\mathcal{N} = \{1, \dots, N\}$ who seek to form cooperative computing coalitions to compute the tasks. And v or $v(S)$ is the profit of the coalition S .

Definition 3: A coalition formation game (\mathcal{N}, v) with transferable utility is *superadditive* if for any two disjoint coalitions $S_1, S_2 \subset \mathcal{N}$, $v(S_1 \cup S_2) \geq v(S_1) + v(S_2)$.

Theorem 1: The coalition formation game for the collaborative computing scenario is not superadditive.

Proof: According to the definition of $v(S)$ in section III-B, if the game is superadditive, then the coalition \mathcal{N} containing all the vehicles will obtain the maximum transferable utility. However, due to the second cost item $c_2(S)$ defined in section III-B are growing at second-order of coalition size which is faster than the revenue $u(S)$, so the transferable utility will decrease when coalition size surpasses a certain value (the value won't be big and are reachable in our simulation).

Definition 4: *Coalition partition.* A *partition* is defined as the set $P := \{S_1, \dots, S_k\}$, where $S_k \subseteq \mathcal{N}$ are mutually disjoint coalitions and $\bigcup_{i=1}^k S_i = \mathcal{N}$.

Definition 5: *Preference order.* For any vehicle $v_i \in \mathcal{N}$, we define preference order \triangleright_i as a kind of comparison relation, where $B \triangleright_i A$ means that v_i prefers to join coalition B instead of coalition A . We use the preference order as follow:

$$B \triangleright_i A \Leftrightarrow x_{B'} > x_A \text{ and } v(B') \geq v(B). \quad (4)$$

where $B' = B \cup \{v_i\}$. This definition means vehicle v_i prefers to join coalition B by leaving coalition A when the new computing power of B will be larger than the origin computing power of A , meanwhile the utility of B is not decreasing.

Definition 6: *Switch rule.* We introduce the simple switch rule as the basic operation of our algorithm as follow [12]:

Switch:

$$A, B \rightarrow A \setminus \{v_i\}, B \cup \{v_i\}, \quad (5)$$

where $A \subseteq \mathcal{N}, B \subseteq \mathcal{N} \cup \emptyset, B \neq A, v_i \in A, B \triangleright_i A$.

Definition 7: *Payoff scheme.* The coalition S needs to divide the utility $v(S)$ for its members, and every vehicle v_i get its payoff ϕ_i . We use the following division scheme:

$$\phi_i = \frac{x_i}{x_S} \cdot v(S). \quad (6)$$

Definition 8: *Partition convergence.* A partition P is considered convergence if for any $A \in P, B \in P \cup \emptyset$ and $B \neq A$, no $v_i \in A$ satisfying $B \triangleright_i A$ exists.

3.2 Transferable utility of the computing coalition

In this section, we design the transferable utility mentioned in section III-A for coalition formation game of collaborative computing.

The revenue for forming coalition S is defined as R_{v_0} , that is:

$$u(S) = R_{v_0}. \quad (7)$$

In addition to getting paid, S also generates some overhead, including computing cost and communication cost. The computing cost is defined as:

$$c_1(S) = \sum_{v_i \in S} x_i * T_S * p_i \quad (8)$$

where T_S is the real computing time of S which is defined as:

$$T_S = \begin{cases} \frac{A_{v_0}}{x_S} & \text{if } x_S > x_0 \\ T_{v_0} & \text{otherwise} \end{cases} \quad (9)$$

and p_i is the unit computing cost of vehicle v_i (such as the power consumption per unit time of CPU).

In the process of forming a coalition, vehicles need to transfer messages directly or by using blockchain, which brings communication overhead. The amount of the communication overhead is related to the size of the coalition and the distance among the vehicles. It is represented by $c_2(S)$ and $c_3(S)$, respectively:

$$c_2(S) = \alpha \cdot |S|^2 \quad (10)$$

where α is a weight factor and $|S|$ is the size of the coalition S .

$$c_3(S) = \beta \cdot \max_{v_i, v_j \in S} d(v_i, v_j) \quad (11)$$

where β is a weight factor, and $d(v_i, v_j)$ is the distance between v_i and v_j in coalition S .

From the above formulas (7) - (11), the utility of the coalition is obtained:

$$v(S) = u(S) - c_1(S) - c_2(S) - c_3(S). \quad (12)$$

Due to $c_2(S)$ and $c_3(S)$ increasing when the coalition expands, forming a grand coalition may not be the solution with maximum $v(S)$, and the vehicles in \mathcal{N} will form several disjoint coalitions instead.

