International Journal of Engineering, Management and Humanities (IJEMH)
Volume 5, Issue 2, Mar.-Apr., 2024 pp: 98-105 ISSN: 2584-2145
www.ijemh.com

Optimizing Edge Collaboration for Task Offloading in MEC-Enabled IoV Networks

Karthikeyan S D, Balachandar B, Sakthiguru.N S, Hariharan B
Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram Campus, Chennai,India.

Date of Submission: 14-03-2024 Date of Acceptance: 28-03-2024

ABSTRACT: Benefiting from its abundant computing resources and low computing latency, mobile edge computing (MEC) is a promising approach for enhancing the computing capacity of the 5G Internet of vehicles (IoV). Because of the high mobility, handover is frequent and inevitable in IoV networks. In this paper, we investigate an edge collaborative task offloading and splitting strategy in MEC-enabled IoV networks, in which the task is splitted on the edge and paralleling executed by each part of the task on several MEC servers when handover is occured. Applications in IoV networks have flexible requirements on latency and energy consumption. To realize the tradeoff between latency and energy consumption, we formulate the task offloading and splitting as an optimization problem with the aim of minimizing the total cost of latency and energy consumption by jointly optimizing the task splitting ratio and uplink transmit power of vehicle terminal (VT). Because the proposed problem is non-smooth and non-convex, we divide the original problem into two convex subproblems, and apply an alternate convex search (ACS) algorithm to obtain the optimized solution with low computational complexity. Numerical simulation results show that the proposed method can adjust the offloading strategy properly according to task preference, and obtain a lower total cost compared with the baseline algorithms.

KEYWORDS: Internet of vehicles (IoV), mobile edge computing (MEC), resource allocation.(RE)

I. INTRODUCTION

The requirements for the Internet of Vehicles (IoV) are changing from data sharing to data sending and processing because to the quick expansion of computation-intensive and latency-sensitive mobile apps. This creates issues for the IoV's computing and communication capabilities. [1], [2]. In order to address the low latency requirements for vehicle applications and supply adequate processing resources for intelligent vehicles, mobile edge computing, or MEC, has emerged as a viable option [3]–[5].

Intelligent vehicles can lower task execute latency and vehicle energy consumption by offloading their large latency-sensitive computing service tasks (L2SC) to MEC servers with the use of MEC networks [6]–[8]. A roadside MEC server cannot handle the increasing and dynamic offloading demand, despite the fact that MEC networks can increase processing performance. Requirements of vehicle terminals (VTs) with high mobility in its coverage. Hence, edge collaboration should be introduced into MEC networks to solve this problem. The existing researches on edge collaboration in IoV networks mainly focused on three collaboration methods: cloud and MEC servers [9]–[13], vehicles [14]–[19], and multi MEC servers. Cloud computing can enrich the computing resources in MEC-enabled IoV networks. However, it can not meet the low latency requirement. Computing resources such as vehicles can utilize idle resources, but they change frequently, which is lack stability. Therefore, the collaboration between multi MEC servers can enrich edge resources and has better stability than vehicle resources, becomes a feasible solution to meet the growing and dynamic e service requirements of IoV networks. The authors of [20] proposed a network that gave users three offloading options to reduce the computation workload of the MEC system. In [21], the authors satisfied the offloading requirements of the vehicle by purchasing computing resources from an alternate MEC server. The work by Xiao et al. [22] provided MEC cooperation based on traffic heat awareness, which can purchase computing resources from low-heat MEC servers to help high-heat MEC servers. The above researches mainly focused on workload balancing between multi MEC servers, but they didn't considered task Considering the wide application of edge collaboration in the IoV-MEC network, splitting tasks and processing them by multiple edge devices will further improve the efficiency of task processing. However, where the task is split and how each subtask is offloaded and

