Sharing Incentive Mechanism, Task Assignment and Resource Allocation for Task Offloading in Vehicular Mobile Edge Computing

Tra Huong Thi Le[1], Nguyen H. Tran[2], Yan Kyaw Tun[1], Oanh Tran Thi Kim[1], Kitae Kim[1], Choong Seon Hong[1]
[1] Dept. of Computer Science and Engineering, Kyung Hee University, Yongin, Korea
[2] School of Computer Science, The University of Sydney, Sydney, NSW 2006, Australia
E-mail:{huong_tra25, ykyawtun7, ttkoanh, glideslope, cshong}@khu.ac.kr, nguyen.tran@sydney.edu.au.

***Abstract*—Vehicular Mobile Edge Computing is a promising technology to leverage the bottleneck at a base station (BS) at peak hours. However, to deploy Vehicular Mobile Edge Computing requires to deal with the challenges in how to incentive vehicles to resource sharing and how to assign tasks and computation resource to minimize the total network delay. In this paper, we develop a two-stage incentive mechanism and task assignment and resource allocation scheme by combining auction game, matching theory, and convex optimization method. In the first stage, we present the incentive problem between the BS and nearby vehicles, which leverages a reserve auction. Then we study the network delay minimization problem. The problem is decoupled into two subproblems for determining task assignment and computing resource allocation, respectively. Finally, numerical results show the effectiveness and efficiency of our scheme.**

***Index Terms*—Mobile Edge Computing, Vehicle, offloading, auction, incentive, matching game**

I. Introduction

The increasing of mobile devices requires the platform to realize the mobile application with low latency and high resource consumption requirement, such as interactive gaming, virtual reality, and natural language processing. Mobile devices with the limit of computation capacity and battery level can not satisfy the requirement of low latency and energy efficiency. The conflict between the high demand in computation resource and the computation capacity limitation of mobile devices makes the implementation of mobile applications more challenging [1], [2].

Mobile Cloud Computing (MCC) can be considered as a potential approach to deal with this challenge. In MCC, the mobile users offload the applications to servers in the core network. Although servers in MCC have large computing capacity, the long-distance between users and servers causes long transmission time of applications. This degrades the efficiency of application offloading.

In order to improve the efficiency of application offloading, Mobile Edge Computing (MEC) is a promising technology. In MEC, servers are located in base stations, closer to users when compared with the servers in the core network. Due to the short distance between the MEC server and users, the MEC can provide low latency, high bandwidth, and computing ability in computation offloading. There are several works studying MEC. Most existing works in MEC focus on the offloading strategy and resource allocation in the homogeneous system [3]–[5].

Covering the broad area requires the deployment and maintenance a large number of MEC servers. However, the MEC servers are high-cost energy inefficient. As the result, the deployment of dense MEC servers causes a significant increase in capital expenditure (CAPEX) and operational expenditure (OPEX) and causes the wastage of resources during the off-peak time. Therefore, it is necessary to consider the deployment of MEC servers to satisfy the ever-increasing demand of users with reasonable cost.

On the other hand, there are huge resources in neighboring equipment, such as vehicles that can be utilized [6], [7]. Several existing papers considered the utilization of vehicles to offload the tasks for BS. Recently, the optimal deployment of vehicles as a cloudlet in offloading has attracted significant attention. In [8], the authors considered the offloading strategy of massive workload tasks in the proximity of vehicles. They aimed to minimize the total delay under three different velocity models. In [9], an approach to offload the workload portion so that energy consumption is efficient was also proposed. The solution was based on the consensus alternative direction method of multipliers. The work in [10] exploited the unused vehicle resource and infrastructure equipment to offload the users' requests. The authors proposed the architecture and presented the interaction between vehicular cloud and MEC server. This approach utilizes fuzzy-based gateways' selection. In [11], the authors optimized the offloading time selection and resource allocations considering the vehicle mobility and the task latency. However, there are two remaining challenges: 1) lack of sharing incentive scheme to vehicles and 2) lack of tasks assignment and corresponding computation resource scheme between vehicles and users to minimize the total network delay.

In this work, we address the two mentioned challenges. For

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing). *Dr. CS Hong is the corresponding author.

