Task Offloading for MEC-V2X Assisted Autonomous Driving

Yilun Zhang
School of Engineering
University of Kent
Canterbury, U.K.
yz312@kent.ac.uk

Changrun Chen
School of Engineering
University of Kent
Canterbury, U.K.
c.chen@kent.ac.uk

Huiling Zhu
School of Engineering
University of Kent
Canterbury, U.K.
h.zhu@kent.ac.uk

Jiangzhou Wang
School of Engineering
University of Kent
Canterbury, U.K.
j.z.wang@kent.ac.uk

Abstract

Mobile edge computing (MEC) and vehicle-to-everything (V2X) communications are two promising technics to support delay-sensitive applications for autonomous driving. In this paper, road side unit (RSU) and assistant vehicle offloading are jointly considered to minimize latency for vehicular tasks in an MEC-V2X network. The impact of Doppler spread caused by high moving speed is also considered. The offloading decision, transmit power, bandwidth, and computation resource allocations are solved by formulating them as a joint optimization problem. The simulation results show that the proposed scheme significantly reduced the task latency, compared to three traditional methods, where task either run locally, or at MEC server, or adaptively between MEC server and local.

***Index Terms*—Mobile edge computing, vehicle-to-everything communication, task offloading, resource allocation**

I. Introduction

The advent of the era of autonomous driving results in an explosive emergence of computation-intensive and delay-sensitive vehicular applications in autonomous vehicles (AVs). These applications require a significant amount of computation resources for processing a huge volume of raw sensing data in real time [1], which impose a great challenge for AVs with limited computation resources. Mobile edge computing (MEC) [2] has been proposed to offload computation-oriented task to edge servers such as road side unit (RSU) through vehicle-to-everything (V2X) communications [3]. With the task being offloaded from the vehicle to MEC server, low task processing latency can be achieved [4].

There have been ralated researches focusing on offloading scheme to balance the allocation of computation and communication resources in MEC system [5]. To further address the computation resource limitation, additional computation resources were introduced in [6]. In [7], computation resources of other users were explored to enhance the offloading performance under the support of cooperative computation where a channel multiplex model was proposed among vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) links.

In the preceding MEC and vehicular scenarios, there is a disregard for the feedback of computation results. Furthermore, the communication model of vehicles has been oversimplified, as demonstrated by [5], [6], and [7], all of which assume all users to be stationary during task offloading. This assumption overlooks the impact of the Doppler spread caused by high moving speeds, as discussed in [8], which can lead to a decrease in communication capacity. Considering the limitation of computation and communication ressources in MEC, we are inspired to utilize the available computation resources of vacant vehicles to assist vehicle task offloading, so that the computation bottleneck is mitigated without deploying new resource. Meanwhile, the transmission bottleneck between the RSU and users can be mitigated with the help of V2V communications in the traditional offloading network.

In this paper, we aim to minimize the latency of vehicular tasks by introducing assistant vehicle offloading in the MEC network and addressing the impact of Doppler Spread. In order to minimize strong interference among V2V communications, a sub-area division scheme has been devised. The MEC-V2X network consists of three offloading decisions: local computation, RSU offloading, and assitant vehicle offloading. We propose a two-steps solution for the formulated mixed integer nonlinear programming (MINLP) problem, where a heuristic algorithm and Lagrange dual method are used. To evaluate the performance, three benchmark schemes are considered for comparison. 1) Local computation: All vehicles complete the overall computing tasks locally. 2) Pure MEC offloading: All vehicles choose to offload the computing tasks to the MEC server [9]. 3) Adaptively MEC offloading: Vehicles addaptively choose to run the task locally or offload to the MEC server [6]. Simulation results show that the proposed scheme outperform the comparison schemes in total delay.