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 5, Issue 2, Mar.-Apr., 2024 pp: 98-105 ISSN: 2584-2145
www.ijemh.com

processed is still an open issue. Existing researches can be categorized into four types: serial offloading serial execution, serial offloading parallel execution, parallel offloading serial execution, and parallel offloading parallel execution. A further offloading strategy was proposed in [23] to jointly optimize latency and energy consumption by serial offloading serial execution. The authors of [24] provided a parallel offloading serial execution strategy to minimize the weighted sum of latency energy consumption. To minimize transmission energy consumption, a parallel offloading parallel execution strategy was provided in [25]. The authors of [26] combined cloud computing with MEC to achieve workload balancing through parallel offloading and parallel execution. The work by Chai et al. [27] proposed a parallel offloading parallel execution scheme, and categorized the tasks into different priorities to minimize the maximum task completion time. The authors of [28] focused on reliability, where task offloading and splitting were controlled through the SDN network. And tasks were serially offloaded to access RSU and split to multi-device for parallel execution. Considering the high mobility and frequent handover in IoV networks. Some works have studied mobility prediction to help avoid handover latency and reduce retransmitting energy consumption, especially in MEC-enabled IoV networks [29]–[31]. There were also some studies focused on information source selection algorithm to enhance transmission efficiency and reliability in the dynamically changing network topology [32], [33]. Moreover, in IoV networks, different applications have different performance requirements. Hence, flexible offloading strategies are essential to satisfy the requirements of various applications while saving energy in IoV networks. The most above-mentioned researches only focused on latency or energy consumption. The authors of [34] proposed the weighted sumof latency and energy consumption in a cellular network. The tradeoff between latency and energy consumption was introduced in IoV networks by the authors of [20], [35]. However, they didn't consider edge cooperation. Motivated by the above analysis, handover is frequent in IoV networks and will cause the decrease of offloading efficiency, but little literature studied the offloading strategy during handover. In order to meet the VT's offloading requirement with high mobility, we propose an offloading strategy using the RSUs located on the VT's pathway, in which RSUs on VT's pathway and other nearby RSUs are participated in computing during handover. This requires task splitting and transmitting subtasks to different RSUs to execute. Because a lot of uplink wireless resources will be occupied by parallel offloading, and tasks such as image recognition in IoV do not need to be executed sequentially [36], we propose a serial offloading parallel executing strategy to enhance task execution efficiency. Furthermore, as flexible offloading strategy is required to meet the various requirement of IoV applications, we propose to use multi MEC cooperation and adjust the weight sum factor based on task preference. Therefore, in this article, to achieve the various requirements of VT's applications with high mobility, we propose a serial offloading parallel execution strategy during the handover with latency energy tradeoff in MEC based IoV network. Specifically, we propose a MEC-based serial offloading parallel execution strategy to enhance L2SC offloading service experience for VT, which can utilize the handover time for edge execution. To reach the tradeoff between user experience and energy efficiency, we jointly optimize energy consumption and latency. The main contributions of this paper are shown as follows.

• We propose a novel cooperative offloading MEC framework in IoV networks, in which the task can be serial transmitted to RSU and parallel executed in several cooperate MEC servers, which can take full advantage of roadside computing resources and utilize the handover to satisfy the VT's low latency requirements in IoV networks.

• We introduce latency and energy consumption tradeoff into the proposed framework, and we use the weighted sum of latency and energy consumption as the cost. This can flexibly adjust the weighting factor for different IoV tasks, to meet the various requirements of IoV applications while saving energy.

• Because of the non-smoothness and non-convexity of the formulated problem, we provide an alternate convex search (ACS)-based algorithm to divide the original problem into two convex subproblems and obtain the suboptimal solution by iteratively solving the two subproblems. The rest of this paper is organized as follows. The system model is presented in Section II. The problem formulation and algorithm design are given in Section III and IV, respectively.The proposed method is verified by simulation results in Section V. Conclusions are drawn in Section VI.

II. SYSTEM MODEL

An MEC-enabled IoV network is considered as illustrated in Fig. 1. There are one vehicle and several roadside units (RSU) equipped with MEC servers, which are called edge nodes. Edge nodes can work cooperatively to provide task

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 5, Issue 2, Mar.-Apr., 2024 pp: 98-105 ISSN: 2584-2145
www.ijemh.com

offloading service to the vehicle. All edge nodes are connected with one high-speed optical fiber, and the transmission rate is equal between any two edge nodes. The vehicle has one task to offload to MEC servers when it is about to move into the coverage of a new RSU. In this paper, we aim to verify the performance of the proposed cooperative offloading MEC framework. Therefore, we only consider a one-car scenario. The multi-car scenarios will be studied in our future works.