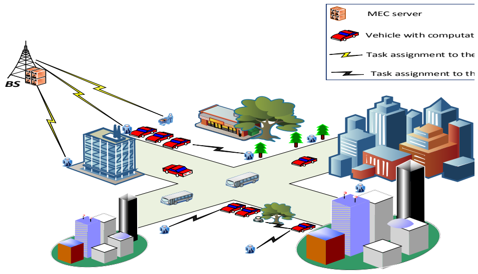


Fig. 1: System Model

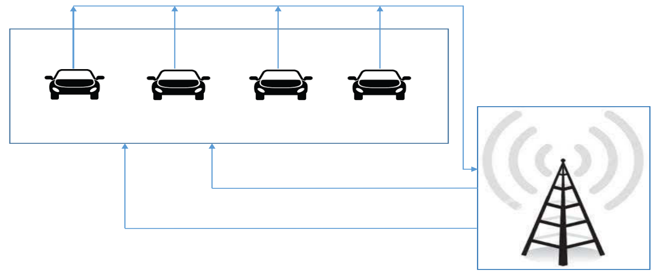


Fig. 2: Auction model between the BS and vehicles

the first challenge, the existing work assumed that vehicles offload the tasks without condition. However, in reality, vehicles need motivation to help BS to offload tasks because it can cause more energy consumption and unnecessarily additional delay for vehicle owner's tasks. Some attempts have been made to study the incentive mechanism for vehicles in offloading. In the paper, the author consider the utilization of vehicle in helping BS to offload the tasks. In addition, vehicle in the future which are produced with the aim of improving driving safety, convenience, and satisfaction will be installed with more computation effective on board computers. In , the author consider of using vehicle to offload the task for BS. In [12], the authors assumed that information between the cloudlet service provider and vehicles was completely known. In [7], a dynamic pricing strategy between the servers and parked vehicles was proposed to minimize the average cost with constraints. The system was modeled as two dimensional Markov chain. However, vehicles may not reveal all its information to the service provider/ servers, which causes information asymmetry problem. To deal with this problem, the work in [13] proposed the contract theory based mechanism to incentivize vehicles to offload the tasks for BS. The BS owner designed the contract for each type of vehicles, which was defined as resource sharing willingness of vehicles. The problem is that the authors assume that the BS knows the probability that one particular vehicle belongs to which type. This information is obtained via long term measurements or historical observations. In this work, we propose a new scheme to drive the nearby vehicles to offload the computation tasks to reduce the network congestion during the peak hour without additional servers deployment. Our scheme is based on a randomized auction theory. In our scheme, the BS acts as an auctioneer, and vehicles act as bidders. Upon the notification of BS, vehicles submit bids to the auctioneer for computation resources and the cost. After receiving bids from the bidders, the auctioneer will decide the winner determination problem and reward. Vehicles are naturally self-interest and reveal the untruthful cost to the BS. Even if vehicles are truthful, the problem of winner selection is NP-hard. Consequently, the VCG [14] type mechanism becomes computationally infeasible and is not applicable. However, the randomized auction can guarantee computationally efficient, individuality rational, and truthfulness. Our work is distinguished from the work in [13] is that the result does not depend on the type of vehicles.

For the second challenge, there lacks a task assignment and resource allocation mechanism which is low-complexity, sub-optimal, and minimize the delay of network. Since users (UEs) are belonged to different parties, they are likely to have different delay tolerance and prefer task assignment decisions. Therefore, all UEs have compromised on the task assignment decision so that each computational task can be executed. In addition, the way to allocate the resource at vehicles to UEs also affects to ensure reliable offloaded task execution. The system delay is formulated as an optimization problem is formulated and then, the problem is decoupled into two sub-problems to determine task assignment and computing resource allocation, respectively. The task assignment problem is modeled as a matching game [15], [16] in which UEs and vehicles are players. UEs rate the vehicles by considering transmission delay, task processing delay, task size. An algorithm for a stable matching between UEs and vehicles is also presented. Then, the computing resource allocation problem is solved via existing convex optimization methods.

The rest of this paper is organized as follows. Section II describes the system model. Section III presents auction-based incentive mechanism design and Section IV shows design of task assignment and resource allocation scheme. Section V provides the numerical results and Section VI concludes the paper.

II. System model

We consider the vehicular mobile edge computing, which is illustrated as Fig. 1. We consider one cell in which there is one base station (BS), M users (UEs), and N vehicles computing offloading ability. The BS can offload the computing application of users in this cell. During the peak time, the high demands of incoming computation can make the BS congested. Therefore, the BS can incentivize nearby vehicles to share their computation resources for task processing. The time-slot model is adopted. The sets of vehicles and UEs within the coverage of the BS are assumed to remain fixed within each slot and vary across different slots. The vehicle is parking. In each slot, each UE m has computation task to process, which characterize by $\{S_m, C_m, L_m^{tol}\}$. S_m is the input data size of task of UE m, C_m is the number of required

TABLE I: Table of notation

Description	Notation
Number of vehicle	N
Number of bids each vehicle can submit	K
Vehicle index	n
Bid index	k
The resource amount of the kth bid of vehicle n	d_{nk}
The claimed cost of the kth bid of vehicle n	b_{nk}
The sharing resource target of BS	W
The weighted coefficient	β^q
The reward that vehicle n can receive if its bid wins	r_n
Number of users	M
The input data size of task of UE m	S_m
The CPU cycles number for executing task of UE m	C_m
The delay constraint	L_m^{tol}
The resource amount allocated to UE m at vehicle n	f_{mn}

CPU cycles for executing task of UE m successfully and L_m^{tol} is the delay constraint. The computation tasks of UEs can be either processed by the base station or offloaded to vehicles that agree to share the computing resource with BS. We assume that the offloading tasks can be finished within one time-slot. We also denote the total sharing computation resource target set by the BS is W. This information can be obtained by the prediction at the beginning of time-slot, which is assumed to be available in this paper.

In this paper, we model the interaction between the BS and vehicles as an auction game in which the BS is an auctioneer. The BS solicits sharing bids (including the planned resource sharing and associated cost) from the vehicles. The BS then selects winning bids, decides reward to the winners, and notifies the vehicles of the auction outcome. Each vehicle n can submit one or many bids to the BS. Denote B_n as the set of bids submitted by the vehicle n. Each bid as two elements, b_{nk}, specifies the claimed cost due to resource sharing; and n_{nk} represents the amount of planned sharing resource. The auction model is presented as Fig. 2.

