Latency Minimization for MEC-V2X Assisted Autonomous Vehicles Task Offloading

Yilun Zhang, Changrun Chen, Huiling Zhu, Yijin Pan, and Jiangzhou Wang, *Fellow, IEEE*

***Abstract*—Delay-sensitive applications for autonomous vehicles (AVs) require a substantial amount of computational resources. However, the onboard computation resources may be insufficient, resulting in long processing latencies. To deal with this critical issue, we jointly consider roadside unit (RSU) and assistant vehicle offloading, along with resource allocation, to minimize latency for vehicular tasks. This approach also takes into account frequency reuse among sub-areas for assistant vehicle offloading. The latency minimization problem can be formulated as a mixed-integer non-linear programming (MINLP) problem. Given the inherent complexity of the MINLP problem, we propose a two-step solution. The first step focuses on the combined decision of assistant vehicle offloading and transmit power allocation. To solve this problem, we propose a particle swarm optimization (PSO) algorithm with low complexity and low average transmit power. The second step deals with RSU offloading/local computation decision, bandwidth allocation, and computation resource allocation. An iterative algorithm is proposed to achieve the optimal solution. Without adding additional computation resources, simulation results demonstrate that the proposed vehicular task offloading approach improves overall delay performance than the adaptive MEC offloading scheme and the pure MEC computing scheme.**

***Index Terms*—Autonomous vehicle, mobile edge computing, vehicle-to-everything communication, task offloading, resource allocation, frequency reuse.**

I. Introduction

THE application of automatic driving has led to a surge in computation-intensive and delay-sensitive vehicular applications, including virtual reality (VR) [1], augmented vehicular reality (AVR) [2], and sensing recognition [3] in autonomous vehicles (AVs). These applications demand substantial computation resources to process vast volumes of raw sensing data in real-time [4], imposing a significant challenge to AVs with limited computation capabilities. At the same time, applications such as cooperative perception [5] raise higher demand for data sharing among vehicles. Data transmission is facilitated through vehicle-to-everything (V2X) communications [6], [7], which encompass vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication. To support these highly demanded computational applications, mobile edge computing (MEC) was introduced [8] [9]. MEC aims to offload computation-oriented tasks from end users to computation servers located at the edge of the radio access network, such as road side units (RSUs) or base stations (BSs) in close proximity. By offloading tasks to MEC servers, lower task processing latencies can be achieved due to the higher processing capabilities of these servers [10].

To mitigate the additional transmission delays introduced by task offloading, several related studies have focused on resource allocation of the MEC systems. In [11] and [12], joint task offloading and resource allocation strategies were investigated in the multiuser MEC system. In [13], partial computation offloading problem was studied in vehicular networks. However, in traditional MEC systems with a large number of users, the computation and communication resources at the MEC server may be insufficient. To address this limitation, additional computation resources and new system architectures were proposed. In [14], computation resources were jointly deployed on unmanned aerial vehicles and ground vehicles to reduced overall latency. Similarly, [15] considered a cloud-assisted MEC system and established a three-tier computing network to enhance the delay performance of vehicular task offloading.

However, the deployment of additional computation resources not only is expensive but also increases the burden on radio resources. Therefore, the utilization of computation resources from other users was explored through cooperative computation. Cooperative computation allows for resource sharing among users and has been extensively studied in device-to-device (D2D) communication scenarios. A cooperative computation-assisted framework was proposed in [16], enabling task offloading between devices within the MEC infrastructure. In [17], a low-complexity iterative algorithm was introduced in multiuser scenarios. However, both [16] and [17] overlooked interference among users by relying on different frequency and time slots, resulting in low spectrum efficiency. In [18], V2I and V2V communications were introduced for AVR data offloading, where convex optimization and Kuhn-Munkres algorithm were used for resource allocation. However, the pairs of V2V an V2I communications assumed a fixed transmission bandwidth, limiting the improvement of overall delay.

In the aforementioned MEC and vehicular scenarios, the feedback of computation results has been disregarded [10]–[18]. However, with the development of vehicular application, the latency requirement for autonomous vehicle is sensitive [19], 10 % of performance improvement can be significant. On the other hand, compared to the original data size, the feedback of some vehicular applications is relatively large and can not be ignored. In data fusion for cooperative sensing [20], LiDAR

This work was supported in parted by the European Commission's Horizon Europe Marie Skłodowska-Curie Postdoctoral Fellowship-UKRI guarantee under Grant EP/Y027558/1.

Y. Zhang, C. Chen, H. Zhu, and J. Wang are with the School of Engineering, University of Kent, Canterbury CT2 7NT, U.K. (e-mail: yz312@kent.ac.uk, c.chen@kent.ac.uk, h.zhu@kent.ac.uk, j.z.wang@kent.ac.uk)

Y. Pan is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 211111, China (e-mail: panyj@seu.edu.cn)

point clouds are extracted as a task for fusion computation. The computation result is the fused point cloud, whose data size is similar to the input data size. In [21], both Voxel Feature Fusion (VFF) and Spatial Feature Fusion (SFF) use features from LiDAR data as the input data which is similar to the output data size. Therefore, the delay of feedback should not be ignored. Additionally, the communication model of vehicles has been oversimplified [13], [15], [18], assuming all users to be stationary during task offloading. [22], [23] showed that the Doppler spread caused by high moving speed degrades the communication capacity. Considering the mobility of vehicle, the impact of Doppler spread on offloading performance cannot be ignored.

Motivated by the significance of serving delay-sensitive vehicular applications, and considering the merits and limitations of existing research, this paper proposes an assistant vehicle offloading approach in MEC networks to minimize the latency of vehicular tasks. Specifically, vacant vehicles' available computation resources are utilized to assist in task offloading through V2V communications, thereby alleviating computation bottlenecks without the need for additional resource deployment or modifications to the traditional RSU-based MEC network. Additionally, the adoption of V2V communications reduces the transmission bottleneck between RSUs and users in conventional MEC offloading system. Similar to cellular D2D communications [24], frequency reuse among V2V communications is possible in case of the short transmission distances involved in task offloading between vehicles. To mitigate interference among V2V communications, a rectangular sub-area division scheme is designed based on the physical positions of vehicles on the road. Each vehicle can only offload its tasks to another nearby vacant vehicle within its assigned sub-area. Furthermore, this paper considers the feedback of task processing and investigates the impact of vehicle movements among sub-areas. In this paper, we examine MEC and V2X for vehicular task offloading, encompassing three decisions: local computation, offloading tasks to RSU through V2I communication, and offloading tasks to assistant vehicles through V2V communication. The main contributions of this paper are summarized as follows

- To enable task offloading to both the RSU and vacant vehicles, we propose a MEC-V2X assisted vehicular task offloading system. By employing a distance-limited sub-area division method, frequency reuse is facilitate in V2V communication which can increase the transmission data rate and reduce transmission delay.
- Considering the movement of vehicles, we formulate the communication model with Doppler spread and dynamically allocate resources for offloading and feedback of the vehicles. A latency minimization problem is formulated which characterizes the tradeoff between communication and computation performance.
- Due to the combinatorial nature of the proposed offloading scheme, we propose a two-step solution for the original mixed integer nonlinear programming (MINLP) problem. First, we propose a heuristic algorithm that jointly determines the offloading decision for assistant vehicles with low complexity and average power consumption. Then, utilizing the Lagrange dual method, we present an iterative algorithm to jointly perform offloading tasks to the RSU. Simulation results show that our proposed MEC-V2X offloading scheme outperforms conventional MEC networks in terms of total latency.
- The sub-area division method is further investigated with the frequency reuse factor to enhance the performance of the proposed MEC-V2X offloading scheme. Numerical results show that with the density of vehicles on the road changing, the number/range of sub-areas can be adjusted to balance the assistant vehicles offloading and RSU offloading. Therefore, optimal sub-area allocation and frequency reuse factor can be obtained and the lowest latency can be achieved.

The rest of this paper is organized as follows. In Section II, we introduce the system model of the proposed MEC-V2X assisted vehicular task offloading system, along with the proposed sub-area division strategy. In Section III, we provide the problem formulation and describe the development of our joint offloading and resource allocation solutions. Section IV presents representative simulation results, and finally, Section V concludes the paper.

II. System Model

A. Network Model

Consider a MEC-V2X assisted vehicular scenario, as shown in Fig. 1. A MEC server with a certain amount of computation resource is integrated into an RSU, covering a one-way road with multiple lanes. Th distribution model of vehicles in each lane follows a homogeneous Poisson point process, denoted by the set $\mathcal{K} = \{1, 2, \cdots, K\}$. At any given instant, each vehicle generates a delay-sensitive task with a probability p_c, while the probability of a vehicle being vacant without a task is $p_v = 1 - p_c$. The driving status of all vehicles is obtained at the RSU as auxiliary information for decisions. The road can be dynamically divided into M sub-areas, and $\mathcal{K}_m$ represents the set of vehicles within sub-area m. Within each sub-area, one pair of proximate vehicles is allowed to operate task offloading through V2V communications when one of the vehicles is vacant. The vehicle that receives and processes the offloaded task is referred to as the assistant vehicle, while the vehicle offloading its task is the requesting vehicle.

Each vehicle's task is modeled as $U_k = \{D_k, C_k\}$, where D_k (bits) represents the task data size, and C_k (CPU cycles) denotes the required computation resource to complete the task. A task U_k generated by vehicle k can be processed locally at vehicle k or offloaded to either the RSU or an assistant vehicle in proximity. The status of all vehicle's computation requirements and their channel state information (CSI) is collected at the RSU where the offloading and resource allocation decision is made. The computation requirements include the onboard computation resource of each vehicle, the task size of each vehicle, and the vacancy status of all vehicles. The communication cost of with transmitting this information is relatively minor [25] and can be obtained at the RSU in very short time. The task offloading decision for

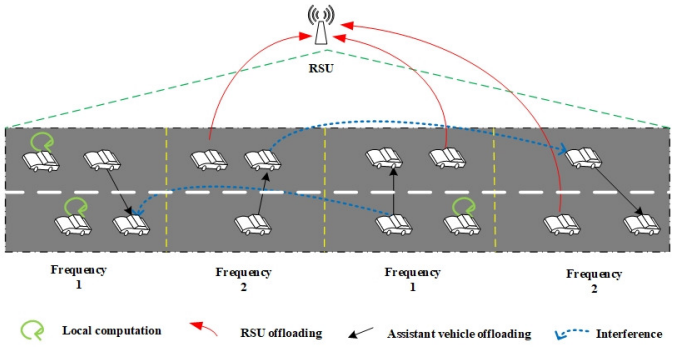


Fig. 1. MEC-V2X assisted vehicular scenario.