3.3 Coalition formation algorithm

Instead of using an optimal centric algorithm, we adopt a distributed algorithm that consists of two phases as shown in Table 1. During the first phase, the vehicles perform switch operations in a random way and form coalitions until no more switch operation can happen, and then the partition P reaches a convergence. In the second phase, v_0 selects the appropriate coalition (or coalitions) from P . In most cases, the coalition with maximum computing power will be chosen to execute the task to minimize the latency. However, sometimes no coalition in P can provide computing power more than x_0 , then v_0 will choose more coalitions to guarantee the sum of computing power to be greater than the required threshold x_0 . In such a case, v_0 needs to divide the whole task into several subtasks and deliver them to different coalitions. Compared with the no-task-division case, v_0 has to coordinate and mix together the work of more coalitions which may bring extra cost and increase the risk for damaging the performance of the task. In the task-division case, the reward for each coalition is calculated with a punishing factor:

$$R_{v_0, S_i}' = \frac{x_{S_i}}{\sum_{S_k \in F} x_{S_k}} \cdot R_{v_0, S_i}. \quad (13)$$

where R_{v_0, S_i} is the origin reward of coalition S_i according to (3). The punishing factor considers the contribution of S_i to the set F . After completing the task, the coalition divides its utility to the vehicles within as described in Definition 7.

4 Simulation results

In our conducted simulations, vehicle v_0 is placed at the center of a 100 m \times 10 m road, and the other vehicles are placed randomly on the road. The number of assisting vehicles N varies from 5 to 15 in the experiment. For each value of N , we run the simulation for 150 times and take the average of the results (utility or computing power described below) as the final value. The other simulation parameters are listed in Table 2.

Table 1. Coalition formation algorithm.

Initialize the vehicles are non-cooperative. The partition of the network is $P = \{v_1, v_2, \dots, v_N\}$.

Phase I - coalition formation:

- 1 **repeat**
- 2 randomly select coalition A, B and v_i , where $A \in P, B \in P \cup \emptyset,$
- 3 $B \neq A, v_i \in A.$
- 4 **if** $B \triangleright_i A$ **then**
- 5 $P = P \setminus \{A, B\} \cup \{A \setminus \{i\}, B \cup \{i\}\}.$
- 6 **until** P reach convergence.
- 7 suppose now $P = \{A_1, A_2, \dots, A_K\}.$

Phase II - coalition selection:

- 8 $\emptyset \rightarrow F.$
- 9 sort P and get $P = \{B_1, B_2, \dots, B_K\},$ where $x_{B_i} > x_{B_j}$ if $i < j.$
- 10 **if** $x_{B_1} > x_0$ **then**
- 11 $\{B_1\} \rightarrow F.$
- 12 **else**
- 13 find k where $\sum_{i=1}^k x_{S_i} \geq x_0$ and $\sum_{i=1}^{k-1} x_{S_i} < x_0.$
- 14 $\{B_1, \dots, B_k\} \rightarrow F.$
- 15 v_0 delegate $Task_{v_0}$ to $F.$
- 16 F completes the task and divide the utility.

Table 2. Simulation parameters.

Parameters	Values
Task Computing Amount A_{v_0}	100 unit
Basic Reward R_{v_0}	200 unit
Maximum Reward Ratio b	1.65
Minimum Task Computing Power x_0	192 unit
Cost Weight for Coalition Size α	3
Cost Weight for Vehicles' Distance β	6
Computing Power of Assisting Vehicle x_i	generated randomly between 48-80 unit
Unit Computing Cost p_i	generated randomly between 0.1-0.15 unit

We will compare our coalition formation game-based algorithm (denoted as CF) with two baseline schemes as follows:

1) *Maximum utility scheme (MU)*. This scheme searches the coalition with maximum utility by using the exhaustive method. However when the number of vehicles is relatively small ($N = 5$ or 6), the coalition with maximum utility can't provide computing power more than v_0 which is similar to our proposed scheme. At this time, v_0 also selects more coalitions beside the one before to fulfill the requirement.

2) *Nearest selection scheme (NS)*. This scheme sorts the vehicles by the distance to v_0 and selects the nearest ones whose total computing power is more than χ_0 . To improve the utility, it will keep adding new vehicles in the selection until the utility of the coalition begins to decrease.