The details for how to model the interactions between UEs and vehicles, and how to derive a low-complexity suboptimal task assignment and resource allocation problem solution will be illustrated in Section III and Section IV, respectively.

The key notation is summarized in Table I.

III. Auction-based Incentive Mechanism Design

A. Problem Formulation

Let denote $\mathcal{X} = \{x_{nk}\}$, a binary matrix, describing the auction outcome. We have

$$x_{nk} = \begin{cases} 1, & \text{if vehicle } n\text{'s } k\text{th bid wins,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Without loss of generality, we assume that each vehicle can submit the same number of bids to the BS. Let K is the number of bids that vehicle n can submit. However, there is only one bid submitted by vehicle can win, that is

$$\sum_{k=1}^{K} x_{nk} \leq 1. \forall n \in \mathcal{N}. \tag{2}$$

Furthermore, the total resource sharing should be exceed the resource sharing target of the BS, that is

$$\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk}x_{nk} \geq W. \tag{3}$$

Sharing the computation resource with the BS can cause more delay for the owner of the vehicle. The vehicle's cost is the difference between the cost due to sharing and the reward receiving, i.e., $\sum_{k=1}^{K} b_{nk}x_{nk} - r_n(\sum_{k=1}^{K} x_{nk})$. The cost of BS is reward that BS gives to vehicles to incentive resource sharing, i.e.,$\sum_{n=1}^{N} r_n(\sum_{k=1}^{K} x_{nk})$. The social cost is the sum of vehicles' cost and the BS's cost, which is equal to $\sum_{n=1}^{N}\sum_{k=1}^{K} b_{nk}x_{nk}$. The social cost represents the negative impact of resource sharing on vehicular mobile edge computing.

Then the computation resource sharing incentive problem can be equivalent to the following social cost minimization problem as following:

$$\min_{x_{nk}} \sum_{n=1}^{N}\sum_{k=1}^{K} b_{nk}x_{nk} \tag{4a}$$

$$s.t. (2)\,(3), \tag{4b}$$

$$x_{nk} = \{0, 1\}. \tag{4c}$$

The solution of the above problem is winner determination. Thus, we denote the above problem as the winner determination problem (WDP). Following, we will show the WDP is NP-hard.

Assuming we have the certificate, the vehicles set $\mathcal{N}$, all the bids submitted to the system, $B_n, k \in \mathcal{N}$, total resource W and value of social cost Π. It takes a running time of $O(NK_n)$ to verify this certificate, which means checking whether: 1) $\sum_{n=1}^{N}\sum_{k=1}^{K} b_{nk}x_{nk} = \Pi$; 2) $\sum_{k=1}^{K} x_{nk} \leq 1 \forall n \in \mathcal{N}$; 3) $\sum_{n=1}^{N}\sum_{k=1}^{K_n} d_{nk}x_{nk} \geq W$ Since the certificate can be verified in polynomial time, the WDP is an NP problem.

In addition, the WDP can be mapped to the knowing NP-hard problem, such as minimum knapsack problems in polynomial time. Therefore, the WDP is NP-hard. When the number of vehicles and bids grow, the complexity of the optima solution will increase exponentially. This makes the VCG mechanism be computationally infeasible to be applied directly. In the following section, we will present a randomized auction-based mechanism.

B. Design of Randomized Auction Mechanism

1) The Randomized Auction Algorithm: Algorithm 1 shows the randomized auction-based mechanism, which consists of three steps. In the following, we will present more detail the proposed mechanism.

Step 1 (The Fractional VCG Auction): The first step is to transform the WDP into the Linear Programming Relaxation (LPR) by relaxing constraint (4c) to

$$0 \leq x_{nk} \leq 1, \forall n \in \mathcal{N}, k \in \mathcal{K}.$$

Algorithm 1: The Randomized Auction Algorithm

Step 1: The Fractional VCG Auction
Relax the constraint $x_{nk} = \{0,1\}$ into $0 \leq x_{nk}$
Compute the optimal solution $\mathbf{x}^*$ and corresponding reward.
Step 2: Decomposition
Decompose the fractional VCG $\mathbf{x}^*$ in to a convex combination of mixed integer solutions $\mathbf{x}^q$,such that ,
$\sum_{q\in\mathcal{Q}} \beta^q \mathbf{x}^q \leq \alpha \mathbf{x}^*$ and $\sum_{q\in\mathcal{Q}} \beta^q = 1$,
Step 3: The Feasible Integer Solution Selecting
Select each feasible integer solution $\mathbf{x}^q$ of the WDP with probability β^q.
Decide the reward for the winning bids.

Here we ignore the upper bound of x_{ij} because we have constraint (2). This LPR problem is linear programmable and can be solved in polynomial time to get an optimal solution.

The reward r_n^* for vehicle n is defined as

$$r_n^* = \sum_{n=1}^{N}\sum_{k=1}^{K} b_{nk}\bar{x}_{nk} - \sum_{\bar{n}=1,\bar{n}\neq n}^{N}\sum_{k=1}^{K} b_{\bar{n}k} x*_{\bar{n}k}, \tag{5}$$

where $\bar{x}_{nk}$ is the output of WDP problem without vehicle n.
Step 2 (Decomposition):

In this step, the decomposition technique used in [17], [18] can provide fractional weight when we decompose the optimal solution of LPR into the combination of integral solutions. We assume that there is an effective polynomial algorithm to solve the WDP approximately, which is presented in Section III-B2.