To make the problem simple, we only consider the single task scenario in this article, and the multi-tasks scenario will be studied in our future works. We assume that the channel status remains unchanged while the task is uploading, for the reason that the coherence time is close to the uplink transmission time. Because of the high speed of vehicles, a vehicle may experience cell switching while offloading a L2SC task. A RSU and its associated MEC server constitute an edge node [37]. We assume that the MEC server will execute the input task as soon as it receives all the data of the task. We also assume that Bf represents the edge node who covered the VT when the task was produced, Bl represents the edge node who covered the VT when task offloading is over, and for other edge nodes who participated in the collaboration, we call them supporter. The task size (bits), the computation workload (cycles/bit) [2], and the latency constrain of the task are represented as C, α, Tmax, respectively. We also assume that the task is bit independent [24] and can be divided into several parts of arbitrary size after transmitted to Bf, one of which is executed at Bf and others are further offloaded to any other edge nodes. Because the task is bit independent, the subtasks do not need to be executed in a certain order. We assume that each edge node begins to execute the subtask at the time they receive it, which we called parallel execute. The results of each subtask will be transmit to Bl at the time it finish execution and combined at Bl , finally Bl transmits the final result to VT

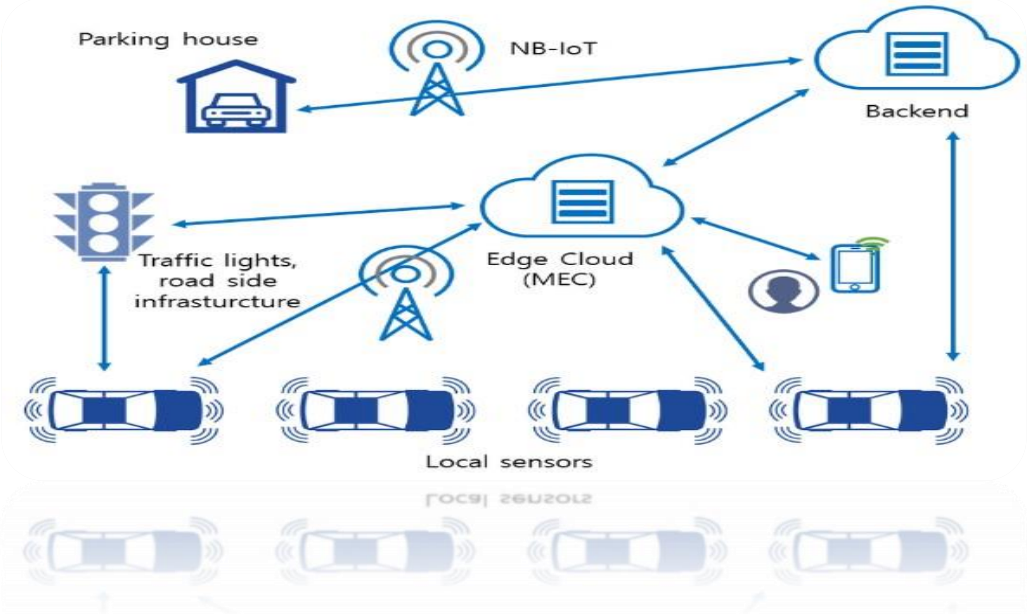

Fig. 1. System Model of multiple MEC-enabled IoV networks.

III. ALGORITHM

- The ACS Based Algorithm

The artificial cooperative search algorithm is a new optimization algorithm designed for solving complex optimization problems. As the structure of ACS algorithm is simpler than the structures of other artificial intelligence algorithms, it is easily programmable and notably faster than the other algorithms.