II. System Model

Our MEC-V2X assisted vehicular scenario is shown in Fig. 1. K vehicles, $\mathcal{K} = \{1, 2, \cdots, K\}$ denoting the set of all vehicles, drop on the road covered by one RSU. At an instant, each vehicle generates a delay-sensitive task randomly with a probability denoted as p_c. The road is virtually divided into 4 sub-areas. $\mathcal{K}_m$ is defined as the set of vehicles within the sub-area $m,\ m \in \{1, 2, 3, 4\}$. Within a sub-area, task offloading is allowed between two proximate vehicles through V2V communications when one is vacant. A vehicle receiving and processing an offloaded task is called an assistant vehicle, while the vehicle offloading its task is called a requesting vehicle. Dedicated bandwidth of V2I and V2V transmissions

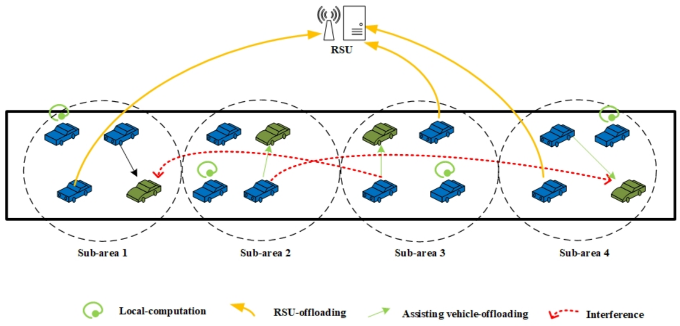

Fig. 1. MEC-Based V2X network

are denoted as B_R and B_V, respectively. Considering the movement of vehicle, delayed channel gain and Doppler spread should be adopted. Thus, channel gain h can be written as $h = \beta\hat{h} + \theta$, where $\hat{h}$ and θ represent the delayed channel gain and the channel discrepancy [10], respectively. Then, the transmission rate from vehicle k to the RSU is given by

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

where P_k and $B_{k,r}$ denote the transmit power of vehicle k and the allocated bandwidth for the V2I trasnmission. $\hat{h}_{k,r}$ is the delayed channel gain from vehicle k to the RSU. σ^2 and σ_θ^2 represent the power of Gaussian noise and estimation error.

With the assistant vehicle offloading enabled in sub-areas, frequency is reused among non-adjacent sub-areas, therefore the frequency reuse is $1/2$. In each sub-area, there is only one assitant vehicle offloading at one instant. Therefore, the transmission rate from the requesting vehicle k to the assistant vehicle i can be obtained as

$$R_{k,i} = \frac{B_V}{2}\log_2\left(1 + \frac{P_k|\hat{h}_{k,i}|^2}{\sigma^2 + \sigma_\theta^2 + \sum_{j\in\mathcal{K}\setminus\mathcal{K}_m} \iota_{k,j} P_j |h_{j,i}|^2}\right), \tag{2}$$

where $\iota_{k,j}$ represents the channel multiplexing factor among different vehicles. When the frequencies are reused between any two vehicle k and j, $\iota_{k,j} = 1$, otherwise 0.

The task in vehicle k is modeled as $U_k = \{D_k, C_k\}, k \in \mathcal{K}$, where D_k denotes the task data size for transmission, and C_k is the required computation resource. The task offloading decision of vehicle k is denoted as $a_k \in \{a_k^l, a_k^r, a_k^v\}$, where a_k^l, a_k^r, and a_k^v represent the decisions of either local computation, RSU offloading, or assistant vehicle offloading. One task can be only processed at one location. That is,

$$C1: a_k^l, a_k^r, a_k^v \in \{0,1\}, \quad C2: a_k^l + a_k^r + a_k^v = 1. \tag{3}$$

Let $\mathcal{A} = \{a_1, a_2, \cdots, a_K\}$ represent the offloading decision of all vehicles. The feedback data size is given by δD_k, where δ is the ratio of feedback data size to the offload data size. Three desicions to process the task is given as follows

1) Local computation: Task U_k only has computation delay when it is processed locally. The delay of local computation can be written as $T_k^l = \frac{C_k}{f_k^l}$, where f_k^l is the on-board computation resource of vehcicle k.