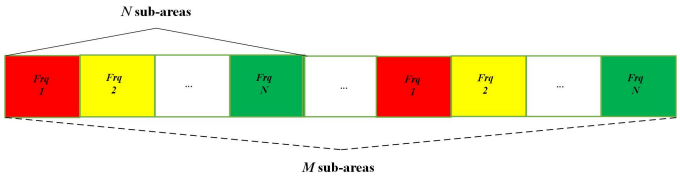


Fig. 2. Sub-area division and frequency reuse.

all vehicles is represented as $\mathcal{A} = \{a_1, a_2, \cdots, a_K\}$, where $a_k \in \{a_k^l, a_k^r, a_k^v\}$. Here, a_k^l, a_k^r, and a_k^v correspond to the decisions of local computation, RSU offloading, and assistant vehicle offloading, respectively, indicating whether the task should be processed locally, at the RSU, or at an assistant vehicle. In certain applications, tasks may not be divisible [26], for simplicity, each task can only be processed at one location [27], satisfying the following conditions

$$a_k^l, a_k^r, a_k^v \in \{0, 1\}, \tag{1}$$

$$a_k^l + a_k^r + a_k^v = 1. \tag{2}$$

B. Communication Model

In the V2X scenario, similar to [28], dedicated spectrum is allocated to V2I and V2V communications. Let B_R and B_V denote the total bandwidths dedicated to V2I and V2V transmissions, respectively. Therefore, there is no interference among V2I and V2V links. Due to the movement of vehicles, delayed channel gain and Doppler spread are considered. The channel gain g can be expressed as [29]

$$g = \beta \hat{g} + \theta, \tag{3}$$

where $\hat{g}$ is the delayed channel gain with a feedback delay of τ, and θ is the error vector with zero mean Gaussian distribution and variance σ_θ^2. The variance σ_θ^2 is defined by [30]

$$\sigma_\theta^2 = \sigma_{\hat{g}}^2(1 - |\beta|^2), \tag{4}$$

where β is the correlation coefficient given by

$$\beta = J_0(2\pi f_D \tau), \tag{5}$$

$J_0(\cdot)$ represents the zeroth order Bessel function of the first kind, and f_D is the maximum Doppler frequency.

Each V2I transmission is allocated a unique channel to avoid interference between V2I communications. In RSU offloading, the offloading transmission rate from vehicle k to the RSU can be obtained as

$$R_{k,r} = B_{k,r} \cdot \log_2\left(1 + \frac{P_k |\hat{g}_{k,r}|^2}{\sigma^2 + \sigma_\theta^2}\right), \tag{6}$$

where P_k is the transmit power of vehicle k, $B_{k,r}$ denotes the bandwidth of the channel allocated for V2I transmission from vehicle k to the RSU, and σ^2 represents the Gaussian noise power. Among all vehicles, the total allocated bandwidth must be equal to or smaller than the specified dedicated bandwidth, given by

$$\sum_{k \in \mathcal{K}} a_k^r B_{k,r} \leq B_R. \tag{7}$$

In (6), $\hat{g}_{k,r}$ is the estimated gain of the channel from vehicle k to the RSU, defined as: $\hat{g}_{k,r} = \alpha_{k,r}^{\frac{1}{2}} |h_{k,r}|$, where $\alpha_{k,r}$ is the path loss determined by the distance $d_{k,r}$ between vehicle k and the RSU, given by $\alpha_{k,r} = 128.1 + 37.6 \log_{10}(d_{k,r})$ [31]. The small-scale fading $h_{k,r}$ is assumed to be exponentially distributed with a unit mean, following a complex Gaussian distribution, i.e., $h_{k,r} \sim \mathcal{CN}(0, 1)$.

After the computation period, let the feedback transmission bandwidth be $B_{r,k}$. The feedback transmit power from the RSU to vehicle k is $P_{r,k}$. Therefore, the feedback transmission rate from the RSU to vehicle k is given by

$$R_{r,k} = B_{r,k} \cdot \log_2\left(1 + \frac{P_{r,k} |\hat{g}_{r,k}|^2}{\sigma^2 + \sigma_\theta^2}\right). \tag{8}$$

To improve spectrum efficiency and avoid strong interference, frequency reuse is adopted among neighboring sub-areas when assistant vehicle offloading is enabled in sub-areas. The frequency reuse factor Δ is defined as $\Delta = \frac{1}{N}$, where N is the number of neighboring sub-areas using different frequencies for transmission. This means that the same frequency is reused in every N sub-areas, as shown in Fig. 2. In each sub-area, only one assistant vehicle offloading is operated at one instant. As there exists interference among different sub-areas, when vehicle k chooses to do assistant vehicle offloading, we use $\{i, j\}, \{i, j\} \in \mathcal{K}$ represent the index of assistant vehicle, and the respective interfering vehicle from another sub-area. Therefore, when the requesting vehicle k offloads to an assistant vehicle i, the transmission rate is given by

$$R_{k,i} = B_v \cdot \log_2\left(1 + \frac{P_{k,i} |\hat{g}_{k,i}|^2}{\sigma^2 + \sigma_\theta^2 + \sum_{j \in \mathcal{K} \setminus \mathcal{K}_m} \delta_{k,j} P_{j,i} |g_{j,i}|^2}\right), \tag{9}$$

where $B_v = B_V \cdot \Delta$ represents the channel bandwidth for each sub-area. Here, $\hat{g_{k,i}}$ is the channel gain from vehicle k to vehicle i, given by: $\hat{g_{k,i}} = \alpha_{k,i}^{\frac{1}{2}} |h_{k,i}|$, with $\alpha_{k,i}$ denoting the path loss, given as $\alpha_{k,i} = 63.3 + 17.7 \log_{10}(d_{k,i})$ [32]. $d_{k,i}$ represents the distance from vehicle k to vehicle i, and $h_{k,i}$ is the small-scale fading, following a complex Gaussian distribution. $\delta_{k,j}$ represents the channel multiplexing factor between different vehicles. When frequencies are reused between any two vehicles k and j, $\delta_{k,j} = 1$, otherwise 0. In equation (10), $P_{k,i}$ is the transmit power of the vehicle k, $P_{j,i}$ is the transmit power of the interfering vehicle j, and $g_{j,i}$ is the gain of the

channel from vehicle j to vehicle i, assuming no consideration of Doppler spread for simplicity.

After computation, the feedback transmission rate from the assistant vehicle i to vehicle k is given by

$$R_{i,k} = B_v \cdot \log_2 \left(1 + \frac{P_{i,k}|\hat{g}_{i,k})|^2}{\sigma^2 + \sigma_\theta^2 + \sum_{j \in \mathcal{K} \setminus \mathcal{K}_m} \delta_{k,j} P_{j,k} |g_{j,k}|^2} \right). \tag{10}$$

In our system, we assume the transmit power of RSU is the same for each vehicle and consider the allocation of transmit power of each vehicle by defining P^{max} as the maximum transmit power for all vehicles. The transmit powers of all vehicles are defined as

$$\mathcal{P} = \{P_1, P_2, \cdots, P_K\}, \tag{11}$$

subject to the constraint

$$0 \leq P_k \leq P^{max}, k \in \mathcal{K}. \tag{12}$$

C. Time Delay Model

To reflect the influence of task feedback, the ratio of the feedback data size to the offloaded data size D_k is defined as a variable ι_k, the feedback data size is $\iota_k D_k$ [33]. Depending on the task type, ι_k can be different. The total delay of completing a task depends on where the task is processed, which includes three cases

1) Local computation: If task U_k is processed locally, the task only experiences computation delay. The local computation delay can be expressed as

$$T_k^l = \frac{C_k}{f_k^l}, \tag{13}$$

where f_k^l is the CPU speed (cycles/s) of the on-board computation resource of vehicle k.

2) RSU offloading: If task U_k is offloaded to the RSU, the delay includes three parts. The first part is the transmission delay of the task offloading from the requesting vehicle k to the RSU, given by

$$T_{k,r}^o = \frac{D_k}{R_{k,r}}, \tag{14}$$

where $R_{k,r}$ is the offloading data rate from vehicle k to the RSU.

The second part is the computation delay at the RSU, given by

$$T_{k,r}^c = \frac{C_k}{f_k^r}, \tag{15}$$

where f_k^r is the allocated RSU computation resource for vehicle k.

The third part is caused by the feedback transmission from the RSU back to vehicle k, given by

$$T_{k,r}^f = \frac{\iota_k D_k}{R_{r,k}}, \tag{16}$$

where $R_{r,k}$ represents the feedback data rate from the RSU to vehicle k.

The total delay for RSU offloading can be calculated as

$$T_k^r = T_{k,r}^o + T_{k,r}^c + T_{k,r}^f. \tag{17}$$

Defining F^R as the total computation capacity at the RSU, the total allocated computation resource must be smaller than the total computation capacity at the RSU, given by

$$\sum_{k \in \mathcal{K}} a_k^r f_k^r \leq F^R. \tag{18}$$

3) Assistant vehicle offloading: Consider that the task of vehicle k in sub-area m is offloaded to one assistant vehicle within its sub-area. Similar to the RSU offloading, the total delay is the sum of offloading transmission delay from the requesting vehicle to the assistant vehicle, computation delays at the assistant vehicle, and the feedback transmission delay from the assistant vehicle to the requesting vehicle. The total delay for assistant vehicle offloading is given by

$$T_k^v = T_{k,i}^o + T_{k,i}^c + T_{k,i}^f, \ k \in \mathcal{K}_m, i \in \mathcal{K}_m, i \neq k, \tag{19}$$

where $T_{k,i}^o$, $T_{k,i}^c$, and $T_{k,i}^f$ represent the transmission delay from the requesting vehicle k to the assistant vehicle i, the computation delay at the assistant vehicle i, and the transmission delay from the assistant vehicle i to the requesting vehicle k, respectively. They are given as follows

$$T_{k,i}^o = \frac{D_k}{R_{k,i}}, \tag{20}$$

$$T_{k,i}^c = \frac{C_k}{f_i^l}, \tag{21}$$

$$T_{k,i}^f = \frac{\iota_k D_k}{R_{i,k}}, \tag{22}$$

where $R_{k,i}$ and $R_{i,k}$ are the offloading and feedback transmission rates between requesting vehicle k and assistant vehicle i, respectively, and f_i^l is the on-board computation resource of vehicle i.

Finally, considering the task offloading decision $\mathcal{A}$ made for vehicle k, the total delay of processing task U_k is given by

$$T_k = a_k^l T_k^l + a_k^r T_k^r + a_k^v T_k^v. \tag{23}$$

III. Problem Formulation And Solution Development

A. Problem Formulation

Considering that the offloading decision $\mathcal{A}$, RSU computation resources $f_k^r | k \in \mathcal{K}$, RSU offloading and feedback bandwidths $B_{k,r}, B_{r,k} | k \in \mathcal{K}$, and transmit power $\mathcal{P}$ of each vehicle are adjustable, we jointly adapt task offloading and

resource allocation to minimize the total delay of all vehicles. This can be formulated as the following optimization problem

$$\min_{\{\mathcal{A},f_k^r,B_{k,r},B_{r,k},\mathcal{P}\}} \sum_{k\in\mathcal{K}} T_k \tag{24}$$

$$s.t.\ C1: a_k^l, a_k^r, a_k^v \in \{0,1\},\ \forall k \in \mathcal{K}, \tag{24a}$$

$$C2: a_k^l + a_k^r + a_k^v = 1,\ \forall k \in \mathcal{K}, \tag{24b}$$

$$C3: \sum_{k\in\mathcal{K}} a_k^r f_k^r \le F^R, \tag{24c}$$

$$C4: 0 \le P_k \le P^{max}, \tag{24d}$$

$$C5: \sum_{k\in\mathcal{K}} a_k^r B_{k,r} \le B_R, \tag{24e}$$

$$C6: \sum_{k\in\mathcal{K}} a_k^r B_{r,k} \le B_R, \tag{24f}$$

$$C7: \sum_{k\in\mathcal{K}} a_k^v \le M, \tag{24g}$$

$$C8: \sum_{k\in\mathcal{K}_m} a_k^v \le 1. \tag{24h}$$

Constraint $C1$ and $C2$ guarantee that each task is processed in only one location. $C3$ ensures that the allocated computation resources at the RSU are within the total computation capacity. $C4$ ensures that the transmit power of each vehicle does not exceed its maximum power. $C5$ and $C6$ ensure that the allocated offloading and feedback bandwidths for RSU offloading are less than the total V2I bandwidth. $C7$ and $C8$ indicate that the number of assistant vehicle offloading should be less than or equal to the number of sub-areas M, and there should be at most one assistant vehicle for task offloading in one sub-area. Due to the combination of the binary constraint on offloading decision $\mathcal{A}$ and other continuous constraints, problem (24) is a non-convex MINLP problem, which is generally NP-hard [34].