Algorithm 1 The alternate convex search based algorithm
Require: Network parameters δ, α, γ, etc; convergence tolerance ζ, iteration index τ = 1
Ensure: ηm, P
1: Initialize starting variables η^0_m, T^0_n, P^0 and a^0
2: repeat
3: Update η^τ_m and T^τ_n according to linear problem (14)
4: Update a^τ by solving convex problem (18)
5: Obtain P according to problem (17)

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 5, Issue 2, Mar.-Apr., 2024 pp: 98-105 ISSN: 2584-2145
www.ijemh.com

6: until $\varepsilon'(\eta^{\tau}_{m}, T^{\tau}_{n}, P^{\tau}) - \varepsilon'(\eta^{\tau-1^{m}}, T^{\tau-1^{n}}, P^{\tau-1}) \leq \zeta$
7: return η^{m}

V. SIMULATION RESULTS

In this section, we provide simulation results to evaluate the performance of the proposed partial offloading strategy by comparing it to three baseline strategies. We assume that there is a vehicle traveling down a one-way straight highway at the speed of about 100 km/h, and the coverage of RSU is 10 m. The transmission frequency between VT

TABLE I
KEY PERFORMANCE INDICATORS

Parameters	Values
δ weighting factor	0–1
α computation workload	40 cycles/bit
γ proportion between task result and itself	0.2
β transmit speed on the high-speed optical fiber	10^{10} bit/s
C task data size	1–10 Mbits
Transmission frequency	5.9GHz
f computation ability of MECs	8 × 109 cycles/s
channel gain	Rayleigh fading channel
N0 power spectral density	3 × 10−13 W
Tmax vehicle's task latency upper bound	30 ms
Pmaxvehicle's transmit power upper bound	0.2 W

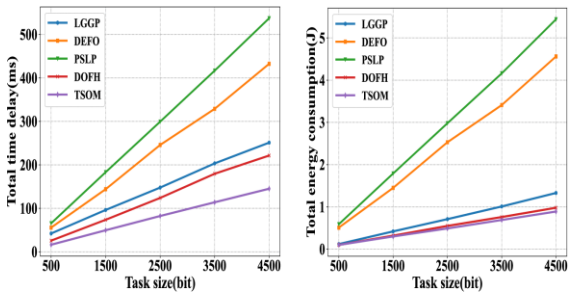

Figure. 2.Total time delay performance versus Total energy consumption

5.9 GHz. During the L2SC task offloading, the vehicle will experience handover, and receive the computing result after the handover. According to the existing researches, we express the existing offloading strategies into three baseline algorithms:
Baseline method 1 (no cooperation strategy): All fractions of the task are computed at the node Bf after the task is offloaded to it, then transmit the result to the node Bl , and then transmit the result to the VT. It represents the recent offloading strategies that didn't consider MEC cooperation. Baseline method 2 (further offloading strategy): The task is first offloaded to the node Bf and transmit all fractions of the task to node Bl , then computing all

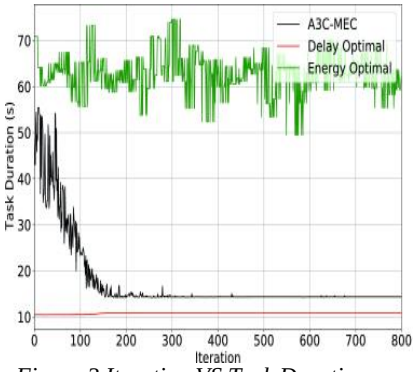

Figure 3 Iteration VS Task Duration

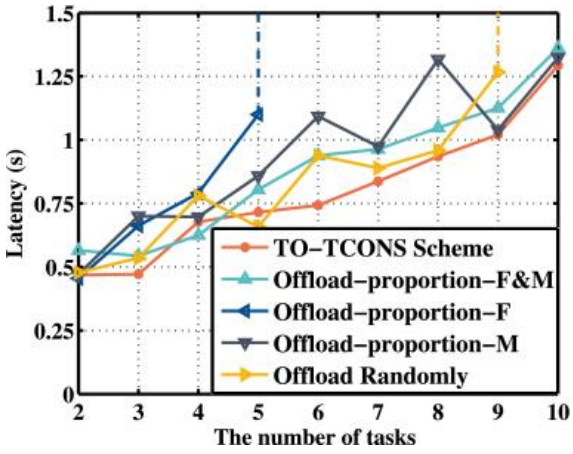

Figure 4 Latency VS Number of Tasks

Baseline method 3 (serial cooperate strategy): The task is split at the VT and offloading to several edge nodes parallelly, then MEC servers execute each part of the task serially in a fixed order. Finally, each part of the task result transmits to node Bl and then transmit the result to the VT. It represents the recent offloading strategies that have considered MEC cooperation by the serial computing of MEC servers. For the parallel

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 5, Issue 2, Mar.-Apr., 2024 pp: 98-105 ISSN: 2584-2145
www.ijemh.com

cooperate offloading which we proposed, we also give the performance comparison between the schemes with different number of edge nodes. The detailed simulation parameters are given in Table I.