2) RSU offloading: If task U_k is offloaded to the RSU, the transmission delay from the requesting vehicle k to the RSU is written as $T_{k,r}^o = \frac{D_k}{R_{k,r}}$, where $R_{k,r}$ is the offloading data rate from vehicle k to the RSU. The computation delay at the RSU is written as $T_{k,r}^c = \frac{C_k}{f_k^r}$, where f_k^r is the allocated RSU computation resource for vehicle k. The feedback delay from the RSU back to vehicle k is written as $T_{k,r}^f = \frac{\delta D_k}{R_{r,k}}$, where $R_{r,k}$ represents the feedback data rate from the RSU to vehicle k. The total delay of RSU offloading is the sum of above three delays, given by $T_k^r = T_{k,r}^o + T_{k,r}^c + T_{k,r}^f$.

3) Assistant vehicle offloading: Let $T_{k,i}^o, T_{k,i}^c, T_{k,i}^f$ stand for the offloading delay, the computing delay and the feedback delay for assistant offloading, written as $T_{k,i}^o = \frac{D_k}{R_{k,i}}, T_{k,i}^c = \frac{C_k}{f_i^l}, T_{k,i}^f = \frac{\delta D_k}{R_{i,k}}$. Then, the total delay for assistant vehicle offloading can be obtained as $T_k^v = T_{k,i}^o + T_{k,i}^c + T_{k,i}^f$.

Therefore, the total delay of processing task U_k can be obtained as

$$T_k = a_k^l T_k^l + a_k^r T_k^r + a_k^v T_k^v. \tag{4}$$

III. PROBLEM FORMULATION

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

$$C3: \sum_{k\in\mathcal{K}} a_k^r f_k^r \le F^R. \tag{5}$$

Defining P^{max} as the maximum transmit power for all vehicles, the transmit powers of all vehicles are defined as $\mathcal{P} = \{P_1, P_2, \cdots, P_K\}$, then the transmit power has constrant as follow

$$C4: 0 \le P_k \le P^{max}. \tag{6}$$

The allocated odffloading and feedback bandwidth of RSU offloading should be less than the total V2I bandwidth, therefore

$$C5: \{\sum_{k\in\mathcal{K}} a_k^r B_{k,r}, \sum_{k\in\mathcal{K}} a_k^r B_{r,k}\} \le B_R. \tag{7}$$

As there is at most one assistant vehicle offloading in one sub-area, following constraints can be obtained

$$C6: \sum_{k\in\mathcal{K}} a_k^v \le M, \quad C7: \sum_{k\in\mathcal{K}_m} a_k^v \le 1. \tag{8}$$

With adjustable variables of offloading decision $\mathcal{A}$, RSU computation resource $\{f_k^r | k \in \mathcal{K}\}$, bandwidth of offloading and feedback in RSU offloading $\{B_{k,r}, B_{r,k} | k \in \mathcal{K}\}$ and transmit power $\mathcal{P}$, the optimization problem can be formulated as follows

$$\begin{aligned} &\min_{\{\mathcal{A}, f_k^r, B_{k,r}, B_{r,k}, \mathcal{P}\}} \sum_{k\in\mathcal{K}} T_k, \\ &s.t.\ C1 - C7. \end{aligned} \tag{9}$$

It can be seen that this is a non-convex MINLP problem because of the binary constraint on offloading decision $\mathcal{A}$ and other continuous constraints [11].

IV. Solution Analysis

When K is large, the complexity of optimal solution is extremely high. Therefore, we solve the problem iteritively with the first step to allocate the assistant vehicle offloading and its transmit power in each sub-area. Then, the allocation of RSU offloading/local computation is less complex.