The complexity of exhaustive search is extremely high, making it very difficult to obtain the optimal solution, especially when K is large. In each sub-area, multiple RSU offloadings can exist with one assistant vehicle offloading at the same time, and the resources of these two parts do not interfere. Therefore, to solve the problem iteratively, the first step of our approach is to find the sole vehicle responsible doing assistant vehicle offloading and allocate its transmit power in each sub-area. Then, the remaining problem of RSU offloading, computation, and bandwidth resource allocation can be obtained with relatively less complexity.

B. Assistant Vehicle Offloading and Power Allocation

As mentioned in the communication model, interference only occurs in assistant vehicle offloading. To achieve the maximum data rate, the RSU offloading employs maximum transmit power. However, in the case of frequency reuse across different sub-areas for vehicle offloading, power allocation becomes crucial to manage interference among vehicles utilizing the same frequency for task offloading. Thus, in order to minimize the total delay of assistant vehicle offloading, careful selection of assistant/requesting vehicles and their corresponding transmit power is critical for each sub-area. Consequently, the original problem can be transformed into

$$\begin{aligned} &\min_{\{\mathcal{A}',\mathcal{P}'\}} \sum_{k\in\mathcal{K}} T_k', \\ &s.t.\ C4, C7, C8, \\ &\quad C9': T_k^v \le T_k^l, \end{aligned} \tag{25}$$

where $T_k^{'} = a_k^v T_k^v$. $\mathcal{A}'$ represents the offloading decisions of vehicles engaging in assistant vehicle offloading, with $a_k^l = 0$ and $a_k^r = 0$. $\mathcal{P}'$ denotes the transmit power of vehicles participating in assistant vehicle offloading. $C9'$ ensures that the delay of assistant vehicle offloading is lower than that of local computation. This is because the nature of the optimization problem (25), which aims to select vehicles for assistant vehicle offloading with the smallest delay. However, there is a risk of choosing the wrong vehicle for assistant vehicle offloading when the offloading delay exceeds the local computation delay, particularly when the task size is very small. e.g. If the task size of one vehicle is significantly smaller than that of others, the delay of performing assistant vehicle offloading may be the smallest among all potential vehicles, even when the offloading delay exceeds that of processing the task locally. Consequently, if such a vehicle is chosen to do assistant vehicle offloading, the overall delay performance will deteriorate, and the computation resource of the assistant vehicle will be erroneously occupied therefore unavailable to assist others. By setting this constrain in the assistant vehicle offloading, this false selection can be prevented.

If multiple vacant vehicles are present in a sub-area for task offloading, all of these vehicles are considered as potential assistant vehicles. The final assistant vehicle is chosen based on the one with the lowest delay.

For each potential vacant assistant vehicle, the remaining vehicles with computation tasks are selected as potential requesting vehicles. By optimizing the offloading decision and transmit power, the total delay $T_k^{'}$ can be minimized, leading to the optimal solution for assistant vehicle offloading. To explore all possible solutions, the total number of combinations for assistant vehicle offloading and transmit power allocation can be calculated as

$$(N_P)^{2M} \prod_{m=1}^{M} [(K_m - L_m)L_m], \tag{26}$$

where K_m and L_m represent the total number of vehicles and vacant vehicles in sub-area m, respectively. N_P denotes the number of transmit power choices for each vehicle, i.e., $P_k \in \{P_k^1, P_k^2, \cdots, P_k^{N_p}\}$.

It can be seen that the complexity of exhaustive searching to find the optimal solution is high and impractical when M and K are large. This is because the selection of vehicles and their transmit power can affect each other due to interference, making the problem challenging to solve using traditional methods. Heuristic algorithms are commonly used search methods to solve complex programming problems with low complexity [35]. Particle swarm optimization (PSO) algorithm [36] is one effective method due to its convergence speed and fewer parameters compared to other algorithms [37]. To

address the solution finding and complexity reduction, we propose a low-complexity heuristic approach inspired by the search concept of the PSO algorithm.

The PSO algorithm operates by employing multiple particles to explore the search space and find the best solution. Each particle represents a potential solution and moves within a d-dimensional search space. Vectors $\mathbf{x}_i = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{id}]$ and $\mathbf{v}_i = [\mathbf{v}_{i1}, \mathbf{v}_{i2}, \cdots, \mathbf{v}_{id}]$ are used to represent the location and velocity of the i-th particle, respectively. The location of each particle represents a potential solution. The velocity of each particle is updated based on: (1) the best location $l^{'}$ the particle has found so far and (2) the global best location l^{*} obtained from all particles in the swarm. By combining the velocity vector with the location vector, PSO progressively explores the potential search space. We define the random function $\mathcal{R}$ to generate parameters between 0 and 1. The updated equations for each particle's schedule are as follows

$$\begin{aligned}\mathbf{v}_i[t+1] =& w[t]\mathbf{v}_i[t] + c_c \cdot \mathcal{R} \cdot (l_i^{'}[t] - \mathbf{x}_i[t]) + \\ & c_s \cdot \mathcal{R} \cdot (l^{*}[t] - \mathbf{x}_i[t]),\end{aligned} \tag{27}$$

$$\mathbf{x}_i[t+1] = \mathbf{x}_i[t] + \mathbf{v}_i[t+1], \tag{28}$$

where c_c and c_s are predetermined cognitive and social factors, respectively. $l_i^{'}[t]$ and $l^{*}[t]$ represent the best location of particle i and the best location among all I particles at the t-th iteration, respectively. The weight factor $w[t]$ controls the step size of the particle's velocity. As the iteration step t increases, $w[t]$ decreases. The weight factor can be computed using

$$w[t] = w_i - \frac{(w_i - w_f) \cdot t}{T}, \tag{29}$$

where w_i and w_f denote the initial and final weight factors, respectively. T represents the maximum number of iteration steps.

As problem (25) is still an MINLP problem, the traditional PSO algorithm cannot be directly applied. This is because the selection of requesting vehicles is an integer set, while PSO generates continuous-valued results. Therefore, the original results of offloading decisions from PSO need to be adjusted based on $\mathcal{A}^{'}$. Another issue with PSO is that the initial particle selection can affect the final results. A good initial particle selection can improve searching accuracy and overall performance. Thus, it is necessary to adjust the random initial particle generation scheme in the original PSO algorithm according to the offloading problem. To address these challenges, we propose a modified PSO algorithm as follows.

The optimization problem is divided into several parts for the PSO algorithm. In order to minimize the total delay, the fitness function is defined as

$$\mathcal{F} = \sum_{k \in \mathcal{K}} T_k^{'}. \tag{30}$$

To formulate the particles in our approach, the next step is to model the offloading decisions and the allocated transmit power. Our objective is to find the best particle that minimizes latency. Let the $4M$-dimensional vector $\mathbf{x}$ represent a solution

$$\begin{aligned}\mathbf{x} =& [r_1, r_2, \cdots, r_M, P_1, P_2, \cdots, P_M, \\ & r_1^{'}, r_2^{'}, \cdots, r_M^{'}, P_1^{'}, P_2^{'}, \cdots, P_M^{'}] \\ =& [\mathbf{x}_r, \mathbf{x}_P, \mathbf{x}_r^{'}, \mathbf{x}_P^{'}],\end{aligned} \tag{31}$$

where $\mathbf{x}_r$ and $\mathbf{x}_P$ represent the set of potential requesting vehicles in M sub-areas and their respective transmit powers. $\mathbf{x}_r^{'}$ and $\mathbf{x}_P^{'}$ are the sets of assistant vehicles and their transmit powers. $r_m = k, r_m^{'} = i, k, i \in \mathcal{K}_m$ means vehicle k/i is the requesting/assistant vehicle. In each particle, to satisfy constraint $C8'$, if $T_k^v \leq T_k^l$, $a_{r_m}^v = 1$. Otherwise, $a_{r_m}^v = 0$.

To enhance convergence behavior and reduce complexity, we select the initial requesting vehicles using roulette wheel selection based on the channel gains

$$p_{k,i} = \frac{g_{k,i}}{\sum\limits_{k,i \in \mathcal{K}_m} g_{k,i}}, \tag{32}$$

where $p_{k,i}$ is the probability of selecting vehicle k as the initial requesting vehicle of $\mathbf{x}_r$ and vehicle i as the initial assistant vehicle of $\mathbf{x}_r^{'}$. The selection of initial transmit power is obtained as follows:

$$P_k = \mathcal{D} \cdot (P^{max}), \tag{33}$$

where $\mathcal{D}$ is a random value generated between 0.5 and 1.

In each iteration, continuous variables are introduced by PSO. As r_m represents the integer index of vehicle within sub-area m, we round r_m using the truncation function $\lfloor \bullet \rfloor$ and let r_m^{min}/r_m^{max} represent the smallest/largest index of vehicle in sub-area m. Then, $\mathbf{x}_r$ can be updated

$$r_m = \begin{cases} \lfloor r_m \rfloor, \lfloor r_m \rfloor \in \mathcal{K}_m, \\ \underset{r_m^{min}, r_m^{max}}{\arg\min} \{|r_m^{min} - r_m|, |r_m^{max} - r_m|\}, \lfloor r_m \rfloor \notin \mathcal{K}_m. \end{cases} \tag{34}$$

Similarly, $\mathbf{x}_r^{'}$ is obtained using the same method.

Next, we check the boundary of the transmit power to satisfy constraint $C4$. Since the requesting vehicles are offloading their tasks for computation, the minimum transmit power should be positive. Let P^{min} be the minimum transmit power among all vehicles. Constraint $C4$ can be rewritten as

$$C4 : P^{min} \leq P_k \leq P^{max}. \tag{35}$$

If the transmit power of a particle exceeds the constraint bound, $\mathbf{x}_P$ is adjusted to the constraint bound as follows

$$P_m = \begin{cases} P^{min}, P_m < P^{min}, \\ P_m, P^{min} \leq P_m \leq P^{max}, \\ P^{max}, P_m > P^{max}. \end{cases} \tag{36}$$

$\mathbf{x}_P^{'}$ is obtained similarly.