The approximation algorithm can provide the integrality gap of α. Therefore, we have:

$$\sum_{n,k} b_{nk} x_{nk} \leq \alpha O_{LPR}, \tag{6}$$

where O_{LPR} is the optimal value of LPR when the solution is $\mathbf{x}^*$. In this step, we aim to find the combination weighting coefficient β^q, such that

$$\sum_{q\in\mathcal{Q}} \beta^q = 1, \text{and} \sum_{q\in\mathcal{Q}} \beta^q \mathbf{x}^q \leq \alpha \mathbf{x}^*. \tag{7}$$

The weight β^q is the probability that feasible integer solution $\mathbf{x}^q$ is selected. In order to obtain the selecting probability β^q for each integer solution $\mathbf{x}^q$, we have to solve the following problem:

$$\begin{aligned} \textbf{Primal:} \quad \min_{\boldsymbol{\beta}} \quad & \sum_{q\in\mathcal{Q}} \beta^q \\ s.t. \quad & \sum_{q\in\mathcal{Q}} \beta^q \mathbf{x}^q \leq \alpha \mathbf{x}^*, \\ & \sum_{q\in\mathcal{Q}} \beta^q \leq 1, \\ & \beta^q \geq 0, \ \forall q \in \mathcal{Q}. \end{aligned} \tag{8}$$

This problem is hard to solve because there is an exponential number of variables. Thus, we transform (8) into the dual form with an exponential number of constraints. The corresponding dual problem of (8) is given as follows:

$$\begin{aligned} \textbf{Dual:} \quad \max_{\boldsymbol{\omega},\,\iota} \quad & \sum_{n,k} \omega_{nk}\, \alpha x_{nk}^* + \iota \\ s.t. \quad & \sum_{n,k} \omega_{nk}\, x_{nk}^q + \iota \geq 1, \ \forall q \in \mathcal{Q}, \\ & \iota \geq 0, \quad \omega_{nk} \geq 0 \quad \forall n,k. \end{aligned} \tag{9}$$

The ellipsoid method is used to solve the dual problem in polynomial time. Specially, we use a polynomial number of hyperplanes to cut the ellipsoid to obtain the optimal solution. Each hyperplane, which produced by our proposed approximation algorithm, matches a feasible integral solution. Each feasible integral solution further corresponds to a variable in the primal problem. This makes the primal problem small-sized with polynomial number of variables, which enables us to solve the primal problem in polynomial time. Hence, we can solve the primal in polynomial time.
Step 3 (choose the feasible integer solution):
In this step, each possible integer solution is chosen with a probability that we can obtain in step 2. Then we have

$$\sum_{q\in\mathcal{Q}}\sum_{n,k} \beta^q b_{nk}^q x_{nk}^q \leq \alpha O_{LPR}. \tag{10}$$

The reward for the vehicle n is defined as

$$r_n = r_n^* \frac{\sum_k b_{nk} x_{nk}^q}{\sum_k b_{nk} x_{nk}^*}. \tag{11}$$

In the following, we prove next that our mechanism guarantees system efficiency in expectation, individual rationality and offers truthfulness in its best effort. We have the expected utility of vehicle n as follows:

$$(r_n^* - \sum_k b_{nk} x_{nk}^*) \frac{\sum_k b_{nk} x_{nk}^q}{\sum_k b_{nk} x_{nk}^*}. \tag{12}$$

Follow the individual rationality property of the fractional VCG, the expected utility of each vehicle is also greater and equal to zero. Therefore, the proposed auction is individual rationality.
The expected social cost is

$$\sum_{q\in\mathcal{Q}}\sum_{n,k} \beta^q x_{nk}^q b_{nk} \leq \sum_{n,k} \alpha x_{nk}^* b_{nk} = \alpha OPT_{LPR}. \tag{13}$$

Therefore, the expected social cost of the randomized auction is bounded by α times the social cost of fractional VCG solutions. Thus, our mechanism guarantees the expected truthfulness.

2) Approximation Algorithm: To use the ellipsoid method, we need a separation oracle to find out a violated constraint in each iteration. Following, we present an approximation algorithm, which is shown as followed:

Step 1: Every vehicle will compute the cost per unit of resource, that is $\theta_{nk} = \frac{b_{nk}}{d_{nk}}$. Here, θ_{kn} shows the cost per unit of computation resource.

Step 2: Resort all the bids according to the non-increasing order of their cost per unit of resource. The bid with minimum cost wins the bidding.

Step 3: Delete vehicle n from the list of bidders. Then go back to Step 2 until satisfying one of the following termination conditions:

i) The BS can achieve its resource sharing target;

ii) All the vehicles already won the auction.

The proposed greedy approximation algorithm is described in detailed in Algorithm 2. After the BS notify which vehicles are winners and pay the reward to winning vehicles, the task of UEs can be transmitted directly to the vehicles. The amount of resources that the vehicles allocate to UEs is the same as the amount declared in their winning bids. The method to determine the vehicles and the amount of allocated resources to execute tasks of each UE will be presented in the following Section.