A. Assistant Vehicle Offloading and Power Allocation

In order to minimize the total delay of assistant vehicle offloading, the assistant/requesting vehicle and their transmit power for each sub-area should be choosen wisely. Problem (9) can be transformed into

$$\begin{aligned} &\min_{\{\mathcal{A}',\mathcal{P}'\}} \sum_{k\in\mathcal{K}} T_k^{'}, \\ &s.t.\ C4, C6, C7, \end{aligned} \tag{10}$$

where $\mathcal{A}^{'}$ denotes the offloading decisions of vehicles doing assistant vehicle offloading with $a_k^l = 0, a_k^r = 0$. $\mathcal{P}^{'}$ is the transmit power of vehicles doing assistant vehicle offloading. $T_k^{'} = a_k^v T_k^v$.

By optimizing the best offloading decision and transmit power, the total delay $T_k^{'}$ can be minimized and the optimal solution of the assistant vehicle offloading can be obtained.

We formulate a low-complexity particle swarm optimization (PSO) algorithm [12] to approch the near-optimal solution as follows. In order to reduce the total delay, the fitness function is given by

$$\mathcal{F} = \sum_{k\in\mathcal{K}} T_k^{'}. \tag{11}$$

By modeling the offloading decision and the allocated transmit power, we formulate the particles as a $4M$ dimension vector $\mathbf{x}$ which represents a solution: $\mathbf{x} = [r_1, \cdots, r_M, P_1, \cdots, P_M, r_1^{'}, \cdots, r_M^{'}, P_1^{'}, \cdots, P_M^{'}] = [\mathbf{x}_r, \mathbf{x}_P, \mathbf{x}_r^{'}, \mathbf{x}_P^{'}]$. $\mathbf{x}_r$ and $\mathbf{x}_P$ represent the set of potential requesting vehicles in M sub-area and their respective transmit power. $\mathbf{x}_r^{'}$ and $\mathbf{x}_P^{'}$ are the set of assistant vehicles and their transmit power. After each iteration, as $\mathbf{x}_r, \mathbf{x}_r^{'}$ are matrix with integer elements inside a sub-area, we round and transform them to the closest index of potential requesting vehicle and assistant vehicle. Through the iteration process, particles which represent different solutions of problem (10) are generated. In each iteration, the particle with the highest revenue which represents the best solution so far is preserved. After the whole iterations, the final solution accepted by all assistant and requesting vehicles can be obtained.

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

After getting the assistant vehicle offloading and power allocation, the second step is to solve the RSU offloading/local computation. The rest set of vehicles is denoted as $\mathcal{K}^*$. Then, the original optimization problem can be formulated as

$$\begin{aligned} &\min_{\{\mathcal{A}^*, B_{k,r}, B_{r,k}\mathcal{P}^*\}} \sum_{k\in\mathcal{K}^*} T_k^* \\ &s.t.\ C1, C2, C3, C5 \end{aligned} \tag{12}$$

where $\mathcal{A}^*$ denotes the decisions of vehicles doing local computation and RSU offloading. $\mathcal{P}^*$ denotes the respective transmit power with $P_k = 0$, when $a_k^l = 1$, and $P_k = P^{max}$, when $a_k^r = 1$. T_k^* is the delay of local computation and RSU offloading, given by

$$T_k^* = a_k^l T_k^l + a_k^r T_k^r = a_k^l \frac{C_k}{f_k^l} + a_k^r \left(\frac{D_k}{R_{k,r}} + \frac{C_k}{f_k^r} + \frac{\delta D_k}{R_{r,k}}\right). \tag{13}$$

From [13], if function $f(x) = q/x$ is convex with respect to x, its perspective function $g(t,x) = tf(t/x)$ is convex too with respect to (t,x). Therefore, we relax the integer $C1$ and adopt the following variable substitutions

$$\begin{aligned} &\varepsilon_k = a_k^l f_k^l, \phi_k = a_k^r f_k^r, \varphi_k = a_k^r B_{k,r}, \psi_k = a_k^r B_{r,k}, \\ &\gamma_k = \log_2\Big(1 + \frac{P_k|\hat{h}_{k,r}|^2}{\sigma^2 + \sigma_\theta^2}\Big), \varpi_k = \log_2\Big(1 + \frac{P_d|\hat{h}_{r,k}|^2}{\sigma^2 + \sigma_\theta^2}\Big). \end{aligned} \tag{14}$$