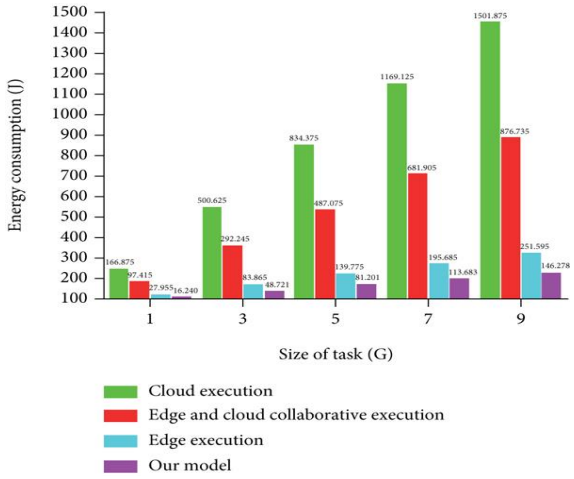

Figure 1 Size of task VS Energy Consumption

executed by different methods while meeting the 20 ms delay
requirement of IoV application. We can also see that latency increases with the increase of task data size and the curve is linear. This is because task data size C is in direct proportion to T_{total}. As we deduced in part II, $T_{total} = T_{edge} + T_s$, and we can obtain from equation (5) and equation (7) that C is in direct proportion to T_{edge} and T_s. Hence, C is in
direct proportion to T_{total}, which makes the curve linear. And proposed MEC cooperation strategy shows less latency than the three baseline strategies. Moreover ,with the increase of the number of cooperation MECs, latency performance shows more superiority. This is because the proposed strategy splits

SOURCE CODE FOR THE EDGE DEVICE:
Let's consider a scenario where a smart vehicle offloads a navigation task to a nearby edge device.
python

```
# Import necessary libraries
import random

# Edge device class
class EdgeDevice:
    def process_task(self, task):
        # Simulate processing time
        processing_time = random.uniform(0.5, 2.0)
        return f"Task '{task}' processed by edge device in {processing_time:.2f} seconds."

# Central cloud server class
class CentralCloud:
    def process_task(self, task):
        # Simulate processing time
        processing_time = random.uniform(2.0, 5.0)
        return f"Task '{task}' processed by central cloud in {processing_time:.2f} seconds."

# Smart vehicle class
class SmartVehicle:
    def _init_(self, task):
        self.task = task

    def offload_task(self, edge_device):
        result = edge_device.process_task(self.task)
        return result

# Example usage
if _name_ == "_main_":
    # Instantiate edge device and central cloud
    edge_device = EdgeDevice()
    central_cloud = CentralCloud()

    # Create a smart vehicle with a navigation task
    navigation_task = "Calculate optimal route"
    smart_vehicle = SmartVehicle(navigation_task)

    # Decide whether to offload task to edge device or central cloud (simplified logic)
    offload_decision = random.choice([True, False])

    if offload_decision:
        result = smart_vehicle.offload_task(edge_device)
    else:
        result = smart_vehicle.offload_task(central_cloud)

    print(result)
```

VI.CONCLUSION

In this paper, we investigate an edge collaborative task serial offloading parallel executing strategy in MEC-enabled IoV networks, which can split the task on the edge and use several MEC servers to paralleling execute each part of the task. And we formulate the problem as the minimization of the weighted sum of the energy consumption and the latency which is non-smooth and non-convex. An alternate convex search algorithm is provided to tackle the problem efficiently, which can converge to a sub-optimal solution. The numerical simulation results show

that the proposed multi-MEC cooperating partial offloading strategy can take advantage of the roadside computing resources properly, and shows superiority in the weighted sum of latency and energy consumption. When latency requirement becomes more relaxed, the proposed strategy can further reduce the total cost. And for different tasks, the proposed strategy can also change dynamically to meet their needs by adjusting the task preference index. Moreover, the impacts of various parameters were revealed, which validate the feasibility of the proposed method in different situations.