During the iteration process, the particle with the highest revenue, representing the best solution of problem (25) so far, is preserved. After the iterations, the final solution, accepted by all assistant and requesting vehicles, can be obtained.

In summary, the process of our proposed modified PSO algorithm is shown in Algorithm 1, and the final solution

Algorithm 1 Modified PSO Algorithm

Initialize: At the iteration step $t = 0$, for particle i, choose one of the vacant vehicles as the assistant vehicle in each sub-area as the assistant vehicle vector $\mathbf{x}_r^{'}$ in (31). Using (32) to choose other vehicles as the requesting vehicles vector $\mathbf{x}_r$ in (31). Using (33) to choose the respect transmit power vector $\mathbf{x}_P$ and $\mathbf{x}_P^{'}$ in (31). In the same way, initialize the other $I-1$ particles repetitively.

Input: $\mathbf{x}$ and the maximum iteration time T_I.

While $t < T_I$ **do**

Fitness value calculation: calculate (30) of each particle.

Local best location: save the particle with the lowest fitness value as the local best location $l_i^{'}[t]$ of each particle i.

Global best location: save the particle with the lowest fitness value as the global best location $l^*[t]$ of the whole particles.

Update the velocity of each particle using equation (27).

Update the location of each particle using equation (28).

Reform $\mathbf{x}_r$ and $\mathbf{x}_r^{'}$ of each particle using equation (34), and $\mathbf{x}_P$ and $\mathbf{x}_P^{'}$ of each particle using equation (36), respectively.

Update the location of each particle $\mathbf{x} = [\mathbf{x}_r, \mathbf{x}_P, \mathbf{x}_r^{'}, \mathbf{x}_P^{'}]$.

Update the iteration time: $t = t + 1$

end while

Output: The global best location $l^*[t]$ which represents the selection and transmit power of the requesting and assistant vehicles.

for assistant vehicle offloading decision and transmit power allocation can be obtained.

For the proposed PSO algorithm, the complexity lies in search steps and number of particles. Let the T_I denote the maximum number of iterations needed for the PSO algorithm. The swarm size is I. The dimension of the particle is $4M$. Then, we can conclude that the complexity of PSO algorithm is $\mathcal{O}(IT_IM)$.

C. RSU Offloading/Local Computation, Computing and Bandwidth Allocation

After obtaining the offloading decisions and power allocations for assistant vehicles $(\mathcal{A}^{'}, \mathcal{P}^{'})$ from problem (27), the next step is to address the RSU offloading and local computation. In this step, we exclude the results obtained for assistant vehicles from the original problem (26). The remaining set of vehicles for RSU offloading and local computation is denoted as $\mathcal{K}^*$. The formulation of the original optimization problem is then as follows

$$\min_{\{\mathcal{A}^*, B_{k,r}, B_{r,k}, \mathcal{P}^*\}} \sum_{k\in\mathcal{K}^*} T_k^* \tag{37}$$

$$s.t.\ C1^*: a_k^l, a_k^r \in [0,1],\ \forall k \in \mathcal{K}^*, \tag{37a}$$

$$C2^*: a_k^l + a_k^r = 1,\ \forall k \in \mathcal{K}^*, \tag{37b}$$

$$C3^*: \sum_{k\in\mathcal{K}^*} a_k^r f_k^r \le F^R, \tag{37c}$$

$$C4^*: \sum_{k\in\mathcal{K}^*} a_k^r B_{k,r}, \le B_R, \tag{37d}$$

$$C5^*: \sum_{k\in\mathcal{K}^*} a_k^r B_{r,k}, \le B_R, \tag{37e}$$

where $\mathcal{A}^*$ represents the decisions of vehicles doing local computation and RSU offloading. $\mathcal{P}^*$ represents the respective transmit power. In the proposed communication model, we assume $P_{r,k}$ is the same for all vehicles. When a vehicle k is engaged in local computation, the transmit power P_k is set to 0 since no transmission is required. Conversely, when a vehicle k is involved in RSU offloading, the transmit power P_k is set to its maximum value P^{max} to achieve the maximum transmission rate. Therefore,

$$P_k = \begin{cases} 0, a_k^l = 1, \\ P^{max}, a_k^r = 1. \end{cases} \tag{38}$$

In (39), T_k^* is the delay of local computation and RSU offloading, given by

$$\begin{aligned} T_k^* =& a_k^l T_k^l + a_k^r T_k^r = a_k^l \frac{C_k}{f_k^l} + a_k^r \left(\frac{D_k}{R_{k,r}} + \frac{C_k}{f_k^r} + \frac{\iota_k D_k}{R_{r,k}}\right) \\ =& a_k^l \frac{C_k}{f_k^l} + a_k^r \left(\frac{D_k}{B_{k,r}\cdot \log_2(1+\frac{P_k|\hat{g}_{k,r}|^2}{\sigma^2+\sigma_\theta^2})} + \right. \\ & \left. \frac{C_k}{f_k^r} + \frac{\iota_k D_k}{B_{r,k}\cdot \log_2(1+\frac{P_{r,k}|\hat{g}_{r,k}|^2}{\sigma^2+\sigma_\theta^2})}\right). \end{aligned} \tag{39}$$

From [38], it is known that if a function $f(x) = z/x$ is convex with respect to x, its perspective function $g(y,x) = yf(y/x)$ is also convex with respect to (y,x). Therefore, in order to handle the integer constraint $C1^*$, we first relax it and then transform problem (37) into a convex problem by making specific substitutions. Subsequently, we propose an iterative algorithm based on the Lagrange dual method to solve problem (37) with small step sizes. The variable substitutions are defined as follows

$$\begin{aligned} &\varepsilon_k = a_k^l f_k^l, \phi_k = a_k^r f_k^r, \\ &\varphi_k = a_k^r B_{k,r}, \psi_k = a_k^r B_{r,k}, \\ &\gamma_k = \log_2(1+\frac{P_k|\hat{g}_{k,r}|^2}{\sigma^2+\sigma_\theta^2}), \varpi_k = \log_2(1+\frac{P_{r,k}|\hat{g}_{r,k}|^2}{\sigma^2+\sigma_\theta^2}). \end{aligned} \tag{40}$$

After applying the introduced substitutions and removing the integer constraint $C1^*$, the optimization problem (37) can be transformed into the following form

$$\begin{aligned} \min_{\{\mathcal{A}^*, f_k^r, B_{k,r}, \mathcal{P}^*\}} & \sum_{k\in\mathcal{K}^*} T_k^* = (a_k^l)^2 \frac{C_k}{\varepsilon_k} + \\ & (a_k^r)^2 \frac{C_k}{\phi_k} + (a_k^r)^2 \frac{D_k}{\varphi_k \gamma_k} + (a_k^r)^2 \frac{\iota_k D_k}{\psi_k \varpi_k} \end{aligned} \tag{41}$$

$$s.t.\ C3^*: \sum_{k\in\mathcal{K}^*} \phi_k \le F^R, \tag{41a}$$

$$C4^*: \sum_{k\in\mathcal{K}^*} \varphi_k \le B_R, \tag{41b}$$

$$C5^*: \sum_{k\in\mathcal{K}^*} \psi_k \le B_R, \tag{41c}$$

$$C6^*: \varepsilon_k = f_k^l. \tag{41d}$$

Here, a new constraint $C5^*$ is introduced to ensure that the computation resource allocated for local computation always corresponds to the remaining local computation resource. This

constraint ensures that all of a vehicle's computation resource is utilized to minimize the computation latency.

Note that Problem (41) is a convex problem, $Proof$: see Appendix A. The Lagrange dual method is employed to solve it. By introducing non-negative variables $\eta_k, \mu_k, \sigma_k, \xi_k$, we can derive the Lagrange function

$$\begin{aligned}\mathcal{L} =& \overset{*}{T_k} + \sum_{k\in\mathcal{K}^*} \eta_k(\varepsilon_k - f_k^l) + \sum_{k\in\mathcal{K}^*} \mu_k(\phi_k - F^R) \\ &+ \sum_{k\in\mathcal{K}^*} \sigma_k(\varphi_k - B_R) + \sum_{k\in\mathcal{K}^*} \xi_k(\psi_k - B_R).\end{aligned} \tag{42}$$

Then, by taking the first order derivatives of $\varepsilon_k, \phi_k, \varphi_k, \psi_k$ respectively, one obtains

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_k} = -(a_k^l)^2 \frac{C_k}{{\varepsilon_k}^2} + \eta_k, \tag{43}$$

$$\frac{\partial \mathcal{L}}{\partial \phi_k} = -(a_k^r)^2 \frac{C_k}{{\phi_k}^2} + \mu_k, \tag{44}$$

$$\frac{\partial \mathcal{L}}{\partial \varphi_k} = -(a_k^r)^2 (\frac{D_k}{{\varphi_k}^2 \gamma_k}) + \sigma_k. \tag{45}$$

$$\frac{\partial \mathcal{L}}{\partial \psi_k} = -(a_k^r)^2 (\frac{\iota_k D_k}{{\psi_k}^2 \varpi_k}) + \xi_k. \tag{46}$$

It can be seen that when ${a_k^l}^* = 0, {\varepsilon_k}^* = 0$, when ${a_k^r}^* = 0, ({\phi_k}^*, {\varphi_k}^*, {\psi_k}^*) = 0$, and from $C1^*$ and $C2^*$, when $(a_k^l, a_k^r) \neq 0$, we can obtain the optimal values of $(\varepsilon_k, \phi_k, \varphi_k, \psi_k)$ respectively

$${\varepsilon_k}^* = {a_k^l}^* \sqrt{\frac{C_k}{\eta_k}}, \tag{47}$$

$${\phi_k}^* = {a_k^r}^* \sqrt{\frac{C_k}{\mu_k}}, \tag{48}$$

$${\varphi_k}^* = {a_k^r}^* \sqrt{\frac{D_k}{\sigma_k \gamma_k}}. \tag{49}$$

$${\psi_k}^* = {a_k^r}^* \sqrt{\frac{\iota_k D_k}{\xi_k \varpi_k}}. \tag{50}$$

Next, by taking the first order derivatives of a_k^l and a_k^r respectively, we get the equations as follows

$$\frac{\partial \mathcal{L}}{\partial a_k^l} = 2\frac{C_k}{\varepsilon_k / a_k^l}, \tag{51}$$

$$\frac{\partial \mathcal{L}}{\partial a_k^r} = 2\frac{C_k}{\phi_k / a_k^r} + 2\frac{D_k}{\varphi_k \gamma_k / a_k^r} + 2\frac{\iota_k D_k}{\psi_k \varpi_k / a_k^r}. \tag{52}$$

Define denotations

$$\varrho_k = \frac{\partial \mathcal{L}}{\partial a_k^l} (\frac{{\varepsilon_k}^*}{{a_k^l}^*}), \tag{53}$$

$$\zeta_k = \frac{\partial \mathcal{L}}{\partial a_k^r} (\frac{{\phi_k}^*}{{a_k^r}^*}, \frac{{\varphi_k}^* \gamma_k}{{a_k^r}^*}, \frac{{\psi_k}^* \varpi_k}{{a_k^r}^*}). \tag{54}$$

From $C2^*$, the optimal offloading decision $\mathcal{A}^*$ can be obtained as

$$\begin{cases} a_k^l = 1, a_k^r = 0, when\ \varrho_k < \zeta_k, \\ a_k^l = 0, a_k^r = 1, when\ \zeta_k < \varrho_k. \end{cases} \tag{55}$$

Sub-gradient method [39] can be used to obtain the values of the variables $\eta_k, \mu_k, \sigma_k, \xi_k$ and it converges to the optimal result within small steps of iterations. The updated parameters are shown in Appendix B. Therefore, Problem (37) can be solved successfully afterwards.