Algorithm 2: The Approximation Algorithm

$\mathcal{X} = \{x_{nk}\} = \mathbf{0}$
for $n \in \mathcal{N}, k \in \mathcal{K}$ **do**
    $\theta_{nk} = \frac{b_{nk}}{d_{nk}}$
end
Resort θ in the non-increasing order
$C = 0;\ l = 0;\ Q = \max\theta$
while $W > C$ *AND* $l < N$ **do**
    $[\mu, \nu] = \text{argmin}\theta;\ C = C + d_{\mu,\nu};$
    $x_{\mu,\nu} = 1; l = l + 1;$
    if $x_{\mu,\nu} = 1$ **then**
        for $\forall k \in \mathcal{K}$ **do**
            $\theta_{\mu,\nu} = Q$
        end
    end
    Resort θ in the non-increasing order
end

IV. Task assignment and Resource Allocation Scheme

In this section, the task assignment and resource allocation problem is introduced firstly. This is followed by problem formulation and then, the problem is decomposed into two subproblems to determine task assignment and resource allocation, respectively. Next, they are solved via the matching game and convex optimization methods. Finally, a joint algorithm is proposed for an approximated solution.

A. Task Assignment Model

1) Delay Constraints: One of essential issue in task assignment for MEC, which needs to be tackle, is the delay constraint. In practice, different tasks have different delay constraints, which is defined as the period of time from one computation task is requested until it is completed. Specifically, the delay threshold of each task of UE m is denoted by L_m^{tol}. The overall latency includes two components: (i) transmission delay; (ii) computation delay. The received signal to interference plus noise ratio (SINR) at vehicle n can, therefore, be expressed as:

$$\Gamma_{mn} = \frac{p_m^{tra}\gamma_{mn}}{N_0 + \sum_{h \neq m} p_h^{tra}\gamma_{mn})}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \quad (14)$$

where p_m^{tra} is the transmission power of UE m and N_0 is the noise power. Then the achievable data rate of from UE m to vehicle n is given by:

$$R_{mn} = B\log_2(1 + \gamma_{mn}), \forall m \in \mathcal{M}, n \in \mathcal{N}. \quad (15)$$

The transmission delay

$$L_{mn}^{tra} = \frac{S_m}{R_{mn}}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \quad (16)$$

where S_m denotes the input data size of task of UE m. In addition, the computational delay is given by:

$$L_{mn}^{com} = \frac{C_m}{f_{mn}}, \forall m \in \mathcal{M}, n \in \mathcal{N}, \quad (17)$$

where C_m represents the number of required CPU cycles for executing task of UE m successfully and f_{mn} is the number of CPU cycle that vehicle n assigns to UE m. Therefore, the total delay of executing task of UE m is given as:

$$L_{mn}^{tot} = L_{mn}^{tra} + L_{mn}^{com}, \forall m \in \mathcal{M}, n \in \mathcal{N}. \quad (18)$$

B. Problem Formulation

In this work, the aim is to relieve the heavy load of the BS and reduce the total network delay by utilizing the computation resources of vehicles. Hence, we model the objective function as the total network delay, i.e., the overall delay of all the UEs. We investigate how to assign the tasks of UEs to vehicles and how to allocate the sharing computation resource of a vehicle to UEs such that the delay is minimized. The task assignment decision between M UE and N is defined as a $M \times N$ matrix $\mathcal{Z}$ in which element z_{mn} is equal to 1 when the computation task of UE m is assigned to vehicle n and equal to 0, otherwise. The problem is formulated as follows:

$$\textbf{TA-RA}: \quad \min_{\boldsymbol{Z},\boldsymbol{f}} \quad \sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn}L_{mn}^{tot} \quad (19a)$$

$$s.t. \quad \sum_{n \in \mathcal{N}} z_{mn} \leq 1, \forall m \in \mathcal{M}, \quad (19b)$$

$$\sum_{m \in \mathcal{M}} z_{mn} \leq q_n, \forall n \in \mathcal{N}, \quad (19c)$$

$$\sum_{m \in \mathcal{M}} f_{mn} \leq F_n, \forall n \in \mathcal{N}, \quad (19d)$$

$$\sum_{n \in \mathcal{N}} z_{mn}L_{mn}^{tot} \leq L_m^{tol}, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \quad (19e)$$

where (19b) guarantees that one UE can be assigned to one vehicle, (19c) guarantees that one vehicle can execute at most of q_n UE. (19d) denotes the maximum computation resource that vehicle n can share according to the agreement in the auction with the BS. (19e) denotes the delay constraints of task assignment. Due to the complex coupling among the optimization variables and nonlinear constraint (19e), solving the above problem is challenging. We develop an approximated approach to decouple the problem (19a) into two subproblems about task assignment and resource allocation as follows.

1) Task Assignment: First, we solve (19a) with respect to given f_{mn}. The outcome is the task assignment decision $\mathcal{Z}$. Therefore, the task assignment can be given as:

$$\begin{aligned} \textbf{TA}: \quad & \min_{\boldsymbol{Z}} \quad \sum_{m=1}^{M}\sum_{n=1}^{N} z_{mn} L_{mn}^{tot} \\ & s.t. \quad (19b), (19c), \text{and } (19e). \end{aligned} \tag{20}$$

To provide a tractable solution, the problem (20) can be transformed into a two-sided matching problem. In this matching game, two separate sets of players are UEs and vehicles. Each UE can be assigned to one vehicle, and one vehicle can execute tasks for at most q_n UEs. The task assignment can be formulated as a one-to-many matching μ with the following definition.