For future investigation, we plan to study the multi-task condition, and schedule the offloading sequence based on priority and task sequence. To solve the multi-task arriving problem, we will also study queuing issues.

REFERENCES

[1]. P. Papadimitratos, A. D. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," IEEE Commun. Mag., vol. 47, no. 11, pp. 84–95, 2009.

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

[3]. J. A. Guerrero-ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and Internet of things technologies," IEEE Wireless Commun., vol. 22, no. 6, pp. 122–128, 2015.

[4]. S. Bitam, A. Mellouk, and S. Zeadally, "'VANET'-cloud: A generic cloud computing model for vehicular ad hoc networks," IEEE Wireless Commun., vol. 22, no. 1, pp. 96–102, 2015.

[5]. F. Spinelli and V. Mancuso, "Toward enabled industrial verticals in 5G:A survey on MEC-based approaches to provisioning and flexibility," IEEE Commun. Surveys Tut., vol. 23, no. 1, pp. 596–630, 2021.

[6]. Y. Wu and J. Zheng, "Modeling and analysis of the uplink local delay in MEC-based VANETs," IEEE Trans. Veh. Technol., vol. 69, no. 4,pp. 3538–3549, 2020.

[7]. S. Xu et al., "RJCC: Reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT," IEEE Internet Things J., vol. 7, no. 9, pp. 8059–8076,2020.

[8]. D. Sabella et al., "MEC-based infotainment services for smart roads in 5G environments," in Proc. IEEE VTC, 2020, pp. 1–6.

[9]. X. Kong et al., "Deep reinforcement learning-based energy-efficient edge computing for Internet of vehicles," IEEE Trans. Ind. Informat.,vol. 18, no. 9, pp. 6308–6316, 2022.

[10]. R. W. L. Coutinho and A. Boukerche, "Modeling and analysis of ashared edge caching system for connected cars and industrial IoT-based applications," IEEE Trans. Ind. Informat., vol. 16, no. 3, pp. 2003–2012,2020.

[11]. S. M. A. Kazmi et al., "Infotainment enabled smart cars: A jointcommunication, caching, and computation approach," IEEE Trans. Veh.Technol., vol. 68, no. 9, pp. 8408–8420, 2019.

[12]. K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control systemcoordinated between cloud and mobile edge computing," in Proc. IEEE SICE, 2016, pp. 1122–1127.

[13]. J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," IEEE Trans. Veh. Technol., vol. 68, no. 8, pp. 7944–7956,2019.

[14]. Y. Wang, X. Hu, L. Guo, and Z. Yao, "Research on V2I/V2V hybrid multi-hop edge computing offloading algorithm in IoV environment," inProc. IEEE ICITE, 2020, pp. 336–340.

[15]. C. Chen, Y. Zeng, H. Li, Y. Liu, and S. Wan, "A multi-hop taskoffloading decision model in MEC-enabled Internet of vehicles," IEEE Internet Things J., p. 1, 2022.

[16]. X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," IEEE Trans. Mobile Comput., vol. 21, no. 2, pp. 598–611, 2022.

[17]. Y. Li et al., "Joint offloading decision and resource allocation for vehicular fog-edge computing networks: A contract-stackelberg

approach,"IEEE Internet Things J., vol. 9, no. 17, pp. 15 969–15 982, 2022.

[18]. S. Olariu, T. Hristov, and G. Yan, "The next paradigm shift: From vehicular networks to vehicular clouds," Mobile ad hoc networking:Cutting edge directions, pp. 645–700, 2013.

[19]. D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," IEEE Wireless Commun. Lett., vol. 8, no. 2, pp. 420–423, 2019.

[20]. H. Wang, Z. Lin, K. Guo, and T. Lv, "Computation offloading based on game theory in MEC-assisted V2X networks," in Proc. IEEE ICC Workshops, 2021, pp. 1–6.

[21]. K. Zhang, Y. Mao, S. Leng, S. Maharjan, Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in Proc.IEEE ICC, 2017, pp. 1–6.