We summarize the above integer constraint relaxation and iterative method in Algorithm 2.

The proposed ICRI algorithm adopts the Lagrangian dual subgradient methods. The complexity comes from two parts. For the first part of optimizing the Lagrangian dual with given dual variables, the main complexity lies in solving the transaction equations (43)-(46), as the rest are provided in closed form. When adopting the Newton method, the required number of iterations in the worst case is denoted by T_L. From $\mathcal{K}^*$, the number of equations to solve is $K-M$. For the second part, the complexity is to find the dual variables that maximize the Lagrangian using the subgradient method. In Problem (37), the number of dual variables that need to be updated is $4K - 4M$. Let T_D be the number of subgradient updates needed. Thus, the total complexity of the ICRI algorithm is given by $\mathcal{O}((K-M)T_L(4K-4M)T_D) = \mathcal{O}((K-M)^2 T_L T_D)$.

Algorithm 2 Integer Constraint Relaxation Iterative (ICRI) Algorithm

Initialize: Set $\varepsilon_k^{(0)} = 0, \phi_k^{(0)} = 0, \varphi_k^{(0)} = 0, \psi_k^{(0)} = 0, a_k^l = 0, a_k^r = 0$ and the precision parameters ϵ, $t = 0$.
Get $\eta_k^{(0)}, \mu_k^{(0)}, \sigma_k^{(0)}, \xi_k^{(0)}$ respectively
While $|{T_k}^{*(t+1)} - {T_k}^{*(t)}| < \epsilon$ **do**
  Using equations (47)-(50) and (55) to obtain $\varepsilon_k^{(t)}, \phi_k^{(t)}, \varphi_k^{(t)}, \psi_k^{(t)}, {a_k^l}^{(t)}$ and ${a_k^r}^{(t)}$, respectively.
  Update $\eta_k^{(t)}, \mu_k^{(t)}, \sigma_k^{(t)}, \xi_k^{(t)}$ by using equations (64)-(67) respectively.
  Update the objective function $T_k^{*(t)}$ from problem (37).
  $t = t + 1$
end while
Output: The value of objective function T_k^*.

D. Decentralized Low Complexity/Overhead Algorithm for RSU Offloading/Local Computation

In the ICRI algorithm, all vehicle's computation requirements such as the onboard computation resource and the task size of each vehicle, and their CSI should be collected at the MEC server. This centralized coordination will increase the overhead for information exchange when the number of vehicles K is large. In order to further reduce the complexity, we proposed a decentralized allocation algorithm where the MEC does not need the computation requirements and CSI of each vehicle.

The decentralized allocation is a sub-optimal matching algorithm based on the update of three parameters, namely offloading gain of each vehicle, index of number of RSU offloading, and total maximum positive offloading gain. For simplicity, we adopt equal distribution for the bandwidth [18] and computation resource between vehicles. The detailed procedure is presented as follows.

Step 1: For each vehicle, compute the delay of local computation with equation (13) and RSU offloading with the given number of RSU offloading with equation (14)-(17). The delay of RSU offloading can be written as

$$T_k^r = a_k^r\left(\frac{D_k}{B_R/N^* \cdot \log_2(1+\frac{P_k|\hat{g}_{k,r}|^2}{\sigma^2+\sigma_\theta^2})} + \frac{C_k}{f_R/N^*} + \frac{\iota_k D_k}{B_R/N^* \cdot \log_2(1+\frac{P_{r,k}|\hat{g}_{r,k}|^2}{\sigma^2+\sigma_\theta^2})}\right), \tag{56}$$

where N^* is the updated number of RSU offloading.

Step 2: Compute the offloading gain of each vehicle using RSU offloading delay T_k^r and local computation delay T_k^l when it performs local computation as

$$G_{k,r} = T_k^r - T_k^l. \tag{57}$$

Update N^* from 1 to $K-M$ and get the $(K-M)\times(K-M)$ dimension vector of offloading gain of all remaining vehicle $\mathbf{G}_{k,r}$.

Step 3: Compute the actual offloading gain of each N^* with the maximum N^* value of $\mathbf{G}_{k,r}$.

$$G_{N^*} = \sum_{k\in\mathcal{K}} \mathbf{G}_{k,r}. \tag{58}$$

Step 4: The number of RSU offloading with the total maximum offloading gain will be selected and these N^* vehicles are selected for RSU offloading with $a_k^r = 1$ and the others for local computation with $a_k^l = 1$. The total delay then can be obtained as

$$\begin{aligned}\sum_{k\in\mathcal{K}} T_k^* &= \sum_{k\in\mathcal{K}}\{a_k^l\frac{C_k}{f_k^l} + a_k^r(\frac{D_k}{R_{k,r}} + \frac{C_k}{f_k^r} + \frac{\iota_k D_k}{R_{r,k}})\} \\ &= \sum_{k\in\mathcal{K}}\{a_k^l\frac{C_k}{f_k^l} + a_k^r(\frac{D_k}{B_R/N^* \cdot \log_2(1+\frac{P_k|\hat{g}_{k,r}|^2}{\sigma^2+\sigma_\theta^2})} + \\ &\quad \frac{C_k}{f_R/N^*} + \frac{\iota_k D_k}{B_R/N^* \cdot \log_2(1+\frac{P_{r,k}|\hat{g}_{r,k}|^2}{\sigma^2+\sigma_\theta^2})})\}.\end{aligned} \tag{59}$$

We summarize the above decentralized offloading allocation method in Algorithm 3.

Algorithm 3 Decentralized Offloading Allocation Algorithm

Initialize: Set $N^* = 1$.
While $N^* \le K - M$ **do**
  Get T_k^l, T_k^r using (13)-(17) respectively. Then calculate the total offloading gain $G_{k,r}$ from (57). Update the offloading gain vector $\mathbf{G}_{k,r}$.
  $N^* = N^* + 1$
end while
for $N^* < K - M$
  Calculate the actual offloading gain from (58).
end for
  Select the N_* with the maximum total offloading gain, and compute the total delay from (59) as the output.

The complexity of the decentralized algorithm lies to the selection of number of vehicle to do RSU offloading. After the assistant vehicle offloading in section B, there are $(K-M)$ remaining vehicles for RSU offloading/local computation. To find the maximum total offloading gain from $\mathbf{G}_{k,r}$, the Bisection method can be adopted. Therefore, the complexity can be obtained as $O((K-M)\log(K-M))$.

E. Offloading Decision and Power Allocation Combination

As the decentralized allocation for RSU offloading/local computation in section D is a sub-optimal solution, we obtain $\mathcal{A}$ and $\mathcal{P}$ by combining $\mathcal{A}^*$ and $\mathcal{P}^*$ with $\mathcal{A}^{'}$ and $\mathcal{P}^{'}$ after obtaining the results of offloading decisions and resource allocations from sections B and C:

$$\begin{cases}\mathcal{A} = \{\mathcal{A}^*, \mathcal{A}^{'}\}, \\ \mathcal{P} = \{\mathcal{P}^*, \mathcal{P}^{'}\}.\end{cases} \tag{60}$$

Together with $f_k^r, B_{k,r}|k\in\mathcal{K}$ from section C, the adjustable variables of the offloading decision $\mathcal{A}$, RSU computation $f_k^r|k\in\mathcal{K}$, RSU offloading bandwidth $B_{k,r}|k\in\mathcal{K}$, RSU feedback bandwidth $B_{r,k}|k\in\mathcal{K}$, and transmit power $\mathcal{P}$ of the original problem (24) can then be obtained.

TABLE I
SIMULATION PARAMETERS [40], [41]

Parameter	Value
RSU coverage	500 m
V2V bandwidth (B_V)	2 MHz
V2I bandwidth (B_R)	[20-80] MHz
Number of sub-area (M)	[4, 6, 8]
Frequency reuse factor (N)	[1-3]
MEC capacity (F^R)	[1-4] $\times 10^4$ GHz
Maximum transmission power (P_{max})	30 dBm
Noise power (σ^2)	-114 dBm
Vacant probability (p_v)	[0.05-0.25]
Ratio of feedback to offloaded data size (ι_k)	[0.1-0.3]
Distance between RSU and the road	35m
Width of lane	4m
Vehicle distribution model	homogeneous Poisson process
RSU antenna height	25m
Absolute vehicle speed	[0-110] km/h

TABLE II
MODIFIED PARAMETERS

Parameter	Value
Number of particles (I)	512
Number of iteration (T)	128
Cognitive factor (c_c)	2.03
Social factor (c_s)	2.03
Weight factor (w_i)	0.9
Weight factor (w_f)	0.5

IV. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

In this section, the performance of our proposed MEC-V2X assisted vehicular task offloading system is evaluated through simulations. The simulation comprises one RSU and several vehicles uniformly scattered along the road. The road is divided into multiple sub-areas. When a task is assigned to a vehicle, its size is uniformly generated between 5 Mbits and 20 Mbits, and the required computation frequency is distributed uniformly in the range of [5, 20] Mcycles [33]. Unless noted otherwise, we set the number of sub-areas $M = 4$, the frequency reuse factor $\Delta = 1/2$, and the vacant probability

$p_v = 0.25$. The simulation utilizes various parameters summarized in Table I. Furthermore, Table II presents the parameters employed in our proposed modified PSO algorithm.

To facilitate comparison, we designate our proposed modified PSO algorithm for assistant vehicle offloading as"V2V-PSO". For the second part of RSU offloading/local computation, the proposed optimal ICRI algorithm is denoted as "ICRI". The decentralized algorithm is labelled as "DEC". When referring to the entire MEC-V2X offloading scheme after combining these two parts, we use the term "MEC-V2X". Furthermore, we present the performance results of the following bench mark methods for comparison

- Local computation scheme (labeled as "Local-fixed"): All tasks of vehicles are executed locally using their own computation resources.
- Pure MEC offloading scheme (labelled as "Pure-MEC") : All tasks of vehicles are sent to the RSU for MEC processing [42].
- Adaptive MEC offloading scheme (labeled as "MEC-Local"): The traditional adaptive binary MEC offloading scheme where vehicles choose to either offload their tasks to the RSU or perform local computation [14].

Simulations of the assistant vehicle offloading, RSU offloading/local computation, and the final MEC-V2X scheme are given in the subsequent sub-sections.