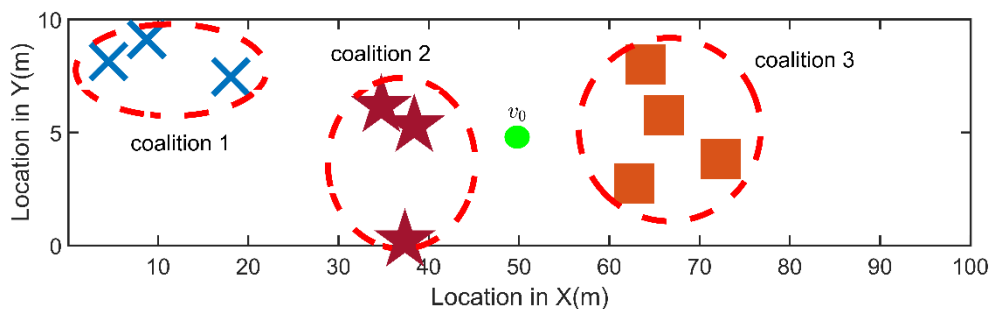


Fig.2. A snapshot example of coalition formation.

Figure 2 shows a snapshot of a network with 10 assisting vehicles. By using the coalition formation game-based algorithm, the vehicles form three coalitions in this example. The utility and computing power are shown in Table 3. v_0 will select coalition 3 because it has the maximum computing power and satisfies the requirement of χ_0 .

Table 3. Coalition parameter result.

	Coalition 1	Coalition 2	Coalition 3
Utility	103.80	109.68	101.37
Computing Power	216.14	173.39	218.25

Figure 3 compares the utilities of the selected coalition(s) generated by three different schemes. From **Figure 3**, we can find that the utility obtained by CF is very close to the one by MU, which indicates that CF can guarantee the assisting vehicles getting well paid from the task. Moreover, we can also see that NS performs much worse than CF and MU. This is because NS forms the coalition in a greedy manner to get enough computing power which includes “join operation” only and does not include “leave operation”.

In **Figure 4**, the computing power of the three schemes are compared. We can find that the computing power of CF and MU are close when N varies from 5 to 10, and then CF begins to perform better than MU. This is because when N is relatively small, CF often finds the optimal coalition with the maximum utility which is the same as MU, and sometimes MU even gets better results by exhaust search (both the utility and computing power are higher) which reduces the difference between CF and MU. But As N keeps growing, CF can perform better by seeking greater computing power at an acceptable cost of utility. It indicates that CF can reach better or at least the same task performance compared with MU. And we also have that NS performs better than CF and MU when N is 5 or 6. This is because NS tries to gather enough computing

power without considering utility at first, and may form the larger coalition which is hard for CF and MU when N is relatively small.

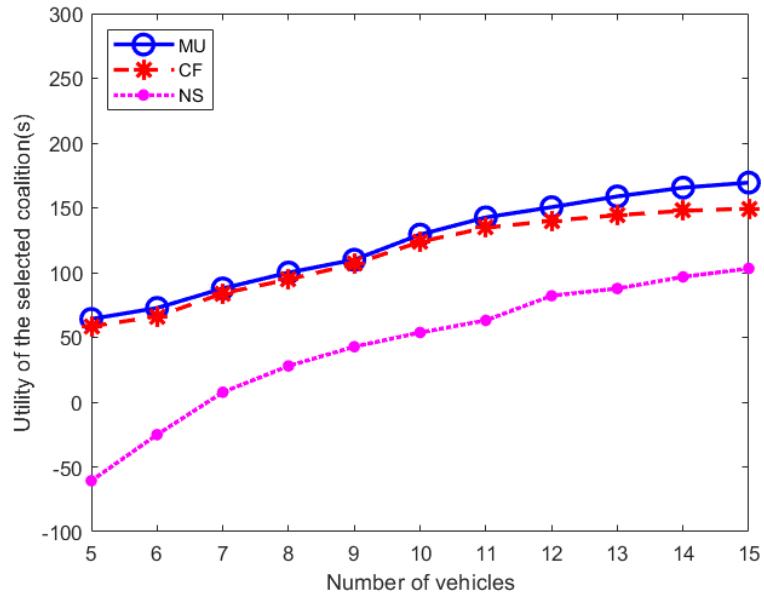


Fig.3. The utility of selected coalition(s).