Removing the integer constraint $C1$ and adopting above substitutions, optimization problem (12) can be transformed to the following one

$$\begin{aligned} &\min_{\{\mathcal{A}^*, f_k^r, B_{k,r}, \mathcal{P}^*\}} \sum_{k\in\mathcal{K}^*} T_k^* = \\ &\quad (a_k^l)^2 \frac{C_k}{\varepsilon_k} + (a_k^r)^2 \frac{C_k}{\phi_k} + (a_k^r)^2 \frac{D_k}{\varphi_k \gamma_k} + (a_k^r)^2 \frac{\delta D_k}{\psi_k \varpi_k} \\ &s.t.\ C1^*: \sum_{k\in\mathcal{K}^*} \phi_k \le F^R, \quad C2^*: \sum_{k\in\mathcal{K}^*} \varphi_k \le B_R, \\ &\quad C3^*: \sum_{k\in\mathcal{K}^*} \psi_k \le B_R, \quad C4^*: \varepsilon_k = f_k^l. \end{aligned} \tag{15}$$

It can be seen that problem (15) is a convex problem so Lagrange dual method can be applied. By introducing the non-negative variables $\eta_k, \mu_k, \sigma_k, \xi_k$ and taking the first order derivatives of $\varepsilon_k, \phi_k, \varphi_k, \psi_k$ respectively, follows are obtained

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_k} = -(a_k^l)^2 \frac{C_k}{{\varepsilon_k}^2} + \eta_k, \quad \frac{\partial \mathcal{L}}{\partial \phi_k} = -(a_k^r)^2 \frac{C_k}{{\phi_k}^2} + \mu_k, \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial \varphi_k} = -(a_k^r)^2 \Big(\frac{D_k}{{\varphi_k}^2 \gamma_k}\Big) + \sigma_k, \frac{\partial \mathcal{L}}{\partial \psi_k} = -(a_k^r)^2 \Big(\frac{\delta D_k}{{\psi_k}^2 \varpi_k}\Big) + \xi_k. \tag{17}$$

We can infer 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 $(a_k^l, a_k^r) \ne 0$, the optimal values of $(\varepsilon_k, \phi_k, \varphi_k, \psi_k)$ can be obtained as follows

$$\begin{aligned} &{\varepsilon_k}^* = {a_k^l}^* \sqrt{\frac{C_k}{\eta_k}}, \quad {\phi_k}^* = {a_k^r}^* \sqrt{\frac{C_k}{\mu_k}}, \\ &{\varphi_k}^* = {a_k^r}^* \sqrt{\frac{D_k}{\sigma_k \gamma_k}}, \quad {\psi_k}^* = {a_k^r}^* \sqrt{\frac{\delta D_k}{\xi_k \varpi_k}}. \end{aligned} \tag{18}$$

Next, we take the first order derivatives of a_k^l and a_k^r respectively

$$\frac{\partial \mathcal{L}}{\partial a_k^l} = 2 \frac{C_k}{\varepsilon_k / a_k^l}, \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial a_k^r} = 2\frac{C_k}{\phi_k/a_k^r} + 2\frac{D_k}{\varphi_k\gamma_k/a_k^r} + 2\frac{\delta D_k}{\psi_k\varpi_k/a_k^r}. \qquad (20)$$

Following denotions are made for simplicity

$$\varrho_k = \frac{\partial \mathcal{L}}{\partial a_k^l}(\frac{\varepsilon_k{}^*}{a_k^l{}^*}), \quad \zeta_k = \frac{\partial \mathcal{L}}{\partial a_k^r}(\frac{\phi_k{}^*}{a_k^r{}^*}, \frac{\varphi_k{}^*\gamma_k}{a_k^r{}^*}, \frac{\psi_k{}^*\varpi_k}{a_k^r{}^*}). \quad (21)$$