A. Assistant Vehicle Offloading

Fig. 3 illustrates the total delay experienced by vehicles performing assistant vehicle offloading under different schemes: local computation (Local-fixed), random selection of offloading vehicle with maximum transmit power (V2V-random-fixed) [43], our proposed PSO-based offloading decision and transmit power allocation (V2V-PSO), optimal offloading decision with maximum transmit power (V2V-optimal-fixed), and optimal offloading decision and transmit power selection using exhaustive search (V2V-optimal-exhausted) [16]. It can be observed that the total delay of vehicles engaged in assistant vehicle offloading decreases, except for the random offloading scheme, as the average number of vehicles in each sub-area increases. As there is only one vehicle offloading in each sub-area, the reduction of the total delay is attributed to the larger pool of potential vehicles available for selection as requesting vehicles when there are more vehicles on the road. Vehicles with better channel conditions are more likely to be chosen for assistant vehicle offloading, resulting in reduced delays. In the case of random offloading, the likelihood of selecting the optimal assistant vehicle is low, particularly when the number of vehicles is large. Hence, the performance remains nearly unchanged. For simplicity and clarity, we assume that all vehicles possess the same amount of computation resources, ensuring that the latency associated with local computation remains constant. Importantly, all the mentioned assistant vehicle offloading schemes outperform local computation, demonstrating the effectiveness of our proposed algorithm in reducing task delays through vehicle selection and transmit power adjustment.

In Section III, the complexity of exhaustive searching for offloading selection with maximum transmit power can be calculated as $(6 \times 2)^4$ when there are 6 vehicles with tasks and 2 vacant vehicles in one sub-area. For exhaustive searching involving both offloading selection and transmit power selection, the complexity can be estimated as $((6 \times 2)^4 \times 5^8)$ when the number of transmit power level N_P is set to 5. In this case, exhaustive searching is considered to provide the optimal solution. In contrast, the searching complexity of our proposed modified PSO algorithm is only 512×128. The PSO algorithm outperforms random offloading selection with maximum transmit power, and its performance is further enhanced by adjusting the transmit power compared to the optimal offloading selection obtained through exhaustive searching with maximum transmit power. Although the PSO algorithm's performance is slightly lower (approximately 4%) than the optimal solution (V2V-optimal-exhausted), it significantly reduces the searching complexity. Consequently, we can conclude that our proposed PSO algorithm achieves near-optimal performance with low complexity.

Fig. 4 illustrates the impact of the vacant ratio p_v on the total delay, with a total of $K = 28$ vehicles. As the ratio of vacant vehicles increases, both random offloading and our proposed modified PSO method exhibit a decrease in total delay. This tendency slows down when the ratio reaches approximately 0.2. This behavior can be attributed to the higher likelihood of having a vacant assistant vehicle in a sub-area, which increases the possibility of offloading to reduce delay. Once the probability is sufficiently high, a vacant assistant vehicle is always available for offloading. When the ratio is 0.05, the performance of random offloading is comparable to that of local computation. This is because a low ratio of vacant vehicles implies fewer opportunities for offloading, resulting in the vehicle executing the task locally in the absence of an assistant vehicle. Furthermore, since the improvement in reducing delay achieved by random offloading is relatively small compared to our proposed algorithm, the overall performance of our PSO method is superior.

Fig. 5 presents the average transmit power, considering different schemes: our proposed modified PSO algorithm (PSO), the maximum transmit power scheme (P-max), and the predefined transmit power [43] obtained through searching (P-exh). It can be seen that the proposed algorithm exhibits lower average transmit power compared to the other two schemes. This reduction can be attributed to the ability of our PSO algorithm to explore a wider range of transmit power options beyond the predefined N_P choices, thereby reducing energy consumption. As the number of vehicles in each sub-area increases, the total power decreases. This is attributed to the larger pool of potential vehicles available for selection, vehicles with better channel conditions are more likely to be chosen for assistant vehicle offloading, resulting in reduced power.

Fig. 6 demonstrates the impact of the average ratio of data transmission and local computation on delay. As the ratio of transmission to local computation increases, the delay also increases. The random offloading scheme outperforms the local computation scheme when the ratio is less than 1.4.

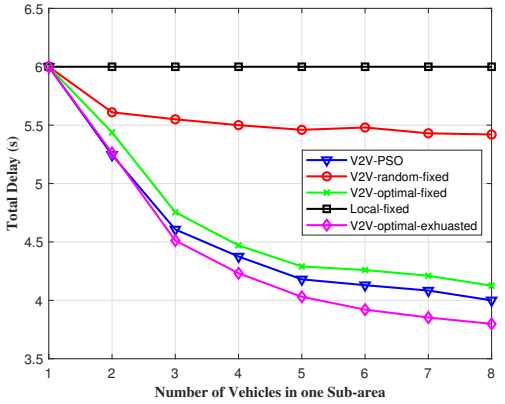


Fig. 3. Total delay of assistant vehicle offloading as a function of the number of vehicles in one sub-area.

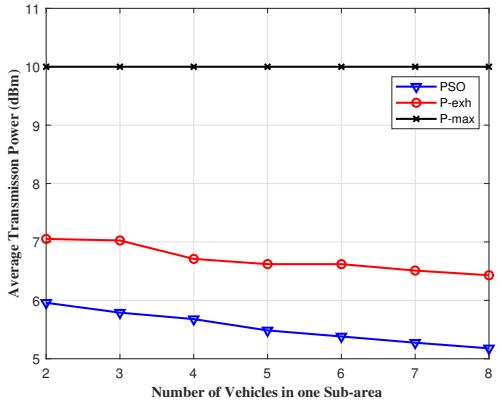


Fig. 5. Average transmission power as a function of the number of vehicles in one sub-area.

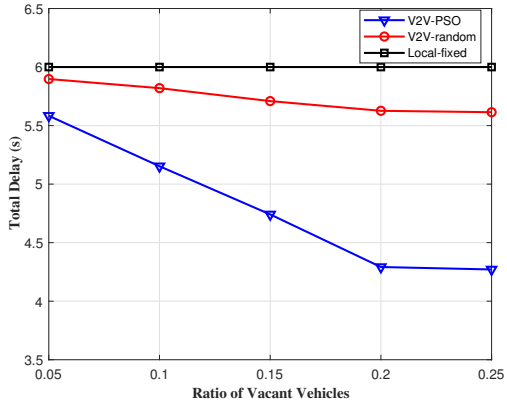


Fig. 4. Total delay of assistant vehicle offloading as a function of ratio of vacant vehicles.

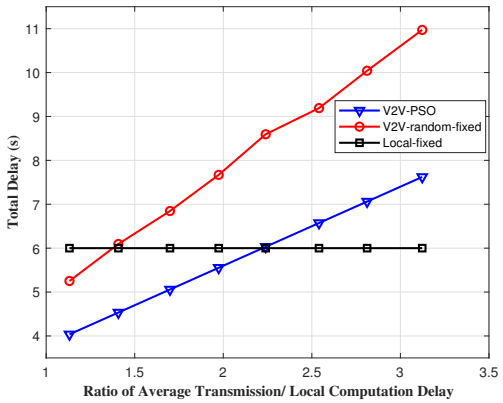


Fig. 6. Total delay of assistant vehicle offloading with the ratio of transmission and computation delay.

However, our assistant vehicle offloading scheme with the modified PSO algorithm improves delay performance when the ratio is less than approximately 2.25. This behavior arises from the fact that when the transmission delay becomes too long, the benefits gained from assistant computation are counterbalanced or even worsened. In other words, selecting the assistant vehicle for offloading is not advisable when the data size exceeds a certain threshold for transmission.

Fig. 7 illustrates the total delay of assistant vehicle offloading as the maximum Doppler frequency varies from 0Hz to 600Hz. With a carrier frequency of 5.9GHz, the relative speed is generated uniformly from 0 to 110 km/h. It can be observed that the total delay increases with the increase in maximum Doppler frequency especially when the speed of vehicle is high. This is attributed to the reduction in channel gain caused by the Doppler effect, resulting in increased transmission delay. When compared to the local computation scheme and random offloading scheme, it is seen that the proposed algorithm effectively reduces the total delay by allocating the reasonable vehicles and their transmit power for offloading.

When the offloading vehicle is at the boundary of the sub-area, there is a probability that the vehicle moves out of its designated sub-area. Fig. 8 illustrates the ratio at which the requesting vehicle moves to another sub-area for different choices of the frequency reuse factor $\Delta = 1/N$ with different Doppler frequency f_D(Hz). To compare the influence of different N, the number of sub-areas M is 6, and N ranges from 1 to 3. From Fig. 8, it is observed that as the ratio increases, the total delay remains nearly stable for $N = 2$ and $N = 3$, while it increases for $N = 1$. This behavior arises because when adjacent sub-areas use different frequencies ($N = 2, N = 3$), the assistant vehicle of the next sub-area does not cause interference as the requesting vehicle moves to it. However, when both sub-areas occupy the same frequency ($N = 1$), strong interference occurs as the requesting vehicle approaches the adjacent sub-area. Consequently, the delay performance significantly deteriorates as the ratio increases. Nevertheless, even when the ratio equals 0.5, the performances of the other two choices show only slight changes, with $N = 2$ consistently exhibiting the best performance. Therefore, we can conclude that the impact of a requesting vehicle moving out of its designated sub-area can be mitigated by selecting $N > 1$, and under the given conditions, the optimal choice for N is 2.

Fig. 9 shows the assistant vehicle offloading delay perfor-

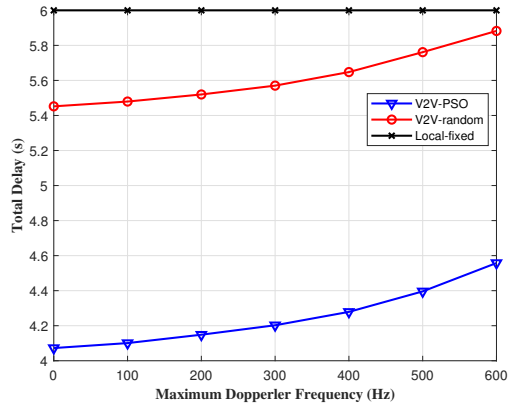


Fig. 7. Total delay of assistant vehicle offloading as a function of maximum Doppler frequency.

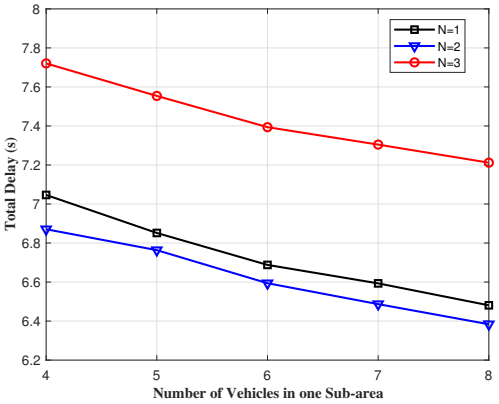


Fig. 9. Total delay of assistant vehicle offloading as a function of number of vehicles in one sub-area.

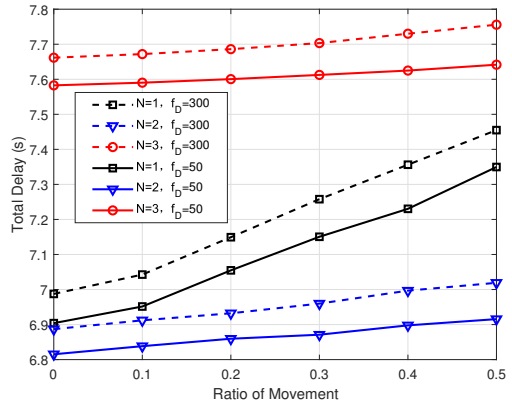


Fig. 8. Total delay of assistant vehicle offloading with the ratio of requesting vehicle moves out to another sub-area.