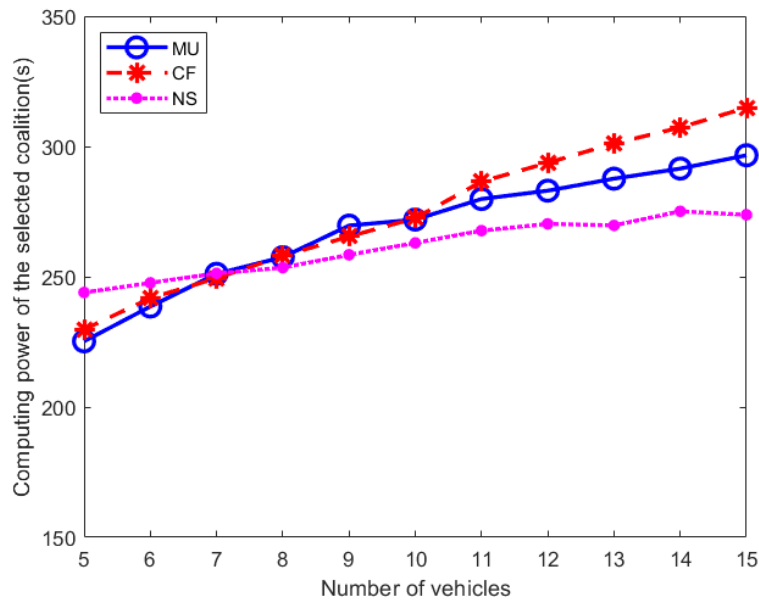


Fig.4. The computing power of selected coalition(s).

5 Conclusions

In this paper, we constructed a two-layer blockchain architecture suitable for reliable collaborative computing among vehicles. We proposed a coalition formation game-based algorithm to seek the appropriate vehicles for collaborative computing, so as to make a tradeoff between task performance and system utility of vehicles. Simulation results show that our algorithm reaches a good performance close to the exhaustive search method.

Acknowledgments. This work is supported in part by the National Key Research and Development Project under Grant 2017YFE0119300, 2019YFB2102300 and 2019YFB2102301, in part by the National Natural Science Foundation of China under Grant 61936014 and 61901302, in part by the Scientific Research Project of Shanghai Science and Technology Committee under Grant 19511103302, in part by the open research fund from Shandong Provincial Key Laboratory of Wireless Communication Technologies (No. SDKLWCT-2019-02), and in part by the Fundamental Research Funds for the Central Universities (China).

References

- [1] Madhok, A.: Global connected car revenues to grow five-fold by 2025. <https://www.counterpointresearch.com/connected-car-revenues-grow-five-fold-2025> (2019)
- [2] Ashok, A.; Steenkiste, P.; Bai, F.: Adaptive cloud offloading for vehicular applications. *Vehicular Networking Conference IEEE*. pp. 1-8 (2016)
- [3] Matzakos, P.; Harri, J.; Villeforceix, B.; *et al.*: An ipv6 architecture for cloud-to-vehicle smart mobility services over heterogeneous vehicular networks. *International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 767–772 (2014)
- [4] Wang, J.; Feng, D.; Zhang, S.; *et al.*: Computation offloading for mobile edge computing enabled vehicular networks. *IEEE Access*. vol. 7, pp. 62624–62632 (2019)
- [5] Zhao, J.; Li, Q.; Gong, Y.; *et al.*: 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)
- [6] Whaiduzzaman, M.; Sookhak, M.; Gani, A.; *et al.*: A survey on vehicular cloud computing. *Journal of Network and Computer applications*. vol. 40, pp. 325–344 (2014)
- [7] Sun, Y.; Guo, X.; Zhou, S.; *et al.*: Learning-based task offloading for vehicular cloud computing systems. *IEEE International Conference on Communications (ICC)*. pp. 1–7 (2018)
- [8] Alahmadi, A. A.; Lawey, A. Q.; El-Gorashi, T. E. H.; *et al.*: Distributed processing in vehicular cloud networks. *International Conference on the Network of the Future (NOF)*. pp. 22–26 (2017)
- [9] Zheng, K.; Meng, H.; Chatzimisios, P.; *et al.*: An SMDP-Based Resource Allocation in Vehicular Cloud Computing Systems. *IEEE Transactions on Industrial Electronics*. vol. 62, no. 12, pp. 7920–7928 (2015)
- [10] Liwang, M.; Dai, S.; Gao, Z.; *et al.*: A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network. *IEEE Internet of Things Journal*. vol. 6, no. 3, pp. 4214–4227 (2019)
- [11] Saad, W.; Han, Z.; Debbah, M.; *et al.*: Coalitional game theory for communication networks. *IEEE Signal Processing Magazine*. vol. 26, no. 5, pp. 77–97 (2009)
- [12] Zhao, N.; Wu, H.; Chen, Y.: Coalition game-based computation resource allocation for wireless blockchain networks. *IEEE Internet of Things Journal*. vol. 6, no. 5, pp. 8507–8518 (2019)