Definition 1. *A one-to-many matching μ is defined as a function from the set $\mathcal{M} \cup \mathcal{N}$ into the set of $\mathcal{M} \cup \mathcal{N}$ such that:*

1) $|\mu(m)| \leq 1$ and $\mu(m) \in \mathcal{N}$,
2) $|\mu(n)| \leq q_n$ and $\mu(n) \in \mathcal{M}$,
3) $m \in \mu(n)$ if and only if $\mu(m) = n$.

Each set of players constructs its preference list by ranking the another side using well-defined preference function. Concerning about the total delay, we can define that the preference function is inversely proportional to the total delay, e.g., L_{mn}^{tot}. The preference of UE m towards vehicle n is calculated as

$$G_{mn} = 1/L_{mn}^{tot} \tag{21}$$

The preferences toward different vehicles is denoted as "$\succ$". For instance, $V_n \succ_m V_n'$ represents that UE m prefers vehicle n to vehicle n' , which is given by

$$V_n \succ_m V_{n'} \Leftrightarrow G_{mn} \geq G_{mn'} \tag{22}$$

Similarly, the preference of vehicle n towards UE m is calculated as

$$P_{nm} = 1/L_{mn}^{tot} \tag{23}$$

and the vehicle n prefers UE m to UE m' when

$$W_m \succ_n W_{m'} \Leftrightarrow P_{nm} \geq P_{nm'} \tag{24}$$

Our goal is to find a stable matching, which is the key concept as optimal result by using the matching game. The definitions of stable matching are presented as follows.

Definition 2. *A task assignment is stable if and only if there is no blocking pair and all vehicles are assigned. Traditionally, a stable matching in the deferred-acceptance algorithm is a state that no player wants to change their partners.*

This implies that the matching is stable, if there is no blocking pair, which is defined as follows:

Algorithm 3: One-to-many matching based task assignment algorithm

- Initial: UEs are associated with max-RSSI vehicles
- Calculate the preference lists of UEs and vehicles by (21), (23)
- Updated quota matrix $\mathcal{Q}$
- Initialize rejected matrix $\mathcal{R}$
while *$\mathcal{R}$ is not empty* **do**
  for *all the UEs in $\mathcal{R}$* **do**
    Propose to the most preferred vehicle
  end
  for *all the vehicles receive the proposal from UEs* **do**
    - Updates its request list
    -Ranks the requests by ((11)) and selects first q_n UEs and rejects the remaining
  end
  Update acceptance matrix $\mathcal{Z}$
  Update rejected matrix $\mathcal{R}$
  for *all the UEs in $\mathcal{R}$* **do**
    Delete the vehicle which rejected its proposal from the preference list.
  end
end

Definition 3. *A matching μ is blocked by a pair of agents (m, n) if there exists a pair (m, n) with $m \notin \mu(n)$ and $n \notin \mu(m)$ such that $m \succ_n \mu(n)$ and $n \succ_m \mu(m)$. Such a pair is called a blocking pair.*

In this paper, the deferred-acceptance [19] based task assignment algorithm is deployed to seek a stable matching as Alg. 3.

2) Resource Allocation: Based on given z_{mn}, vehicle n is required to allocate the computing resource f_{mn} for task of UE m. Let $\mathcal{U}_n$ as the set of tasks assigned to the vehicle n. A computation resource allocation problem for the tasks in vehicle n is formulated as

$$\begin{aligned} \textbf{RA}: \quad & \min_{f_{mn}} \quad \sum_{m \in \mathcal{U}_n} \frac{C_m}{f_{mn}} \\ & s.t. \quad (19d) \text{ and } f_{mn} \geq f_{mn}^{min}, \forall m \in \mathcal{U}_n, \end{aligned} \tag{25}$$

where f_{mn}^{min} is calculated based on (19e). We first define the optimal solution for the above problem as $f_{mn} = f_{mn}^{min} + \Delta f_{mn}$, where Δf_{mn} is a nonnegative value. We also set a temporary variable $\Delta F_n = \sum_{m=1}^{M} \Delta f_{mn} = F_n - \sum_{m=1}^{M} f_{mn}^{min}$ and have $0 \leq \Delta f_{mn} \leq \Delta F_n, \forall m \in \mathcal{M}$. The problem 25 can be written as follow:

$$\begin{aligned} \textbf{RA2}: \quad & \min_{\Delta f_{mn}} \quad \sum_{m \in \mathcal{U}_n} \frac{C_m}{\Delta f_{mn} + f_{mn}^{min}} \\ & s.t. \quad \Delta f_{mn} \geq 0, \forall m \in \mathcal{U}_n, \\ & \qquad \sum_{m=1}^{M} \Delta f_{mn} \leq \Delta F_n. \end{aligned} \tag{26}$$

The Lagrangian of (26) is

$$L(\boldsymbol{f},\lambda)=\sum_{m\in\mathcal{S}_n}\frac{C_m}{\Delta f_{mn}+f_{mn}^{min}}+\lambda(\sum_{m\in\mathcal{U}_n}\Delta f_{mn}-\Delta F_n) \tag{27}$$

Thus, we can obtain the dual problem of (26) as

$$\max_{\lambda\succ 0}\min_{\boldsymbol{f}\succ 0} L(\boldsymbol{f},\lambda) \tag{28}$$