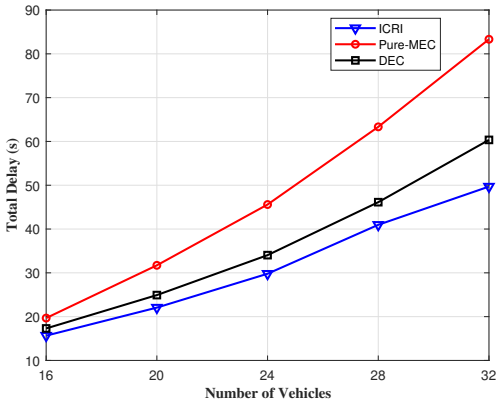


Fig. 10. Total delay of RSU offloading as a function of number of vehicles.

mance of adopting different frequency reuse factor $\Delta = 1/N$. As the number of vehicles increases, the total delays of all three gradually decrease due to the availability of more choices for offloading. The lowest latency is achieved when $N = 2$. However, when $N = 1$, where all sub-areas use the same bandwidth, the delay performance degrades due to strong interference. On the other hand, for $N = 3$, the performance is the worst. This is because while it mitigates interference, it reduces spectrum efficiency and prolongs task transmission time. Regardless of number of vehicle in one sub-arean, these results indicate that $N = 2$ (frequency reuse factor is 1/2) strikes the best balance between interference and frequency efficiency for our assistant vehicle offloading scheme.

B. RSU Offloading/Local Computation

Fig. 10 illustrates the delay performance of our proposed ICRI and decentralized allocation algorithm as the total number of vehicles varies. It can be observed that the ICRI algorithm outperforms the decentralized algorithm and the pure MEC offloading scheme. As the number of vehicles increases, the total delay gradually increases due to the higher task load. However, the ICRI algorithm performs the best with the lowest delay and the decentralized algorithm second, particularly when the number of vehicles is relatively large. This is achieved by balancing the offloading decision, where some vehicles offload their tasks while others perform local computation, and the ICRI can further reduce the delay along with the allocation of bandwidth and MEC computation resources. By doing so, our proposed ICRI algorithm obtains the optimal result and mitigates the impact of computation insufficiency at the MEC server.

The impact of MEC computation resources and V2I bandwidth is depicted in Fig. 11 and Fig. 12, respectively. It can be observed that the total delay decreases with an increase in the computation capacity of MEC and the V2I bandwidth. This is because both parameters enhance the delay performance of MEC offloading: bandwidth reduces transmission delay, while MEC capacity reduces computation delay. Furthermore, as these two parameters increase, the performance gap between the three schemes narrows, as more tasks are executed at the MEC server. Our proposed ICRI algorithm consistently outperforms the decentralized algorithm and the pure MEC scheme, especially when the MEC computation and bandwidth resources are limited, which is more realistic, considering that these resources for MEC are usually limited.

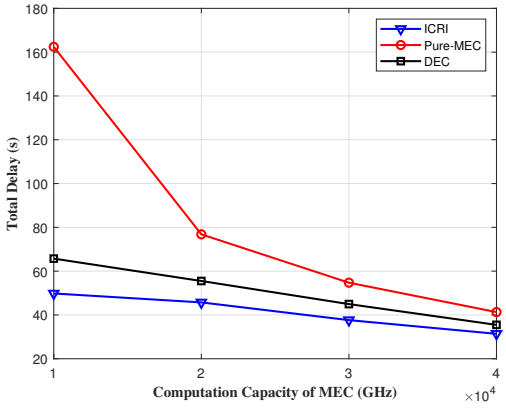


Fig. 11. Total delay of RSU offloading as a function of computation capacity of MEC.

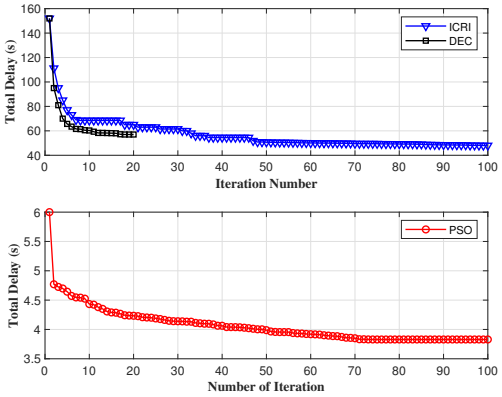


Fig. 13. The convergence performance of ICRBI algorithm, decentralized algorithm, and PSO algorithm.

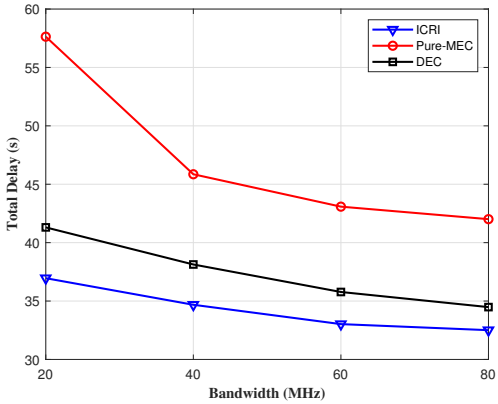


Fig. 12. Total delay of RSU offloading as a function of bandwidth.

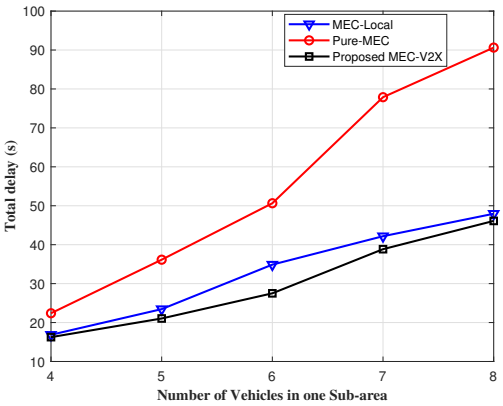


Fig. 14. Total delay of vehicles as a function of number of vehicles in one sub-area.

C. Proposed MEC-V2X Scheme

The convergence performance of the proposed ICRI algorithm, the decentralized algorithm, and the PSO algorithm is illustrated in Fig. 13. With 32 vehicles and the road divided into 4 sub-areas, it is evident that the decentralized algorithm achieves the fastest convergence, completing in 20 iterations, but it exhibits the worst delay performance. In comparison, the PSO algorithm converges in approximately 80 iterations, which is relatively quick and within the specified maximum iteration limit. Thus, we can conclude that the proposed algorithms converge rapidly while maintaining good delay performance.

The total delay performance of our proposed MEC-V2X scheme, which combines assistant vehicle offloading and RSU offloading/local computation, is shown in Fig. 14. This is compared with the adaptive MEC offloading scheme and the pure MEC computing scheme. Without adding additional computation resources, our proposed MEC-V2X offloading scheme exhibits the best performance. Under the given conditions, the improvement gap between our scheme and the adaptive MEC offloading scheme initially increases as the average number of vehicles in one sub-area reaches 6, and then decreases. This can be attributed to two factors. Firstly, in the assistant vehicle offloading part, the improvement is limited when the number of vehicles in one sub-area is small, as shown in Fig. 3. Secondly, in the RSU offloading part, the computation resources of the MEC for all other vehicles are relatively sufficient compared to the vacant assistant vehicle. Therefore, the improvement from assistant vehicle offloading is also limited. However, as the number of vehicles increases, the computation resources of the MEC become less powerful, and the additional assistant vehicles can enhance the overall performance. On the other hand, when there are too many vehicles in one sub-area, the improvement achieved by adding one assistant vehicle in each sub-area becomes relatively small. Consequently, we conclude that the largest improvement in our proposed MEC-V2X scheme occurs when the number of vehicles in one sub-area is 6.

Finally, Fig. 15 illustrates the delay performance of our proposed MEC-V2X scheme with different sub-area allocations. It can be observed that all three allocations outperform the adaptive MEC offloading scheme. The sub-area allocation with the smallest delay varies as the total number of vehicles K increases. When K is 24, $M = 4$ exhibits the smallest delay, while $M = 6$ and $M = 8$ rank second and third, respectively. This behavior can be attributed to the interaction between the number of vehicles and the sub-area allocation.

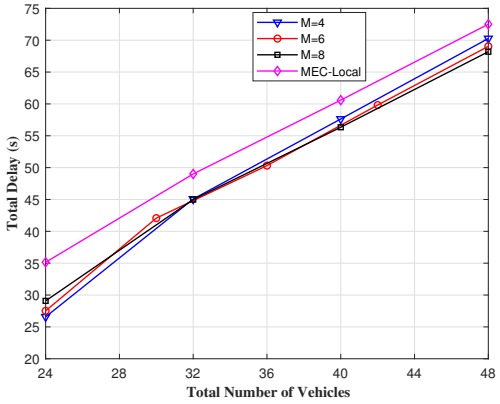


Fig. 15. Total delay of all vehicles as a function of total number of vehicles.

When M is large and K is small, there are insufficient vehicles in each sub-area. Consequently, there is a high chance of not having any vacant assistant vehicles, leading to a lack of assistant vehicle offloading. Even if one assistant vehicle is available, as shown in Fig. 3, the improvement achieved through assistant vehicle offloading is limited. As K increases to 40 and 48, $M = 6$ and $M = 8$ perform the best, respectively. These findings indicate that the optimal sub-area allocation for achieving the smallest delay depends on the vehicle density. The lowest delay can be obtained when there are approximately 6 vehicles in each sub-area. Therefore, we can conclude that based on the total number of vehicles, the optimal number of sub-areas should be adaptively adjusted to allocate 6 vehicles per sub-area.

V. Conclusion

In this paper, we investigated the latency minimization problem of task offloading and resource allocation in the context of vehicular task offloading with the assistance of MEC and V2X communication. We proposed a MEC-V2X assisted scheme for task offloading and resource allocation to enhance the crucial latency performance. To specify, we first focused on assistant vehicle offloading, which is tackled with our proposed modified PSO algorithm. It can achieve near optimal performance with low complexity and average transmit power consumption. Then we formulated an ICRI algorithm to solve the RSU offloading/local computation. The ICRI algorithm can solve the problem optimally. To further reduce the complexity, a decentralized sub-optimal algorithm is proposed. We studied the movement of vehicles by applying Doppler spread and the moving ratio of vehicles across sub-areas, and took into consideration of the non-negligible feedback of the offloaded task. The following conclusions are drawn

- The proposed MEC-V2X offloading scheme outperforms other comparative offloading schemes in terms of total delay performance.
- The impact of Doppler spread cannot be ignored, especially when the speed of the vehicle is high, as it introduces higher transmission delay.
- The optimal frequency reuse factor for assistant vehicle offloading to enhance frequency efficiency and reduce interference is obtained as 1/2.
- There exists an optimal number of sub-areas division depending on different number of vehicles on the road. By adjusting the number of sub-areas, assistant vehicle and RSU offloading can be balanced and lower latency can be obtained.