[22]. Z. Xiao et al., "Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method," IEEE Internet Things J., vol. 7,no. 3, pp. 2038–2052, 2020.

[23]. W. Fan, Y. Liu, B. Tang, F. Wu, and Z. Wang, "Computation offloading based on cooperations of mobile edge computing-enabled base stations,"IEEE Access, vol. 6, pp. 22 622–22 633, 2018.

[24]. M. Qin et al., "Service-oriented energy-latency tradeoff for IoT task partial offloading in MEC-enhanced multi-RAT networks," IEEE Internet Things J., vol. 8, no. 3, pp. 1896–1907, 2021.

[25]. M. Zeng and V. Fodor, "Parallel processing at the edge in dense wireless un. Soc., vol. 3, pp. 1–14, 2022.

[26]. W. Zhang, G. Zhang, and S. Mao, "Joint parallel offloading and load balancing for cooperative-MEC systems with delay constraints," IEEE Trans. Veh. Technol., vol. 71, no. 4, pp. 4249–4263, 2022.

[27]. R. Chai, M. Li, T. Yang, and Q. Chen, "Dynamic priority-based computation scheduling and offloading for interdependent tasks: Leveraging parallel transmission and execution," IEEE Trans. Veh. Technol., vol. 70, no. 10, pp. 10970–10985, 2021.

[28]. X. Hou et al., "Reliable computation offloading for edge-computingenabled software-defined IoV," IEEE Internet Things J., vol. 7, no. 8, pp. 7097–7111, 2020.

[29]. X. Wang, Z. Ning, and L. Wang "Offloading in Internet of vehicles: A fog-enabled real-time traffic management system," IEEE Trans. Ind. Informat., vol. 14, no. 10, pp. 4568–4578, 2018.

[30]. S. Zhou et al., "Short-term traffic flow prediction of the smart city using 5G internet of vehicles based on edge computing," IEEE Trans. Intell. Transp. Syst., pp. 1–10, 2022.

[31]. S. D. A. Shah, M. A. Gregory, S. Li, R. d. R. Fontes, and L. Hou,"SDN-based service mobility management in MEC-enabled 5G and beyond vehicular networks," IEEE Internet Things J., vol. 9, no. 15, pp. 13425–13442, 2022.

[32]. J. Wang et al., "Vehicular sensing networks in a smart city: Principles, technologies and applications," IEEE Wireless Commun., vol. 25, no. 1, pp. 122–132, 2018.

[33]. J. Wang, C. Jiang, Z. Han, Y. Ren, and L. Hanzo, "Internet of vehicles: Sensing-aided transportation information collection and diffusion," IEEE Trans. Veh. Technol., vol. 67, no. 5, pp. 3813–3825, 2018.

[34]. J. Zhang et al., "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," IEEE Internet Things J., vol. 5, no. 4, pp. 2633–2645, 2018.

[35]. H. Wang, Z. Lin, K. Guo, and T. Lv, "Energy and delay minimization based on game theory in MEC-assisted vehicular networks," in Proc.IEEE ICC Workshops, 2021, pp. 1–6.

[36]. O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and ˜computational resources for energy efficiency in latency-constrainedapplication offloading," IEEE Trans. Veh. Technol., vol. 64, no. 10,pp. 4738–4755, 2015.

[37]. Y. Wu and J. Zheng, "Modeling and analysis of the downlink local delay in MEC-based VANETs," IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 6619–6630, 2020.

[38]. C. Song et al., "Hierarchical edge cloud enabling network slicing for 5G optical fronthaul," J. Opt. Commun. Netw., vol. 11, no. 4, pp. B60–B70,2019. [Online]. Available: https://opg.optica.org/jocn/abstract.cfm?URI=jocn-11-4-B60

[39]. W. He et al., "Latency minimization for full-duplex mobile-edge computing system," in Proc. IEEE ICC, 2019, pp. 1–6.

[40]. S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in Proc.IEEE INFOCOM, 2016, pp. 1–9.

[41]. V. Chvatal, et al., Linear programming. Macmillan, 1983.

[42]. I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, maxcut and complex semidefinite programming," Mathematical Programming, vol. 149, no. 1, pp. 47–81, 2015.