We have

$$f_{mn}^{*}=\frac{\sqrt{C_m}}{\sqrt{\lambda}}-f_{mn}^{min} \tag{29}$$

Substituting (29) into (28), we have

$$\begin{aligned}\varphi(\lambda)&=\min_{\boldsymbol{f}\succ 0} L(\boldsymbol{f},\lambda)\\&=\sum_{m\in\mathcal{U}_n}\sqrt{\lambda C_m}+\lambda(\sum_{m\in\mathcal{U}_n}\frac{\sqrt{C_m}}{\sqrt{\lambda}}-\sum_{m\in\mathcal{U}_n}f_{mn}^{min}-\Delta F_n)\end{aligned} \tag{30}$$

By solving $\varphi(\lambda^*)=0$, we obtain the optimal Lagrangian multiplier λ^* as

$$\lambda^*=\frac{4(\sum_{m\in\mathcal{U}_n}C_m)}{F_n^2} \tag{31}$$

Substituting (31) into (29), we have

$$f_{mn}^{*}=\frac{F_n}{2(\sqrt{\sum_{m\in\mathcal{U}_n}C_m})}\sqrt{C_m}-f_{mn}^{min} \tag{32}$$

V. Numerical Results

This section evaluates the performance of the proposed scheme. The number of vehicles is 5 and the number of bids that each vehicle can submit to the BS varies from 1 to 8. Before submitting the bids to the BS, each vehicle predicts how much resource that the BS requires to fulfill the BS users' request. To model for the random character of the proposed model, the resource sharing amount and reward are randomly chosen between 17 Gb and 20 Gb and between 1 and 10, respectively. The sharing resource target of the BS is 30 Gb. Other parameters are summarized in Table II.

TABLE II: Simulation Parameters

Parameter	Value
Data size of UE's task	400-800 kb
Computation size of UE's task	1-5 Gb
Cell Radius	1000m
UE's coverage	200m
Transmission power of UEs	30 dBm
Bandwidth of UEs	10 MHz
Noise power	-174 dBm

A. Evaluation of the approximation algorithm -Algorithm 2

Fig. 3a presents the social cost performance of Algorithm 2 and Optimal solution. We can see that when the number of bids submitted to the BS increases, the social cost decreases. This is due to the BS have more opportunity to choose bids with lower cost so that the resource sharing amount can be satisfied. We also see that the performance of Algorithm 2 is 9% different from the performance of the optimal solution when the number of bids is 5.

B. Evaluation of Prediction Error

Moreover, we study the performance of the proposed scheme when the prediction of sharing resource target is not perfect. To model this, we assume that each vehicle may underestimate its near future resource target by a percentage between 5% and 40%. As illustrated in Fig. 3b, the social cost increases linearly as the prediction error grows. The reason is that as the prediction error increases, there is a corresponding increase in the bidding reward, which will increase in the social cost. Furthermore, Fig. 3b also shows that the social cost increases with a decrease in the number of bids. This is reasonable because a larger number of bids indicates more favorable bids with lower $\theta_{\mu,\nu}$ will win the bidding.

C. Comparison The Randomized Auction based Incentive Scheme with Renting Scheme

In the following, we compare our proposed auction-based scheme with the renting scheme. We use the same planning sharing resource and cost. Hence, in this renting scheme, the BS first decides the available sharing resource amount by querying vehicles and then sets the expected renting price for its users to purchase. In this simulation, the expected price for a unit resource is used as the renting price. The simulation result is illustrated in Fig. 3c. As a comparison, the social cost of our scheme is lower than that of renting scheme.

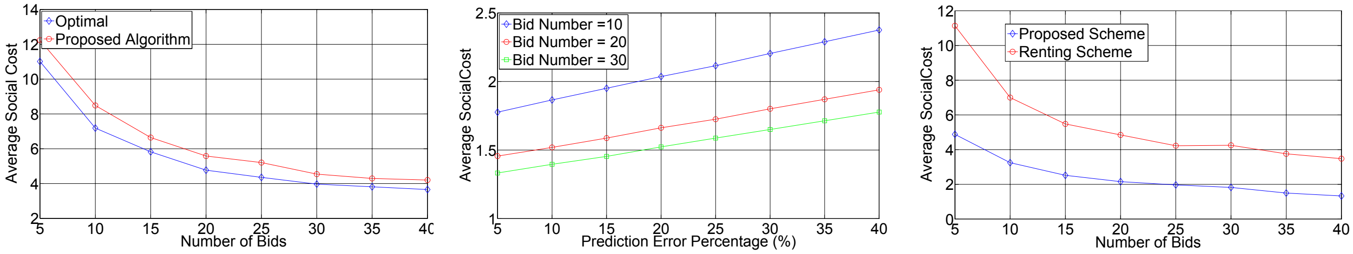
D. Evaluation of Proposed Task Assignment and Resource Allocation Scheme

Fig.3c shows the performance of the proposed task assignment and resource allocation scheme when the quota of a vehicle are 3, 5 and 7. The number of vehicles are 5. The total delay increase when the number of users increases because there are more tasks to process. In addition, when the quota of a vehicle increases, the total delay also increases. This is because the number of users assigned to the same vehicle increase, which makes the processing delay increases.

Table III presents the comparison in total delay between the proposed scheme and RSSI scheme. In RSSI scheme, the task assignment is determined based on RSSI maximization and the resource allocation is decided based on (32). The performance of RSSI scheme is much worse than our proposed scheme in terms of reducing the total delay. Furthermore, when the delay threshold increases, the total delay increases. This is due to the UEs have more freedom to choose the vehicle to offload the task.