Appendix A

From (41), T_k^* is the sum of four sub-function

$$T_k^* = U_1(a_k^l, \varepsilon_k) + U_2(a_k^r, \phi_k) + U_3(a_k^r, \varphi_k) + U_4(a_k^r, \psi_k) \tag{61}$$

The Hessian matrix of U_1 can be written as:

$$\begin{aligned}\nabla^2 U_1 &= \begin{bmatrix} 2C_k/\varepsilon_k & -2C_k a_k^l/(\varepsilon_k)^2 \\ -2C_k a_k^l/(\varepsilon_k)^2 & 2C_k(a_k^l)^2/(\varepsilon_k)^3 \end{bmatrix} \\ &= (2C_k/\varepsilon_k)\begin{bmatrix} 1 \\ -2C_k a_k^l/\varepsilon_k \end{bmatrix}\begin{bmatrix} 1 & -2C_k a_k^l/\varepsilon_k \end{bmatrix} \succeq 0.\end{aligned} \tag{62}$$

Therefore, U_1 is convex. Similarly, we can obtain

$$U_2 \succeq 0, U_3 \succeq 0, U_4 \succeq 0, \tag{63}$$

and U_2, U_3, U_4 are convex.

From [38], it is known that if a function $f(x)$ and $g(x)$ are convex with respect to x, their perspective sum function $h(x) = f(x) + g(x)$ is also convex. As a result, Problem (43) is convex, which can be solved by Lagrange dual method optimally.

Appendix B

To update the variables, the $(t+1)$-th iteration of variables $\eta_k, \mu_k, \sigma_k, \xi_k$ can be formulated as follows

$$\eta_k^{(t+1)} = [\eta_k^{(t)} + \tau_k^{(t)}(\sum_{k\in\mathcal{K}^*}(\varepsilon_k^{(t)} - f_k^l))]^+, \tag{64}$$

$$\mu_k^{(t+1)} = [\mu_k^{(t)} + \nu_k^{(t)}(\sum_{k\in\mathcal{K}^*}(\phi_k^{(t)} - F^R))]^+, \tag{65}$$

$$\sigma_k^{(t+1)} = [\sigma_k^{(t)} + \varsigma_k^{(t)}(\sum_{k\in\mathcal{K}^*}(\varphi_k^{(t)} - B_R))]^+, \tag{66}$$

$$\xi_k^{(t+1)} = [\xi_k^{(t)} + \chi_k^{(t)}(\sum_{k\in\mathcal{K}^*}(\psi_k^{(t)} - B_R))]^+, \tag{67}$$

where $[i]^+ = \max\{0, i\}$, and $\tau_k^{(t)}, \nu_k^{(t)}, \varsigma_k^{(t)}, \chi_k^{(t)}$ represent the learning step-sizes of t-th iteration.

References

[1] Z. Liu, S. Ishihara, Y. Cui, Y. Ji, and Y. Tanaka, "Jet: Joint source and channel coding for error resilient virtual reality video wireless transmission," *Signal Processing*, vol. 147, pp. 154–162, 2018.

[2] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "Avr: Augmented vehicular reality," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 81–95.

[3] Y.-D. Kim, G.-J. Son, C.-H. Song, and H.-K. Kim, "On the deployment and noise filtering of vehicular radar application for detection enhancement in roads and tunnels," *Sensors*, vol. 18, no. 3, 2018.

[4] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 000 059–000 066.

[5] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 1852–1864, 2022.

[6] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by lte-based systems and 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[7] S. A. Ashraf, R. Blasco, H. Do, G. Fodor, C. Zhang, and W. Sun, "Supporting vehicle-to-everything services by 5g new radio release-16 systems," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 26–32, 2020.

[8] M. Cui, S. Zhong, B. Li, X. Chen, and K. Huang, "Offloading autonomous driving services via edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 535–10 547, 2020.

[9] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. S. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

[10] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. S. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Network*, vol. 32, no. 1, pp. 80–86, 2018.

[11] C. Wang, C. Liang, F. R. Yu, Q. Chen, and L. Tang, "Computation offloading and resource allocation in wireless cellular networks with mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 4924–4938, 2017.

[12] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.

[13] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. S. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," *IEEE Access*, vol. 7, pp. 62 624–62 632, 2019.

[14] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning-based joint resource scheduling algorithms for hybrid mec networks," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6252–6265, 2020.

[15] Z. Xiao, X. Dai, H. Jiang, D. Wang, H. Chen, L. Yang, and F. Zeng, "Vehicular task offloading via heat-aware mec cooperation using game-theoretic method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020.

[16] Y. Kai, J. Wang, and H. Zhu, "Energy minimization for d2d-assisted mobile edge computing networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[17] X. Cao, F. Wang, J. Xu, R. Zhang, and S. Cui, "Joint computation and communication cooperation for energy-efficient mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4188–4200, 2019.

[18] B. Dai, F. Xu, Y. Cao, and Y. Xu, "Hybrid sensing data fusion of cooperative perception for autonomous driving with augmented vehicular reality," *IEEE Systems Journal*, vol. 15, no. 1, pp. 1413–1422, 2021.

[19] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[20] D. Qiao and F. Zulkernine, "Adaptive feature fusion for cooperative perception using lidar point clouds," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 1186–1195.

[21] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, ser. SEC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 88–100.

[22] S. Ye, R. Blum, and L. Cimini, "Adaptive modulation for variable-rate ofdm systems with imperfect channel information," in *Vehicular Technology Conference. IEEE 55th Vehicular Technology Conference. VTC Spring 2002 (Cat. No.02CH37367)*, vol. 2, 2002, pp. 767–771 vol.2.

[23] M. Souryal and R. Pickholtz, "Adaptive modulation with imperfect channel information in ofdm," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 6, 2001, pp. 1861–1865 vol.6.

[24] Y. Li, Y. Liang, Q. Liu, and H. Wang, "Resources allocation in multicell d2d communications for internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4100–4108, 2018.

[25] "Study on Enhancement of 3GPP Support for 5G V2X Services," *3rd Generation Partnership Project (3GPP); TR 22.886 V15.0.0*, 2018.

[26] Y. Cheng, Y. Liao, and X. Zhai, "Energy-efficient resource allocation for uav-empowered mobile edge computing system," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*, 2020, pp. 408–413.

[27] Z. Sun, G. Sun, Y. Liu, J. Wang, and D. Cao, "Bargain-match: A game theoretical approach for resource allocation and task offloading in vehicular edge computing networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1655–1673, 2024.

[28] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.

[29] N.-S. Kim and Y. H. Lee, "Effect of channel estimation errors and feedback delay on the performance of closed-loop transmit diversity system," in *2003 4th IEEE Workshop on Signal Processing Advances in Wireless Communications - SPAWC 2003 (IEEE Cat. No.03EX689)*, 2003, pp. 542–545.

[30] P. Wang, W. Wu, J. Liu, G. Chai, and L. Feng, "Joint spectrum and power allocation for v2x communications with imperfect csi," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 12, pp. 16 338–16 353, 2023.

[31] X. Fan, T. Cui, C. Cao, Q. Chen, and K. S. Kwak, "Minimum-cost offloading for collaborative task execution of mec-assisted platooning," *Sensors*, vol. 19, no. 4, 2019.

[32] J. Karedal, N. Czink, A. Paier, F. Tufvesson, and A. F. Molisch, "Path loss modeling for vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 323–328, 2011.

[33] Y. Pan, H. Jiang, H. Zhu, and J. Wang, "Latency minimization for task offloading in hierarchical fog-computing c-ran networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[34] Y. Pochet and L. A. Wolsey, *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.

[35] F. Guo, H. Zhang, H. Ji, X. Li, and V. C. M. Leung, "An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2651–2664, 2018.

[36] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[37] C.-m. Yan, G.-y. Lu, Y.-t. Liu, and X.-y. Deng, "A modified pso algorithm with exponential decay weight," in *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2017, pp. 239–242.

[38] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[39] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[40] "Technical Specification Group Radio Access Network; Study LTE-Based V2X Services," *3rd Generation Partnership Project (3GPP); 3GPP TR 36.885 V14.0.0*, 2016.

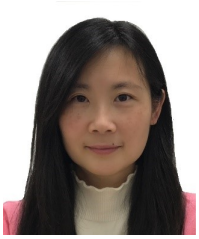
[41] "WF on SLS Evaluation Assumptions for eV2X," *3rd Generation Partnership Project (3GPP) TSG RAN WG1 Meeting 85; R1-165704*, 2016.

[42] J.-B. Wang, H. Yang, M. Cheng, J.-Y. Wang, M. Lin, and J. Wang, "Joint optimization of offloading and resources allocation in secure mobile edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8843–8854, 2020.

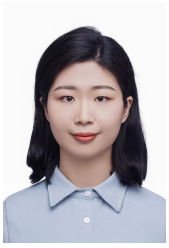
[43] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

Yilun Zhang received the B.S. and M.S. degrees in telecommunication engineering from Xidian University, Xi'an, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in electronic engineering, University of Kent, UK. His research interests include the wireless communication and mobile edge computing.

Changrun Chen received the B.E degree in automation and the Ph.D. degree in control science and engineering from the Guangdong University of Technology, Guangzhou, China, in 2017 and 2022, respectively. From 2022 to 2023, he was a lecturer with the School of Internet Finance and Information Engineering, Guangdong University of Finance, Guangzhou, China. He is currently a Marie Curie Research Fellow with the School of Engineering, University of Kent, U.K. He was awarded the EU Marie Curie Postdoctoral Fellowship in 2023. His research interests include integrated sensing and communication, statistical signal processing, and rank correlation theory.



Huiling Zhu (SM'17) received the B.S degree from Xidian University, China, and the Ph.D. degree from Tsinghua University, China. She is currently a Reader (Associate Professor) in the School of Engineering, University of Kent, United Kingdom. Her research interests are in the area of wireless communications. She was holding European Commission Marie Curie Fellowship from 2014 to 2016. She received the best paper award from IEEE Globecom 2011. She was Symposium Co-Chair for IEEE Globecom 2015 and IEEE ICC 2018, and Track Co-Chair of IEEE VTC2016-Spring and VTC2018-Spring. Currently, she serves as an Editor for IEEE Transactions on Vehicular Technology.



Yijin Pan is an associate professor in the School of Information Science and Engineering at Southeast University. She was selected as a Newton Fellow by the Royal Society of the United Kingdom from 2019 to 2021. Dr. Pan Yijin serves as a reviewer and the TPC member for prestigious international journals and conferences including IEEE Transactions, Globecom, and ICC, etc. Her research focuses on key technologies for the future communication networks, especially for the edge intelligence enhanced communication schemes.

Jiangzhou Wang (Fellow, IEEE) is a Professor with the University of Kent, U.K. He has published more than 400 papers and four books. His research focuses on mobile communications. He was a recipient of the 2022 IEEE Communications Society Leonard G. Abraham Prize and IEEE Globecom2012 Best Paper Award. He was the Technical Program Chair of the 2019 IEEE International Conference on Communications (ICC2019), Shanghai, Executive Chair of the IEEE ICC2015, London, and Technical Program Chair of the IEEE WCNC2013. He is/was the editor of a number of international journals, including IEEE Transactions on Communications from 1998 to 2013. Professor Wang is a Fellow of the Royal Academy of Engineering, U.K., Fellow of the IEEE, and Fellow of the IET.