From $C2$, the optimal offloading decision $\mathcal{A}^*$ can be obtained as $a_k^l = 1, a_k^r = 0$, when $\varrho_k < \zeta_k$, $a_k^l = 0, a_k^r = 1$, when $\zeta_k < \varrho_k$. The values of the variables $\eta_k, \mu_k, \sigma_k, \xi_k$ can be obtained by using sub-gradient method [14]. Therefore, optimal solution of problem (12) can be achieved successfully with fast convergence.

After getting the results of offloading decisions and resource allocations from section A and B, we combine $\mathcal{A}^*, \mathcal{P}^*$ with $\mathcal{A}^{'}, \mathcal{P}^{'}$ to get $\mathcal{A}$ and $\mathcal{P}$, respectively. Then, the 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 transmist power $\mathcal{P}$ of the original problem (9) can be obtained.

V. Simulation Results

In our simulations, the K vehicles are randomly deployed in the MEC-V2X network with one RSU. The road is divided into 4 sub-areas by distance. As for the tasks of vehicles, when a task emerges to a vehicle, the task size and the required computation frequency are uniformly generated in [5-20] Mbits, and in [5-20] Mcycles. The number of particles and iteration of the propsed PSO algorithm are 512 and 128. Other parameters used in the simulation are summarized as follows, $B_V = 2$ MHz, $B_R = 20$ MHz, $p_c = 0.75$, $F^R = 1 \times 10^4$, $P_{max} = 30$ dBm, $\sigma^2 = -114$ dBm, $\iota = 0.2$, and the maximum Doppler frequency is 300Hz.

For the sake of comparison, our proposed PSO algorithm for assistant vehicle offloading is marked as "V2V-PSO", the second part of RSU offloading/local computation is marked as "propose-V2I", and we use "MEC-V2X" to represent the whole MEC-V2X offloading scheme after combining these two parts. The performances of the following typical traditional methods are also presented for comparison.1) Local computation scheme (labelled as "Local-fixed"). 2) Pure MEC offloading scheme (labelled as "Pure-MEC"). 3) Adaptive MEC offloading scheme (labelled as "MEC-Local").

Fig. 2 shows the total delay of vehicles doing assistant vehicle offloading between local computation, random selection of offloading vehicle with maximum transmit power (V2V-random-fixed), our proposed PSO based offloading decision and transmit power allocation, optimal offloading decision with the maximum transmit power (V2V-optimal-fixed) and optimal offloading decision and transmit power selection (V2V-optimal-exhuasted) using exhausted searching. With the average number of vehicles in one sub-area increases, the total delay of all vehicles doing assistant vehicle offloading decreases except the random offloading scheme. This is because when there are more vehicles on the road, more potential vehicles can be selected as the requesting vehicle. Vehicles which have better channel conditions may be selected

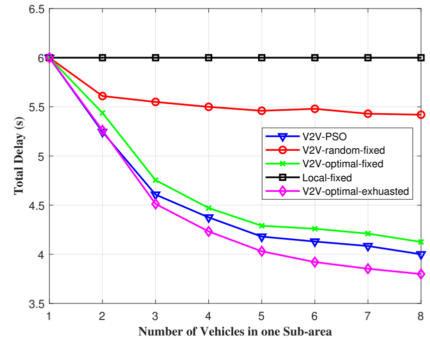

Fig. 2. Total delay of assistant vehicle offloading versus the number of vehicles in one sub-area

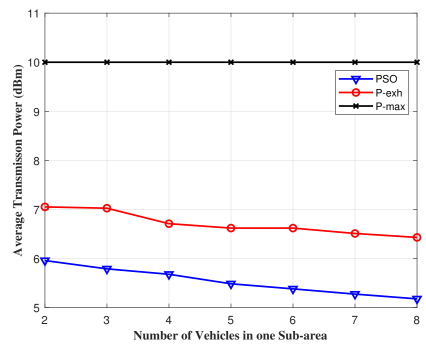

Fig. 3. Average transmission power versus the number of vehicles in one sub-area

and will lead to smaller delay. Regardless of the number of vehicles, our proposed PSO algorithm can achieve near optimal performance with low complexity and reduce the total delay efficiently.