VI. Conclusion

In this paper, we studied the problem of offloading incentive from BS to nearby vehicles and task assignment and resource allocation between the users and vehicles. We proposed a randomized auction-based incentive mechanism in order to minimize the social cost with the sharing resource target. The task assignment and resource allocation were formulated as a delay minimization problem. The problem was decomposed



(a) (b) (c)

Fig. 3: Numerical results of a) Comparison between proposed Aprroximation Algorithm and Optimal Solution, b) Performance under sharing resource prediction error. c) Comparison between proposed auction based scheme and renting scheme.

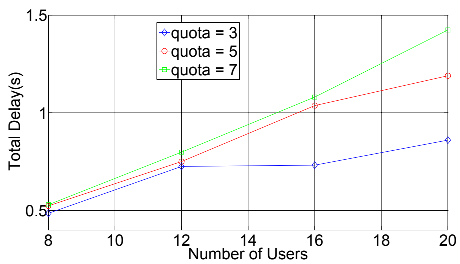


Fig. 4: Total network delay vs. number of users.

TABLE III: Comparison between RSSI based scheme and proposed task assignment and resource allocation scheme

Quota q_v	$L^{tol} = 1$		$L^{tol} = 3$	
	RSSI	Proposed	RSSI	Proposed
$q_v = 1$	15180	0.1634	10038	0.3489
$q_v = 2$	1411	0.33	753.93	0.7949
$q_v = 3$	60715	0.4895	2184	1.2334
$q_v = 4$	59503	0.7169	32003	1.8495

into two subproblems and can be solved based on matching theory and convex optimization technique. Evaluation results show the performance of the proposed scheme can reduce the social cost compared with the renting scheme and reduce the total delay compared with the baseline. In future work, we will study how to apply machine learning to predict the demand and optimize the resource sharing.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[2] J. Z. W. S. N. K. H. Guo, J. Liu, "Mobile-edge computation offloading for ultra-dense iot networks," *IEEE Internet Things Journal*, vol. 5, no. 6, pp. 4977–4988, May 2018.

[3] G. U. o. T. G. C. Chao Yang ; Yi Liu ; Xin Chen ; Weifeng Zhong ; Shengli Xie Guangdong Key Laboratory of IoT Information Technology, School of Automation, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26 652–26 664, Feb. 2019.

[4] J. Jingyun Feng ; Zhi Liu ; Celimuge Wu ; Yusheng Ji National Institute of Informatics, Tokyo, "Mobile edge computing for the internet of vehicles: Offloading framework and job scheduling," *IEEE Vehicular Technology Magazine*, vol. 14, no. 1, pp. 28–36, Dec. 2019.

[5] F. G. . V. S. . D. S. . M. C. F. . S. M. . W. F. D. Munaretto, "Multi-access edge computing: The driver behind the wheel of 5g-connected cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, Oct. 2018.

[6] L. Z. Y. Lai, F. Yang and Z. Lin, "Distributed public vehicle system based on fog nodes and vehicular sensing," *IEEEAccess*, vol. 6, pp. 22 011–22 024, 2018.

[7] D. H. . W. C. . Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 420–423, Oct. 2018.

[8] P. L. . J. L. . Z. Sun, "Matching-based task offloading for vehicular edge computing," *IEEE Access*, pp. 27 628–27 640, Feb. 2019.

[9] Z. Z. . J. F. . Z. C. . X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus admm approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5087–5099, Mar. 2019.

[10] T. M. . I. J. . A. R. . M. B. Jemaa, "Towards multi-access edge based vehicular fog computing architecture," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6.

[11] J. W. . D. F. . S. Z. . J. T. . T. Q. S. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62 624–62 632, May 2018.

[12] X. L. . J. L. . W. Y. . J. W. . Z. Z. . X. Wang, "Vehicle-to-cloudlet: Game-based computation demand response for mobile edge computing through vehicles," in *89th Vehicular Technology Conference*, Apr. 2019, pp. 1–6.

[13] Z. Z. . P. L. . J. F. . Y. Z. . S. M. . J. Rodrig, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3113–3125, Jan. 2019.

[14] P. Maille and B. Tuffin, "Why vcg auctions can hardly be applied to the pricing of inter-domain and ad hoc networks," *Next Generation Internet Networks, 3rd EuroNGI Conference on*, pp. 36–39, 2007.

[15] D. Gale and L. S. Shapley, "College admissions and the stability of marriage," *American mathematical monthly*, vol. 69, no. 1, pp. 9–15, Jan. 1962.

[16] C.-P. T. J. Sethuraman and L. Qian, "Many-to-one stable matching: Geometry and fairness," *Mathematics of Operations Research*, vol. 31, no. 3, pp. 581–596, Aug. 2006.

[17] R. Carr and S. Vempala, "Randomized metarounding," *Random Structures & Algorithms*, vol. 20, no. 3, pp. 343–352, May 2002.

[18] J. Li, Y. Zhu, Y. Hua, and J. Yu, "Crowdsourcing sensing to smartphones: A randomized auction approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2764–2777, Jan. 2017.

[19] A. E. Roth, "Deferred acceptance algorithms: history, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, no. 3–4, pp. 537–569, Mar. 2008.