The average transmit power is presented in Fig. 3, considering our proposed modified PSO algorithm (PSO) with the maximum transmit power scheme (P-max) and the above transmit power by using exhausted searching (P-exh). It can be seen that the proposed algorithm has less average transmit power than the other two schemes. This is because the transmit power is a continuous variable and the exhausted searching can only search a given number of choices. Howerver, our proposed PSO algorithm can explore more transmit power and therefore can reduce the energy consumption.

Fig. 4 indicates the delay performance of our proposed RSU offloading/local computation scheme when the number of vehicles changes. It can be observed that our proposed RSU offloading/local computation scheme is better than the pure MEC offloading scheme. With the increasing number of vehicles, our scheme can reduce the total delay significantly when the number of vehicles is relatively large. This is because the computation resource of MEC will be insufficient when too many vehicles offload their tasks. By balancing the whole offloading decision with the bandwidth and the MEC computation resource allocation, the influence of inadequate

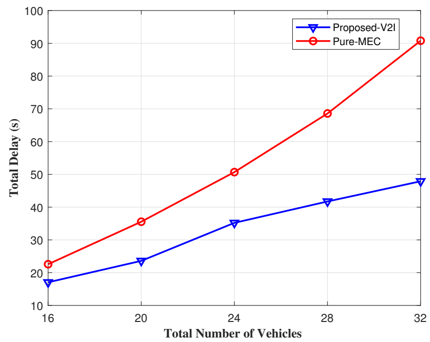

Fig. 4. Total delay of RSU offloading versus number of vehicles

computational resources of MEC can be reduced, improving the delay performance.

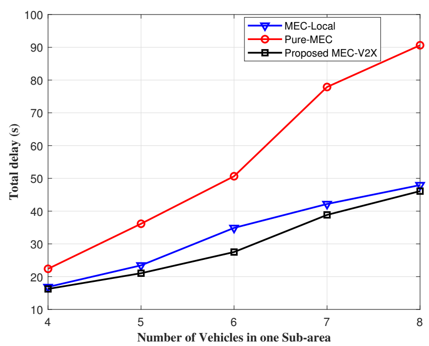

Fig. 5. Total delay of vehicles versus number of vehicles in one sub-area

The total delay performance of our proposed MEC-V2X scheme is shown in Fig. 5. It can be seen that our proposed MEC-V2X offloading scheme performs the best. With the given conditions, the improvement gap between our scheme and the traditionnal MEC offloading scheme increases at first when the average number of vehicles in one sub-area increases to 6 and shrinks afterwards. The reason comes from both the assistant vehicle and RSU offloading. When the number of vehicles in one sub-area is small, the improvement of assistant vehicle offloading is limited. Meanwhile, when the number of vehicles is large, the computation resource of MEC for RSU offloading is limited which will also decrease the improvement in reducing delay. Therefore, we can infer that the balance point of assistant vehicle offloading and RSU offloading is when there is 6 vehicles in one sub-area.

VI. Conclusion

In this work, the assistance of MEC and V2X communication is applied for the task offloading and resource allocation problem in the autonomous driving scenario . It is showed that the MEC-V2X scheme can improve the delay performance without deploying additional computation resources. In addiation, the influence of number of vehicle on the offloading selection is studied and the balance opint of assistant vehicle offloading and RSU offloading is inferred.

References

[1] B. Yang, X. Cao, K. Xiong, C. Yuen, Y. L. Guan, S. Leng, L. Qian, and Z. Han, "Edge intelligence for autonomous driving in 6g wireless system: Design challenges and solutions," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 40–47, 2021.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

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

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